Ecole doctorale mathématiques, sciences de l'information et de l'ingénieur

Laboratoire ICube UMR 7357, Equipe TRIO, INSA Strasbourg

International Doctorate in Civil and Environmental Engineering, University of Florence

THÈSE

présentée par

Eugenio Pellis

Soutenue publiquement le 25 septembre 2023 à l'université de Florence

pour obtenir le grade de
Docteur de l'université de Strasbourg
Discipline Sciences de l'ingénieur
Spécialité Géomatique and Computer Vision

Segmentation sémantique des nuages de points du patrimoine bâti : une approche multi-vues

THÈSE dirigée par

M. Pierre Grussenmeyer	Professeur des universités, INSA Strasbourg, France
Mme Grazia Tucci	Professeur, Université de Florence, Italie

RAPPORTEURS

M. Pierre Charbonnier, Président	Directeur de recherche, Cerema Est, France
M. Gabriele Milani	Professeur des universités, Polytechnique de Milan, Italie

AUTRES MEMBRES DU JURY

M. Andrea Masiero	Professeur, Université de Florence, Italie
M. Michele Betti	Professeur, Université de Florence, Italie

International Doctorate in Civil and Environmental Engineering, University of Florence

Ecole Doctorale Mathématique, Sciences de l'Information et de l'Ingénieur

A multiview approach for the semantic segmentation of heritage building point clouds

Ph.D. Dissertation

Eugenio Pellis

Defended in public on the 25th of September 2023 at the University of Florence

Doctoral Examination Committee:

Gabriele Milani, external reviewer, *Professor, Technical University in Milan*

Pierre Charbonnier, external reviewer, President, *Research Director, CEREMA Est, Strasbourg*

Grazia Tucci, thesis supervisor, *Associate Professor, University of Florence*

Andrea Masiero, *Associate Professor, University of Florence*

Michele Betti, *Associate Professor, University of Florence*

Pierre Grussenmeyer, thesis supervisor, *Professor, National Institute of Applied Sciences, Strasbourg*

University of Strasbourg
University of Florence
2023

Declaration

I hereby declare that the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Eugenio Pellis

2023

Abstract

This dissertation arises from context of the digitization of cultural heritage, and from the emerging and growing need to define standards, procedures and workflows to operatively contribute at the conservation, protection, and dissemination of cultural heritage in the world throughout digital technologies. Building Information Modeling (BIM) has become increasingly significant in managing and documenting cultural heritage, and nowadays, Heritage Building Information Modeling (H-BIM) has become a new design methodology paradigm. It consists of an as-built digital representation of an existing building, which includes a wide range of useful information, and it turned out to be a powerful tool for the management and the preservation of cultural heritage. The overall process of creating an as-built model is called Scan-to-BIM, and it encompasses all the phases starting from building data acquisition, up to the 3D modeling and the creation of a digital twin. However, the production of as-built models is still an open problem in real-world and large-scale applications, and it still presents numerous issues and challenges. Currently, one of the main issues of the Scan-to-BIM process is the management of the large-scale data resulting from the acquisition campaign. The high level of detail and automation achieved by the latest acquisition technologies, like 3D laser scanner or photogrammetry, allow collecting a large amount of data in short time with an impressive accuracy, but properly and efficiently processing such data is still a challenging procedure. The effective production of BIM-based models has not yet reached an adequate level of automation, and it still requires time consuming manual interventions by specialized operators. This research focuses on supporting and improving the automation in the Scan-to-BIM pipeline, providing an effective strategy for the management of large-scale point clouds, and an efficient tool to improve the automatic transition from point cloud data to 3D digital models. One of the key point to support 3D model generation from point cloud is the process of semantic segmentation. It involves dividing the raw point cloud data into smaller, meaningful segment, and assigning a label or category to each segment, according to the objects present in the scene and to the list of categories of interest. It is a step towards the machine interpretation and understanding of the 3D environment, which can be exploited for the automatic execution of other complex tasks. Over the last years, the recent progress on artificial intelligence, machine learning, and deep learning turned out in a new era, characterized by the availability of powerful algorithms for semantic segmentation, which have already shown to ensure remarkable results in several applications, such as autonomous driving, robotics and medical diagnosis. Such recently developed methods are still not fully exploited for heritage buildings semantic segmentation: and at this time, few research works have explored the potential of artificial intelligence in this field. Hence, the main goal of this research is to investigate the effectiveness of artificial intelligence, and more specifically the deep learning branch, on the problem of semantic segmentation of heritage building point clouds. To this aim, a novel semantic segmentation workflow for 3D heritage point cloud is proposed. It is based on a deep learning multiview approach, in which the segmentation is carried out at first on a set of images coming from a photogrammetric survey, and then the determined labels are projected on the related point cloud by exploiting the interior and exterior camera parameters, already estimated in the photogrammetric workflow. Three main contributions can be identified in this dissertation. First, a new image-point dataset for heritage building semantic segmentation has been produced. It is composed by five building point cloud scenes, and the related photogrammetric images, both with their respective ground truth segmentation. All the phases of the dataset generation are illustrated in detail, including acquisition, processing, annotation standards and the labelling procedure. Secondly, three state-of-the-art image segmentation architectures, namely Fully Convolutional Network, SegNet and Deeplabv3+, have been trained, tested and compared on the new dataset. Finally, a labelling projection procedure, based on the majority vote principle, has been developed and tested. It leverages on the exterior and interior camera parameters calculated during the photogrammetric workflow in order to transfer the labels, outputted by the deep network, to the point cloud, producing a 3D segmented scene. Several tests and experiments are shown and discussed in detail. The obtained results are quite promising, as they showed a quite good robustness and functionality of the overall process on most of the conducted tests. However, the procedure still needs some improvements: despite modern deep learning networks often guarantee an impressive capability, the implemented segmentation step still provides not fully satisfactory results on unseen scenarios, probably due to the inadequate number of buildings in the training set, which hence should be enlarged, and due to the high complexity and variety of heritage scenes.

Contents

List of Figures

Chapter 2

Chapter 3

Chapter 4

Chapter 5

List of Tables

Chapter 4

Chapter 5

Chapter 1

Introduction

1.1 Digitization of cultural heritage

Over the last years, computer-aided digitization has emerged as a powerful technology to enhance documentation and preservation of cultural heritage, producing new knowledge forms and deeper comprehension levels. Digital technologies create new opportunities, providing innovative ways for the public to access, discover, explore, and enjoy cultural assets, and they increase the possibilities for reusing cultural assets for original and creative services and products in various sectors. Nowadays, the development of advanced digital technologies, such as 3D modeling, artificial intelligence, cloud computing, virtual and augmented reality, has brought new opportunities for digitization, online access, and digital preservation. They lead to a more efficient execution of processes such as the automated generation of metadata, knowledge extraction, automatic features recognition, computer-aided simulations and, in general, a deeper understanding and an improved analysis level. Since the extraordinary perspectives of digitization, on 26 January 2022, the European Commission proposed an inter-institutional solemn declaration on digital rights and principles for the Digital Decade. To contribute to the objectives of the Digital Decade the Commission has published the recommendation 2011/711/EU (European Commission, 2011) on a common European data space for cultural heritage. The aim is to accelerate the digitization of all cultural heritage monuments and sites, objects, and artefacts for future generation, to protect and preserve those at risk, and boost their reuse in domains such as education, sustainable tourism, and cultural creative sectors. With further recommendation (EU) 2021/1970 (European Commission, 2021), the Commission encourages EU Member States to digitize by 2030 all monuments and sites that are at risk of degradation and half of those highly frequented by tourist, and it encourages Member States to put in place appropriate frameworks to enhance the recovery and transformation of the cultural heritage sector to become more resilient in the future (European Commission, 2019). With the 2019 Declaration of Cooperation on advancing the digitization of cultural heritage, the European Commission's Expert Group on Digital Cultural Heritage and Europeana (DCHE Expert Group) contributed to the development of common guidelines for comprehensive documentation of European 3D cultural assets, providing a list with 10 basic principles for 3D digitization of tangible cultural heritage (European Commission, 2019). Among the guidelines provided for the different principles, those that mostly summarize the general aim of the project are the following ones:

- Define the rationale or purpose of the digitization project, considering the target user groups, examining the features of what is digitized, and defining the required strategy.
- Take into consideration long-term preservation from the beginning, including all aspects such as formats, storage, future migrations and re-use, ongoing maintenance, and the corresponding long-term costs.
- Select an archive able to accept incoming digital data in multiple formats, including raw data and metadata with the necessary storage space, making contents easily accessible, and supporting open format.
- Determine the minimum quality needed for the highest affordable, collecting and including rich metadata and annotations throughout the workflow, aiming for the highest 3D capture quality for the largest number of assets, investigating how high the capture resolution could be, and what the costs in time, money and skills needed are.
- Protect the assets both during and after digitization, avoiding as much as possible direct handling of the assets in question, using instead the digital twin created.
- Use the right equipment, methods, and workflow, promoting the use of advance acquisition technologies that match the category of the cultural heritage involved and the quality needed.

This dissertation arises from the general context of the digitization of cultural heritage, and from the growing necessity to define standard procedures and workflows to operatively contribute at the principles and aims of Digital Decade for the preservation, protection and maintenance of built heritage. Built heritage represents a crucial part of our collective inheritance as a society. It not only provides physical evidence of our past but also serves as a reminder of the cultural and economic achievements of previous generations. These monuments and buildings represent our history, cultural values and traditions. Therefore, preserving and protecting heritage buildings should have major importance in our society.

1.2 Motivation and challenges

In recent years, Building Information Modeling (BIM) began to play a significant role in managing and documenting heritage buildings, as proved by the new *Heritage Building Information Modeling* (H-BIM) paradigm that has been recently established. It consists in a digital representation of an existing building at present (as-built), which includes a wide range of information such as geometry, materials, technological systems, quantities, performance, documentation, maintenance information and many others. Several recent works have shown that this design methodology turned out to be a very powerful tool for the digitization of heritage buildings, and they proved the effectiveness of this modeling procedure in a wide range of applications. In addition, H-BIM is consistent with the principles of the European guidelines. The common pipeline for the creation of an as-built model is called *Scan-to-BIM*: such pipeline includes the whole process, starting from the data acquisition, up to the modeling phase. However, the reconstruction of as-built models still presents a number of issues and challenges in real-world and large-scale applications. One of the main issues of the Scan-to-BIM process is the management of the large-scale data resulting from the acquisition campaigns. The high level of detail and automation achieved by the latest acquisition technologies, like 3D laser scanner or photogrammetry, allow collecting a large amount of data in short time with an impressive accuracy, but efficiently and effectively processing such data is still a challenging procedure. The acquired data are usually represented by means of *3D point clouds*, set of points in a three-dimensional coordinate system, often containing information on colour or reflectance. Several steps are required to transform a point cloud in a 3D standard BIM object. They need to be carefully supervised, and they necessitate time-consuming and manual operations. Improving the automation of such process is a key point to improve and speed up the Scan-to-BIM procedure.

Among the operations that can be executed in order to support the generation of H-BIM models, the *Point Cloud Semantic Segmentation* (PCSS) is one of the most challenging ones. Nevertheless, lots of benefits can derive from its automation. It involves dividing the point cloud data into smaller, meaningful segments, and assigning a label or category to each segment, according to the objects present in the scene. It can be considered as a step towards the machine understanding of the 3D scene.

The other processing operations, such as registration, cleaning, or down-sampling, are relatively simple operations, and several implementations are already available and exploitable to automatize and speed-up the Scan-to-BIM. Instead, point cloud semantic segmentation still usually requires a significant amount of human interaction in order to obtain a semantically structured point cloud. Despite 3D parametric elements are part of the final output of the Scan-to-BIM procedure, the development of an automatic procedure to build the related 3D shape geometries, meshes, or surfaces, always starts from a proper segmented element. By separating and detecting each constructive element from the context and by defining its relationship within the other constructive elements, the algorithms could be more easily able to extract the features necessary to a proper transformation of the implicit point cloud data to a solid parametric object. In addition, several works have shown that the quality and the accuracy of the final extracted geometry are strictly related to the segmentation quality, and such relation is even more strong when dealing with complex elements like curved surfaces or irregular shapes. Heritage buildings are mainly composed by complex and irregular constructive elements, such as columns, vaults, arches or mouldings, and without a proper semantic segmentation of these elements the development of automatic modeling algorithms remains a challenging task. Currently an algorithm able to proper modeling complex elements in a fully automatic way is still not available, and the state-of-the-art approaches allow only the automatic reconstruction of simple elements, such as planar surfaces, (walls, roofs, or floors), openings (door or windows), and is some cases curved surfaces, like regular cylinders or spheres. In most of the practical H-BIM applications, the shift from point clouds to object-oriented elements is still obtained by means of manual operations, with only a partial support provided by some semi-automatic tools. Semantic segmentation of the raw point cloud is a fundamental operation even in the manual modeling case since it supports the operator during the modeling phase. It helps the operator by increasing the understanding of the building, enhancing the analysis of its components, improving the management of complex and detailed point clouds, improving the visualization of the point cloud in the CAD or BIM environment, and reducing the computational power required to manage massive clouds often composed by millions of points.

Over the last years, recent achievements on artificial intelligence (AI), machine learning (ML) and deep learning (DL) turned out in a new era, characterized by the availability of powerful algorithms for semantic segmentation, which have already shown to ensure remarkable results in several applications such as autonomous driving, robotics, medical diagnosis, and many others. Given the fundamental role played by semantic segmentation in several applications, this research area is particularly active, with numerous algorithms that are proposed every year. Currently, such interesting developments are still poorly applied in the field of heritage buildings semantic segmentation, and at this time, few research works have explored the potential of AI in the heritage field. This dissertation aims at investigating the effectiveness of this approach, in particular the branch of DL, on the problem of semantic segmentation of point clouds, to improve the automation of the Scan-to-BIM process. Despite PCSS may appear a limited contribution to the complex process of digitization, this task is the key point towards the automation of the Scan-to-BIM process, and hence its improvement can play an important role in making easier the digitization of cultural heritage.

1.3 Overall goal, objectives, and contributions

1.3.1 Overall goal

The overall goal of this thesis is to improve the automation in the digitization of cultural heritage, providing an effective strategy for the management of large-scale point clouds, and providing an efficient tool to facilitate the automatic transition from point cloud data to 3D digital models. These challenges should be critically investigated, and innovative methodologies should be proposed to address them in a constructive way. The target objective should be the development of a functional tool, adequately stable and robust, usable for a wide range of real-word buildings, and able to generalize among different building typologies. The proposed approach should be a powerful easy-to-use instrument, easily reproducible, open-source accessible and available for other researchers for future developments. It intends to overcome the current state-of-the-art approaches, providing a faster way to generate 3D heritage digital models.

1.3.2 Objectives

Considering the overall goal, a summary of the main aim of this thesis can be found in the following questions:

"Which are the main bottlenecks in the Scan-to-BIM workflow that make challenging the creation of 3D models starting from point cloud data?"

"Can machine learning or deep learning be leveraged to facilitate 3D model generation, and can they provide an effective help for the architectural heritage digitization?"

"Can semantic segmentation support 3D model generation, and can it help the development of tools and techniques to automatize the Scan-to-BIM?"

"Is there an effective procedure for the semantic segmentation of 3D point clouds, and can it be successfully applied in the context of heritage buildings?"

"Considering the more advanced techniques for heritage building data acquisition, what might be an ideal semantic segmentation pipeline globally relevant for the cultural heritage domain?"

"Can this segmentation pipeline be applied to a wide range of architectural heritage, and can it be able to generalize among several building typologies, multiple constructive elements, complex and non-standardize geometries?"

To address the resulted issues of these questions, the following objectives have been defined:

Objective 1. Identify the major issues and challenges in the Scan-to-BIM process, analysing each step that leads to the creation of 3D digital models, and providing an exhaustive literature review of the state-of-the-art algorithmic approaches to address each phase of the Scan-to-BIM workflow.

Objective 2. Explore how the recent advances in machine learning and deep learning can be exploited to support the 3D model generation in the Scan-to-BIM process. To this end, the main ML and DL semantic segmentation techniques should be reviewed and compared, underlying the strengths and the weakness of each method, and identifying the best strategies applicable to the heritage building domain.

Objective 3. Propose an effective semantic segmentation procedure suitable for the heritage building point clouds. The main aim is to create an approach applicable to a wide range of real-world scenarios with different conditions that fully exploits the data resulting from advanced acquisition technologies.

Objective 4. Create a specific dataset to develop and test the segmentation procedure of Objective 3. The new dataset should be composed by several buildings relevant to the heritage domain and guarantee an appropriate level of generalization. It should allow the integration with existing similar dataset, and it should be easily extendable with new data. To guarantee future developments and improvements it should be freely available, user-friendly and easily accessible by the research community.

Objective 5. Test the proposed procedure of Objective 3 on the new dataset, assessing and optimizing the performances in the case of unseen scenarios. The critical aspects and the limitations of the approaches should be pointed out, and the performance should be compared with existing methods or other approaches. The proposed procedure should overcome the state-of-the-art performances.

1.3.2 Main contributions

The main contributions of this thesis can be summarized:

- An overview of Heritage Building Information Modeling (H-BIM), looking in particular at the algorithmic approaches for the creation of 3D informative models. To this end, the various phases of the *Scan-to-BIM* process are identified and widely analysed. The state-of-the-art methods and techniques to address each of these phases are presented. They include point cloud acquisition, point cloud registration, point cloud sub-sampling, point cloud segmentation, and the BIM modeling from point cloud (**Objective 1**).
- An exhaustive review of the state-of-the-art algorithms for the 3D semantic segmentation of point cloud, including reprojection-based approaches and point-based approaches. Among them, the multiview approaches are widely analysed and discussed to better understand their principles and impact, along with the major open challenges (**Objective 2**).
- The development of a new multiview semantic segmentation pipeline, suited for the semantic segmentation of photogrammetric point clouds. It leverages on the features extraction from the images by deep neural networks, and their reprojection on the 3D point cloud by means of the intrinsic and extrinsic camera parameters (**Objective 3**).
- A new image and point based dataset for the semantic segmentation of heritage buildings has been introduced. It is composed by five heritage buildings labelled according to the guidelines of *ARCHdataset*. It has been used to train the deep models at the core of the segmentation pipeline (**Objective 4**).
- The development of a semi-automatic image labelling procedure, that enables to automatically label a set of photogrammetric images starting from a manual segmentation of the related point cloud. It allows to produce simultaneously hundreds of images ground truth, with a remarkable decrease of the labelling time and manual interventions (**Objective 4**).
- The implementation of three semantic segmentation deep architectures, *Fully Convolutional Network*, *SegNet*, and *Deeplabv3+*. They have been trained with the generated image dataset, and their performances have been assessed to find out the most efficient on the new heritage benchmark (**Objective 5**).
- The development of a labels transfer procedure from the 2D images to the 3D point cloud, leading to the completion of the segmentation pipeline. Assuming that the interior and exterior camera parameters are known, it allows to project

the labels predicted on a set of images to a point cloud. It has been tested with the available buildings of the dataset, and its accuracy has been assessed (**Objective 5**).

1.4 Thesis outline

Chapter 2. *Semantic Modeling of Heritage Buildings* introduces the notion of semantic modeling, and how this designing approach can be applied to heritage buildings modeling. The concept of Heritage Building Information Modeling (H-BIM) is introduced, and the goals, strengths and drawbacks of such modeling technique are analysed. The chapter focuses then on the *Scan-to-BIM* workflow, and an exhaustive literature review on the state-of-the-art algorithmic approaches to automatize the Scan-to-BIM is provided. The problem of point cloud semantic segmentation is comprehensively review, and the motivations that led to address this research topic are pointed out.

Chapter 3. *Semantic Segmentation Algorithms* introduces at first the concepts of Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL), and it briefly explains the functioning of the Convolutional Neural Networks (CNNs), since they are at the core of the further proposed segmentation procedure, and they are going to be widely used in the context of this thesis. The state-of-the-art algorithms for the semantic segmentation of 3D point clouds are then exhaustively presented. They are categorized into two main approaches: *projection-based* and *point-based.* Both the approaches are presented and discussed, focusing in particular on the *multiview approaches*, a subset of the projection-based methods that uses images as intermediate representation of the point cloud. The proposed procedure for the semantic segmentation of heritage buildings is then presented. The procedure is particularly well suited for the segmentation of photogrammetric point clouds, since it uses images for the feature extraction, and it leverages on the camera intrinsic and extrinsic parameters for the label projection on the point cloud.

Chapter 4. *The Dataset* presents in detail the image/point-based benchmark specifically designed to test the developed procedure. At first a review of the existing datasets, including image-based (2D), RGB-D based (2.5D) and point-based (3D) datasets is provided. In the second part, the buildings, their acquisitions and processing are illustrated in detail. The dataset is structured following the classification guidelines of the *ARCHdataset,* an existing large-scale benchmark for point cloud segmentation. A semi-automatic procedure to speed-up the image labelling is presented, and its functioning is widely explained. Finally, the structure and statistics of the dataset are shown.

Chapter 5. *Semantic Segmentation Results* provides exhaustively the outcomes of the semantic segmentation procedure applied to the buildings of the developed dataset. The first part is focused on image segmentation, and it illustrates the model implementations, the neural networks settings, the hyperparameters tuning, and the structure of the experiments. The second part is focused on the second step of the procedure, the reprojection of the labels from the images to the point cloud. The parameters used as settings of the procedure are described, and the final results are reported. Finally, the results are critically discussed.

Chapter 6. *Conclusion and Future Development* is the final chapter, and it summarizes the work presented in this thesis. The main contributions are recapped, and according with the obtained results, the main challenges and limitations are discussed, and future directions and developments are given.

Chapter 2

Semantic Modeling of Heritage Buildings

In this chapter the concept of *Semantic Modeling* and how this designing approach could be applied to the heritage building modeling is introduced (§2.1). In the second paragraph (§2.2) the concept of Building Information Modeling (BIM) applied to heritage constructions is illustrated and discussed, defining what H-BIM (§2.2.1) is, which are the goals of H-BIM (§2.2.2), and finally, illustrating the most relevant applications and case studies (§2.2.3). The creation of "as-built" informative models, a process generally called *Scan-to-BIM*, is a labour-intensive procedure that requires lot of manual intervention and time-consuming operations. In the paragraph §2.2.4 the state-of-the-art algorithmic approaches developed to support and speed-up the Scan-to-BIM process are widely analysed. Paragraph (§2.3) is focused on the point cloud processing phase, one of the most challenging step in the Scan-to-BIM, and a key point to improve the automation in such process. The state-of-the-art algorithms and methodologies for each phase of the point cloud processing are analysed in detail, including point cloud acquisition (§2.3.1), point cloud registration (§2.3.2), point cloud down-sampling (§2.3.3), point cloud segmentation (§2.3.4), and the BIM modeling from point cloud (§2.3.5). The chapter ends with a general summary (§2.4).

2.1 Introduction

Semantic modeling is a type of modeling procedure that is used to define and describe the semantic meaning of a physical objects. Semantic modeling is a method of structuring data in order to represent it in a logical way through an organized set of models, predefined rules, and data sources. This modeling technique is often used to segment, describe, and analyse various components of physical objects, as well as to represent the meanings associated to different objects and their relationships. The purpose of semantic modeling is to create a model that is highly detailed, complete, and accurate, so that physical objects can be accurately described in terms of their significance and in terms of how they are related and connected with their context. For example, semantic modeling can be used to describe the meaning of a building or structure, such as representing the structure historical importance, its cultural values, or its architectural features. This means that the model can provide information about the components of an object in terms of its physical parameters, such as its shape, size, orientation, etc. Additionally, semantic modeling can be used to represent more abstract aspects of a physical object, such as its human-related meaning, its connection to its cultural context, or its symbolic significance. The use of semantic modeling has become increasingly popular in a variety of fields, such as architecture, engineering, and urban planning. In these areas, semantic modeling is used as a basis for making decisions regarding the design and construction of structures and environments. By creating detailed models of physical objects that contain layers of information, this modeling technique can be utilized by architects, engineers, and urban planners to gain a greater understanding of their target object and optimize the design and construction process.

The importance of preserving and protecting heritage buildings is quickly becoming evident, and as such, there is a growing need for effective modeling techniques. *Semantic modeling* provides a solution to this need, as it is able to represent the physical and abstract characteristics of a heritage building in a meaningful way. By quantifying and assessing the various components and features of a heritage building, it is possible to gain a detailed understanding of the cultural and historical significance of the site. By creating a detailed model of a heritage building, it is possible to gain valuable insight into its physical characteristics, its spatial parameters, and its semantic meaning. This information can then be used when making decisions regarding the design and construction of structures and environments. The use of semantic modeling to study heritage buildings can also be invaluable when it comes to monitoring and preserving their cultural value. By understanding the meaning behind certain components of the buildings, such as their history, their characteristics, their symbolic value, it is possible to form effective conservation strategies to protect them and ensure their longevity. In the last years, the use of semantic modeling in the architectural field, and in particular to study and analyse heritage buildings, is becoming increasingly popular, and one of the most popular tools that changed completely the approach of representing and managing the cultural heritage is the use of Heritage Building Information Modeling.

2.2 Heritage Building Information Modeling (H-BIM)

2.2.1 What is H-BIM?

The term Building Information Modeling (BIM) was introduced in the latter part of last decade, when BIM replaced 3D digital modeling and computer aided design (CAD) as the expression generally used to describe the use of information and communication technology (ICT) for the design of the modern built environment. *Heritage Building Information Modeling* (H-BIM) is the extension of Building Information Modeling in the Heritage or Historical environment. According to the Centre for Digital Built Britain (CDBB), BIM is a digital approach that enables faster and more efficient creation, analysis and management of 3D buildings, as well as supporting decision-making in conservation of existing infrastructures. The utilization of the ICT design digital approach is increasingly prevalent, and it is well defined and normatively regulated in most country. ISO 19650-1:2018 defines BIM as "*Use of a shared digital representation of a built asset to facilitate design, construction and operation processes to form a reliable basis for decisions"*. The US National Building Information Model Standards Project Committee provides the following definition *"Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition"*. From April 2016, with the D.lgs. 50/2016, Italy has included the European directives 2014/24/EU on Public Procurement, that promote the *"rationalization of designing activities and of all connected verification processes, through the progressive adoption of digital methods and electronic instruments such as Building and Infrastructure Information Modeling"*. The Getty Conservation Institute's Recording, Documentation, & Information Management (RecorDIM) Initiative (2003 - 2007) provided one of the earliest definitions of Heritage Information Modeling, along with its principles and purpose (Eppich & Chabbi, 2007). With H-BIM models, historical buildings are represented as a digital twin, allowing virtual simulations and analysis of building performance over the life cycle of the physical building. In the last decades, the application of BIM in heritage contexts is increasing, as the benefits of using a digital approach are becoming more and more recognised. It offers the advantage of connectivity, accuracy and continuity across different phases of a building life-cycle and the ability to integrate data from different sources. H-BIM project files provide a live document, which allows for collaboration and consensus between all stakeholders, as well as providing a richer description of buildings, increasing traceability and data exchange. H-BIM offers further properties that are particularly well adapted to the needs of heritage buildings, such as the ability to carry out 3D reconstructions at different scales, visualisation in augmented or virtual reality, high-definition digital terrestrial scanning and the rendering of traditional features and details. It can create digital replicas of historical buildings and sites, to support better conservation, intervention and development, as well as sharing information among various stakeholders. Its accuracy and efficiency of data exchange among stakeholders are also useful in working with archaeological sites, allowing for a workflow that achieved previously unimaginable levels of detail in the spatial and temporal resolution of data. Its use is increasing in the last years, its efficiency and accuracy are increasingly recognised, and it is likely to continue to grow in the future. The implementation of Heritage BIM should be primarily focused on realising the value and significance of cultural heritage assets in the built environment and supporting the long-term sustainable conservation of these assets for all stakeholders. However, there is a lack of research on user engagement in the specifications or development of H-BIM, and few published prototypes for Heritage-focused BIM, in contrast with the definitions of "new construction" BIM. Consequently, there is a need for research that addresses the differing requirements of Heritage-focused BIM from new construction BIM, making this an area worthy of study.

2.2.2 What are the goals of H-BIM?

Currently, Building Information Modeling (BIM) protocols are being developed to make construction more efficient, with most of the focus on new construction, and, in most of the cases, these protocols might not be appropriate to approach BIM for existing buildings. As-built BIM protocols are still variously defined and described, so it is not surprising that most of applications and approaches show equal or greater diversity, even more in the case of heritage or historical constructions. The author in (Linning, 2014), in the context of Sydney Opera House management project, defined H-BIM as "*an information resource for future generation*" underlining the main goal of "*ensuring the effective sustainable conservation and management of the heritage complex for a projected further lifespan of 250 to 300 years",* and pointing out that *"an integrated information model opens up the way for more automated intelligence in the model incorporating rules and best practices".* The American Institute of Architects defined the broad goal of BIM as *"a* *collaborative alliance of people, systems, business structures and practices into a process that harnesses the talents and insights of all participants to optimize project results, increase value to the owner, reduce waste, and maximize efficiency through all phases of design, fabrication, and construction"* (AIA, 2007). A useful summary of the value and aims of BIM as process can be found in (Kemp, 2014): (i) it converges information production with engineering judgement and design, (ii) it provides wider, faster access to comprehensible and integrated information, (iii) it fosters instinctive but rigorous collaboration and better decision making, (iv) it harnesses innovative technologies and harvests intelligence from big data, (v) it enables reflective, adaptive thinking to incorporate whole life and integrated systems approach within the wider geographic context. According to ICOMOS guidelines the "*record of a building should be seen as cumulative with each stage adding both to the comprehensiveness of the record and the comprehension of the building that the record makes possible.*", and "*recording should therefore so far as possible not only illustrate and describe a building but also demonstrate significance.*" Despite several definitions, and many works that tried to define a common design workflow, lot of questions in the H-BIM approaches are still open, and there is a lack of a clear vision on end objectives. Some key points could be summarized in these four goals:

Preserve Heritage Buildings: H-BIM provides a holistic understanding of a heritage building's condition and its associated risks, allowing stakeholders to make informed decisions about preservation and maintenance. This information can also be used to develop conservation strategies that aim to protect the building's fabric and integrity.

Improve Management: By digitizing the entire building, H-BIM can provide stakeholders with an up-to-date view of the building's condition. This can help to identify and address any issues in a timely manner, preventing further damage. It can also be used to improve the management of the building's resources, including energy and materials.

Enhance Collaboration: H-BIM enables stakeholders to collaborate more effectively on the restoration and maintenance of a heritage building. By providing a shared platform to discuss, plan and monitor the building's condition, stakeholders can work together to ensure it is preserved to the highest possible standard.

Create a Digital Record: H-BIM provides a digital record of a building's condition, which can be used to monitor its progress over time. This can be invaluable for future generations, allowing them to access detailed information about the building's history and condition.

2.2.3 Recent applications of H-BIM

Several works have been proposed in literature, and in this section the most relevant and recent applications in the context of Heritage Building Information Modeling (H-BIM) will be presented. The concept of H-BIM was introduced for the first time in the work of (Murphy et al., 2009). The authors proposed a new design methodology for historical structures that leverages on parametric BIM objects. The workflow involves several stages: data collection and processing, identifying historic detail from architectural pattern books, building of parametric components, mapping these onto scan data, and producing engineering survey drawings and documentation. The product is a 3D model containing detail about the object's construction and material make-up, which automatically produces engineering drawings for conservation purposes, including 3D documentation, orthographic projections, sections, details and schedules. In (Baik et al., 2014) the authors created a parametric library to model the architectural elements of the constructions in Jeddah City, focusing on Hijazi architectural elements. The main aim of the work was to offer a rich digital architectural element library to be used in any heritage projects in Old Jeddah, reducing time to complete the model and ensuring a high-level standard of detail. As part of this process, three stages are involved, starting with the capture of data using range/image-based methods, then the processing of data, and finally the definition and modeling of the historical objects as parametric components. The work presented in (Quattrini et al., 2015) showed the possibility to develop a high-quality 3D model semantic-aware, able to connect geometrical and historical survey with descriptive thematic databases. For this purposes they started from point clouds by TLS, they built a centralized H-BIM model of the Church of Santa Maria at Portonovo, and they developed a procedure for its semantic management. Dore et al., 2015 presented the research output to date of a H-BIM model of the Four Courts, an historical building in Dublin City. After creating a 3D model using laser scans, they developed simulations of structural damage and decay for documentation and conservation analysis using the H-BIM model. In this work (Stober et al., 2018) the authors applied recording technologies, laser scanning, and thermal scanning, as support for H-BIM. Simulation of non-existent constructive elements is presented as the preceding step of creating a H-BIM library that allows for broader dissemination of heritage information. Through the modeling logic, which is closely related to the logic of construction, the results demonstrate the advantages of the model building approach to valorisation and interpretation of constructive changes over time. To overcome the complexity of conservation practices and the lack of knowledge of historical buildings, this research (Osello et al., 2018) aims to ensure the preservation of relevant information through the use of BIM methodology. A H-BIM model that ensures the accuracy of values related to space management and component conservation was developed based on the application of the methodology to a real case study. Using the modeling approach, the management and maintenance processes of the building could be optimized in line with the project's goal. This paper (Oreni et al., 2014) described the generation of the H-BIM of the Basilica di Collemaggio in L'Aquila, and its use in the on-going restoration project with a particular attention to the procedures used to preserve the complexity given by photogrammetric and laser scanning data. In order to achieve a detailed H-BIM, it was necessary to exploit the photogrammetric and laser scanning survey, interpret and model the structural behaviour of the building, and perform economic evaluations of the project. In the work of (Nieto et al., 2016) the Pavillion of Charles V was selected to set up a H-BIM model and to propose an innovative methodology of analysis and treatment of the information based on a representative 3D graphic model of the flooring and wall tiling of a historic building. BIM generates graphic models of parametric objects that enable refined systematization and efficient data management. The authors in (Castellano-Román & Pinto-Puerto, 2019) developed a H-BIM model of the Charterhouse of Jerez considering all the information required for the strategic planning for heritage management including research, protection, conservation and dissemination. They transferred the two concept of BIM Dimensions and Level of Development (LOD) to the heritage environment introducing the Level of Knowledge (LOK). In (Sztwiertnia et al., 2021) the authors showed how effective H-BIM can be in accurate spatial documentation of small-scale heritage site. For this purpose, they developed an accurate 3D model of the Wand Temple of Karpacz, in Poland, evaluating the use of Grafisoft ArchiCAD software regarding the interaction with point cloud, the accuracy of the obtained model, the level of detail that is possible to obtain in the case of modeling old wooden structures and the type of data that can be store using this platform. The H-BIM methodology has been used recently by the authors in (Conti et al., 2022), in which they test the use of BIM as design tool for the modeling of Carlo III bridge in Moiano. The bridge, almost without previous drawings and documents, was surveyed with an integrated approach using laser scanner, photogrammetry and topography. Based on the data, a metrically reliable H-BIM model was produced, along with graphical and non-graphical information, for use in maintenance and restoration. They used Autodesk Revit to model the bridge, exploiting group of parametric objects, and "in-place models" without parametric proprieties, to better represent the irregularities of the structure. This article (Rocha et al., 2020) addressed the creation of an H-BIM model of heritage assets using photogrammetry and 3D laser scanner. Based on point cloud data, the authors described the modeling phase of the House Pacos Reais in Lisbon, that was carried out using Revit mostly with manual operations.

In recent years, several works have explored methods and strategies to automatize and speed-up the process of creating 3D and parametric objects starting from survey data, such as point clouds, images or vector drawings. In this paper (Chiabrando et al., 2017) the authors proposed a workflow to automatize and speed-up the construction of H-BIM models of two case studies, Palazzo Sarmatoris and Smistamento RoundHouse. In order to avoid the manual modeling from scratch of the parametric objects, they tested PointSense, a plug-in for Revit that allows to easily extract ortho-view from the point cloud and section to simplify the 3D modeling phase. (Dore & Murphy, 2013) developed a semi-automatic method for generating façade models using an existing parametric library of object built using the Geometric Description Language (GDL) in ArchiCAD. By using procedural modeling techniques, the procedure automatically combines library objects based on architectural rules and proportions, creating the façade. In the work of (Costantino et al., 2021) a procedure to transform the point cloud into parametrized object was developed. The procedure consists in a two-way transformation of the object between the modeling software of Rhinoceros and the BIM software Revit, by using the plug-in Grasshopper. The procedure has been tested on a religious heritage building, and it showed a remarkable efficiency and potentiality, simplifying the modeling step. In a similar way, the authors in (Andriasyan et al., 2020) developed a workflow that enables the automatic conversion of TLS and SFM point cloud data into textured 3D meshes and then in parametric H-BIM objects, exploiting the combination of three software: Rhinoceros, Grasshopper and ArchiCAD. They tested the procedure on a study case demonstrating a good interoperability between the software. The authors in (Pepe et al., 2020) developed a procedure to build H-BIM models starting from geomatics surveys, using the support of Rhinoceros. At first, they imported the point cloud in the software environment, and secondly, they built the NURBS surfaces from point cloud exploiting the plug-in EvoluteTools to generate highly complex and sophisticated surfaces. The model was then parametrized by using Grasshopper and imported in the Revit environment. Such as this last works, the paper (Antón et al., 2018) proposed an approach for the generation of H-BIM model, including manual and automatic processes. The workflow is composed by three main steps: the laser scanning, the meshing process, and the 3D solid modeling. The meshing process has been performed automatically with Rhinoceros and the Mesh Flow plug-in, and the modeling process involves converting the mesh into closed NURBS. Such objects were then imported in the BIM environment considering the IFC file format. The authors in (Rolin et al., 2019) proposed a semi-automated geometrical H-BIM-oriented modeling step with Rhinoceros 5 software and VisualARQ plugin that has allowed the construction of a hybrid model by reverse engineering from the point clouds in a semi-automatic way, using cross-sections and edges extracted directly from the point cloud inspired by the tomography process.

2.2.4 Algorithmic approach to BIM

Since a long time, parametric models have successfully been used in the design of new buildings as they contain a high level of semantic information added or implanted by the designer during the modeling phase. For as-built BIMs however, this level of knowledge is not available, and, in most of the cases, it is not cost-effective or time expensive achieving a good level of knowledge and modeling this information in detail. Despite these drawbacks, it is sometimes useful to create as-built models, and, in order to obtain as much detailed geometrical information as possible, this is usually done by using a point cloud as reference, and by modeling the building features and elements. This process in generally called *Scan-To-BIM.*

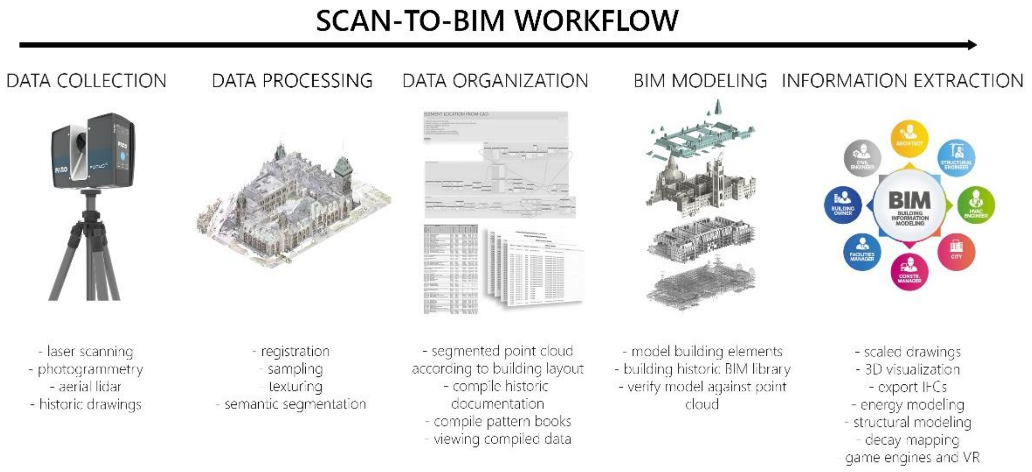


Figure 2.1 – The Scan-to-BIM workflow.

As illustrated in Figure 2.1 the *Scan-to-BIM* workflow is composed by five main phases:

Data collection. This is the first step in the process, and it involves collecting all the information useful for reaching a good level of understanding of the building, strictly related to the Level of Knowledge (LOK) and the Level of Detail (LOD) that we want to achieve for the as-built model. The geometrical and space data are usually collected using reality capture technologies such as laser scans, photogrammetry, often mounting the acquisition sensors on Unmanned Aerial Vehicle (UAVs). These methods allow both a fast acquisition and a high level of detail. More detailed information about data collection techniques will be provided in the next paragraph (§2.3.1). Data collection also includes capturing metadata to support other tasks such as cost estimation, energy analysis, structural analysis, etc. In the case of heritage buildings, it is often useful to carry out a detailed historical analysis and the recovery of historic drawings, past interventions, or monitoring data that allow a clever and careful interpretation of the construction.

Data processing. It involves processing all the acquired data in order to produce as output a reliable and easy-to-use dense point cloud. Therefore, the raw point cloud needs to pass through a series of processing steps so that the post-processed data can be used for further 3D modeling. Commonly the processing procedures consists of these phases: (i) *data registration*, it aims to align multiple point clouds collected from different locations in a common reference system, (ii) *data sub-sampling*, it consists of reducing the dimension of the data to make it more handle, (iii) *data cleaning*, it aims to remove noise, outliers and gaps generated by environmental or technical limitations of sensors, and lastly (iv) *semantic segmentation* that allows to group points that share similar features in continuous regions, spatially related and organized. Data processing will be analysed and discussed in detail in the next paragraph (§2.3).

Data organization. This phase allows to ensure that all the acquired information are accurate and consistent by properly comparing and organizing the various data into sections or structured layers, according to a predefined standard or system. It can help to save time and money, as errors and discrepancies can be identified and addressed early in the workflow, and it allow to easily access to the data during the modeling phase.

BIM modeling. This is the phase in which the point cloud and all the other collected information are used and combined for the creation of the 3D model in the virtual environment of the BIM platform using parametric objects and defining the attributes and the relationships for each elements of the model. The modeling involves three tasks (Tang et al., 2010): (i) modeling the geometry of the components, (ii) assigning the object category and the material propriety to the component, and (iii) establishing relationships between components. BIM modeling will be discussed in the next paragraph (§2.3.5).

Information Extraction. This is not a proper phase of the Scan-to-BIM process, but it involves the use of the BIM model for practical applications, such as the 3D visualization of the model, the information extraction for energy evaluation, structural analysis, etc. However, interventions on the building, or some applications such as the monitoring, can involve the transformation of the construction or the collection of new data that could be taken into account by updating the BIM model.

It is well known that Scan-to-BIM is an error-prone procedure that often requires manual and time-consuming interventions (José López et al., 2017), (Bruno et al., 2018). In order to speed up the production of as-built models in the Scan-to-BIM workflow and improve the accuracy of the results, algorithmic improvements are constantly developed to automate the translation of point data to parametric models. Automatic algorithms that could quickly and accurately generate BIM models directly from a raw scanned geometry, would represent an incredibly powerful tool, but parametric model generation from point cloud is currently a bottleneck hard to overcome. To turn a point cloud into a BIM model, the data must be properly interpreted and separated, that is separated taking into account their characteristics and structured in a machine-interpretable context. Human brains are powerful enough to translate neuron pulses directly into parametric information based on previous experiences, but machines need a series of additional and easier operations. The development of algorithms that can interpret point data into a parametric model involves giving the machine the ability to interpret data with the same level of precision and understanding that humans have. To do that, the machine needs a frame of reference and encoding capabilities to compare the data with, not only to reading, classifying, and interpreting the context of the data, but also to retain information and apply it to future tasks. However, the development of such algorithms would not solve completely the issues of the interpretation of data, that is subjective and often depends on the human designer, as specified by (Adan & Huber, 2011) "modeling of surface shapes is an especially labour-intensive and error prone operation, and even trained modellers sometimes produce significantly different results".

Despite these drawbacks, over the last few years, several commercial software and academic research works have proposed various workflow and algorithms to automate the reconstruction of existing buildings in various phases and steps of the Scan-to-BIM. From the academic point of view, the most popular and consolidated works are the research in (Jung et al., 2014), (Hong et al., 2015), (Wang et al., 2015) and (Zheliazkova et al., 2015). These research works proposed several approaches to extract geometrical features of the construction objects by segmenting the point cloud. Further detail will be provided in the section "*Automatic approaches for as-built BIMs"* in the paragraph §2.3.5. On the other hand, in recent years, several commercial software has investigated the automatic reconstruction of existing buildings from point clouds, and currently they allow various degrees of automation in various phase of the process. Following are reported some of the most popular and available packages. EdgeWise® was developed as complement software for Autodesk Revit, and it allows to classify and separate the point cloud into uniform surfaces. When the points are separated, the application automatically detects candidates points between pairs of similar horizontal planes, and it builds the parametric element based on the extracted shape. Scan-to-BIM® is a plug-in for Autodesk Revit developed by IMAGINiT Technologies. It allows to automatically model walls or other simple elements surfaces in parametric objects, by detecting and adjusting points with similar features. Leica CloudWorx® is a popular plug-in for AutoCAD, and it provides various tools to manage and work with as-built point cloud directly in the CAD environment. It allows to automatically detect and recognize geometric features in the point cloud, that then can be manually modelled.

Other popular software are Trimble RealWorks, that allows the semiautomatic creation of geometry by manually segmenting the clouds Intergraph Smart3D used to automatically detect and model pipes, Kubit PointSense Buildings and PointFuse from Arithmetica. Over the last years, an important step towards the automatic modeling has been done by commercial and academic research. However, the fully automation of Scan-to-BIM is still in its infancy, and remarkable progress still need to be done. There are currently no methods or workflows to automate Scan-to-BIM that have gained wide acceptance in the AEC community (Giel & Issa, 2016).

2.3 Point clouds and their manipulation

In recent years, the tools and the technologies used to acquire 3D data or point clouds have remarkably been improved, enabling fast high resolution 3D geometric information acquisition and extraction. In addition, the available computing power has kept growing, being now sufficient to run even complex analysis algorithms on these data. Scanning methods may vary, but the result is usually a point cloud containing coordinates (x, y, z), colour (RGB), reflection intensity of the transmitted signal and gravity direction. An important distinction between 3D parametric models and point clouds concerns the knowledge on certain subject characteristics, which the raw point cloud geometry does not provide explicitly. Therefore, the implicit data concerning the position and reflective value of points must be evaluated and manipulated to determine its meaning compared to the values in the rest of the data set. By comparing available values of points (x-coordinate, y-coordinate, z-coordinate, red, green, blue, intensity and orientation), the human brain can detect patterns and interpret what the points represent onscreen. For example, a scanned scene can represent a room with a window, door, and some pipework on the ceiling. An experienced eye would recognize the type of window or the pipe diameter. This knowledge can be used to determine where the window was purchased or what type of liquid the pipes are likely to carry. However, the machine still has a limited capability in detecting pattern and interpret points, and for this reason there is an increasing interest and need to develop an efficient strategy of transferring the human ability to recognize and parameterize features in a point cloud onto a machine. This is already being done in broad terms and in specific environments, but the procedure to extend it to Building Information Models is still in its early stages. The process of interpreting point clouds to obtain parametric models is usually done manually, which is an expensive and time-consuming task. In the next paragraphs, the steps of the Scan-to-BIM workflow involving the use and process of point clouds will be illustrated. For each step the state-of-the-art algorithms to automatize, speed-up, and improve each phase will be analyzed and discussed. The paragraphs are organized as follow: (§2.3.1) *Point Cloud Data Acquisition*, (§2.3.2) *Point Cloud Down-sampling*, (§2.3.3) *Point Cloud Registration*, (§2.3.4) *Point Cloud Segmentation*, and (§2.3.5) *BIM Modelling from Point Cloud.*

2.3.1 Point cloud data acquisition

Four main methods are used for point cloud acquisition: Image-derived methods, Light Detection And Ranging (LiDAR) systems, Red Green Blue Depth (RGB-D) cameras, and Synthetic Aperture Radar (SAR) systems. Each of these techniques has its distinct data features and applications due to the varying survey principles and platforms used. A brief description of these techniques is provided in the following.

Image-derived methods. Image-derived point clouds are generated indirectly from imagery, using electro-optical systems such as cameras in order to acquire stereo or multi-view images. Photogrammetric principles are then used to calculate 3D point information, either automatically or semi-automatically. There are four distinct platforms for creating this type of point clouds, including airborne, space-borne, UAV-based, and close-range. Traditionally, aerial photogrammetry was used to create 3D points with a semi-automatic human-computer interaction in digital photogrammetric systems, where high survey accuracy was required. However, this was a time-consuming and expensive process, making it difficult to generate dense spatial information for large areas. In surveying and remote sensing, these early point clouds were used in mapping and producing Digital Surface Models (DSMs) and Digital Elevation Models (DEMs). Due to resolution limitations and the inability to process multiple view images, traditional photogrammetry could only acquire close to nadir views from aerial/satellite platforms, making the generated point cloud a 2.5D, rather than full 3D. Close-range photogrammetry can also be used to determine the position of certain points on objects on small-area scenes, but manual editing is still required in the point cloud generating process. In recent years, the development of processes such as Dense Matching (Hirschmüller, 2005), (Hirschmüller, 2008), Multi-View Stereo (MVS) (Furukawa & Ponce, 2010), (Nex & Remondino, 2014), and Structure from Motion (SfM) (Westoby et al., 2012), (Snavely et al., 2006), (Snavely et al., 2008) have revolutionized the generation of image-derived point clouds and opened a new era of vision-based reconstructions. Through these methods, it is now possible to easily generate large 3D dense point clouds in city-scale areas. SfM has the power to automatically estimate camera positions and orientations, enabling the simultaneous processing of multi-view images, and dense matching and MVS algorithms enable to generate large point clouds. However, the use of SfM and MVS can be critical in certain applications, for instance when covering large areas (Xiao, Owens, et al., 2013a), not matching that of LiDAR and traditional photogrammetry techniques. In comparison, a satellite stereo system may not have the same spatial resolution or availability of multi-view imagery as airborne photogrammetry, but it can map large spaces in a short period of time and at a lower cost. Along with the advancements in dense matching techniques, satellite imagery is slowly becoming an important source for image-derived point clouds due to increased spatial resolution.

LiDAR systems. Light Detection And Ranging (LiDAR) is a powerful surveying and remote sensing technique, based on the use of laser in order to determine object distances from the sensor. LiDAR is typically pulse-based, and its point density or resolution can greatly vary depending on the sensor and platform, from less than 10 points per m^2 (pts/m^2) to thousands of points per m^2 (R. Qin et al., 2016). LiDAR systems are divided into four distinct categories: Airborne Laser Scanning (ALS), Terrestrial Laser Scanning (TLS), Mobile Laser Scanning (MLS), and Unmanned Laser Scanning (ULS). Early ALS point clouds compared with traditional photogrammetric point clouds are more expensive to acquire and they lack spectral information. In addition, they have a low density when the distance from the ground is large. Multispectral airborne LiDAR, which uses different wavelengths, is well-suited for extracting water, vegetation and shadows. TLS (also known as static LiDAR scanning) is based on the use of a sensor mounted on a stationary tripod. Its usage in a middle- or close-range environment provides high point cloud density and real, high quality 3D models. So far, TLS has typically been used for modelling small urban or forest sites, heritage or artwork documentation, and for other similar tasks. MLS (mobile LiDAR scanning), on the other hand, is an acquiring process performed from a moving vehicle on the ground, most commonly a car. This technology is used in the creation of HD maps, which is currently an ongoing research topic due to its application in the development of autonomous driving. ULS (unmanned LiDAR scanning) systems are usually mounted on drones or other unmanned vehicles. They are relatively cost-effective and very flexible, making them increasingly popular in recent times. Compared to ALS (aerial LiDAR scanning) which works above objects, ULS can provide shorter-distance LiDAR surveys with higher accuracy. Additionally, its small platform also allows for high operational flexibility, making it a suitable choice for tasks involving agriculture and forestry surveys. When it comes to LiDAR scanning, it is essential to combine point positions with Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU) data, due to the system being constantly in motion with the platform (Le Chang et al., 2020). LiDAR has been the most widely utilized source of point cloud data and has been often used to provide benchmarks to assess the accuracy of point clouds generated with other techniques.

RGB-D cameras. An RGB-D camera is a type of sensor that can acquire both RGB and depth information. Its working principles can be based on three different kinds of approaches: structured light (Han et al., 2013), stereo (Mattoccia & Poggi, 2015), and time of flight (Lachat et al., 2015). In contrast to LiDAR, an RGB-D camera is much cheaper and can measure the distance between the camera and objects pixel-wise. Microsoft's Kinect is a well-known and widely used example of an RGB-D camera. The are several sensors that use different technologies to measure the depth, and the relative orientation between or among the different sensors is calibrated and known, so synchronized RGB images and depth maps can be easily acquired. From the known position of the camera's optical centre, the 3D space position of each pixel in a depth map can be used to create a point cloud. RGB-D cameras have three main applications: object tracking, human pose or signature recognition, and SLAM-based environment reconstruction. As they are usually employed in close-range indoor environments, they can often be seen in indoor point cloud segmentation benchmarks (Jinyu et al., 2021), (Zhu et al., 2022).

SAR point cloud. Interferometric Synthetic Aperture Radar (InSAR) is a key technique in remote sensing, providing maps of surface deformation and digital elevation through analysing multiple SAR image pairs. In recent years InSAR-based point clouds have become increasingly valuable, creating new possibilities in a multitude of point cloud applications (Zhu & Shahzad, 2014a), (Schmitt et al., 2015). Two main InSAR point cloud generating techniques are Persistent Scatterer Interferometry (PSI) (Bamler et al., 2009), and Synthetic Aperture Radar tomography (TomoSAR) (Zhu & Bamler, 2010). TomoSAR has demonstrated greater accuracy when reconstructing and monitoring infrastructure, particularly in urban areas. The point density of TomoSAR point clouds is comparable to ALS and can be employed for building reconstruction purposes (Zhu & Shahzad, 2014b). These point clouds have several features which make them advantageous for use in such applications. TomoSAR point clouds reconstructed from spaceborne data reach a moderate 3D positioning accuracy, around 1 m, with accuracy reaching even decimetre level when using geocoding error correction techniques. By comparison, ALS systems provide accuracy typically on the order of 0.1 m. Due to their coherent imaging nature and side-looking geometry, TomoSAR point clouds bring different advantages with respect to LiDAR systems. They specifically provide rich facade information, as pixel-wise TomoSAR was used for the high-resolution reconstruction of complex buildings with a very high level of detail from spaceborne SAR data. Temporarily incoherent objects, such as trees, cannot be reconstructed from multipass spaceborne SAR image stacks, however, and the full structure of individual buildings from space requires facade reconstruction using TomoSAR point clouds from multiple viewing angles. InSAR point clouds offer a unique ability compared to LiDAR and optical sensors: the capacity to provide fourth dimension information from space, i.e., temporal deformation of the building complex and microwave scattering properties of the facade. Two main shortcomings impede their accuracy, though. Limited orbit spread and the small number of images lead to an anisotropic location error of TomoSAR points: elevation error is typically one or two orders of magnitude higher than in range and azimuth. Additionally, multiple scattering may give rise to ghost scatterings, appearing as outliers far away from a realistic 3D position. Due to the remarkable advancements in image-derived, LiDAR-based, and RGB-D-based point clouds, the utilization of Synthetic-Aperture Radar (SAR) data has yet to be widely explored. However, mature SAR satellite such as TerraSAR-X, has collected a large amount of global SAR data, allowing for the generation of InSAR-based reconstructions at a global scale (Shi et al., 2019). In the future, SAR are expected to play an increasingly important role in point cloud acquisition.

2.3.2 Point cloud down-sampling

Due to the typically huge size of LiDAR and photogrammetric point clouds, processing them tend to be a difficult and time-consuming operation, usually requiring a multi-step procedure involving tasks such as registration, segmentation and data interpretation. To make these processes easier, down-sampling is often employed as a pre-processing step, alongside other operations such as filtering, smoothing and outlier removal. Quick and efficient down-sampling of 3D point clouds is essential for the sake of the computational efficiency of the data processing procedure: various algorithms exist for this purpose. For instance, software such as CloudCompare, Leica and Z+F Laser Control software, Autodesk, and Geomagic Suite can be employed to down-sample 3D point clouds. In the last years, several approaches have been proposed to sample point clouds. The authors in (Al-Durgham, 2019) suggested an adaptive down sampling approach that preserves points in low-density areas, while eliminating redundant points in high-density areas. This technique has been used in a variety of research and case studies. (Y. J. Lin et al., 2016) developed this method based on planar neighbourhoods, while the authors in (Al-Rawabdeh et al., 2020) combined planar adaptive down sampling and Gaussian sphere-based down sampling in order to work with irregular point clouds. Unfortunately, this method is time-consuming as it evaluates local density using neighbouring information, thus limiting its speed. The Spectral Decomposition Filter (SpDF) proposed in (Labussiere et al., 2018) is a novel sampling method aimed at reducing the number of points in large-scale point clouds while preserving geometric details with a non-uniform density. First, the input point clouds are analysed to identify geometric primitives and their saliencies. Then, density measures are computed for each geometric primitive based on their saliencies. If the geometric primitive density is higher than the desired density, the primitive is subsampled. This process is repeated until the density is less than the desired density, ultimately providing an output as a uniformly sampled point cloud that can be used for efficient large-scale applications. Octree-based sampling, developed in (El-Sayed et al., 2018), is an approach that combines octree-balancing with down-sampling and principal component analysis (PCA). The initial step in this process is to divide the point cloud into small cubes using the octree approach. After the cubes are created, they are then down-sampled based on their local densities. NSS (Normal-Space Sampling) and DNSS (Dual Normal-Space Sampling) are two methods that work by down-sampling points within normal spaces. NSS (Rusinkiewicz & Levoy, 2001) focuses on the point translational components, while DNSS (Kwok, 2019) takes into account both translational and rotational components. These methods are simple and cost-effective, but do not perform well with large-scale point clouds due to their disregard for spatial distribution. (Błaszczak-Bąk, 2016) developed a method called OptD to reduce large datasets, specifically for digital terrain applications in ALS point-cloud processing. OptD is a fully automated reduction technique that produces an optimal result by meeting optimization criteria.

2.3.3 Point cloud registration

Registration is the process of properly aligning or fitting a point cloud or data set, and it is a fundamental step to process point clouds in the Scan-to-BIM workflow. This alignment typically occurs in relation to a local grid, another point cloud, or global grid. In practical cases, a point cloud is often made up of multiple scans that need to be put together to create a complete representation of the object. The goal of registration is to find a correct transformation that optimizes data position in relation to the model (Gelfand et al., 2005). Terrestrial laser scanning often requires the scanner to be moved in order to capture a full view of larger objects. In these cases, a common coordinate system is established, and pair-wise registration is the standard procedure to merge the scans (Mitra et al., 2004). The cloud could be georeferenced, and in this case the coordinate system corresponds to the real-word position. Several methods have been proposed in literature, and they can be divided in two main group: *traditional methods* and *deep learning methods*.

Traditional methods. According to (L. Cheng et al., 2018) traditional point-cloud registration methods are divided into two parts: an initial coarse registration and a subsequent fine registration. The coarse registration is designed to match the 3D features of two rough point clouds and is classified into point-based, line-based and surface-based methods. The coarse registration is used to roughly align the point clouds, followed by a fine registration using iterative approximation methods to improve accuracy, a process commonly referred to as *coarse-to-fine* registration. The fine registration method is used to attain the best alignment between two point clouds by automatically minimizing an error function. This is accomplished by employing iterative closest point (ICP) algorithms (Segal et al., 2009a) , RANSAC (Fontanelli et al., 2007) and normal distribution transform (NDT) methods (Biber et al., 2003). The iterative approaches are utilized for fine registration, and they allow for a more precise transformation of the two point clouds. By minimizing a properly defined error function, the method determines a somehow optimum transformation matrix, thereby leading to a more accurate outcome. Iterative approximation is one of the most widely used techniques for accurate and stable 3D point-cloud data registration. This method starts by identifying correspondences between the two point sets and then determines the rigid transformation between them by minimizing the average distance between one of the two sets and the other one properly transformed. Unfortunately, classical ICP algorithm may end up on a local minimum, in particular when the initial condition is far from the correct registration (Y. He et al., 2017). Several approaches have been developed to deal with this issue and improve the ICP algorithms, such as the point-to-line (Censi, 2008), point-to-plane (Grant et al., 2012), point-to-surface (Makovetskii et al., 2017), Generalised-ICP, and GO-ICP. Generalised-ICP (Segal et al., 2009b) is a method exploiting the combination of ICP and point-to-plane ICP in a single probabilistic structure, whereas GO-ICP (Yang et al., 2016) uses a branch-and-bound approach to address the global optimization. An evaluation and a comparison of the various iterative methods is presented in the work of (Li et al., 2020). RANSAC is another suitable method for fine registration, being usable also for pre-processing and segmenting point cloud data. By randomly selecting different sets of points to register it then fit a predefined model efficiently in the presence of noise and outliers. It has a high computing efficiency but due to its randomised nature it is not capable to guarantee a globally optimal solution (C. Yu & Ju, 2018). NDT is another method for fine registration, based on the probability density function. A 3D grid is used to represent point-cloud data, and a probability distribution is applied to each grid point to achieve optimal fine registration. It is faster and more reliable in real-time applications than ICP (Magnusson et al., 2009) because it does not require a good initial solution. However, due to the large number of calculations needed for this method, it is labour-intensive. ICP is the most used registration method for 3D point clouds, however, high-density data is necessary to achieve accurate results. This method is not well-suited for airborne LiDAR due to its large and noisy data. (Gressin et al., 2013) compared ICP algorithms on airborne, mobile and terrestrial LiDAR platforms and found that the highest accuracy was obtained on TLS and MMS datasets, while the results on ALS were quite unsatisfactory. So far, NDT has only been implemented on TLS datasets and it is not suitable for complicated and large environments. RANSAC, on the other hand, is used for ALS, TLS, and MLS datasets mainly as a pre-processing step to eliminate outliers and blockages.

Deep learning methods. The registration of 3D point clouds is still a challenging task due to their unordered and sparse nature. In recent years, deep learning has become increasingly important in point-cloud registration, leading to the development of state-of-the-art deep-learning-based methods. Two of the most well-known geometric deep learning methods for 3D point clouds are PointNet and Graph Neural Networks. These methods have inspired the development of various deep learning registration methods. For example, the authors in (Deng et al., 2018b) proposed the Point Pair Feature Network (PPFNet), which learns 3D local feature descriptors from unorganised point sets. However, a major limitation of this approach is that it requires a considerable amount of annotated data. To address this issue, the authors in (Deng et al., 2018a) developed PPF-FoldNet, which employs unsupervised learning of 3D local descriptors. PointNetLK, developed by Aoki et al. (2019), combines the Lucas and Kanade algorithm with a global feature descriptor based on PointNet. It employs iterative approximation techniques to estimate the relative transformation PCRNet (Sarode et al., 2019) is another deep-learning method that also uses PointNet for the extraction of global features. This approach uses a Siamese architecture comprising five multi-layered perceptrons that are used to generate the global features, which are then fed into five fully connected layers, together with an output layer of the desired dimension for the pose. Despite being faster and more robust than methods that rely on iterations, PCRNet is less accurate. Fully Convolutional Geometric Features (FCGF) (Choy et al., 2019) is an effective method for extracting geometric features by computing a full convolution network. To improve the accuracy of this method, DeepGPMR (Yuan et al., 2020) was developed, which combines Gaussian Mixture Model (GMM) registration with neural networks and does not require costly iterative procedures. Deep Globalisation Registration (Choy et al., 2020) is a robust deep-learning method that aligns 3D scans of the real world by using a 6D convolutional network to estimate the point sets correspondence and then applying the weighted Procrustes method for global optimisation. For real-time object tracking, AlignNet-3D (Groß et al., 2019) was developed, which learns the predicted frame-to-frame alignments for estimating the relative motion between 3D point clouds. The most recent developed works are PREDATOR (Huang et al., 2021), RGM (Fu et al., 2021), and POintDSC (Yuan et al., 2020).

2.3.4 Point cloud segmentation

The task of segmenting 3D point clouds is an essential step in the processing of point clouds and in the Scan-to-BIM process (Rashdi et al., 2022). The goal of the segmentation process is that of partitioning the point cloud in subsets that share common characteristics, e.g. homogeneous regions. The point regions determined in this way should be meaningful enough to be useful when analysing the scene in different ways, such as to locate and recognize objects, classify them, and extract features. This operation is a fundamental step to make machine-interpretable an implicit set of data. To avoid misunderstanding a brief clarification is necessary. The term "*Point Cloud Segmentation* (PCS)" refers to unsupervised methods used to group raw 3D points into non-overlapping regions, that correspond to specific structures or geometrical rules in the scene. The existing algorithms are mainly based on hand-crafted features derived from statistical properties and geometric constraints. Since the segmentation approaches are unsupervised, the results do not have any remarkable semantic information. The term "*Point Cloud Semantic Segmentation* (PCSS)" is widely used in computer vision, particularly in recent deep learning applications. It is also referred to as "point cloud classification" or, in some cases, "point labelling", especially in photogrammetry and remote sensing applications. Similarly to PCS, PCSS aims at partitioning the point cloud, but, differently from PCS, PCSS techniques aim at generating rich semantic information for every point of the scene. Therefore, PCSS is often implemented by using PCS algorithms as a pre-segmentation step followed by a semantic information extraction phase, or, in recent years, using directly supervised learning methods. Since the high level of semantic information needed to develop a workflow to transform point clouds into parametric objects, PCSS should be at the core of any automatic approach (Tang et al., 2010), (Volk et al., 2014). Given the core role of this task in the as-built model development, 3D point cloud semantic segmentation applied to heritage constructions is the central topic of this thesis, and the PCSS state-of-the-art techniques will be largely discussed in the Chapter 3. Instead, this paragraph will illustrate and analyse the state-of-the-art PCS algorithms, which can be grouped into four categories: edge-based, region growing, model fitting, and clustering-based.

Edge-based. Edge-based approaches were used in the early stage of PCS, and they were transferred directly from 2D images to 3D point clouds. The segmentation is carried out by detecting edges or discontinuities in the scene, and by locating points that have a rapid change in intensity or in geometrical features. The algorithms are based on a two-step procedure: (i) edge detection, where the boundaries are extracted, and (ii) grouping points, where the segmented region is defined by selecting the point inside the boundaries. For example, the authors in (Bhanu et al., 1985) developed a gradient-based method for edge detection by recognizing change in the direction of the unit normal vector on the surface, the work in (Sappa & Devy, 2001) proposed a method to extract close contours from binary edge map, or the authors in (Wani, 2003) a parallel edge-based segmentation algorithms extracting three type of edges. Edge-based algorithms are simple and fast, but they have a good performance only with straightforward scenes and with low-noise and low-density point clouds.

Region growing. Region growing is a classic PCS technique that is still widely used. It uses growing criteria, combining features between two points or two region units in order to measure the correlation among 2D pixels, 3D points, or 3D voxels, and combine them together if they are spatially nearby and have similar surface characteristics. These algorithms are composed by two main steps: (i) the selection of the seed points or seed units, and (ii) the region growing driven by determined features or principles. To develop an algorithm three factors should be taken into account: criteria, growth unit, and seed point selection. For the criteria are commonly used the normal vector (Ning et al., 2009), the distance between two points (Dong et al., 2018) or the distance of the neighbouring points to the adjusting plane (Tóvári & Pfeifer, 2005). Three options are normally considered as growth unit: single points (Rabbani et al., 2006), region units with a *K-d* tree search in raw data (Deschaud & Goulette, 2010), or hybrid units (Xiao, et al., 2013). Seed points are usually selected by designing a fitting plane for a certain point and its neighbours first, then selecting the point with the lowest residual to the fitting plane (Rabbani et al., 2006). Region growing techniques has been successfully applied in the segmentation of building plane structures (Xiao et al., 2013), (Dong et al., 2018). In order to be accurate, these algorithms need to be adjusted for different datasets based on seed growth criteria and locations. Furthermore, these algorithms are computationally demanding and may require a reduction in data volume to achieve efficiency and accuracy.

Model fitting. Model fitting approaches are normally used as shape detectors, since their purpose is to match point clouds to different geometrical shapes, such as planes, cylinders, etc. They can be used as segmentation approaches when parametric geometric shapes need to be extracted. The most used fitting methods are either based on the Hough Transform (HT) or on the RANdom Sample Consensus (RANSAC). HT is a technique introduced in (Hough, 1962) and it is composed by three main steps: (i) converting each input sample into a discretized parameter space, (ii) creating an accumulator with a cell array on the parameter space and then, for each input sample, voting for the basic geometric element of which they are included in the parameter space, and (iii) picking the cells with the local maximal score, of which parameter coordinates are used to represent a geometric segment in original space. More detailed information could be found in (Limberger et al., 2015). This method is mainly used for planes, or other basic elements segmentation (Tarsha-Kurdi et al., 2007), (Hulik et al., 2014). RANSAC techniques are very popular, and detailed information could be found in literature (S. Choi et al., 2009), (Raguram et al., 2008). The algorithm has three main phases: (i) hypothesis generation, in which N sample points are randomly chosen and a set of model parameter values are estimated using the sampled points. After N repetitions of step (i), in the (ii) hypothesis evaluation step, the most probable hypothesis among the N considered ones is determined based on a majority voting criterion. Finally, morel parameter values are estimated based om the inliers found at the previous step. RANSAC methods are very efficient, and they do not require complex optimization. They can process data with a high amount of noise or outliers, and they are widely used in building segmentation applications with remarkable results (Adam et al., 2018), (D. Chen et al., 2014), (Li et al., 2017).

Unsupervised clustering-based. It involves different methods that share a similar goal of grouping points with similar geometric features, spectral features, or spatial distributions into homogeneous patterns. Such patterns are usually not known in advance, in contrast with region-growing and model fitting methods. Hence, clustering-based algorithms can be utilized for irregular object segmentation. The main unsupervised clustering algorithms are K-means (Shahzad et al., 2012), mean shift (Shahzad et al., 2015), fuzzy clustering (Sampath & Shan, 2010), DB-Scan (Ester et al., 1996) and graph-based ones.

2.3.5 BIM modeling from point cloud

Given a reference point cloud of a structure or a building, the development of a BIM model involves three main tasks. At first, modelling the 3D geometry of the component or element, secondly, assigning the proprieties to the object, such as the category, the family, material characteristics, etc., and finally establishing relationships between the various components and elements. The aim of the geometric modelling task is to develop simplified representations of building elements by configuring geometric primitives to the point cloud data. These geometric primitives can either be surfaces or volumetric shapes. For instance, a basic wall can be modelled as a flat patch, or it could be a cube. Surfaces like carvings or moldings may not be accurately represented by a basic geometric primitive. In such cases, different modelling plans can be used. To build linear structures such as moldings, a cross-section of the object can be modelled by connecting splines to the data and then swiping the cross-section along a trajectory to form the object model (De Luca, 2006). More complex structures such as carvings, can be modelled non-parametrically, using triangle meshes or from a database of already known object models (Campbell et al., 2001). Since BIMs are normally established using solid shapes, surface-based representations must be transformed into solid models. As pointed out by (Patraucean et al., 2015) the creation of an as-built model cannot be expected to be as rich as an as-designed BIM.

Manual creation of as-built BIMs. In most of the practical applications the creation of as-built BIMs is still a largely manual operation (Fai et al., 2011), (Maietti et al., 2018).

Depending on the complexity of the building, completing a project requires several months of work by one or more skilled operators. Due to the relatively new concepts encompassed by as-built BIMs, the software supporting the Scan-to-BIM process are in continuous development, but currently they still have limited tools and functionalities. A single platform that covers all the aspects is not yet available, and while reverse engineering programs are great at creating detailed surfaces, they often lack the volumetric capability and BIM-specific features to create semantic models. On the other hand, BIM design systems have difficulty managing large data sets obtained by laser scanning. To tackle this challenge, modellers often transfer data between various programs during the modelling process, which may lead to data loss because of limitation in data exchange standards or implementation issues with the software tools.

There are two main approaches for geometric modelling. The first method involves directly fitting geometric primitives like planes, cylinders, spheres, and cones to 3D data. Many software packages include specific tools suitable for this purpose, such as those designed for modelling pipes. These tools are not automated, and they require user input and decision making. For example, a plane may be fitted to a patch of data points chosen by the user, and then extended using a region growing algorithm. However, this can lead to irregular, imprecise boundaries. To achieve more regular boundaries, multiple primitives can be intersected; for instance, a corner of a room may be formed by intersecting three orthogonal planes representing two walls and a floor. Geometric modelling may be carried out on either point clouds or polygonal (usually triangular) surface meshes. Currently, many BIM packages are unable to convert geometric primitives created via reverse engineering into BIM objects directly. It is thus common to manually re-model the geometry in the BIM environment with the reverse engineered model serving as a guide. This data transferring process between several software packages may lead to data interoperability problems. The second geometric modelling approach utilizes both cross-sections and surface extrusion. Initially, both horizontal and vertical cross-sections are taken from the data, with lines fitted to the respective cross-sections to represent walls and slabs in plan views. Next, vertical cross-sections are taken to determine the heights of walls, doors and windows in relation to the floor and ceiling. Finally, walls are modelled through extruding the horizontal cross-section vertically as per the constraints specified by the vertical cross-sections. This approach is less computationally intensive than the surface-fitting approach, yet it can lead to erroneous results when components do not comply with the idealized geometries, such as when the wall is not exactly vertical. Speeding up the modelling process can be achieved through various techniques. For example, when dealing with repeated components, such as a window, the initial model can be used as a template to generate the rest of them. However, this carries the risk of errors due to differences in geometry between components. To avoid this, prior knowledge regarding component geometry, such as the diameter of a column can be used as constraints. Alternatively, a component library with known characteristics may be used. The category of a BIM component is determined by the modeller when the object is created in the design software. Relationships between components are then established manually or in a semi-automated way. In some cases, software may automatically connect two components that are created in touching positions.

Automatic approaches for as-built BIMs. Several applications and case studies have shown that the creation of as-built BIMs needs manual, labour-intensive process that is long, tedious, and subjective, and it requires personnel with specialized training and skill. Geometric primitive modelling can be achieved rather quickly, however when it comes to modelling a complete building, it can take thousands of primitives and take months to finish for an average-sized building. The repetitive, tedious steps make this process the slowest part of the BIM creation project (Hajian et al., 2009). Though the modelling tools are complex, they are still not enough to address the uniqueness of each situation, which is why skilled personnel are often required. On top of that, due to the subjective nature of the manual work, there can be a wide range of models that could be built by different individuals. The need to optimize the as-built Building Information Modelling (BIM) process by utilizing semi-automated and automated techniques has been highlighted by several works. To further enhance the modelling process, the development of a system that takes a point cloud of any given facility as its input and creates a fully annotated as-built BIM of the same facility as its output would be the ideal framework. To reduce the manual operations, many commercial tools and algorithms have been developed in the last years. The most used ones are the Scan-to-BIM plug-ins for the Autodesk Revit environment, such as *ClearEdge3D Edgewise, IMAGINit Scan to BIM, Pointsense* and Leica *CloudWorx*. Some works proposed their own algorithms to automatize the BIM elements extraction or reconstruction. The authors in (Macher et al., 2015) proposed an approach for the 3D reconstruction of indoors of existing building from point clouds. At first, they identified walls, ceilings, and floors thanks to a segmentation algorithm, and then, in the reconstruction step of the procedure, they automatically modelled the elements into the .obj file format using planes or volumes with the assumptions of horizontal slabs and vertical walls. In a second phase, they used FreeCAD to generate the Industry Foundation Class (IFC) files starting from the .obj object, and they imported them in the BIM environment. The verification and the assessment of the procedure is carried out in (Macher et al., 2019). The work of (Thomson & Boehm, 2015) proposed the applicability to full automated reconstruction of object-based “intelligent” BIM geometry from point cloud. At first the horizontal plane representing floor and ceiling are detected using the RANSAC algorithm. Therefore, they applied a Euclidean clustering step to separate the contiguous elements out, defining a tolerance. Once that the cluster are well defined the points are projected onto the RANSAC-derived plane. To create the BIM objects, they initialized an empty related IFC element, and they used the extracted information to define the boundary, the extrusion depth, or the thickness. The authors in (Croce et al., 2021) proposed a semi-automatic approach to the 3D reconstruction of Heritage BIM from point clouds. They leverage the RANSAC shape detection algorithm as proposed by Schnabel et al., (2007), with a hierarchically structured sampling strategy for candidate shape generation. Once the raw point cloud is segmented according to the chosen categories, this method decomposes it in a set of primitive shapes with associated point sets, it randomly samples minimal subsets of points to generate candidate shapes, and then outputs the best fit primitive by means of a probabilistic score function. For more complex shapes, new parametric objects were created from scratch. Although these works proved good results in modelling simple planar elements such as walls or floors, they are still quite inconvenient when dealing with the complex objects and shapes.

2.4 Summary

In this chapter, the concept of semantic modelling has been explained and the paradigm of Building Information Modelling applied to heritage building (H-BIM) has been introduced. The main aims, the goals, and the most recent applications of the H-BIM have been illustrated and discussed. The process that leads to the creation of an as-built model is called Scan-to-BIM: it is composed by several phases, and it starts from data acquisition up to the final 3D model generation. In this chapter, the various steps of the Scan-to-BIM have been analysed in detail, focusing in particular on 3D point cloud processing. The algorithmic approaches to address each step of point cloud processing have been summarized, including point cloud acquisition, registration, down-sampling and segmentation. As turned out from the previous paragraphs, one of the most challenging phases of Scan-to-BIM is the point cloud segmentation, i.e. the process of partitioning a point cloud in subsets, where the points of each subset share common features and similar characteristics. Since the point cloud semantic segmentation problem plays a central role in this dissertation, it is comprehensively introduced and widely discussed in the next chapter.

Chapter 3

Semantic Segmentation Algorithms

In this chapter, the state-of-the-art methodologies used to address the semantic segmentation problem are reported. First, paragraph (§3.1) introduces the concepts of artificial intelligence, machine learning and deep learning, and, afterwards, the basic structure of artificial neural networks (§3.1.1) and convolution neural networks (§3.1.2) are explained in more details. Paragraph (§3.2) focuses on the semantic segmentation task, analysing and discussing and the main approaches for image (§3.2.1) and point cloud semantic segmentation (§3.2.2). Paragraph (§3.2.3) presents the existing multi-view approaches for point cloud segmentation, which are based on image segmentation and reprojection of the extracted classes or categories on the original point cloud. Finally, in the last paragraph (§3.3) the developed semantic segmentation approach is explained and discussed in detail. The proposed two-step workflow is particularly well suited for photogrammetric point clouds, for which the geometric relation between images and point cloud has already been established during the photogrammetric reconstruction. Semantic segmentation of the images used in the photogrammetric workflow is performed using the most popular convolutional neural networks, once properly trained. Then, the determined pixel-wise labels are reprojected from the images to the related photogrammetric point cloud, exploiting the already available intrinsic and extrinsic camera parameters. The chapter ends with a general summary (§3.4).

3.1 AI/Machine Learning/Deep Learning

Over the past few years, artificial intelligence (AI) has been subject of an intense media hype. Studies on machine learning (ML), deep learning (DL) and AI led to a countless number of articles, publications, and academic researches. Nowadays, they are used in a wide range of different applications, also far from each other: economics, psychology, linguistics, philosophy, music, and many others. They underlie most of the common technological applications to solve complex tasks and alleviate problems such as self-driving cars, image processing, web search, robotics, automated decision making, and many others. But firstly, what are artificial intelligence, machine learning and deep learning? How do they relate to each other? Artificial intelligence was born in the 1950s, when few pioneers form the emerging fields of computer science started asking whether computers could be able to "think". A formal definition of AI is as follows: *the ability of a computer or a robot controlled by a computer to do tasks that are usually done by humans because they require human intelligence and discernment.* AI encompasses machine learning and deep learning, but it also includes many other approaches that don't involve any learning. Machine learning is a subset of AI, and a first definition was given by Arthur Samuel in 1959: *machine learning is the field of study that gives computers the ability to learn without being explicitly programmed.* More recently Tom Mitchel (1998) defines machine learning by saying that a well-posed learning problem is defined as follows: *a computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.* While in classical programming, humans input rules and data to come out with an answer, with machine learning humans input data as well as the answer expected for such data, and the expected outcome are the relationships. These rules can then be applied to new data to produce original answers. A machine learning algorithm is *trained* with significant examples, aiming at allowing it finding the statistical structure in these examples, and hence allowing the algorithm to *learn* the rules for automating the task. This simple idea allows to solve a wide range of complex tasks, and ML has quickly become the most successful subfield of AI. Deep learning is a specific subset of machine learning

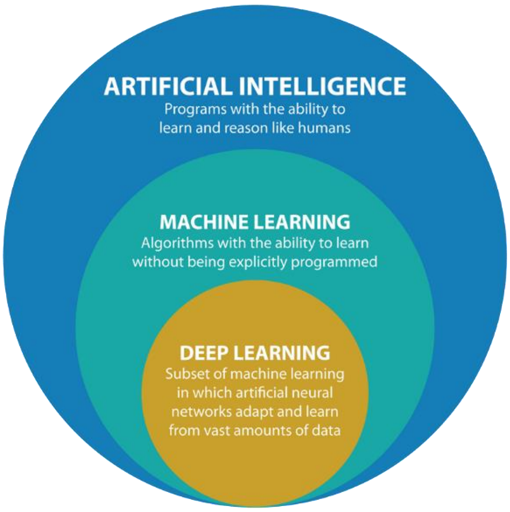

ARTIFICIAL INTELLIGENCE
Programs with the ability to learn and reason like humans
MACHINE LEARNING
Algorithms with the ability to learn without being explicitly programmed
DEEP LEARNING
Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data

Figure 3.2 – AI/Machine Learning/Deep Learning and their relationship (from datacatchup.com).

in which the learning process puts an emphasis on learning successive layers of increasingly meaningful representation. The term was introduced in the machine learning community by Rina Detcher (1986). The adjective *deep* in deep learning refers to the use of multiple layers in the model, and the term *depth* is used to define how many layers contribute to the model. Deep learning is particularly suited to contexts where the data are complex and where there are large datasets available. In modern deep learning the layered representations are learned using models called *neural networks.*

3.1.1 Artificial Neural Networks (ANNs)

Despite some learning-based models have been proposed in the first part of the 20th century (McCulloch & Pitts., 1943), (Kleene, 1956), (Rosenblatt, 1958), Artificial Neural Networks (ANNs), which are a class of artificial intelligence algorithms, emerged in the 1980s from developments in cognitive and computer science research inspired by the biological neural networks that constitute animal brains. A standard ANN is based on a set of connected basic units or nodes called artificial neurons, which aim at mimicking the neurons in a biological brain. Each neuron can send a signal to the others through the available connections, similarly to the functioning of the brain synapses. The power of an ANN to model complex relations emerges from the interactions between large sets of simple neurons. Figure 3.2 illustrates the structure of a standard neural network, in which the neurons are organized into layers.

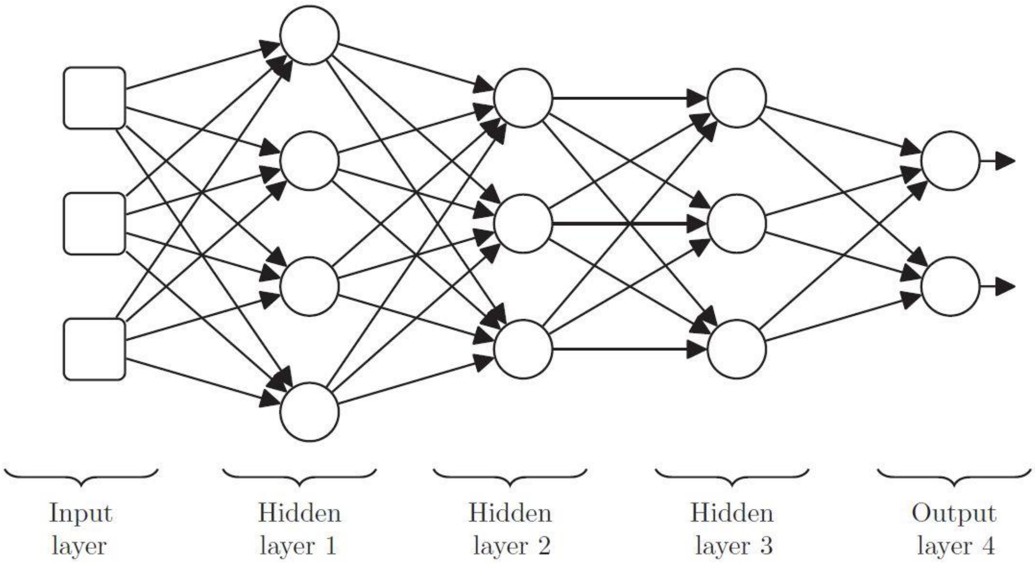


Figure 3.3 – The basic structure of an Artificial Neural Network (ANN) (from *Deep Learning*, J.D. Kelleher).

The represented network has five layers: one input layer, three hidden layers, and one output layer. Deep learning networks are neural networks that have much more than two hidden layers. The circles in the figure represent the information processing neurons in the network. Each of these neurons takes a set of numeric values as input and maps them into an output value. Each input of a processing neuron is either the output of a sensing neuron or the output of another processing neuron. The arrows show how information flows through the network from a neuron to another one and from a layer to another one. Each connection has a *weight* associated, i.e. a scalar number. Weights are very important: they affect how a neuron processes the information it receives, and their estimation is the goal of learning. Training an artificial neural network entails the pursuit of the most suitable weight values: in the process of end-to-end learning, this exploration is achieved by minimizing an objective function that assesses how well the model output aligns with the correct values. How does a neuron process the input information? A neuron implements a two-stage process to map inputs to an output. The first stage of processing involves the calculation of a weighted sum of the inputs to the neuron. Then, the result of the weighted sum calculation is passed through a second function that maps the results of the weighted sum score to the neuron's final output value. Typically, this second function is known as an *activation function.* Figure 3.3 illustrates how these stages of processing are elaborated in the structure of an artificial neuron. The symbol ∑ represent the calculation of the weighed sum, and the symbol φ represent the activation function generating the output. The neuron receives *n* inputs from *n* different connections, and each connection has an associated weight. The weighted sum is as follows:

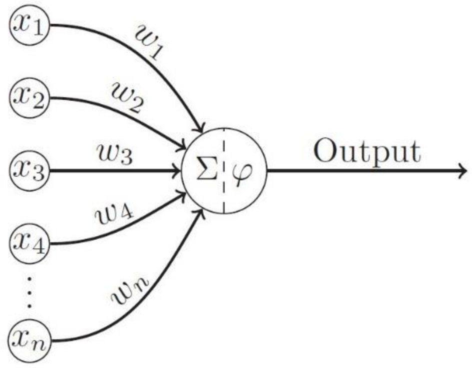


Figure 3.4 – The structure of a neuron (from *Deep Learning*, J.D. Kelleher).

$$z = (x_1 \times w_1) + (x_2 \times w_2) + \cdots + (x_n \times w_n) \tag{3.1}$$

The weighted sum value is used as input of the *activation function*, whose outcome is the final output of the neuron. Examples of the typical activation functions used in modern deep networks are shown in Figure 3.4. The most used activation functions are the *threshold*, *logistic*, *tanh*, and the *rectifier*. Currently the most used activation function is the Rectifier Linear Unit (ReLu), that showed to enable better training in modern deep networks. Such activation functions have a nonlinear behaviour: they apply a nonlinear mapping to the output of the weighted sum. This is the reason why these functions are used: the introduction of a nonlinear behaviour enables a neural network to learn more complex relations and to create more effective models. A neural network may use different activation functions, but generally all the neurons in the same layer use the same function.

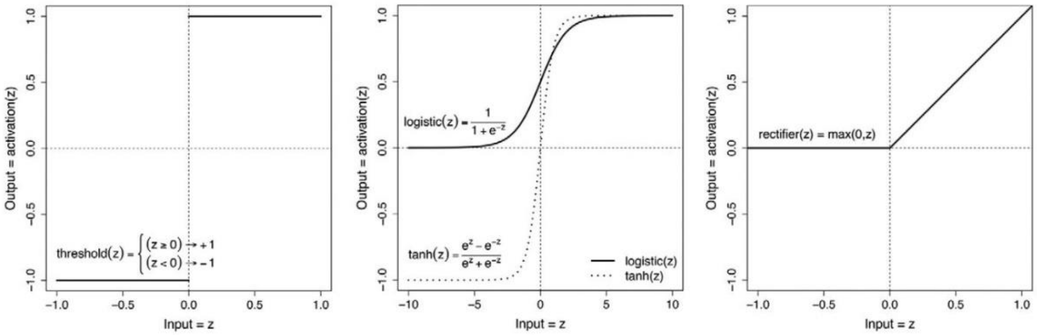


Figure 3.5 – Typical activation functions used in modern deep learning (from *Deep Learning*, J.D. Kelleher).

3.1.2 Convolutional Neural Networks (CNNs)

Convolution Neural Networks (CNNs) are a class of artificial neural networks designed to work with multidimensional inputs: they are commonly used to analyse visual imagery. Similarly to ANNs, which are inspired by the connectivity between the human brain neurons, the architecture of a CNN was inspired by the organization of the visual cortex of the human brain, in which individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. They were introduced and successfully applied for the first time for handwritten digit recognition (LeCun et al., 1989). Currently they are the most used neural networks in the computer vision domain, and they are used for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Image segmentation, Object detection etc. The basic idea of the CNN functioning is the detection of the local visual features, whose extent is limited to a small patch, a set of neighbouring pixels, in an image. CNN architecture extracts the local visual features in the early layers and combine these features to form higher-order features in the later layers. Since the precise location of a feature is often not relevant to the image processing task, feature detection must work in a translation invariant manner. This property is achieved by using weight sharing between neurons of the same layer, leading to a filter-like interpretation of the layer behaviour. A typical CNN architecture is composed by a stack of specific layers that transform the input volume into an output volume through a differentiable function. *Convolutional layer* is the core building block of a CNN. It includes a set of filters (or kernels) which contain the parameters learned during the training. Each filter convolves with the image sliding across the height and the width of the image, and at every spatial position is calculated the dot product between every element of the filter and the input. The output of this operation is called feature or activation map. The output volume is generated by stacking the feature maps of every filter along the depth dimension. After the convolution, in the *Nonlinearity layer*, a non-linear map, the activation function (often the ReLu), is applied to each single value of the feature. *Pooling layer* is a form of non-linear down-sampling, and it serves to progressively reduce the spatial size of the representation, to avoid the curse of dimensionality, to reduce the amount of computation and to control overfitting. There are several functions to perform the down-sampling, *max pooling*, which returns the maximum value, *average pooling,* which returns the average, or *RoI pooling,* in which the input rectangle is a parameter.

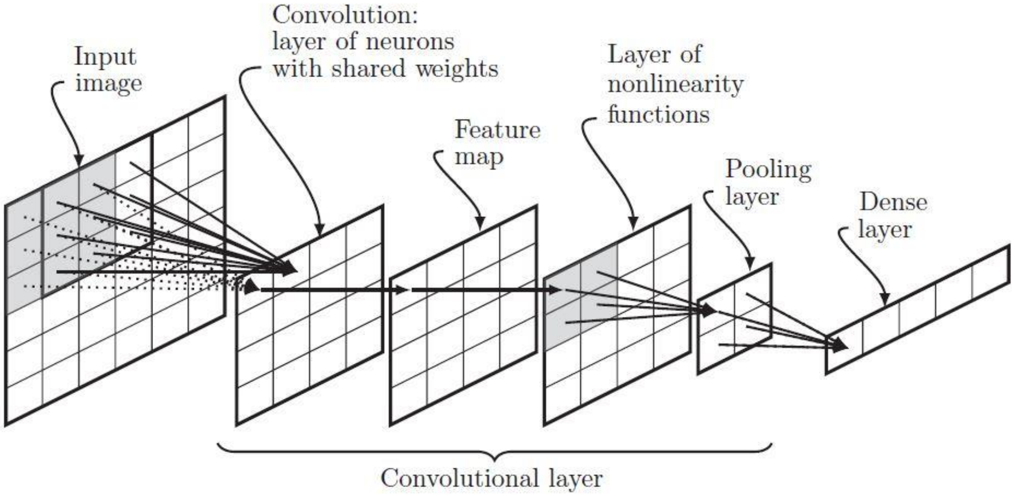

Input image
Convolution: layer of neurons with shared weights
Feature map
Layer of nonlinearity functions
Pooling layer
Dense layer
Convolutional layer

Figure 3.6 – The sequence of a basic convolutional layer (from *Deep Learning*, J.D. Kelleher).

This sequence of layers is relatively common across most CNNs, and they define a complete Convolutional layer, showed in Figure 3.5. At the end of the CNN, a *Dense layer,* that operates like a standard layer of a fully connected network, is usually used. Nowadays there are a lot of different CNN architectures. One of the most well-known CNN classification architectures is AlexNet (Krizhevsky et al., 2012). It has a depth of 8 layers, and it is composed by five convolutional layers followed by three fully connected layers. The first layer has 96 different kernels, the second layer 256 kernels and the last layer 384 kernels. In total AlexNet has 650,000 neurons and sixty million weights, but the sharing of the weights reduces the parameters to learn. VGG-16 and VGG-19 (Simonyan & Zisserman, 2014) are two other popular networks that have respectively the depth of 16 and 19 layers, and 138 and 144 million weights. In 2015 Microsoft developers introduce ResNet (K. He et al., 2015), and the technique of the skip-connections. A skip-connection feeds the output of one layer directly into a deeper layer in the network, and it allows to train very deep networks and learning fewer parameters. The deepest version of ResNet has a depth of 152 layers and it allows to learn very complex relationships.

3.2 Existing semantic segmentation methods

3.2.1 Image semantic segmentation

Image semantic segmentation is a key topic in many computer vision applications and a fundamental component in many visual understanding systems. In the last years, numerous algorithms have been developed in literature to address the task of image segmentation. The earlier methods were based on thresholding (Saleh Al-amri & Kalyankar, 2010), region-growing (Nock & Nielsen, 2004), K-means clustering (Dhanachandra et al., 2015). More advanced methods were based on active contours (Kass & Witkin, 1988), graph cuts (Boykov et al., 2001), conditional and Markov random fields (Plath et al., 2009) and sparsity-based methods (Starck et al., 2005). Over the last years deep learning methods have yielded a new generation of segmentation methods, with an impressive performance improvement, overcoming the results of the former approaches.

The first family of deep models for image semantic segmentation is based on *fully convolutional networks* (FCN). The first work was proposed by (Long et al., 2014). It is considered a milestone and it is the first work that introduced an end-to-end workflow to segment images of different size. FCN produces the segmentation map of the same size of the input by replacing the final fully connected layers with the fully convolutional layers, and by changing the classification scores output by a base network like VGG or ResNet with the segmentation map. Despite its effectiveness FCN showed some disadvantages: it does not consider the global context information, and it is not fast enough for real-time inference. To face these issues the authors in (W. Liu et al., 2015) proposed ParseNet. The model adds an extra global context to FCNs by using the average feature for a layer to augment the features at each location. The feature map is then pooled over the whole image resulting in a context vector, then normalized and unpooled to produce the new feature map.

Another popular family of models is based on *encoder-decoder* architecture. They were introduced by (Noh et al., 2015) with the transposed convolution. Their model is composed by two parts: an encoder, based on the VGG-16 convolutional network, and a deconvolutional network that starting from the feature vector generates a map of pixel-wise class probabilities. The deconvolution network is composed by the deconvolution operation and the unpooling layers, which predict the segmentation mask from the pixel-wise class labels. A well-known encoder-decoder network initially proposed for the segmentation of medical images is U-Net (Ronneberger et al., 2015). Its architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. The up-sampling step uses up-convolution, reducing the number of feature maps while increasing their dimensions. The maps from the down-sampling part are copied to the up-sampling part to avoid the loss of pattern information. To improve its performance various extensions have been developed, for example nested U-Net (Z. Zhou et al., 2018) or U-Net for 3D images (Çiçek et al., 2016). Another popular network is SegNet (Badrinarayanan et al., 2017). The main novelties are that this network has no fully connected layers, and the manner the decoder up-samples its lower resolution input features map: it uses pooling indices calculated in the max-pooling step of the corresponding encoder to perform non-linear up-sampling. This decreases the computing time because learning parameters is not needed in the up-sample operation. An interesting network for 3D image segmentation is V-Net (Milletari et al., 2016) in which a new objective function based on the Dice score (see eq. 5.7) was introduced, allowing the model to face situations in which there is an unequal distribution of pixels or regions among different classes or categories in the training dataset, for example between the background and the foreground. This situation in normally defined as class imbalance.

One of the currently most used family of segmentation models is based on the *dilated convolution* or *"atrous" convolution.* These models introduce the dilatation rate, defined as $y_i = \sum_{k=1}^{K} K\, x[i + rk]w[k]$, where *r* in the dilatation rate that defines a spacing between the weights and the kernel *w*. Numerous recent models used this technique, and the most important ones belong to the DeepLab family, e.g. Deeplabv1 (L.-C. Chen et al., 2014) and DeepLabv2 (L.-C. Chen et al., 2016). In this kind of model three stages can be distinguished: first, the dilated convolution is used to address the problem of the decreasing resolution caused by pooling and striding. Secondly, Atrous Spatial Pyramid Pooling (ASPP), which searches initial convolutional feature layer with filters at multiple sampling rates, thus capturing objects as well as image context at multiple scales to robustly segment object at multiple scales. Finally by combining probabilistic graphic models and deep CNNs the localization of the object boundaries can be improved. These models reached the state-of-the-art performance on the 2012

PASCAL VOC challenge and on the Cityscape challenge. Two improvements have been done recently. Deeplabv3 (L.-C. Chen et al., 2017), that combines cascaded and parallel modules of dilated convolutions, and Deeplabv3+ (L.-C. Chen et al., 2018) that uses an encoder-decoder architecture, including separable convolution, a spatial convolution for each channel of the input (depth-wise convolution) and then a 1x1 convolution on the output (pointwise convolution). The model extends DeepLabv3 by adding a simple decoder module to refine the segmentation boundaries. This model obtained an 89.0% mean Intersection over Union (mIoU) (see eq. 5.9) on the 2012 PASCAL VOC challenge, currently the best performance ever achieved.

Another class of models is based on *multi-scale analysis*. One of the most remarkable architectures is the Feature Pyramid Network (FPN) developed in (T.-Y. Lin et al., 2016) for object detection, but it can also be applied for segmentation. The model is composed of a bottom-up pathway, a top-down pathway, and lateral connections that are used to merge the high and low-resolution features output by a pyramidal hierarchy of deep CNNs. Finally, two multi-layer perceptrons (MLPs) are used to generate the segmentation masks. In this work (Zhao et al., 2017), the authors proposed Pyramid Scene Parsing Network (PSPN), a multi-scale algorithm to better learn the global context representation of a scene by extracting several patterns from the input image using a residual network (ResNet) and a dilated network as feature extractor. To distinguish patterns of different scales the extracted features are then fed into a pyramid pooling module that works on four scales, each one corresponding to a pyramid level. Each level processes the features by means of a 1x1 convolutional layer to decrease the dimension. Finally, is generated the pixel-wise prediction by up-sampling and concatenating the output with the initial feature maps to capture the local and the global features.

The last class of models is based on *regional convolution network* (R-CNN), originally used for object detection, but applied with success also for instance segmentation. One of the most successful is Faster R-CNN (Ren et al., 2015) that uses a regional proposal network (RPN) to propose the object bounding box. It extracts the Region of Interest (RoI) and RoIPool layer computes the features from these proposals to assign the coordinates of the bounding box and the object class. Mask R-CNN (K. He et al., 2017) is an extension of this model and it is the state-of-the-art network for instance segmentation. The model detects objects in an image with bounding boxes and simultaneously it creates a high-quality segmentation mask. It is composed by three branches: the first detect the bounding boxes, the second the associated classes and the third computes the binary mask. The loss function of the algorithm combines all the three losses and trains all of them jointly. Many other algorithms for image semantic segmentation have been developed in literature, and other categories of model could be group in *graphical based, recurrent neural network based, attention-based generative models, adversarial training,* and *active control based* (Ulku & Akagunduz, 2019).

3.2.2 Point cloud semantic segmentation

3D point cloud semantic segmentation (PCSS) is attracting increasing interest due to its applicability in a wide range of different applications (Xie et al., 2020). Despite the term semantic segmentation is widely used in computer vision, in photogrammetry and remote sensing applications, the following nomenclature is also often used for similar purposes: "point cloud classification" or "point labelling" (Boulch, Le Saux & Audebert, 2017). Given a point cloud, the goal of semantic segmentation is to partition it into several subsets according to the semantic meaning of the points. Artificial Intelligence (AI), in particular the branch of machine learning, has become the basic building block for these tasks and nowadays PCSS is usually realized by supervised learning methods, including "regular" supervised machine learning (A), and deep learning (B) (Guo et al., 2019).

A) Regular Supervised Machine Learning

Regular supervised machine learning methods for semantic segmentation of point clouds can be divided into two main groups (Weinmann et at., 2015):

- Individual PCSS methods, which classify each point based only on its individual features. Four stages can usually be identified in these methods: neighbouring selection, feature extraction, feature selection, and semantic segmentation. These methods are usually computationally efficient, but their results are often affected by a significant level of noise. The most used classifiers in PCSS methods are Random Forest, AdaBoost, Support Vector Machine.

- Methods based on statistical contextual models, which focus on point cloud statistics and relational information over different scales. Differently from individual PCSS they take into account contextual features. The most widely used model in this category is Conditional Random Fields (CRF) (Weinmann, 2014; Vosselman, Coenen & Rottensteiner, 2017).

B) Deep Learning

Deep learning has been recently successfully used on several 2D vision problems, becoming more and more popular during the last years, in particular after the introduction of Convolutional Neural Networks (CNN) (He et al., 2016), and nowadays it can be considered as a predominating technique in AI. Given the typical high performance of deep learning-based solutions, there is an increasing interest of the civil engineering sector on the extension of the use of such techniques also to data related to construction and building models. For instance, they may be used to extract information from 3D point clouds, for 3D Shape classification, 3D Object Detection, 3D Object Tracking, and 3D Point Cloud Segmentation (L.-C. Chen et al., 2018), (Shelhamer & Darrell, 2015). Three-dimensional data provide richer spatial and geometrical information compared to two-dimensional data and could better characterize complex scenes. However, the use of deep learning methods on point clouds still faces several significant challenges, due for instance to:

- the large data size, which implies long computing time.
- the unstructured nature of 3D point clouds, which complicates the use of network architectures commonly used for 2D data,
- the unavailability of large-shared datasets, which makes the results of the training process hardly exportable to scenarios different from the one that motivated the network realization.

According to the literature, semantic segmentation methods for 3D point cloud can be divided into two groups: (i) projection-based methods and (ii) point-based methods (J. Zhang et al., 2019), which are going to be described in the following.

Projection-based methods. The main issues to be solved for using standard neural networks, such as Convolutional Neural Networks (CNNs) or Fully Connected Layers (FCs), are the unstructured nature of point clouds and to the presence of orderless data. To this aim, projection-based methods first apply a transformation to convert 3D point clouds on data with regular structure, then they perform the semantic segmentation task by applying standard approaches, and finally they re-project the extracted features on the original shape or point cloud (Lawin et al., 2017a). The advantage of projection-based methods is that they leverage on well-established networks. However, any kind of transformation and intermediate representation involves inevitably a loss of information, in particular geometrical and spatial. Depending on the type of used representation, it is possible to distinguish four categories among these methods: a) multi-view, b) volumetric, c) spherical, and d) lattice.

a) Multi-view representation. These methods first project the 3D shape into multiple views, then apply 2D image segmentation methods to extract information from each image. The results obtained on such images are compared and analysed, and eventually re-projected on the original scene to obtain a semantically segmented point cloud. How to aggregate the multiple views in a global representation is still a key challenge for this method. MVCNN (Su et al., 2015) is a pioneering work, which proposed the use of Convolutional Neural Networks (CNN) with multiple perspective of the 3D object. It is suitable for individual objects rather than complex scenes because it ignores spatial relations between objects. Another important work is SnapNet (Boulch et al., 2018), that, in order to address the problem of information loss, selects some snapshots of the point clouds to generate RGB and depth images, and then it uses the marked points to project the segmentation on the 3D cloud. The more recent SnapNet-R improves the process of image generation and the overall accuracy. These networks ensure excellent image segmentation results, but the transposition of such results on the 3D cloud entails a large loss of spatial and geometrical information.

b) Volumetric representation. Volumetric representation or voxelization of point clouds consists in the transformation of the unstructured 3D cloud into a regular spatial grid, and then the information distributed on such regular grid is exploited to train a quite standard neural network to properly perform the segmentation task. VoxNet (Daniel Maturana, 2015) converts the 3D clouds in a grid in which CNN operations can be applied and use CNN to predict the classes directly on the order grid. PointGrid uses the same transformation of VoxNet, but it addresses the problem of information loss and change of scale, and it has less memory requirements. SEGCloud (Tchampi et al., 2017), to reduce the computational cost has introduced the methods of spatial partition such as K-d tree or Octree. In conclusion, the mentioned methods and others like OctNet (Riegler et al., 2017), VV-Net (Qi et al., 2016), ScanComplete (Dai et al., 2018) ensure the achievement of a reasonable segmentation of non-structured relatively small point clouds. Unfortunately, they are still unsuitable for the semantic segmentation of complex scenarios.

c) Spherical representation. Spherical representation typically refers to a mathematical representation of data on a sphere surface or in a spherical coordinate system. This representation is often used in tasks related to spherical data, such as 360-degree images, 3D point clouds, or orientation data. These types of representation, compared with the multi-view representation, retain more geometrical and spatial information. However, they have some issues such as discretization errors and occlusion. The most important works are SqueezeNet (Iandola et al., 2016; Milioto et al., 2019), and RangeNet++ (Milioto et al., 2019) for real-time LiDAR data semantic segmentation.

d) Lattice representation. Volumetric representation is naturally sparse, and it is inefficient to apply dense convolutional neural networks (DCNN) on spatially sparse data. Lattice representation converts a point cloud into discrete representation such as sparse permutohedral lattice (A. Adams et al., 2010). This method can control the sparsity of the extracted features and reduces memory requirements and computational costs reducing the convolution output. One of the main works is SPLATNet (Su et al., 2018). It interpolates a raw point cloud to a sparse lattice and then a Bilateral Convolutional Layers (BCL) is applied to convolve on occupied parts of lattice. Other works are LatticeNet (Alexandru Rosu et al., 2020), which achieves efficient processing of large point clouds, and MinkowskiNet (Choy, Gwak & Savarese, 2018), a 4D spatio-temporal convolutional neural network for 3D video perception.

Point-based methods. Point-based methods, or direct methods, work directly with point clouds and they do not introduce explicit information loss with intermediate representations. This direct approach leverage on the full use of the characteristic of the raw point cloud data and consider all the geometrical and spatial information. Despite point-based methods are still in development, they seem the most promising in the future and a series of networks have been proposed recently. Overall, these methods can be divided into four categories: a) pointwise MLP methods, b) convolution methods, c) RNN-based methods and d) graph-based methods.

a) Pointwise methods. These methods usually use shared Multi-Layer Perceptron (MLP) as the basic unit in their network. The pioneering work for this method is PointNet (Qi et al., 2017a), it learns per-point features using shared MLPs and global features using symmetrical polling functions. However, MLP cannot capture local geometry in mutual interaction between points. In order to capture wider context and learn more local structures, a lot of network based on PointNet have been developed recently. These methods are based on neighbouring feature pooling such as PointNet++ (Qi et al., 2017b), PointSIFT, PointWeb, RandLA-Neton attention-based aggregation such as Gumbel Subset Sampling (GSS) or Local Spatial Aware (LSA), and on local-global concatenation such as EdgeConv and NetVLAD.

b) Convolution methods. These methods tend to propose effective convolution operators for point clouds (Hua, Tran & Yeung, 2018). PointCNN (Wang et al., 2018) is a network based on parametric continuous convolution layers and kernel function of this layer is parametrized by MLPs. KP-FCNN is based on Kernel Point Convolution (KPConv), and the convolution weights are determined by the Euclidean distances to kernel points, and the number of kernel point is not fixed. ConvPoint proposed a point-wise convolution operator, where the neighbouring points are binned into kernel cells and then convolved with kernel weights.

c) RNN-based methods. Recently Recurrent Neural Network (RNN) have been used for semantic segmentation, in particular to capture inherent context features from point clouds. G+RCU first transformed a block of points into multi-scale blocks and grid blocks to obtain input-level context. Then, the block wise features extracted by

PointNet are sequentially fed into Consolidation Units (CU) or Recurrent Consolidation Units (RCU) to obtain output-level context. 3DCNN-RNN (F. Liu et al., 2017) first learns spatial distribution and colour features using a 3D CNN, and then the final concatenated feature vector is fed into a residual RNN to obtain the final segmentation. However, these methods lose geometric features and density distribution from point clouds when aggregating the local neighbouring features with global structures.

d) Graph-based methods. To improve the results and capture richer geometrical structures several methods leverage on graph networks. Graph Neural Network (GNN) is a type of Neural Network which directly operates on the Graph structure. The most important works are DGCNN (Wang et al., 2018), PyramNet based on Graph Embedding Module (GEM), and GACNet.

Finally, a summary of the main network architectures with their typology, year, and accuracy (mIoU) on the ModelNet40 dataset (Fig. 3.6).

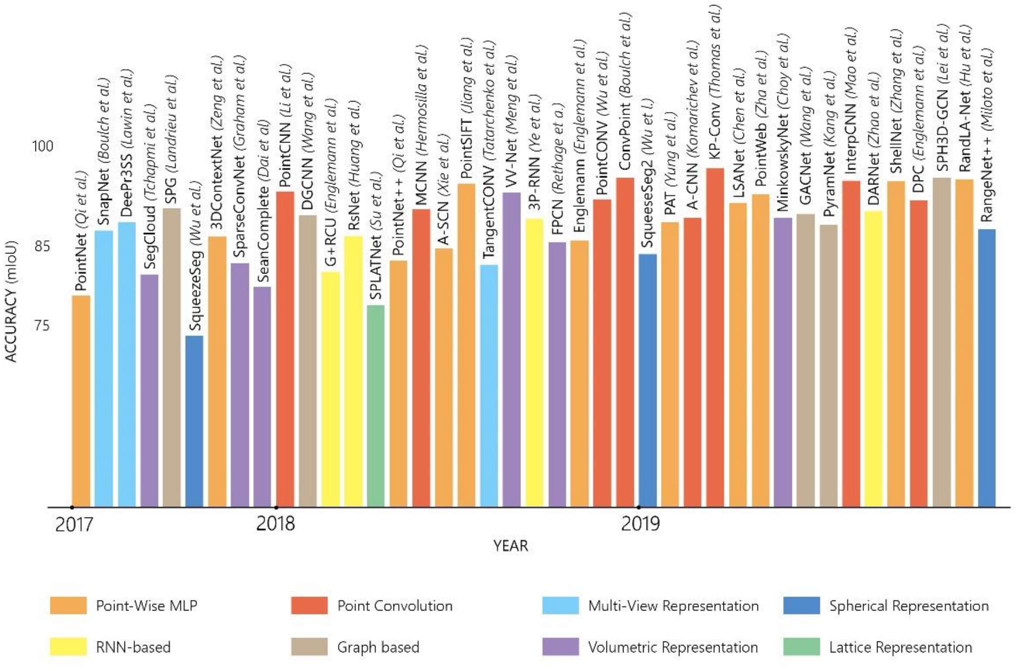


Figure 3.7 – Main network architectures with typology, year and mIoU on ModelNet40.

3.2.3 Multiview approaches

Multiview approaches are a class of projection-based algorithms that use a set of images as intermediate representation of the 3D object, shape or point cloud. These models could be used for different tasks, 3D shape classification, 3D object detection & tracking or 3D point cloud semantic segmentation. For shape classification they extract view-wise features and then fuse these features into a discriminative global representation. For object detection these models fuse proposal-wise features from different view maps to obtain a 3D rotated box. For semantic segmentation they extract a feature pixel-wise map for each image and then they aggregate the map information on the initial 3D representation by calculating the relationship between the 2D pixels and the 3D space. Despite the use of an intermediate representation could introduce a geometrical, spatial, and dimensional information loss on the 3D shape or point cloud, these approaches have shown remarkable results, and they are an effective strategy to deal with 3D semantic segmentation. At first, they allow to exploit the standard 2D segmentation architectures, and to leverage on the greater simplicity and clarity of image-based algorithms. As shown in the previous paragraph there are several available architectures, and the state-of-the-art networks have reached remarkable performance on different data typologies. Secondly, they benefit from the availability of several datasets and benchmarks for image semantic segmentation, enabling the use of pre-trained networks, the use of transfer learning, and reducing the training time, the computing power required and the craving of training data. The most challenging step of these approaches is the re-projection or the transfer of the 2D features on the 3D object. In most cases, label transferring results are affected by obstructions or occlusions, and by the low quality of the 2D segmentation boundaries, resulting in certain cases in a weak 3D segmentation quality, which can decrease the 2D segmentation performance. Depending on the application and the available input data, several techniques have been tested to address these issues, and they differ according to the strategy used to connect the 2D environment with the 3D space: for example, some methods exploit depth information or intrinsic and extrinsic camera parameters, other methods leverage on Bayesian updates and close pairwise Conditional Random Fields (CFRs), others on graph-based approaches. In the last year several methods have been proposed in literature, and this section will provide an exhaustive summary of the main developed ones, both for semantic segmentation (A) and 3D object detection and recognition (B).

A) Semantic Segmentation

Some of the pioneering works dealing with a multi-view approach are based on join segmentation, the simultaneous segmentation of registered 2D images and 3D points reconstructed from multiple view images. In their work (Xiao et al., 2007) treat the segmentation as a two-stage weighted graph labelling problem: first they constructed a graph for the joint 3D and 2D points, and then used a hierarchical sparse affinity propagation algorithm to segment 2D images and group 3D points. (Quan et al., 2007) revisited the quasi-dense approach to structure from motion, and they proposed a probabilistic framework for the joint segmentation of 3D points and 2D pixels into groups of meaningful objects. In their work, (Xiao & Quan, 2009) proposed a multi-view semantic segmentation framework for labelling street images captured by a camera mounted on a car. They used Structure for Motion to reconstruct the scene geometry across multiple views, and, with both 2D and 3D information available, they exploited a Markov Random Field (MFR) representing superpixels as nodes, and the smoothness across superpixels as edges. For smoothness terms, they make use of colour differences to identify accurate segmentation boundaries, and dense pixel-to-pixel correspondences to enforce consistency across different views. They tested the procedure on a manually segmented dataset achieving a high overall accuracy, but an unsatisfactory performance on small objects. The authors in (R. Wang et al., 2010) developed an automatic method to segment building outlines from multiple city-scale street view images. Different from joint segmentation this approach makes individual image segmentation obtained with a graph-cut-based algorithm consistent across views, by re-projecting the features from the neighbouring views using the 3D information and voting to find conflicting results. The proposed method is robust and efficient, and it allows precise pixel labelling despite inaccuracies of the 3D models and misalignments in the data. In (Hermans et al., 2014a) the authors addressed the problem of image sequences segmentation performing an efficient 2D semantic segmentation of RGB-D frames based on Randomized Decision Forests (RDFs) and then developed a novel way to transfer the 2D image labels into the 3D point cloud based on Bayesian updates and dense pairwise Conditional Random Field (CFRs) that allows to enforce temporal and spatial constraints. (Riemenschneider et al., 2014) proposed an alternative approach that exploits the geometry of a 3D mesh model obtained from multi-view reconstruction. Instead of clustering similar views, they predict the best view before the actual labelling reducing the inherent data overlapping. For this, they find the single image part that best supports the correct semantic labelling of each face of the underlying 3D mesh. (Vineet et al., 2015) presents a framework for real-time dense large-scale reconstruction and semantic segmentation. The segmentation pipeline extracts 2D features from stereo images based on random forest classifier, and it transfers the predictions into 3D volume, where they define a densely connected CFR. The authors in (Pan & Taubin, 2016) proposed a graph-cut based method for segmenting point clouds from multi-view reconstruction removing the unwanted background points. The method is based on the observation that the objects of interest are usually located in central area of the image. The segmentation is carried out in two steps: first they built a weighted graph, whose nodes represent points and edges that connect each point to its k-nearest neighbours. Secondly, graph-cut is used to find the initial binary segmentation, and then it is refined with Gaussian mixture models (GMMs) using colour and density information. An interesting work is SemanticFusion (McCormac et al., 2016a), a pipeline for mapping RGB-D video. Firstly, a Simultaneous Localization and Mapping (SLAM) system provides the correspondences between the frames and a globally consistent map of fused *surfels*. Secondly, a CNN receives the 2D images and returns a set of per pixel class probabilities, and finally a Bayesian scheme updates the probabilities according to the correspondences of the SLAM system. In (Boulch et al., 2017), the authors introduce SnapNet, a framework which applies CNNs on multiple snapshots of the point cloud. It is composed by three core ideas: firstly, they generate two type of snapshots, RGB views and depth views containing geometric features. Secondly, for each snapshot, a fully convolutional network (FCN) is used to obtain a pixel-wise labelling. Finally, they perform fast back-projection using efficient buffering to label 3D points. (Hazirbas et al., 2017) proposed FuseNet investigating a solution on how to incorporate complementary depth information into a semantic segmentation framework by making use of convolutional neural networks (CNNs). They propose an encoder-decoder network type, where the encoder part is composed of two branches of networks that simultaneously extract features from RGB and depth images and fuse depth features into the RGB feature maps as the network goes deeper. In (L. Ma et al., 2017), the authors proposed a deep neural network approach to predict semantic segmentation from RGB-D sequences. The key innovation is to enforce consistency by warping CNN feature maps from multiple views into a common reference view using the SLAM trajectory and to supervise training at multiple scale. The network is inspired by FuseNet and it contains two branches to learn features from RGB and depth. The feature maps from depth are fused into the RGB branch at each scale. The authors in (Jaritz et al., 2019) propose Multi-View PointNet (MVPNet) where they aggregated 2D multi-view image features, calculated with an encoder-decoder network, into 3D point clouds using the camera intrinsics and poses. Complementary 3D geometry and 2D image features are fused in 3D canonical space using PointNet++, which predicts the final semantic labels. In (Lawin et al., 2017b), the authors proposed a framework for 3D semantic segmentation that exploits the advantages of deep image segmentation approaches. The point cloud colour, depth and normal are first projected onto a set of synthetic images, which are then used as input to the deep network. The resulting pixel-wise segmentation scores are re-projected into the point cloud according to the intrinsic and extrinsic virtual camera information. In their work (Dai et al., 2018) the authors present 3DMV a novel method for 3D semantic scene segmentation of RGB-D scans in indoor environments using a joint 3D multi-view prediction network. The network is composed of a 3D stream and several 2D streams that are combined in a joint 2D-3D network architecture. The 3D part takes as input a volumetric grid representing the geometry of a 3D scan, and the 2D stream stake as input the associated RGB images that are aligned with respect to their world coordinate system. In this contribution (Antonello et al., 2018) propose a batch approach and a novel multi-view fusion technique to exploit multiple views for improving the semantic labelling results. The batch approaches rely on 3D Entangled Forest Classifier (3DEF), an extension of Random Forest that is able to model complex contextual features, firstly over-segmenting the scene in a way that each segment contains one object, and secondly classifying each segment by means of the 3DEF depending on geometric relationship. In this work the approach is improved with a novel multi-view frame fusion technique at the end of the workflow which improves the semantic segmentation of single-frames and allows the creation of accurate semantic maps. (Kundu et al., 2020) propose an approach for the semantic segmentation of meshes based on multiple synthetic images. At first a 2D CNN model is used to make the prediction on the images, and then the features are fused on 3D mesh vertices. To project the 2D labelling to 3D, they rendered a depth channel for each view, and they accumulated the image feature only if depth of the pixel matched the point-to-camera distance. This paper introduces several new ideas that significantly improve labelling performance: virtual views with additional channels, back-face culling, wide field-of-view, multiscale aware view sampling. As a result, it overcomes the 2D-3D misalignment, occlusion, narrow view, and scale invariance issues that have vexed most previous multiview fusion approaches. In this paper (Gerdzhev et al., 2021), they introduce TORNADO-Net, a neural network for 3D LiDAR point cloud semantic segmentation. They incorporate a multi-view (bird-eye and range) projection feature extraction with an encoder-decoder ResNet architecture with a novel diamond context block. To better utilize the local neighbourhood information and reduce noisy predictions, they introduce a combination of Total Variation, Lovász-Softmax, and Weighted Cross-Entropy losses. One of the more recent works that gained the new state-of-the-art for large-scale indoor/outdoor semantic segmentation on S3DIS and KITTI-360 datasets is proposed in (Robert et al., 2022). They proposed a multi-view aggregation model for the semantic segmentation of 3D scenes that uses an attention-based scheme to select and merge the most significant 2D features. The method starts computing an occlusion mapping between pixels and points, and then uses viewing conditions through an attention scheme to aggregate relevant image features for each 3D point. This approach allows to learn both from point clouds and images in an end-to-end workflow. (Wang et al., 2022) proposed FSDCNet, a multiview 3D point cloud classification methods based on dynamic and static convolution fusion neural network. It devises a view selection method with fixed and random viewpoints, and a local feature extraction operator of dynamic and static convolution adaptive weight fusion was designed to improve the model adaptability. Compared with other methods it achieved state-of-the-art classification score on ModelNet40 and Sydney Urban Object datasets, and the results demonstrated that it could extract fine-grained detailed information, and it is suitable also for sparse point clouds with noise and local block defects. Like other CNNs based models, it requires a large dataset to support training and testing.

B) Object Detection and Recognition

One of the pioneering and most popular works in this class is MVCNN, a method proposed by Su et al., (2015) to recognize 3D shapes using multiple perspectives of the object. They used at first a standard CNN to extract features for each image, and then a pooling layer to aggregate the features from different perspective. The aggregated features are then input in a second CNN for processing and receiving the final classification or segmentation result. This approach achieved efficiency, compactness and a high performance compared with existing methods, but is suitable only for individual shapes because it ignores the spatial relationship between objects. (Qi et al., 2016) improved the MVCNN introducing a multi-resolution extension to capture information at multiple scales and performing a sphere rendering at different volume resolution. In their work (Pang & Neumann, 2016) used a multiview framework to perform object detection on 3D point clouds. They transformed the 3D representation in a set of multiple images, and they exploited Convolutional Neural Networks, that can easily handle all viewpoints and rotations for the same class, to predict the object class. Lastly, all 2D detection results are re-projected back into 3D space for a fused 3D object location estimation based on depth information. (X. Chen et al., 2016) proposed Multi View 3D networks (MV3D), a sensory-fusion framework that uses both LiDAR point cloud and RGB images as input and predicts 3D bounding box for autonomous driving applications. The network takes three inputs, the bird's eye view, the front view of point cloud and the images. It first generates 3D object proposals from bird's eye view map and project them to the three views. A deep fusion network is used to combine region-wise features obtained via ROI pooling for each view. The fused features are used to jointly predict object class and do oriented 3D box regression. (Papadakis, 2017) presented a study targeting the application of multi-view hypothesis fusion scheme for the purpose of 3D object classification. For this purpose, they benchmarked a number of schemes for hypothesis fusion under different environment assumptions and observation capacities. Their experimental results highlighted significant aspects that should be considered in the design of a multiview-based recognition pipeline for 3D shape detection. By exploiting the relationship between polynomial kernel and bilinear pooling (T. Yu et al., 2018) obtained an effective 3D object recognition by aggregating local convolutional features through bilinear pooling. They harmonize different components inherited in the bilinear feature to obtain a more discriminative representation. To achieve an end-to-end trainable framework, they incorporate the harmonized bilinear pooling as a layer of a network, constituting the proposed Multi-view Harmonized Bi-linear Network (MHBN). (N. Qin et al., 2018) presented a novel deep learning framework for 3D terrain scene recognition using 2D representation of point cloud. It is composed by two key components: Initially, several suitable discriminative low-level local features are extracted from airborne laser scanning point cloud, and 3D terrain scene is encoded into multi-view and multimodal 2D representation. Secondly, A two-level fusion network embedded with feature and decision-level fusion strategy is designed to fully exploit the 2D representation of 3D terrain scene, which can be trained end-to-end. (C. Wang et al., 2019) improved the view-based strategies for 3D object recognition introducing a view clustering and pooling layer based on dominant sets. The pooled feature vectors are then fed as inputs to the same layer. In addition to the grey-scale representations the model uses also depth and surface information. This module, once inserted in the pretrained CNN, boosted the performance achieving a new state of the art accuracy on ModelNet40 databased. (Liu et al., 2019) proposes a multi-view hierarchical fusion network (MVHFN) for retrieval and classification of 3D object exploiting the relevance and discrimination among multiple view. This approach consists primarily of two essential modules. The initial module, focused on visual feature learning, employs 2D CNNs to extract visual features from multiple views generated around the specific 3D object. Subsequently, they employ the multi-view hierarchical fusion module that they have developed to combine these multiple view features into a concise descriptor. This module can fully exploit the relevance among multiple views by aggregating the view features in the same cluster and discover the content discrimination by learning information of the cluster-level features. (Q. Yu et al., 2020) developed a novel network called Latent-MVCNN that recognize 3D shape using multiple view-images from pre-defined or random viewpoints, overcoming the difficult to make prediction with a small number of images. It is composed by three types of CNNs: the first one outputs the category probability, the second one outputs a latent vector, the third one outputs the transition probabilities between the views. LMVCNN performs well and remains competitive with other related methods for the pre-defined and random viewpoints and achieves a promising performance when the number of view-images is quite small.

3.3 Proposed segmentation workflow

In this paragraph the developed methodology used to address the problem of semantic segmentation of heritage building point clouds is illustrated. The procedure is based on a deep learning multi-view approach workflow, in which the segmentation is carried out at first on an intermediate image representation of the cloud, and then the extracted features are projected on the original point cloud. As already mentioned in the previous paragraph, working directly with the 3D point cloud could provide an opportunity for a better understanding of spatial and geometrical information. However, the choice to leverage on a multi-view approach can be an effective strategy. On one hand, it allows to exploit the existing models and networks for image segmentation, in particular the CNNs, that in recent years have reached remarkable results. On the other hand, the proposed procedure could be integrated in the standard photogrammetric pipeline, since it uses a set of images as input for the creation of a dense point cloud. Hence, it allows to develop an automatic workflow for the creation of a directly segmented clouds starting from the images acquired for the photogrammetric reconstruction. In addition, at this time, a multiview approach on heritage data has never been tested, and it is interesting to explore that approach. The segmentation workflow is shown in Figure 3.7, and the main steps of the procedure are five: (1) the photogrammetric survey, (2) the camera calibration and parameters estimation, (3) the dense cloud construction and preparation, (4) the semantic segmentation of all the images of the photogrammetric survey, used to create the related dense point cloud, and (6) the projection of the extracted 2D labels output by the segmentation system on the 3D reconstruction. The steps are comprehensively described in the following paragraphs.

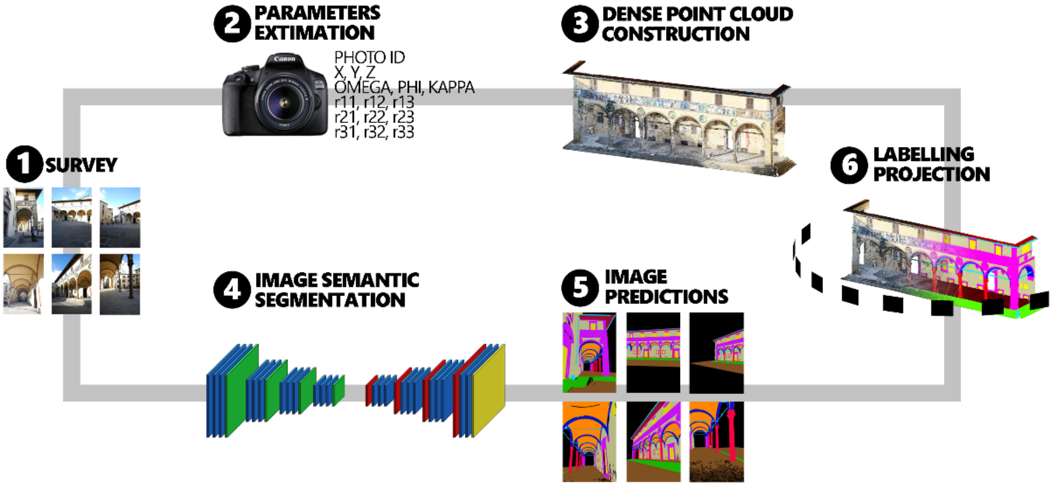


Figure 3.8 – Proposed semantic segmentation pipeline.

3.3.1 Photogrammetric survey

The global procedure starts form the building image acquisition. The collected images are the basis of the entire semantic segmentation workflow, since they allow in parallel the generation of the point cloud, and the detection of the predetermined building categories or classes. Hence, a well-planned and structured photogrammetric survey is essential to achieve good and reliable results. A photogrammetric survey requires the collection of multiple overlapping photographs from different point of views to create an accurate three-dimensional representation of a subject. To obtain a reliable survey, several key points must be considered. These include using a high-quality camera and lenses, proper camera setup and calibration, sufficient image overlapping and coverage, establishing accurate ground control points, using advanced image processing software, and conducting accuracy assessments. However, achieving a good survey can be challenging due to a variety of factors. Some of the main challenges include dealing with varying lighting conditions, controlling for camera motion, managing image noise and distortion, and ensuring accurate and consistent measurement of ground control points. Additionally, the accuracy of the final model can be affected by errors in camera calibration, or image processing. Therefore, it is important to carefully plan and execute the survey, and to take measures to minimize these sources of error to achieve a high-quality result. Despite several types of photogrammetric surveys can be distinguished, two categories are usually those of major interest: aerial and close-range photogrammetry. Aerial photogrammetry is used to map large areas of land. It usually uses high-resolution aerial photographs taken from a plane (or, more recently by an Unmanned Aerial Vehicle, in the UAV photogrammetry case). Close-range photogrammetry is based on photos taken from the ground, such as those taken by a handheld camera. The datasets used in this dissertation falls in the close-range photogrammetry case, even if the adopted procedure could also be applied for instance in the UAV photogrammetry case.

3.3.2 Camera calibration and exterior orientation estimation

The following step of the pipeline is the geometric camera calibration, also known as camera resectioning. It is the process of determining the values of the camera parameters (in particular for what concerns its imaging system) so that measurements taken from photographs can be reliably related to real-world locations. Knowledge of the camera parameter values is a *sine qua non* condition for dense point cloud generation, and, in the proposed workflow, it is also required in the labelling projection on the point cloud. Camera *intrinsic* and *extrinsic* parameters can be distinguished: the first describe the camera behaviour in the camera reference system, whereas the second ones are used to express camera measurements in a different reference system. To be more precise, the *intrinsic camera parameters* describe the internal workings of the camera, such as the size and shape of the imaging sensor, the lens focal length, and the optical centre of the camera. The *extrinsic camera parameters* describe the relative position and orientation of the camera with respect to the reference system used in the survey. This includes the position and orientation of the camera in 3D space, as well as the parameters of the camera viewing frustum, such as the field of view and the viewing direction. Extrinsic parameters are usually determined based on a known set of points in the 3D space, which should be visible in the acquired images.

Calibration of intrinsic camera parameters is traditionally performed using an ad hoc procedure, involving the collection of a set of calibration images often of a specific pattern of points, even if recently self-camera calibration if also quite often used, i.e. exploiting images collected for the 3D reconstruction of an object also for determining the intrinsic camera parameters. Hence, in the *Self-calibration* case the camera parameters are estimated without the need for a calibration object, by exploiting feature points in the scene to estimate the camera parameters. Despite this kind of procedure is quite convenient in terms of easiness of use, the obtained results are usually less accurate and robust than those obtained with ad hoc calibration methods. It is clearly fundamental when an ad hoc camera calibration procedure cannot be performed. Self-calibration is usually performed within the *Structure from Motion* (SfM) workflow: SfM identifies and robustly matches key feature points in multiple images, leading to the joint estimation of camera parameters and of the positions of a sparse set of points, namely the tie points, derived from the matched feature points.

Direct Linear Transformation (DLT) is a simple and efficient algorithm that estimates the camera matrix by using a set of corresponding points in the image and world coordinate systems. The algorithm assumes a pinhole camera model and estimates the camera matrix by solving a linear system of equations. DLT is widely used due to its simplicity and efficiency, but it is sensitive to noise and can result in inaccurate parameter estimates. *Bundle Adjustment* (BA) is a widely nonlinear optimization approach to refine the camera parameters and the 3D positions of the considered points by minimizing the reprojection error, i.e. the distance between the observed 2D points and the projections of the corresponding 3D points on the image views. BA is computationally more expensive with respect to linear approaches, but it leads to more accurate and robust results.

Zhang's Method is a widely used calibration technique that uses a planar calibration object to estimate the camera parameters. The algorithm requires at least two images of the planar object with known dimensions. Zhang's Method can estimate both the intrinsic and extrinsic parameters of the camera, including distortion coefficients. The algorithm is simple and efficient but requires a calibration object with known dimensions. More detail about camera calibration algorithms can be found in several reviews (Salvi et al., 2002), (Q. Wang et al., 2010), (D'Emilia & Di Gasbarro, 2017), (Long & Dongri, 2019).

The parameters obtained from the camera calibration process can be used to accurately estimate the 3D coordinates of the objects visible in the images of the considered scene.

3.3.3 Dense cloud construction and preparation

There are several algorithms that can be used to build a dense point cloud. Here are reported some of the most commonly used ones:

- *Patch-based multi-view stereo* (PMVS): PMVS is a popular algorithm for generating dense point clouds from multiple images. It works by dividing the images into small patches, and then searching for matching patches in neighbouring images. These matches are used to generate depth estimates for each pixel in the images, which are then combined to produce a dense point cloud.
- *Semi-Global Matching* (SGM): SGM is a stereo matching algorithm that can be used to generate dense point clouds from pairs of stereo images. It works by comparing the intensities of corresponding pixels in the two images, and then using these comparisons to estimate the depth of each pixel. SGM can be very accurate, but it can also be computationally expensive.
- *Multi-View Stereo* (MVS): MVS is another photogrammetric technique that uses multiple images to create a dense point cloud. It involves the computation of depth maps from multiple images and the fusion of these depth maps into a single 3D point cloud. MVS algorithms use dense image matching techniques to compute the depth maps.

Once the point cloud is generated, it requires some preliminary processing operation, in order to make the point cloud more suitable for the segmentation, and to prepare it for the reprojection phase. This operations include the cleaning and the denoising of the point cloud, in order to remove unwanted points, or elements in the scenes that are not relevant, for instance: vegetation, background buildings, people, etc. Despite this operations are not fundamental, they guaranteed a better accuracy and performance of the segmentation procedure.

3.3.4 Image semantic segmentation

Image semantic segmentation is the core step of the entire procedure. It allows to extract the target features of the building by producing a segmentation map for each image of the photogrammetric survey, according to the chosen categories or classes. Image semantic segmentation can be carried out using various methods, already described in the previous paragraphs. This dissertation focuses on exploring the deep learning-based methods. Hence, image semantic segmentation can be obtained exploiting the most popular state-of-the-art convolutional neural networks. As turned out in the previous paragraph, several CNN architectures are available, and new architectures are constantly proposed every year. Currently, the most popular models are Fully Convolutional Network (FCN), U-Net, SegNet, the DeepLab family, and Mask-RCNN. However, the proposed segmentation procedure is not restricted to a single type of architecture: the image segmentation building block can be easily changed with new or more performing models, based on different image segmentation strategies.

In order to be included in the overall pipeline, the deep learning-based image segmentation system should be properly set up. This involves several steps, reported in the following:

- *Data preparation*: The first phase is to prepare the data to be used in the deep learning model. This involves collecting and labelling a dataset of images that will be used to train the model. The dataset should contain images with the desired semantic labels, such as object categories or scene types. Nowadays several existing datasets are freely available, but they are not so frequently relevant to the required task. Currently, a specific dataset for heritage building image segmentation is missing. Therefore, the next chapter (§4) is completely focused on the generation of a new dataset suitable for the specific purpose.

- *Model architecture selection*: The second phase of the pipeline is to select the appropriate model architecture for the image segmentation task. There are many different deep learning models that can be used for image segmentation. In this dissertation FCN, SegNet and Deeplabv3+ are tested, but as mentioned previously, other networks could be used as well.

- *Model training*: The third phase of the pipeline is to train the selected model on the labelled dataset. This is the most critical phase since during training, the model learns to recognize the patterns and features that correspond to the semantic labels in the images. It involves adjusting the network weights and biases through the process of backpropagation, which involves calculating the error between the network output and the expected one. The training goal is to find the optimal set of weights and biases that allows the network to accurately classify or predict new data. Training a deep learning model can be a time-consuming and computationally intensive process, often requiring specialized hardware.

- *Model validation*: The fourth phase of the pipeline is to validate the trained model on a separate dataset of images. This is done to evaluate the performance of the model on images that were not used during training or were partially used in a preliminary test phase. Usually, the entire available dataset is split in training, validation and test set. The first two sets are used to train the network, and to select its parameter values, while the test set to evaluate its performance.

Once the model is correctly trained and validated, it can be integrated in the overall 3D point cloud segmentation pipeline, and it should be able to output a correct segmentation map for each input image. Once all the images of the photogrammetric survey are labelled, they could be projected onto the point cloud.

3.3.5 Labelling projection

The last step of the procedure is the projection of the labels on the 3D point cloud, in order to obtain the final cloud segmentation. The label transferring from the 2D representation to the 3D space is one of the most critical aspects in the multiview approaches. The procedure becomes more challenging when dealing with complex scenarios like the case of heritage buildings, in which complex shape, element uniqueness and irregular geometries require careful modeling. During the last years, several methods have been proposed to solve the problem of label projection from 2D images to 3D space to obtain a consistent 3D point cloud segmentation from labelled images. For example, (Y. Wang et al., 2013) design an approach to propagate the pixel-wise image labels from ImageNet to point clouds. In the first step they used *Exemplar SVMs* to over segment individual images into "superpixels", and then propagate their labels onto the visually similar superpixels in the reference images of point cloud. In the second step they used a graphical model to aggregate superpixel label candidates to jointly infer the point cloud labels. Some works on semantic mapping (McCormac et al., 2016b), (Hermans et al., 2014b) typically aggregated pixel-wise semantic features onto 3D reconstructed surfaces via Bayesian fusion and used Conditional Random Field (CRF) models to regularize the resulting 3D segmentation. In (B. H. Wang et al., 2019), the authors present *Label Diffusion Lidar Segmentation* (LDLS), a method for instance segmentation of 3D point clouds which leverages a pretrained 2D image segmentation model. They obtain 2D segmentation prediction by applying Mask-RCNN, and then link the image to a 3D LiDAR point cloud by building a graph of connections among 3D points and 2D pixels. (R. Zhang et al., 2018) addressed the issue of the semantic segmentation of large-scale 3D scenes by fusing 2D images and the last step of the procedure is the projection of the labels on the 3D point cloud, in order to obtain the semantically segmented cloud. Label transferring from the 2D representation to the 3D space is one of the most critical aspects in the multiview approaches. The procedure becomes more challenging when dealing with complex scenarios like the case of heritage buildings, in which complex shapes, element 0uniqueness and irregular geometries require careful modeling. During the last years, several methods have been proposed to face the problem of label projection from 2D images to 3D space to obtain a consistent 3D point cloud segmentation from labelled images. For example, (Y. Wang et al., 2013) design an approach to propagate the pixel-wise image labels from ImageNet to point clouds. In the first step, they used *Exemplar SVMs* to over segment individual images into "superpixels", and then propagate their labels onto the visually similar superpixels in the reference images of point cloud. In the second step they used a graphical model to aggregate superpixel label candidates to jointly infer the point cloud labels. Some works on semantic mapping (McCormac et al., 2016b), (Hermans et al., 2014b) typically aggregated pixel-wise semantic features onto 3D reconstructed surfaces via Bayesian fusion and used Conditional Random Field (CRF) models to regularize the resulting 3D segmentation. In this work (B. H. Wang et al., 2019), the authors present *Label Diffusion Lidar Segmentation* (LDLS), a method for instance segmentation of 3D point clouds which leverages a pretrained 2D image segmentation model. They obtain 2D segmentation prediction by applying Mask-RCNN, and then link the image to a 3D lidar point cloud by building a graph of connections among 3D points and 2D pixels. (R. Zhang et al., 2018) addressed the issue of the semantic segmentation of large-scale 3D scenes by fusing 2D images and 3D point clouds. According to this work the preliminary segmentation results with 2D images obtained by a DeepLab-Vgg16 based model, are mapped to 3D point clouds according to the coordinate relationship between the images and the point cloud calculated with DLT algorithm. More recently, (Genova et al., 2021) proposed a novel network 2D3DNet, that uses multi-view fusion to make best-guess semantic labels for as many 3D points as possible via back-projection and voting from labels of the corresponding pixels. (Mascaro et al., 2021) presented *Diffuser*, a novel framework that leverages 2D semantic segmentation to produce a consistent 3D segmentation. They formulate the 3D segmentation task as transductive label diffusion problem on a graph, where multi-view and 3D geometric proprieties are used to propagate semantic labels from the 2D space to the 3D map. They show a significant accuracy compared to probabilistic fusion methods. The approach developed in (Lertniphonphan et al., 2018), propagate object label from 2D image to a sparse point cloud by matching a group of points that corresponds to the area within the 2D bounding box in the image. The method was used for producing training data, and it demonstrates that the label propagation can be used to train a classifier with a good average precision. In the specific context of building segmentation, (Murtiyoso et al.,

2021) proposed an approach for the segmentation of 3D building façade based on orthophoto. The XY coordinates of each pixel in the orthophoto was used to determine the corresponding planimetric coordinates of the point in the point cloud and finally a winner-takes-all approach was applied to annotate the 3D points with the respective 2D pixel class. In a more recent work, (Murtiyoso et al., 2022) introduced semantic classification at the beginning of the classical photogrammetric workflow in order to automatically create a classified dense point cloud. In this regard, several image masks obtained by a trained neural network are employed during dense image matching in order to constrain the process into the respective classes. In the same context, (Stathopoulou & Remondino, 2019) proposed a semantic photogrammetry workflow, in which the label back-projection is based on the projection matrix P which connects the 3D with the 2D space. The segmented images are automatically generated using neural networks, and then the labels are used as constraints in the photogrammetric process. Giving the correspondence, all the images contribute to the labelling projection on the cloud, and if the assigned labels to each back-projected point do not match, the most weighted label wins

The proposed methodology aims at projecting the labels, predicted by a deep learning-based image semantic classifier on a set of N 2D images, on a 3D point cloud. The interior and exterior parameters of the images input in the deep-learning classifier are assumed to be known: despite such parameters could be computed aside of the point cloud generation, their availability comes for free when the point cloud is the outcome of a photogrammetric reconstruction procedure, and the images input in the classifier are taken among those used in the reconstruction. Hence, this could be considered as a quite ideal working condition for the proposed method. In accordance with the above consideration, hereafter the considered images are assumed to have already been aligned, and the exterior parameters are assumed to be expressed in a reference system compatible with the point cloud one. Then, the labels of the N predicted images are properly transferred to the point cloud, as described in the following.

1) 3D points of the cloud are projected on the N images, by means of the known interior and exterior camera parameters.

2) For each image I_j, each point class is assessed, if visible.

3) For each point, the mostly voted class is selected.

Let (u_j,v_j) be the pixel coordinates of the projection of point p on the image I_j. A straightforward implementation of step 2) is the assignation of the label of pixel (u_j,v_j) in I_j (if inside the image extent) as its vote to point p class. Despite being very simple, such a strategy does not take into account the obstructions, leading to unreliable outcomes in complex scenarios: the implementation of an effective procedure to check obstructions is of vital importance for ensuring a good performance of the overall algorithm in a wide range of working conditions. Assume that the point cloud density is sufficiently high to ensure that at least one 3D point is projected in all the adjacent pixels, in image I_j, describing the same object surface. Down-sampling the image size, or, equivalently, enlarging the pixel size, could be necessary in order to ensure the validity of such assumption. According to the above hypothesis, at least two points should be projected on the same pixel (u_j,v_j) when an obstruction occurs. When such event is detected, a simple check on the distance between the camera and the points projected on the same pixel is used in order to determine if any of such points probably obstructs the others. Image I_j votes only for the non-obstructed points. The main advantages of such procedure are the implementation simplicity and the quite effectiveness in most of the examined conditions. Nevertheless, a more complex strategy will be considered in our future investigations in order to improve the semantic segmentation results in critical conditions.

3.4 Summary

This chapter widely illustrated the algorithms for point cloud semantic segmentation, including also the approach investigated more in this dissertation. In the first paragraph (§3.1) the concept of artificial intelligence, machine learning, and deep learning have been introduced, including more in detail the functioning of the artificial neural networks (ANNs) (§3.1.1) and the convolutional neural networks (CNNs) (§3.1.2), which are at the basis of most of the semantic segmentation architectures. First, the image semantic segmentation algorithms have been presented (§3.2.1). They include four main classes: fully convolutional networks, dilated or "atrous" convolutional networks, multi-scale analysis, and regional convolution networks. In (§3.2.2), point cloud segmentation networks have been presented, including machine and deep learning-based ones. The main architectures can be grouped in two categories: projection-based and point-based networks. Both typologies have been illustrated and the state-of-the-art models have been comprehensively shown and discussed. Furthermore, (§3.3) presented the proposed procedure for the semantic segmentation of heritage building point clouds has been explained. It is based on a multiview approach, in which the features are extracted on the images, and then are projected to the point cloud. It is composed by five main steps: (i) photogrammetric survey, (ii) camera calibration and exterior orientation estimation, (iii) dense cloud construction and preparation, (iv) image semantic segmentation, and (v) label projection to the point cloud. Each of the five phases has been explained and discussed.

Chapter 4

The Dataset

In this chapter a new benchmark dataset developed to improve machine learning and deep learning methods that leverage on image segmentation in the heritage sector is presented. Such dataset can be used for the training and the validation phase of a machine learning system, and for the comparison of new and already existing segmentation approaches. In the first paragraph (§4.1) the importance, the motivation, and the challenges in the creation of a new dataset are introduced. In the second paragraph (§4.2) the existing datasets are shown and analysed, both for image and point cloud semantic segmentation. In the third paragraph (§4.3) the structure of the dataset is analysed, and the buildings that are going to compose the dataset are illustrated and described (§4.3.1). Currently the dataset is composed by five buildings, from different historical periods and architectural styles, mainly located close to Florence. In the next sections (§4.3.2, §4.3.3), the data acquisition procedures and the pre-processing operations are illustrated. The standards and the categories chosen to generate the ground-truth are then pointed out (§4.3.4). Since a proper and well-functioning dataset should be composed by thousands of images, a semi-automatic procedure to quickly label all the images of the same building has been developed, assuming that a manual segmentation of the related photogrammetric point cloud is available. Such procedure, that significantly reduces the manual intervention and the labelling time, is illustrated and discussed in detail (§4.3.5). In the next section (§4.3.6) the statistics and the properties of the dataset are shown accurately. Finally in paragraph (§4.4) two main techniques to improve quality and size of a dataset are illustrated, i.e. data augmentation and synthetic data generation. The chapter ends with a general summary (§4.6).

4.1 Introduction

Datasets play a central role in AI based applications, especially when talking about machine and deep learning applications (Dekker, 2006), (Koesten et al., 2020a). Given the huge number of parameters to be properly set, deep learning-based models are data-hungry, and they require a large amount of data to ensure a high level of reliability of the trained network (X. W. Chen & Lin, 2014), (W. Wang et al., 2016). But at first, what is a dataset in machine learning? A dataset is a collection of several typologies of data stored in a digital and structured format used to train and validate the models. Common types of data include texts, images, video sequences, audio sequences, points, numerical values, etc. Data are usually labelled or annotated in order for the algorithms to understand what the outcome needs to be. Dataset preparation, setting and understanding are certain of the most important aspects in a machine learning application lifecycle, and these operations underlie the success or the failure of a machine learning project (Jain & Nicholls, 2008). According to *The State of Data Science 2020* data scientists and AI developers spend nearly 70% of their time analysing and creating a properly functioning dataset, and only the 30% of the remaining time in other processes such as training, testing, model selection and tuning. The importance of data can be understood following the concept of "Garbage in, Garbage out" (GIGO), a popular expression in the early era of computing, '*if we feed low-quality data to ML model it will deliver a similar result*'. Nowadays several open-source datasets are available to solve real-world problems in many fields (Gregory et al., 2019), but often they are not directly suitable for the specific application that we are working on, or they reveal some limitations that can compromise the success of the developed model. For this reason, solving a new problem statement can be quite challenging. The performance of a learning system strictly depends on four key points: quantity, quality, usability, and scalability of the training dataset (Koesten et al., 2020b) (Figure 4.1). *Quantity* is important because an algorithm needs enough data to be trained and to create robust predictions, especially the deep learning models. The lack of a large dataset could be the cause of overfitting and the model could be performed poorly when applied to new examples. There is no perfect recipe for how much data the model needs, but in general more complex is the task more data the model demands. *Quality* is essential for avoiding problems with bias and blind spots in the data. Low-quality data could be cause of overfitting problems, and eventually leading low-quality output predictions. High-quality can be achieved cleaning and denoising the data and making it uniform and manageable before the annotation and the training processes. A key quality point is the balance of the dataset, that refers to the propriety of the data to represent all the classes or the categories of the problem with the same weight. *Usability* describes how much the data are easy to use, and how they are relevant for the specific task that we are working on. The problem statement needs to be well-defined, and the data need to be well-representative of the problem. *Scalability* is important because to accurately represent all aspects a dataset needs to be scalable.

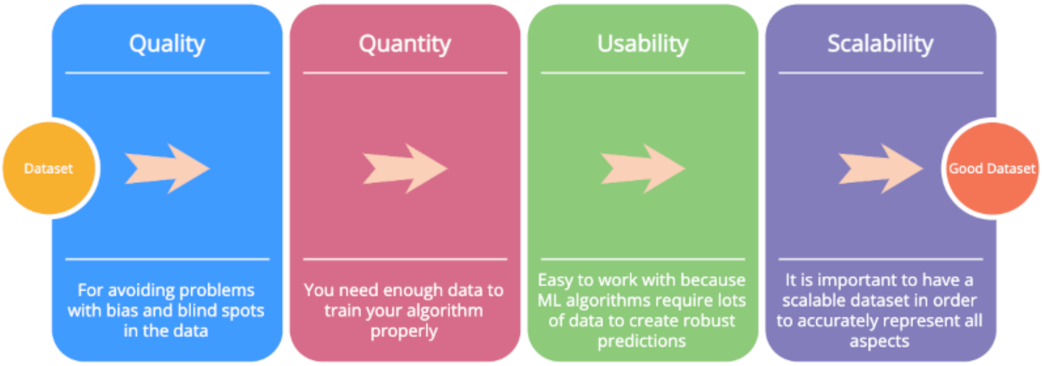

Quality
Quantity
Usability
Scalability
Dataset
Good Dataset
For avoiding problems with bias and blind spots in the data
You need enough data to train your algorithm properly
Easy to work with because ML algorithms require lots of data to create robust predictions
It is important to have a scalable dataset in order to accurately represent all aspects

Figure 4.1 – The four key points for a good dataset designing (from clickworker.com).

Structuring a new dataset needs to face many challenges: *(i) insufficient data*, in many cases the collection of multiple and different data can be difficult due to time restrictions or due to the non-availability of large samples in the real world, *(ii) bias and human error,* instruments and tools used for data acquisition lead to human errors or biases towards some aspects, *(iii) quality,* the real problems are often complex and giving a formal structure to data leads inevitably to a loss of quality and information, *(iv) privacy and compliance,* in some cases the sources cannot share their data due to privacy and compliance regulations, such as medical or security applications, *(v) data annotation process*, generally the labelling or the annotation of the data require manual and human interventions that are time-consuming, expensive and often prone to error. Nowadays there are many different platforms that allow to search and download open-source data for machine learning tests and experiments. The most popular platforms are Kaggle Dataset, UCI Machine Learning Repository, AWS Public Datasets, Google Dataset Search. Publicly available datasets should be well organized and regularly updated, they should provide a high-quality data for several tasks and applications, and they should be directly and easily suitable for training. However, despite several datasets are currently available, in many cases they do not fit properly into a custom-built model, and they are not relevant enough to describe and represent a specific problem. For these reasons a highly specific task, such as the mentioned semantic segmentation of heritage building point clouds, requires the careful construction of a dedicated dataset from scratch. In the next paragraphs, all the decisions and the procedures that have led to the definition and the creation of the new dataset are illustrated.

4.2 Existing Dataset

In this section a summary of the most used and popular datasets and benchmarks for semantic segmentation are provided, including details about the structure, characteristics, and tasks in each case. The datasets have been grouped into three categories: 2D image datasets, 2.5D image datasets, including the datasets with depth information in addition to RGB colour, and 3D point cloud datasets. Moreover, a summary of the existing datasets in the specific context of heritage building is provided. It is worth noticing that is currently missing a precise dataset for the semantic segmentation of heritage images (Fiorucci et al., 2020). This reason led to the creation of a custom and personalized dataset.

4.2.1 Image Datasets (2D)

Since most applications deal with 2D image segmentation, more than two hundreds open-source datasets for different tasks are available. In this paragraph just the most popular ones are shown and briefly discussed.

PASCAL VOC – Visual Object Classes (Everingham et al., 2010) is one of the most popular datasets. Images are annotated for 5 different tasks: classification, segmentation, detection, action recognition and person layout. Each image has a pixel-level segmentation annotation, bounding boxes and object class annotations. For the segmentation task there are 21 classes of object labels, including vehicles, household, animals, and other common objects. This dataset is divided into three sets, training and validation, with 1,464 and 1,449 images, respectively and a private testing set.

MS COCO – Microsoft Common Object in Context (T.-Y. Lin et al., 2014) is a large-scale collection of images for object detection, segmentation, key-point detection, and captioning. It is composed by 328k images with a total of 91 object types and 2.5 million labelled instances, mainly representing everyday scenes and common objects in their natural contexts. Objects are labelled using per-instance segmentation to aid in precise object localization.

ADE20K (B. Zhou et al., 2016) is a scene-centric parsing benchmark with 150 object categories, which include stuffs like sky, road, grass, person, etc. For the task of semantic segmentation, the images are finely labelled with a pixel-wise annotation. There are 20,210 images in the training set, 2000 images in the validation set, and 3000 images in the test set.

The Cityscapes Dataset (Gählert et al., 2020) is a large-scale collection of diverse set of stereo sequences recorded in street scenes from 50 cities during several months, daytimes and in good weather conditions, mainly focus on semantic understanding of urban scenarios. It consists of 30 classes grouped in 8 categories (flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void) in 5k fine annotated images, and 20k coarse annotated ones. It was originally a video sequence, so the images were created selecting manually the frames in order to obtain varying background, varying layout and a large number of dynamic objects.

CamVid – Cambridge-driving Labelled Video Database (Brostow et al., 2009) is a road/driving scene understanding database composed by 701 images sampled from a video sequence with a resolution of 960x720 pixels, captured by a camera mounted on the dashboard of a car. The images were manually annotated in 32 classes, including street categories (road, car, train, vegetation, lane markings, etc.) to support self-driving car applications.

CMP Façade Database (Tyleček & Radimšára, 2013) is a dataset of façade images assembled at the Centre for Machine Perception of the Technical University of Prague. It includes 606 rectified images of façade from various sources, which have been manually annotated using a set of overlapping rectangles with a class label assigned. The façades are from different cities around the world and diverse architectural styles, labelled in 12 classes, including the main architectural elements such as window, pillar, door, balcony, etc.

KITTI – Karlsruhe Institute of Technology and Toyota Technological Institute (Geiger et al., 2013) is one of the most popular datasets used in autonomous driving and mobile robotics. It contains traffic scenarios recorded with high-resolution RGB cameras, grey scale stereo cameras, and 3D laser scanners. Initially it did not contain the ground truth for image segmentation, however various researchers have manually labelled parts of the datasets for their purposes. For example, Zhang et al. (2021) annotated 252 acquisitions with ten object categories, or Alvarez et al. (2012) generated the ground truth for 323 images with three categories: road, sky, and vertical.

Aerial Semantic Segmentation Drone Dataset focuses on semantic understanding of urban scenes for increasing the safety of autonomous drone flight and landing procedures. The images were acquired with a high-resolution camera at an altitude from 5 to 30 meters above ground, and at a size of 6000x4000 pixels. The images were labelled with a fine pixel-level annotation including 20 classes representing the main ground elements such as tree, grass, vegetation, gravel, etc. In addition, the dataset includes the bounding boxes for the task of person detection, thermal images, ground control points, and fish-eye stereo images with synchronized IMU measurements.

Other popular datasets are SYNTHIA, Stanford background, Youtube-Objects, Adobe Portrait Segmentation, SiftFlow, Berkeley Segmentation Dataset (BSD).

4.2.2 RGB-D Image Datasets (2.5D)

In recent years RGB-D sensors and range scanners have become more affordable, and RGB-D images have become popular in many research and industrial applications. They provide a per-pixel depth information aligned with corresponding image pixel, and they usually allow to better understand spatial features. Currently very little data are available, and they cover a small range of scenarios and annotations. The following datasets are the most popular benchmarks for this category.

NYU-D V2 (Silberman et al., 2013) is composed of video sequences from a variety of indoor scenes recorded by both the RGB and Depth cameras from the Microsoft Kinect. It includes 1,449 RGB and depth images from 464 different indoor scenes of commercial and residential buildings recorded in three different US cities. Each image was labelled using Amazon Mechanical Turk with a fine and dense per-pixel annotation. The dataset contains 35,064 objects, spanning 894 different classes, and if the scene contains multiple instances, each instance received a unique instance label, to uniquely identify them.

SUN-3D (Xiao, Owens, et al., 2013b) is a dataset composed by 8 annotated sequences from a large-scale RGB-D video database. The annotation was carried out with a semi-automatic tool that uses a partial reconstruction to propagate labels from a frame to another. Semantic segmentation of the objects and information about the camera position are provided for each frame. It is composed by 415 sequences captured in 254 different spaces, in 41 different buildings.

SUN RGB-D (Song et al., 2015) is a benchmark similar to the PASCAL VOC composed by more than 10,000 RGB-D images captured by four different sensors (Intel RealSense, Asus Xtion, Kinect v1, Kinect v2). For each image they annotated the objects with both 2D polygons and 3D bounding boxes. The whole dataset contains 146,617 polygons and 58,657 3D bounding boxes, there are 14 objects in each image on average, and, in total, there are 47 scene categories and about 800 object categories. The dataset can be used for six different tasks, including semantic segmentation, object detection, scene categorization and object orientation.

ScanNet (Dai et al., 2017) is very large instance-level indoor RGB-D dataset that contains 2,5M views in 1513 scenes acquired in 707 distinct spaces. What make this dataset interesting is its annotation with estimated calibration parameters, camera poses, textured meshes, 3D surface reconstructions, and dense object-level semantic segmentation. The frames were captured at a resolution of 640 x 480 pixels and colour at 1296 x 968 pixels. To collect the sequences, they used the Structure sensor, a sensor similar to the Microsoft Kinect v1, attached to a portable device such as an iPhone or iPad. The considered categories include the main indoor elements such as chair, table door, window, bed etc.

UW RGB-D Object Dataset (Lai et al., 2011) is a dataset recorded using a Kinect style 3D camera that records synchronized and aligned 640 x 480 pixels RGB and depth images at 30Hz. The images contain 300 distinct objects from multiple views. The chosen objects are commonly found in home and office environments.

Other popular datasets are InteriorNet, SUNCG, Hypersim, OCID and TICaM.

4.2.3 Point Cloud Datasets (3D)

Nowadays 3D datasets are becoming more and more popular, and a lot of applications in robotics, remote sensing, and construction, leverage on these data typologies. Three-dimensional data could be provided via meshes, shapes, voxel representation or, in many cases, point clouds. Building a 3D dataset is always challenging due to the difficult of retrieve large amount of 3D data, the complexity of processing these types of data, and the time-consuming operations to finely annotate the scenes. In the following, some of the most popular 3D datasets are mentioned.

ShapeNet (Chang et al., 2015) is a large-scale repository for 3D CAD models developed by researchers from the Princeton University and the Stanford University. It contains a multitude of semantic categories organized according to the WorldNet taxonomy.

S3DIS – Stanford 3D Indoor Scene Dataset (Armeni et al., 2016) is a collection of RGB coloured 3D scans of indoor areas of large buildings with various architectural style, containing 6 large-scale areas with 271 rooms. The entire point clouds were automatically generated without any manual intervention using the Matterport scanner. All the points in the dataset are properly annotated, selecting their class among the available 13 semantic categories, including the main structural elements, and commonly found items and furniture. Compared with other 3D indoor point dataset the classes are more fine-grained and challenging.

Semantic3D (Hackel et al., 2017) is a large point cloud outdoor dataset which covers a range of diverse urban scenarios: churches, streets, railroad tracks, squares, etc. It is composed by 15 training and 15 test scenes, with over four billion points acquired with static terrestrial laser scanners. The point clouds were manually labelled in 8 categories (terrain, pavement, grass, vegetation, tree, building, car, etc.) following both 2D and 3D annotation techniques.

STPLS3D (M. Chen et al., 2022) is a large-scale aerial photogrammetry dataset with synthetic and real annotated 3D point clouds for semantic and instance segmentation.

To create the synthetic scene the authors developed a pipeline using Computer Generated Architectural (CGA) shape grammar od CityEngineering tools, that create 3D buildings starting from their footprints. The precise annotations were generated fully automatically while rendering the 2D images. The dataset covers more than 16 square kilometres of landscape and up to 18 fine-grained semantic categories.

SceneNet (Handa et al., 2015) is a synthetic dataset of indoor scenarios created starting from 10 3D scenes. Each scene is composed of 15-250 objects, but the complexity can be controlled algorithmically. The granularity of the annotations can be adapted by the user depending on the type of application. The scene can be classified in 11 categories, following the guidelines of NYU-V2 dataset.

Paris-Lille-3D (Roynard et al., 2017) is a large and high-quality ground truth urban point dataset for automatic segmentation and classification. The dataset consists of around 2Km of Mobile Laser Scanner (MLS) mounted at the rear of a truck, acquired in Paris and Lille, reaching totally 143,1M of points with a density between 1000 and 2000 points per square meter on the ground. The clouds were segmented and classified by hand using CloudCompare software.

Sydney Urban Objects Dataset (De Deuge et al., 2013) contains a variety of common urban road objects annotated across classes of vehicles, pedestrian, signs and trees. The acquisitions were made with terrestrial laser scanner in the central business district of Sydney.

Other popular datasets are Toronto-3D, SensatUrban, InteriorNet, SemanticPOSS, KITTI Road.

4.2.4 Heritage Datasets

In the context of heritage environment, few datasets are available, they are often small and not publicly available, and a specific dataset for semantic segmentation of image of historical buildings (Fiorucci et al., 2020) is still missing. The most remarkable heritage datasets are reported below.

ArCH – Architectural Cultural Heritage dataset (Matrone et al., 2020) is a benchmark for large scale heritage point cloud semantic segmentation. It is composed of 17 fine manually annotated scenes, derived from the union of several scans and their integration with photogrammetric surveys. In addition to the point coordinates the dataset provides the RGB values for each point, and the point normal *Nx, Ny, Nz,* calculated with CloudCompare. The point clouds are labelled in 10 classes, which include the main BIM standard elements: column, arch, moulding, floor, window, wall, stair, vault, roof and other. The dataset was created to support the development of machine learning and deep learning models in the heritage environment and is one of the most promising dataset in this context.

CHAS – Cultural Heritage Architectural Segmentation (Pavia et al., 2019) is a point cloud dataset from cultural heritage aimed to provide data for semantic segmentation techniques. The data were generated by terrestrial laser scanning and UAV photogrammetric surveys. The dataset comprises relevant buildings representing religious and colonial Brazilian architecture.

AHE – Architectural Heritage Elements (Llamas et al., 2017) is an image dataset developed for the task of classification of architectural heritage images. The dataset consists of 10235 RGB images classified in 10 categories, including some construction elements like Domes, Altars or Bell towers. Most of the images have been obtained from Flickr and Wikimedia Commons, all of them under creative common license.

MonuMAI – Monument with Mathematics and Artificial Intelligence (Lamas et al., 2021) is a public image dataset labelled using two annotation types, which make it useful for several tasks, such as monument style classification, for the detection of key elements, and other potential applications. It contains 1514 RGB images grouped in four architectural styles. Some key elements are also identified using bounding boxes, which report element names and locations.

Cultural Heritage Dataset – Orthodox Churches consists of 128 x 128 pixels images representing Christian Orthodox churches grouped in four categories: (i) chandelier, (ii) dome, (iii) frescoes, and (iv) lunette. There are 200 images per category, totalling 800 images.

UNESCO Heritage sites (2021) simply provides the spatial data of 1121 World Heritage Sites that were listed by UNESCO. The dataset can be used to catalogue, preserve sites and enhance the protection of the value of these sites.

4.3 Dataset Structure

The main aim of the dataset creation is to design a large scale image-based benchmark for the semantic segmentation of heritage building images. The dataset will be used to develop and train a deep neural network model designed to be incorporated into a wider point cloud segmentation workflow. The model should be capable to output a high-quality per-pixel feature map of a new and never seen set of images. Hence, the dataset should be composed by as much images as possible (*quantity*), and, to guarantee a high level of generalization, the images need to represent multiple and variable scenarios, several buildings typologies, and different types of architectural styles (*relevance*), and they should be annotated with a fine-grained per-pixel map, avoiding inaccuracy and lack of precision (*quality*). Most of the time, however, it is not easy to maintain both *quality* and *quantity* at the same time. Creating a fine and accurate ground-truth often requires manual intervention and a careful supervision, rarely applicable to large scale datasets. A well-structured dataset should guarantee a good balance between both the proprieties. In this section the dataset structure is illustrated in detail, including the buildings, the data acquisition, and the processing phase. Furthermore, a semi-automatic labelling procedure developed to speed up the annotation of multiple images is shown, and its accuracy is assessed. Finally, the dataset is analyzed and discussed.

4.3.1 The Buildings

The first decision to be made in order to structure the benchmark is the choice of the scenes and the buildings to be included in the dataset. Since the main aim of the benchmark is to support machine learning models development in the heritage/historical sector, the first question is which kind of buildings could be included, and which attributes, qualities or characteristics should have a building to be defined as 'historical'. A historical building is generally defined as a building or structure with an 'historical value' or an 'historic interest' such as the national or society value, the construction methods, the design, the architectural significance and so on. The current choice turned on Italian monumental buildings, with an important historical value, including churches and chapels, but future integration with different typologies will be taken into account. The initial goal was to collect and process nine 3D scenes from nine historical buildings, but due to the long time for the acquisition, the challenging processing phases, and due to time limitation, currently the dataset is composed by five buildings. Despite it could be a reasonable number of case studies to start training a neural network, the generalization and the capability of the model strongly depend on the variety of the scenes, and training a more robust and reliable model certainly requires a larger number of buildings. For these reasons, future developments of the dataset will be focused firstly to increase the number of the study cases. These five buildings are located in Tuscany (Italy), they were built in different historical periods, and they are characterized by different architectural styles and designs. Nevertheless, they share some common features and design structures, such as the presence of loggia in the facade, the presence of classic orders, the proportion between different elements. These buildings and their characteristics are typical of the Florentine renaissance style. On one hand, the presence of common features could facilitate the model during training to match between the various classes. On the other hand, not providing the model with a wide range of heterogeneity throughout the same classes, could constitute a drawback in the development of a well-generalizing model across multiple typologies of elements in the same category. This is a key and challenging point in the dataset definition, since historical and heritage buildings are generally characterized by a wide range of different element typologies without a clear standardization, many times unique or different in shape, geometry and dimension. Such a demanding context requires more case studies to be included in the benchmark, which increases the probability that the model will be successful. Below is reported a list with the buildings included in the dataset and a brief description of their characteristics.

(1_SC) Spedale del Ceppo

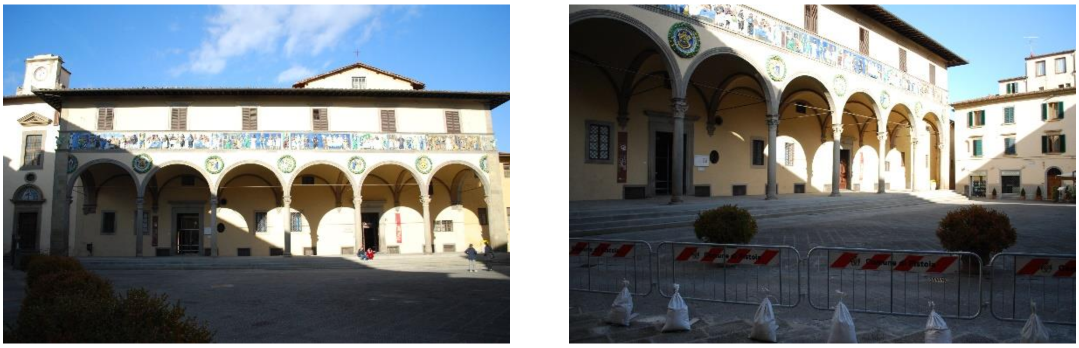


Figure 4.2 – (1_SC) Spedale del Ceppo, Pistoia.

Spedale del Ceppo (Figure 4.2) is a medieval hospital founded in 1227 in Pistoia, Tuscany, but the current complex is the result of a series of additions and modifications dated back to 15th and 16th century. The symmetric façade is composed by a renaissance loggia with six arcades, and it is decorated by a ceramic glaze frieze and a series of ceramic medallions at the springers of the arches. The five stone columns are in Corinthian style, and the interior of the loggia, opened in both the sides, is composed by a series of sail vaults ribbed by stone arches.

(2_OSA) Ospedale Sant'Antonio

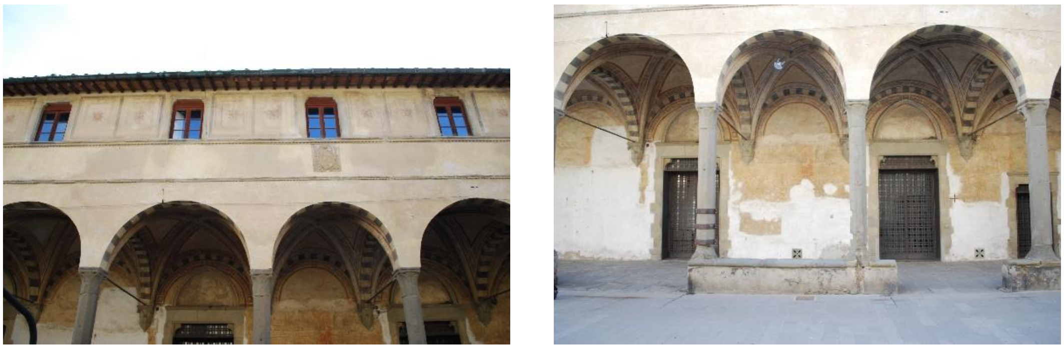


Figure 4.3 – (2_OSA) Ospedale Sant'Antonio, Lastra a Signa (FI).

The Ospedale Sant'Antonio (Figure 4.3) is located in Lastra a Signa, close to Florence, Tuscany, and its construction started around 1410. The façade is composed by seven arcades, one of which is blind. The loggia, composed by seven octagonal stone pillars, is blind on both sides and is closed on top by a series of cross vaults, ribbed by decorated arches.

(3_SS) Basilica della Santissima Annunziata

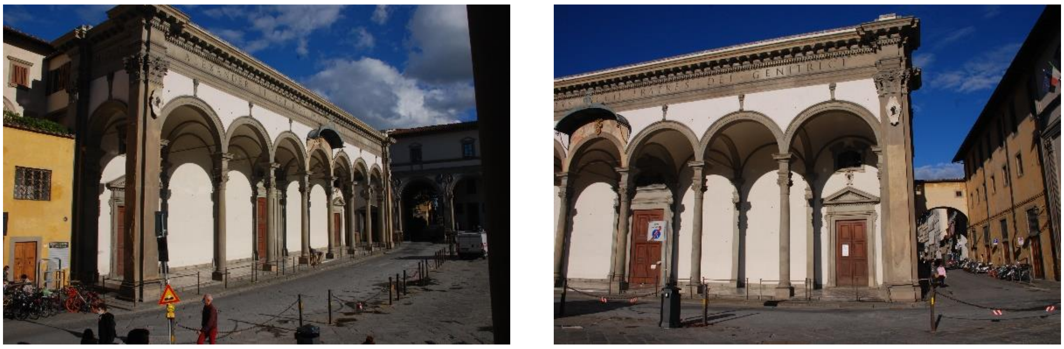


Figure 4.4 – (4_SS) Basilica della Santissima Annunziata, Firenze.

The basilica of Santissima Annunziata (Figure 4.4) is one of the most important churches in Florence, and it was built between 1440 and 1481 based on a project of the architect Michelozzo. The façade is inspired by the close Ospedale degli Innocenti and it is composed by a loggia with seven large arcades and six high stone columns. The interior of the loggia is surmounted by sail vaults ribbed by stone arches.

(4_CG) Certosa del Galluzzo

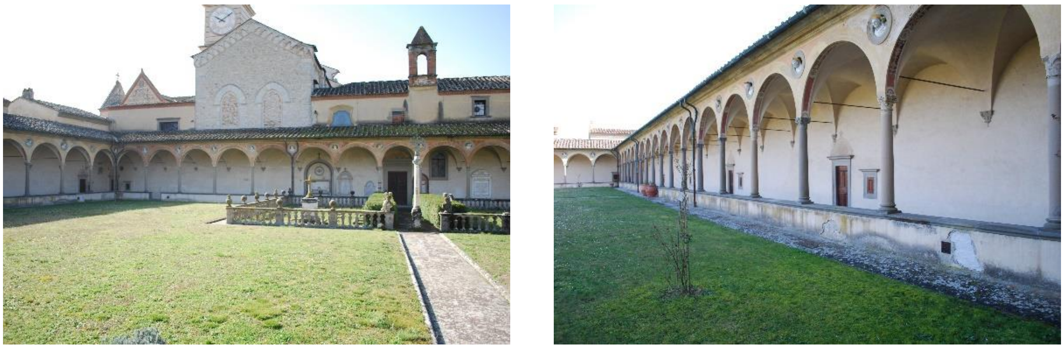


Figure 4.5 – (4_CG) Certosa del Galluzzo, Firenze.

The Certosa of Galluzzo (Figure 4.5) is a charterhouse, or Carthusian monastery, located in the Florence suburb of Galluzzo, Tuscany. It was built starting from 1341, and it was expanded and reconstructed over the centuries. The dataset includes a portion of the Chiostro Grande, a large square cloister built around 1520, which each of the four sides is articulated in a loggia with columns and round arches with sixty-six ceramic glaze medallions above each column. The interior of the loggia is surmounted by cross vaults.

(5_CB) Cappella Buontalenti

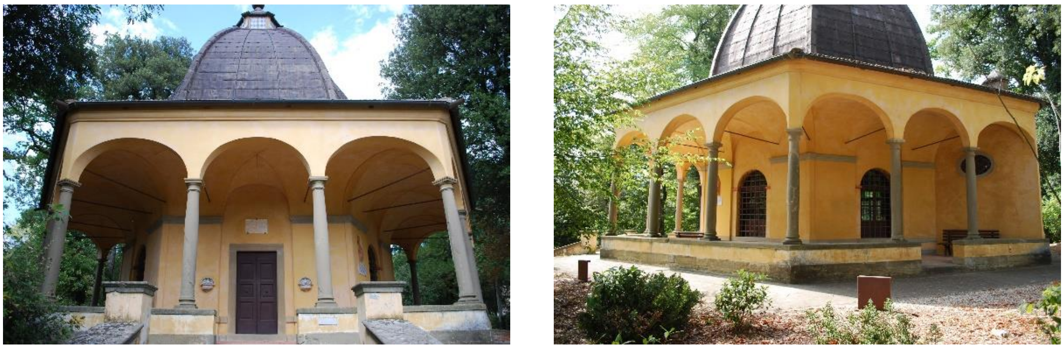


Figure 4.6 – (5_CB) Cappella Buontalenti, Firenze.

The Cappella Buontalenti (Figure 4.6) is a little and elegant renaissance chapel built in 1580 by the architect Bernardo Buontalenti, and it is located inside the large park of Villa Demidoff, not far from Florence. The chapel has a hexagonal plan, and is surrounded by a loggia, composed by twelve columns and arches, closed on one side. The chapel, accessible by a large stone staircase, is surmounted by a cloister dome on the top.

4.3.2 Data Acquisition

The terrestrial laser scanner data, collected over the past years in an educational context by the GECO laboratory (Geomatics and Conservation group of the Department of Civil and Environmental Engineering, University of Florence headed by professor Grazia Tucci), were already available for all the mentioned buildings. As a result, a georeferenced 3D point cloud was already available for all the considered buildings. The main aim of the new acquisition campaigns was to integrate these data with close-range photogrammetric surveys, in order to collect a large number of images that are going to compose the new image dataset. Depending on the building, the photogrammetric survey was done using two digital single-lens reflex (DSLR) cameras, a Nikon D60 and a Nikon D80, both with 10.2 MP, and provided with a 23.6 mm x 15.8 mm Nikon DX format RGB CCD sensor, 1.5 x FOV crop, with a maximum resolution of 3,872 x 2,592 pixels. Both the cameras were equipped with a zoom lens AF-P DX Nikkor 18-55 mm f/3.5-5.6G. The images were acquired in the .JPEG format with a horizontal and vertical resolution of 300 dpi. All the images were acquired with a focal length of 18 mm, ISO-200, and a fixed aperture not larger than f/14 to guarantee a good depth of field, helping the photogrammetric process, and increasing the number of usable pixels in the point cloud reconstruction. In the following table (Table 4.1) are reported some details for each acquisition.

Table 4.1 – Photogrammetric acquisition information for the five buildings.

Building	TLS cloud	Camera	N° of photo	Resolution
1_SC	yes	Nikon D60	**748**	3872x2592
2_OSA	yes	Nikon D60	**755**	3872x2592
3_SS	yes	Nikon D80	**473**	3872x2592
4_CG	yes	Nikon D60	**1102**	3872x2592
5_CB	yes	Nikon D80	**166**	3872x2592

After completing the acquisitions, the following operation is constructing the ground-truth starting from the RGB images. The current total number of images considering all the five buildings is 3,244, that makes challenging and time consuming the manual annotation of each single image. For this reason, I developed a semi-automatic procedure that allows to label all the images of the photogrammetric survey starting from a manual segmentation of the related point cloud, less challenging and time consuming. In the next paragraphs, all the steps of the procedure will be explained and detailed.

4.3.3 Data Processing

The collected data were processed firstly to create the 3D scene, and secondly to set it up for the projection phase. Data processing was the most time-consuming stage in dataset creation, it requires long manual and specialized interventions, and it is not easy to be automated. In addition, the performance of the image labelling procedure is strongly related to the quality of the processing phase, hence it requires a careful supervision. For each scene the processing operations followed these steps:

- Photogrammetric 3D point cloud generation
- TLS and photogrammetry cloud alignment
- cleaning/denoising
- subsampling
- annotation
- integration of missing points.

The first processing step was the *cloud construction* starting from the images. This operation was performed using Agisoft Metashape™, one of the most used proprietary licenced software for the photogrammetric pipeline. Alternatively, there are also some open-source software solution, such as PhotoCatch, Meshroom, MicMac and many others. Metashape workflow for the 3D reconstruction is composed by four main steps.

- *Feature matching across the photos*. At the first stage Metashape detects points in the source photos which are stable under viewpoint and lighting variations and generates a descriptor for each point based on its local neighbourhood. These descriptors are used later to detect correspondences across the photos. This is similar to the well-known SIFT approach but uses different algorithms for a little bit higher alignment quality.

- *Solving for camera intrinsic and extrinsic orientation parameters*. Metashape uses a greedy algorithm to find approximate camera locations and refines them later using a bundle-adjustment algorithm. Camera self-calibration is also implemented in the Metashape processing workflow. Ad hoc calibration of the intrinsic parameters can be considered in order to improve the overall performance of the proposed method.

- *Dense surface reconstruction*. At this step several processing algorithms are available. Exact, Smooth and Height-field methods are based on pair-wise depth map computation, while Fast method utilizes a multi-view approach.

- *Texture mapping*. At this stage Metashape parametrizes a surface possibly cutting it in smaller pieces, and then blends source photos to form a texture atlas.

In the following table (Table 4.2) some details on the reconstructions of the five buildings are reported, including the final number of points of the dense reconstruction.

Table 4.2 – Results of the point cloud construction with Metashape.

Building	N° of Images	Tie Points	Quality/Filtering	Dense Cloud
1_SC	748	413,405	High/Mild	**43,839,637**
2_OSA	755	465,021	High/Mild	**87,421,205**
3_SS	473	323,013	High/Mild	**41,220,646**
4_CG	1102	746,884	High/Mild	**149,000,912**
5_CB	166	142,082	High/Mild	**65,677,457**

Clouds alignment. Since the photogrammetric surveys were made with no targets and reference points, the result of the reconstruction is a dimensionless and not georeferenced point cloud. However, it is possible to exploit the TLS point cloud to scale and georeferenced the photogrammetric reconstruction, via the alignment of the two clouds. The alignment was obtained with a two-step procedure using CloudCompare, a GPL open source software for point cloud processing. At first, the two clouds were roughly aligned picking manually some equivalent reference points, and minimizing an error metric, in this case the sum of squared differences between the coordinates of the matched pairs. The minimum number of points that need to be selected is 3, but, depending on the complexity of the scene, the selected points were around 5-10. At the end of the procedure will be output the Root Mean Square (RMS) calculated on the picked points. It is possible to see also the error contribution for each pair of points, that can help to improve the result if not satisfactory, removing or adding new points. The first alignment has been considered acceptable if the RMS was less than 0.1 m. Secondly, the alignment was improved and refined by means of the Iterative Closest Point (ICP) algorithm (Y. Chen & Medioni, 1991), (Besl and McCay, 1992). At the end of the procedure CloudCompare output the transformation matrix, the scale factor, and resulting final RMS. In the following table (Table 4.3) the alignment results are reported for all the five buildings composing the dataset.

Table 4.3 – Results of the alignment procedure after the ICP alignment.

Building	Picked Points	Scale Factor	ICP	RMS (m)
1_SC	5	1.570	yes	**0.19**
2_OSA	5	0.605	yes	**0.09**
3_SS	5	0.977	yes	**0.19**
4_CG	8	1.344	yes	**0.16**
5_CB	10	1.273	yes	**0.12**

Is not surprising that the final RMS is higher than the first obtained with the manually picked points, in fact the error of the ICP tool is computed on up to 50,000 points, while the first only with the picked points. However, ICP tool always improves the overall alignment, and it makes sure the overlap parameter is realistic.

The *subsampling* operation was necessary to face the high number of points in each scene, and to make all the scenes more homogeneous and regular. To perform this operation was used a random subsampling, setting a minimum space distance between the points. The minimum distance needs to be carefully selected, avoiding a loss of the level of detail, but at the same time reducing considerably the number of points and the size of the file. Moreover, the point cloud will be used to project the labels on the images, and a low density of points could make the process unsuccessfully. At the same time, a too large density could turn the process computational costly. After some tests a minimum space of 0.01 m turned out to be a reasonable distance to ensure the above mentioned properties.

The *cleaning/denoising* operations was fundamental to eliminate the unwanted portion of the scene, to remove obstacles and obstructions, and to obtain a more precise and reliable point cloud. These operations were performed with two methods: manually, picking the unwanted points and deleting them from the scene, or by semi-automatic features selection, such as colour, particular suited for sky or vegetation removal, or geometric features like planarity, distance, altitude, etc. Another method used to remove noise and inaccuracy from the photogrammetric cloud was to exploit the more reliable TLS cloud, setting a maximum space between the clouds, and filtering out the points which do not respect the distance. This procedure could be performed during the following annotation stage, and it is explained more in detail in the next section.

The *annotation* is the key operation of this stage: it consists in assigning to each point of the cloud a label according to the chosen categories. It is a labour-intensive procedure, that requires a lot of manual work. This operation was performed using CloudCompare and is based on two main steps. The manual annotation was carried out firstly on the TLS point cloud, which is usually more accurate and less noisy. The annotation was carried out manually picking the points with selection bounding boxes, or exploiting common geometrical features such as distance, planarity, altitude, symmetry, etc. Secondly the labels set on the TLS cloud were transferred to the aligned photogrammetric reconstruction, based on a closest point criteria. Several tests were made to find the optimal distance to transfer the labels between the two clouds. A longer-range distance allows to select a larger part of the scene, and to avoid excluding some significant points, but at the same time it may generate inaccuracy especially in the connections between different elements. A shorter-range distance may cause loss of information, and a decreasing of the number of points, but simultaneously a positive denoising and regularizing effect, that increase the precision of the photogrammetric point cloud. Figure 4.7 reports three labelling examples for the columns of (1_SC) Spedale del Ceppo with three different distances: a) 0.1m, b) 0.05m, and c) 0.02m. Based on the experiments the transfer distance was set to 0.05 m, a good balance between inclusion, precision and denoising, but depending on the specific case may be changed to refine the segmentation.

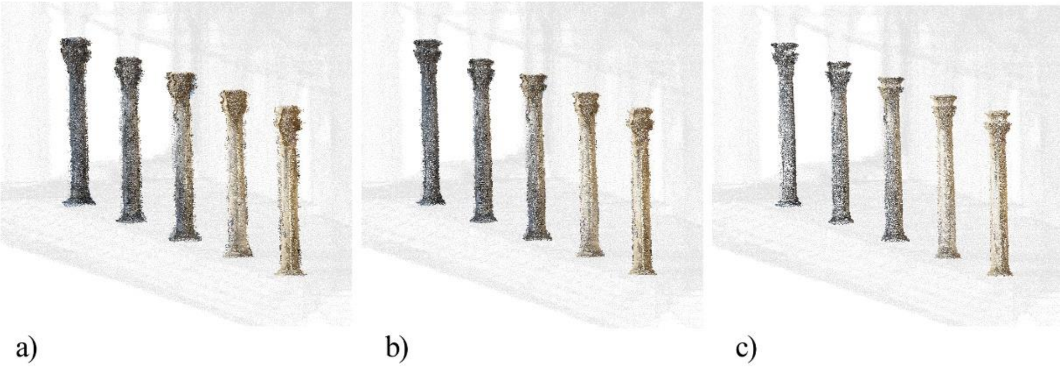


Figure 4.7 – Label transferring for (1_SC) columns, with three distance ratios: a) 0.1m, b) 0.05m, c) 0.02m.

Missing points integration. At the end of the processing some clouds have revealed some missing parts or some parts with a low point density. This issue could have been the consequence of (i) problems or errors during acquisitions, (ii) the presence of obstacles/obstructions or visibility constraints, (iii) the difficulty to match points between images with low-contrast or uniform textured surfaces. The absence of some elements could negatively affect the labelling projection, generating a low-quality image ground-truth. For this reason, the final stage consists in the *integration of the missing points,* in which the point cloud was fixed to make it more suitable for the labelling projection, adding the missing parts or increasing the density where required. In some cases, it was achieved integrating the photogrammetric reconstruction with the TLS point cloud, overlapping the two clouds where necessary. In other cases, when the problem occurs in both the clouds, this method is not applicable, and a reconstruction of the missing part is necessary. For this purpose, two tools were used. Rhinoceros®, a commercial 3D computer graphics and computer-aided design (CAD) application software, and Grasshopper®, a visual scripting language add-on for Rhinoceros. At first a subsampled reference point cloud was imported in the 3D software environment, and with various modeling tools the missing parts were filled out with meshes or NURBS surfaces. Secondly the meshes/surfaces were imported in CloudCompare, populated with points according to the required density, and finally added to the initial point cloud. Figure 4.8 shows an example of integration of points for the vaults of (1_SC) Spedale del Ceppo.

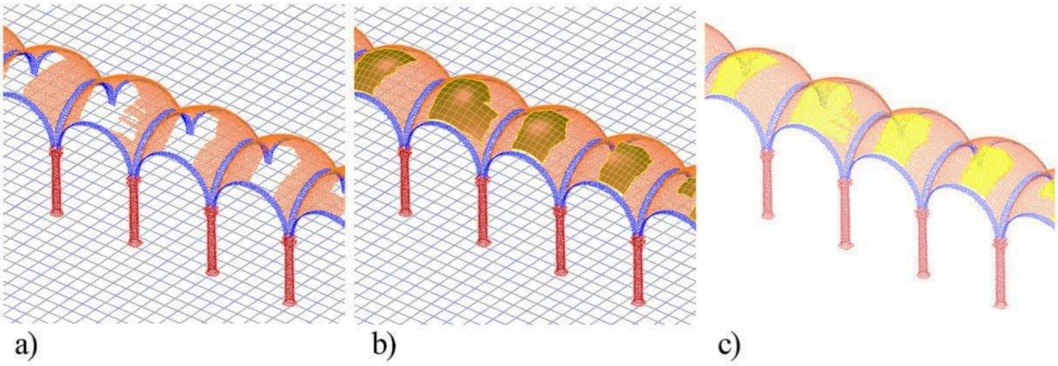


Figure 4.8 – Points integration workflow: a) import, b) mesh creation, c) points population.

4.3.4 Standards and Class Definition

The main aim of this dataset is to support the development tools for the automatic determination of heritage architectural elements starting from a raw point cloud. Automatic recognition allows to separate a constructive element from the wider building context, and to examinate all the geometric features necessary for the 3D reconstruction in a CAD environment. Since the main aim of these procedures is to support 3D model generation in a BIM environment, it is essential to choose the segmentation output according to the standards and the element category of the main BIM-based or object-oriented software (Simeone et al., 2019). Several standards have been developed, and they allow to guarantee the interoperability, the project continuity, and the interchange of information (Cursi et al., 2022). The main BIM interchange file format is the Industry Foundation Class (IFC), an open format founded by BuildingSMART in 1996, that allows interoperability between various software. CityGML is an open standardized format for storing and exchanging digital 3D models of cities and landscapes. It defines methods to describe the city object and their relationships, and it also define the concept of Level of Detail (LOD) of a 3D object. Other standards are the Building Topology Ontology (BOT), a minimal ontology for describing the core topological concepts of a building, or the Art & Architecture Thesaurus (ATT) of Getty Institute, a controlled vocabulary used for describing items of art, architecture, and material culture. One of the first work that deals with the issue of associating semantics in the domain of heritage element recognition using machine learning and deep learning techniques, was the work proposed by (Malinverni et al., 2019). The authors showed the results of the application of PointNet++ to heritage segmentation, using some scenes annotated according to the IFC format and CityGML standard in 9 classes. The authors in (Grilli & Remondino, 2020), following the idea proposed by this previous work, added other three classes*, moulding*, *drainpipe* and *other*. In (Croce et al., 2021), the authors identified an articulated set of segmentation categories, composed by seventeen classes, that reproduce the architectural decomposition illustrated by Scamozzi in the 17th century in the *'L'Idea Dell'Architettura Universale'.* One of the most recent and interesting work in this context, is the dataset developed in (Matrone et al., 2020). The authors introduced the ARCHdataset, in which the point clouds are labeled in 10 classes, according with the IFC file format, the CityGML Level of Detail 3/4, and the Art & Architecture Thesaurus (ATT) of the Getty Institute. They selected the following classes from all the three standards: arch, column, moulding, floor, door/window, wall, stair, vault, roof, other (Figure 4.9).

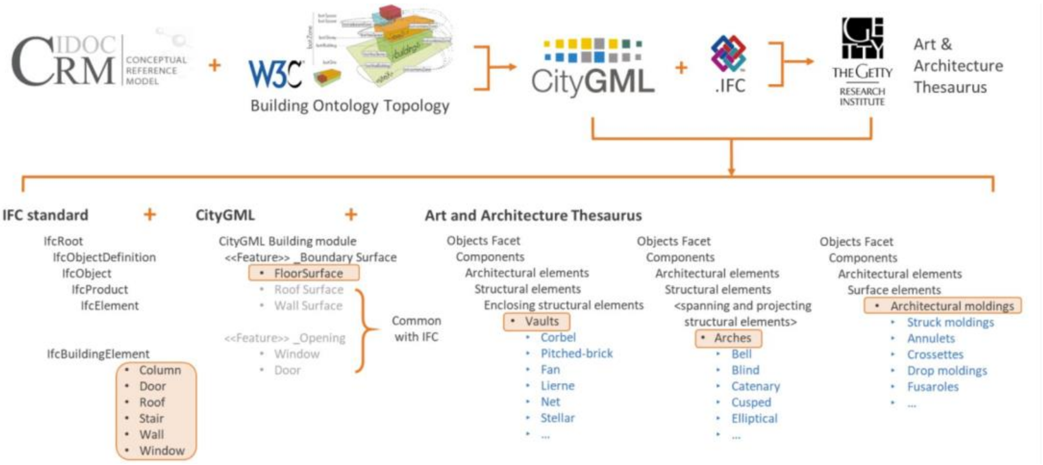


Figure 4.9 – Segmentation classes of the ARCHdataset (Matrone et al., 2020).

Given the heterogeneity of the architectural elements, the dataset also provides some annotation guidelines to allow other researchers to contribute to the expansion of the benchmark. The segmentation categories considered in this dataset are structured following the conventions and the guidelines defined in the ARCHdataset, hence the images of the benchmark will be segmented in 10 classes. Differently from point clouds, background is always present in the images, hence a new class was introduced:

it includes all the pixels that cannot be classified as part of the previously defined classes. Such class is conventionally named “background”. Consequently, the class “other” comprises all the elements that are not included in the previous classes, but, at the same time, are included in or close to the building. Figure 4.10 shows an example of image segmented according to these classes.

To the best of the current knowledge, ARCHdataset is currently the only benchmark realized to deal with point cloud-based machine and deep learning tools in the heritage field, and it is promoting crowdsourcing to enrich the already annotated scene. Consequently, the choice of maintaining the same class definition of ARCHdataset can be convenient for: (i) comparing the performance of an approach in both datasets, (ii) enabling a potential integration of the two datasets, hence increasing the number and typologies of labelled buildings, (iii) promoting the development of new image-point based models. Nevertheless, this dataset may be distributed in the future also with annotations made according to different class definitions or standards, according with a different Level of Detail, or according with different task, such as object detection or instance segmentation. Regarding this last point, making the dataset suitable for instance segmentation could be an interesting future development. However, the annotation procedure in this case is longer and more challenging: indeed, each instance of a category should be segmented with a specific pixel-level annotation, and each object or element requires a bounding box to define its relative position in the image.

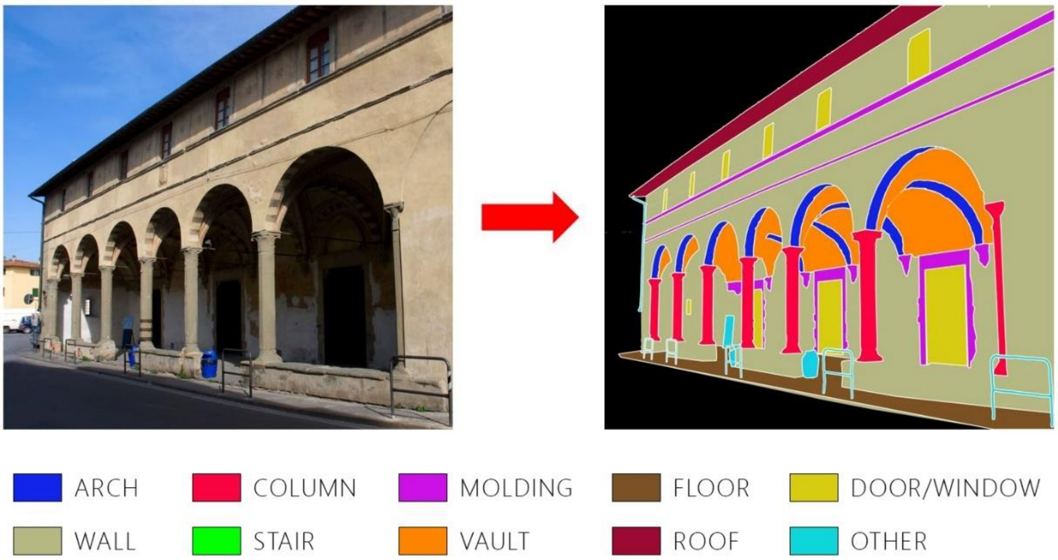


Figure 4.10 – An image segmented according to ARCHdataset guidelines.

4.3.5 Labelling Projection Procedure

As already mentioned, a fundamental characteristic of a machine learning dataset is the *quantity*, and training a NN semantic segmentation model in particular, requires assembling a very large dataset, composed by thousands of different labelled images. For this reason, it is fundamental to choose an appropriate tool to speed-up the annotation phase. There are several labelling tools for various Computer Vision tasks, and here are reported some of the most popular. *Labelme,* can be used for various tasks, but since it involves manual labelling is suitable for small dataset. *LabelImg,* fits for object detection task, and it is easy and fast to use. *Hasty.ai*, has a built-in assistant that allows to automatize the labelling starting from 10 manually annotated images. *CVAT,* is suitable for teams looking to automate labelling with their model. *Labelbox* a versatile platform with useful label functionality. Other tools are *V7, SuperAnnotate, Dataloop, Scale AI,* and *SAM.* Every mentioned platform or tool has its key features or functionalities, and its special tools to speed-up and to make easier the annotation, but, in every case, the manual operations are still essential, and they always require long time and tedious manual processing, especially to segment complex and articulated scene such as the heritage building ones. To give an idea, the manual labelling of an image of the proposed dataset typically requires between 1h and 3h of a specialized operator work, depending on the scene complexity. A medium-scale dataset should be composed by 3000-4000 images, and, according to this timing, the dataset construction would require around 3 man-years. Hence, the development of an automatic image labelling procedure is essential to reduce the time to produce the ground-truth labelling and minimize the time-consuming manual operations. The developed procedure that will be described, allows to automatically project the labels set manually on the 3D space of the photogrammetric point cloud directly on the totality of the images used to generate the dense point cloud. Despite the segmentation of the 3D scene is still a manual and tedious operation, the procedure leads to a considerable saving of time, enabling the annotation of hundreds of images from just one labelled 3D scene. For instance, considering the scenes composing the dataset, their annotation required on average from 8h to 24h of manual work each. Each scene allowed to automatically annotate around 400 images that, according with the above-mentioned timing, they would have required from 400h to 1200h of manual work. It should be noted that the method has two weak points. Firstly, despite the large number of annotated images, they are representative of only one scene from different angles and perspective. Consequently, the images lack in generalization and they are representative of a small range of building typologies. Secondly, the direct manual segmentation of the image is more accurate and precise, and it allows to carefully process each image and its peculiarity. The automatic procedure leads inevitably to a loss of quality. The first issue can be address acquiring and introducing as many scenes as possible, also integrating existing datasets and already annotated scenes. The second issue will be analysed and discussed in the next sections, and the performance of the procedure will be assessed and compared with the manual segmentation.

In order to establish a precise correspondence between the points in the 3D space and their projection in the 2D image plane, a mathematical model should be defined. The most common and easiest geometric camera model is the *pinhole* one (Figure 4.11).

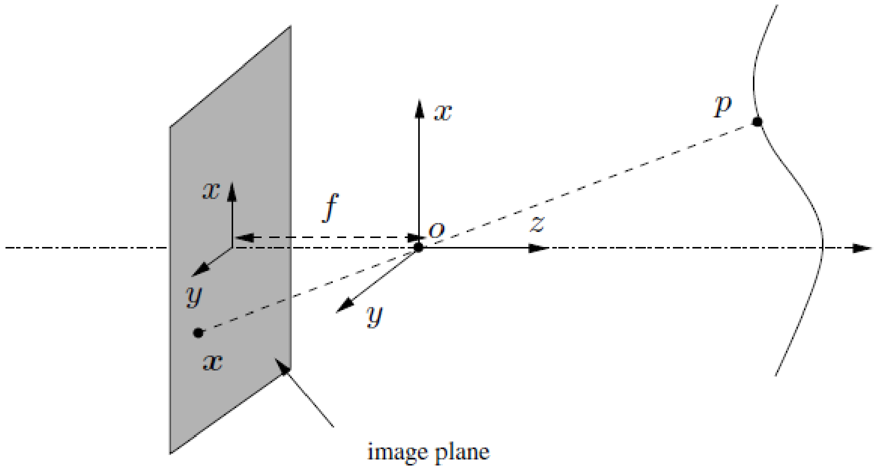


Figure 4.11 - Camera Pinhole model.

Given a point *p* of coordinate $[X, Y, Z]^T$, and its projection on the image plane *x* of coordinate $[x, y]^T$, their relationship is described by the following expression, called ideal *perspective projection:*

$$x = -f\frac{X}{Z}, \qquad y = -f\frac{Y}{Z} \tag{4.1}$$

where *f* is referred to as the *focal length*. By using homogeneous coordinates, the projection can be expressed with a linear mapping:

$$Z\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} -fX \\ -fY \\ Z \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.2}$$

A realistic camera model that describes the transformation from 3D coordinates to 2D pixel coordinates, in addition to the prospective transformation, should consider two other aspects: (i) the rigid transformation between the scene and the camera, and (ii) the pixelization, including the CCD sensor shape and dimension, and its position with respect to the optical centre. The generic geometric relationship between a point of coordinates $\boldsymbol{X_0} = [X_0, Y_0, Z_0, 1]^T$ relative to the world frame and its pixel coordinates $\boldsymbol{x'} = [x', y', 1]^T$is captured by the following equation:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} [R \quad t] \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} \tag{4.3}$$

Where s_x and s_y are the size in metric units of the pixels in the *x* and *y* directions, o_x and o_y are the pixel coordinates of the principal point relative to the image reference frame, and s_θ is the *skew factor*. $\boldsymbol{R}$ and $\boldsymbol{t}$ are respectively the rotation matrix and the translation vector that describe the rigid-body transformation between the camera coordinates and the world coordinates. The expression is written with an arbitrary positive scalar $\lambda \in \mathbb{R}_+$. In matrix form the expression could be written as follow:

$$\lambda \boldsymbol{x'} = [KR, Kt]\boldsymbol{X}_0 \tag{4.4}$$

where $\Pi = [KR, Kt]$ is generally called *projection matrix*. The matrix K collects all the parameters that are 'intrinsic' to a particular camera, and is called *intrinsic parameter matrix*, or *calibration matrix* of the camera. R and t represent the external parameters of the camera and are called *extrinsic parameters* of the camera. The camera pinhole is an ideal model, and in real applications the use of wide-angle lenses may introduce a deviation from rectilinear projection and the problem of distortion, which can make the pinhole model fail. Various lens distortion models have been proposed in literature to correct such distortion. In this procedure the Brown-Conrady model has been used, and it corrects both for radial distortion and for tangential distortion. According to this mathematical model, the procedure was developed with Matlab®, with the support of the Camera Calibration Toolbox and the Computer Vision Toolbox. The procedure requires two main inputs: (i) the segmented 3D scene, and (ii) the camera parameters of the images used to generate the scene, calculated during the photogrammetric workflow. It outputs a set of images, corresponding to the original ones, but containing the ground-truth labelling, obtained according to the 3D scene annotations. This procedure is thought for the photogrammetric point clouds whose intrinsic and extrinsic camera parameters are known and already calculated during the photogrammetric workflow.

Let us consider the generation of just one of the output (annotated) images, and, for the sake of notation simplicity, let us assume that the image distortion has already been corrected. The process, which is basically a label projection procedure from the annotated 3D scene to the image plane, is composed by several steps that are designed in order to take into account the point visibility and possible issues related to the too low point cloud density on certain locations. First, the 3D points are projected on the image plane according to (4.4). Ideally, in this way the labels should be reported on the corresponding locations on the $N \times M$ output image. However, in the output image generation procedure the following rules should be taken into account:

(i) the point should be projected, and hence its label potentially transferred to the output image, only if it is in front of the camera, e.g. positive coordinate along the optical axis.

(ii) locations computed with (4.4) are rounded to the closest integers and only those within the image domain are considered.

(iii) the point should be considered only if not obstructed by other objects (e.g. points) in the scene.

(iv) gaps in the output image, if present, should be filled in accordance with the labels in the neighbours of the consider location.

The first two rules can be easily checked as follows:

$$(i)\ Z > 0, \qquad (ii)\ 0 \leq u \leq M - 1, 0 \leq v \leq N - 1 \tag{4.6}$$

Indeed, some more effort is needed for the remaining two.

In order to consider possible obstructions, if two different 3D points are mapped on the same 2D image position, then the value of the point closer to the camera is chosen. Therefore, the 3D point labels are transferred to the 2D image based on a distance hierarchy, starting from the closer points. Not sufficiently dense point clouds may lead to gaps on the image where the labels have been projected. For this reason, the area

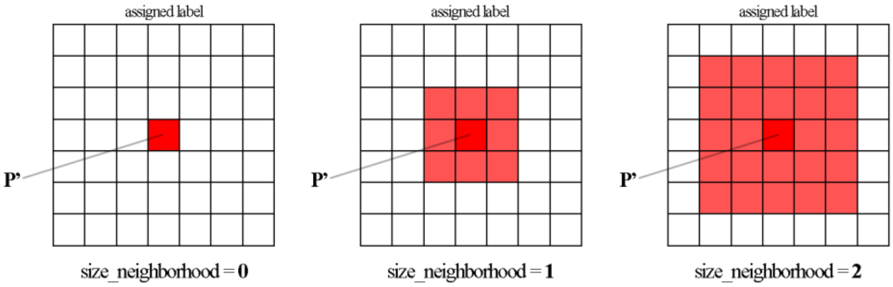


Figure 4.12 – The functioning of the *size_neighborhood* parameter.

covered by a point is enlarged from one pixel to (2 *size_neighborhood* +1) × (2 *size_neighborhood* +1) pixels. In practice, to increase the number of image pixels covered by a 3D point on the image, the procedure allows to fill also the corresponding neighbourhood, up to distance threshold named *size_neighborhood*, as shown in Figure 4.12. While the assignment proceeds, if a pixel label has already been set, it is not modified, because in order to take into account of the occlusions labels of the points closer to the camera are considered first. Hence, the parameter *size_neighborhood* also determines to which extent the foreground points hide the back positioned ones. At the end of the procedure, when all the *N* points are assessed and transferred, if a label has not been assigned to a pixel, its class is set to 'none'. Figure 4.13 shows the output of the procedure with 3 different values of the parameter *size_neighborhood* for the same image. The three outputs reveal the benefits of the neighbourhood filling. It allows to decrease the number of unclassified pixels addressing the issue of point density, and it allows to properly hide the back positioned elements improving the assessment of occlusions due to points in the foreground. The value of this parameter has a remarkable impact on the quality of the ground-truth image segmentation generation: the choice of a proper value for such parameter is fundamental to obtain a high-quality output. Small values guarantee high labelling precision but, at the same time, a high percentage of unclassified pixels. Large values guarantee a good covering of the image, but they could cause labelling inaccuracy, making the occlusion problem worse than in reality. Its choice and impact will be assessed in the next sections. Despite the use of this parameter usually improves the output quality, in certain cases the generated image ground-truth may still show some inaccuracies, mainly caused by (i) unwanted non-classified pixels, and (ii) isolated pixels with incorrect label. These issues are mostly related to the low density of the point cloud, which affects in particular the pixels close to the camera, to missing parts of the point cloud, that could reveal hidden elements, and to inaccuracy and noise that bias the projection with incorrect labels. To overcome

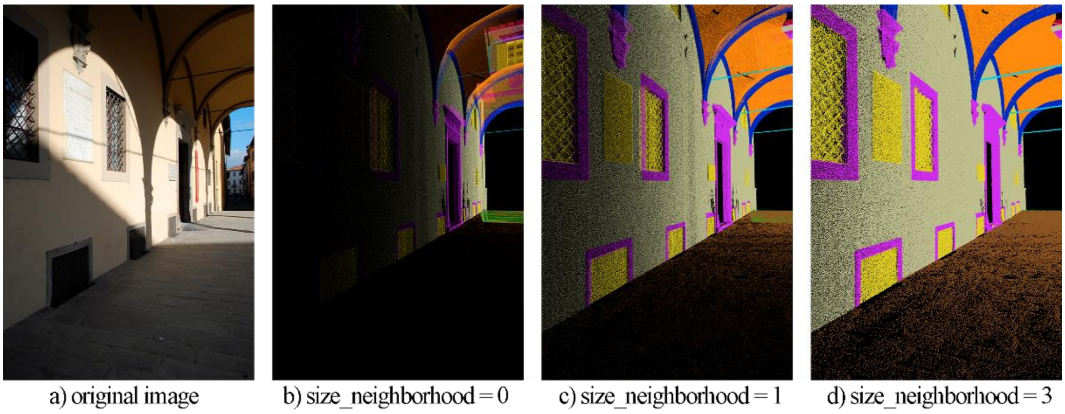


Figure 4.13 – The influence of the *size_neighborhood* on the procedure output.

these issues and improve the quality of the final ground-truth, the output is processed by two other functions.

The first function, *SubstituteWithMostPopularVote( ),* allows to address the problem of the isolated areas of pixels with incorrect annotation, by substituting labels in small connected regions, with the most popular label in their neighbourhood. The process is regulated by two parameters *size_area* and *size_neighborhood,* and their functioning is shown in Figure 4.14.

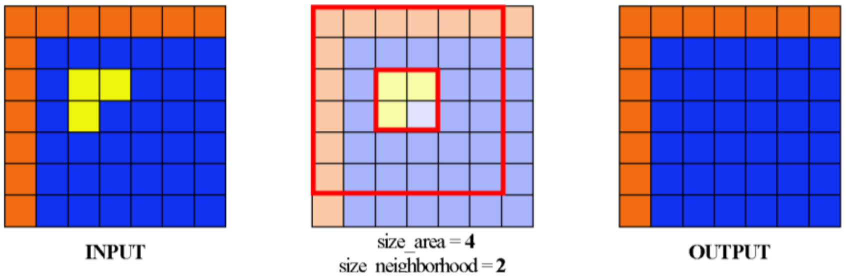


Figure 4.14 – The functioning of the *SubstituteWithMostPopularVote( )* function.

A uniform label over a large image region is expected to be a quite reliable class projection. Instead, there is a quite high chance that the presence of a certain label over a very small region is incorrect, e.g. error due to noise. For this reason, *Size_area* determines the maximum size of the areas that have to be checked for label substitution: if the area of the connected region associated to a certain label value is smaller than the *Size_area* threshold, then such region is considered for a label substitution. Instead, *Size_neighborhood* selects the size of the neighbourhood area, i.e. the area that has to be considered in order to determine which is the label to be used for the substitution, i.e. the most frequently label in the neighbourhood. If at least one of the pixels of the region is not classified with 'none', the most popular label of that neighbourhood region is assigned to the investigated area, whereas if the most popular label is 'none', the second popular label is assigned.

The second function, *FillWithMostPopularVote( ),* allows to fill small areas of pixels classified as 'none'. As in the previous function, the parameters that rule the process are *size_area* and *size_neighborhood*, which have roles similar to those described before: they indicate the maximum size of an isolated area to be checked, and the size of the neighbourhood region. Even in this case, the most popular label, excluding 'none', within the neighbourhood region is assigned to the unclassified pixels. Figure 4.15 shows the final output of the procedure after the use of the two functions, called with different parameter value combinations, keeping the initial *size_neighborhood* fixed to the value of 5 pixels. As shown in Figure 4.15, the use of the two functions has a quite positive effect on the quality of the image annotations, decreasing the percentage of unlabelled regions, removing noisy pixels, and regularizing the general label map.

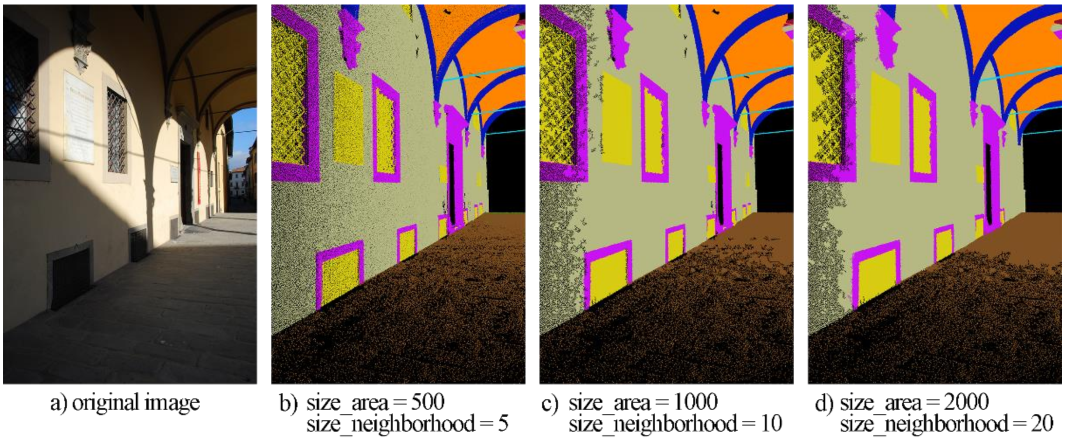


Figure 4.15 – The influence of the two functions on the final procedure output.

To preserve the quality and the accuracy of the first generated label, the value of the two parameters needs to be carefully assigned. Increasing their value, the isolated areas and the unlabelled pixels decrease remarkably, but, at the same time, they could cause a loss of information, especially favouring and increasing the labels of the large areas at the cost of the small elements. This issue is shown for instance, in Figure 4.15d, in which, increasing excessively the parameter values, the small windows and mouldings at the end of the loggia disappear, and they are classified like wall.

Given the remarkable influence of the parameters on the final quality of the ground, several tests were performed to find the optimal value combination, in order to guarantee the highest quality level and, at the same time, as many pixels as possible correctly annotated. In addition to several visual analysis on a wide range of images, a more rigorous test was performed by comparing two fine-manually annotated images

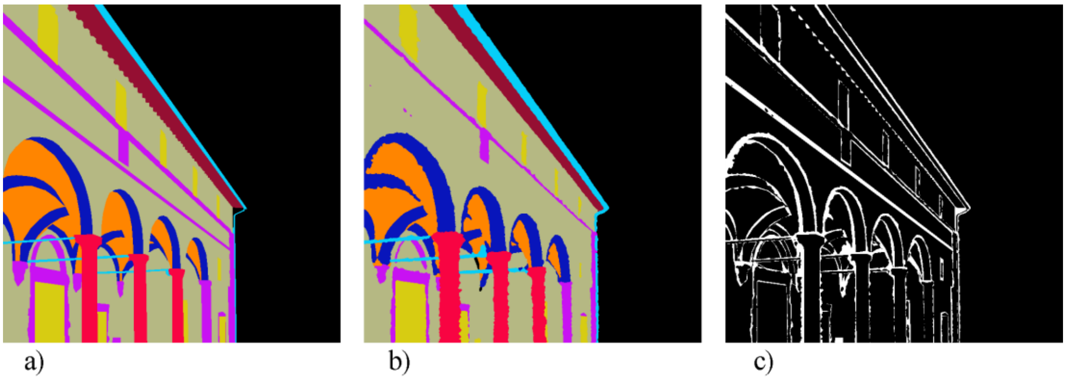


Figure 4.16 – Automatic labelling assessment: a) fine-manually segmented image, b) automatic segmented image, c) overlay between the two images.

with several generations obtained with different parameter combinations and assessing the percentage of correctly annotated pixels. Figure 4.16 shows an example of manually segmented image (Figure 4.16a), the corresponding automatically segmented image (Figure 4.16b), and the overlay-difference between the two: in black the pixels classified in the same way, whereas pixels wrongly classified by the automatic labelling procedure are shown in white (Figure 4.16c). Considering the two testing images, the highest percentage of classified pixels obtained was around 96-97%, which turned out to be useful to define an optimal range of the value of the various parameters. Table 4.4) reports the optimal ranges of the parameters obtained from the test, which will be used for the further generation of the dataset. However, the range is still wide, and the precise value of the parameters will be set depending on the specific scene or depending on the single images.

Table 4.4 – Optimal range of the parameters of the labelling procedure

Labelling	**SubstituteWithTheMostPopular**		**FillWithTheMostPopular**	
Size (pixels)	Area (pixels)	Size (pixels)	Area (pixels)	Size (pixels)
2-4	500-1500	5-15	500-3000	5-30

In conclusion, the procedure has shown a positive overall performance, proving an outstanding save of time, which justifies the small loss of accuracy of the automatic labelling compared with a manual annotation. However, the results have still shown some issues and limitations, which are mainly caused by: (i) low density of the point cloud, that affects in particular the image areas close to the camera, (ii) missing points in some parts of the building, that could expose hidden points or elements, (iii) local noise of the point cloud, which could generate incorrect classified pixels, (iv) presence of objects and obstacles in front of the buildings, present in the images but not properly reconstructed in the point cloud, and (v) local differences between the LiDAR and the photogrammetric cloud, which, during the label transferring between the two clouds, cause incorrect classified points. Some of these issues, (i), (ii), (iii) in particular, could be easily corrected automatically during the procedure with a proper choice of the transfer parameters, which can be adapted depending on the scene, or depending on the single images. The issues describe in (iv) and (v), could be addressed by a careful processing of the point cloud, including the reconstruction of the background buildings and elements, and integrating the missing part with synthetic points if necessary. However, despite the correction parameters, the final accuracy is strictly related to the quality of the initial point cloud, and it is worth generally to spend more time processing the initial point cloud than to work directly on the parameter setting. At the end of the entire process, for each image of the survey the procedure output the ground-truth with the size of $N \times M$ pixels, and the relative undistorted input RGB image with the same size, both in a *.png* or *.jpeg* file format. Such generated images are ready to feed a machine learning system, and to train, validate and test a semantic segmentation model.

4.3.6 Dataset Trend and Statistics

In this section the structure and the composition of the dataset will be illustrated. For each building composing the dataset three types of data are available: (i) the manually annotated TLS point cloud, (ii) the annotated photogrammetric cloud obtained with annotation transfer, and (iii) the survey images annotated with the labelling projection procedure. At first, both the input point clouds will be shown together with the generated ground-truth. The number of points for each class are reported in the following tables and histograms, along with the percentage of the class on the total number of points of the scene for both the clouds (Fig 4.18 - 4.26). These data are useful to reveal the class balancing in the scene, and to assess the proper functioning of the annotation transfer between the point clouds, both TLS and photogrammetric. Secondly, the results of the labelling procedure will be illustrated for each building, and even in this case the number of pixels for each class and the percentage of the class on the total number of pixels will be shown (Fig. 4.27 - 4.36).

The final structure of the dataset will be organized following the standards of the main semantic segmentation datasets. A set of RGB images and the corresponding set of labelled images with the same size, both in a *.png* file format will be provided. The labels in the ground-truth file are compatible with those of the ARCHdataset: 0 arch, 1 column, 2 moulding, 3 floor, 4 door/window, 5 wall, 6 stair, 7 vault, 8 roof, 9 other. Differently from point clouds, background is always present in the images, hence a new class was introduced: it includes all the pixels that cannot be classified as part of the previously defined classes. Such class is conventionally named "background", and it is labelled with the index 10.

The starting size of the images is 2592x3872 pixels with a resolution of 300dpi. The generation of the ground-truth was performed while maintaining the initial size of the images. On one hand, that choice allows to guarantee the highest accuracy and versatility of the dataset, giving the possibility to resize the image just before inputting them into the machine learning model, and finding the optimal size during the training. On the other hand, it allows to perform other additional processing operations, such as cropping or rotation, without a remarkable loss of resolution. However, the size of the generated images could be variable, allowing the process of any type of images acquired with different cameras or sensors, and making easier the integration and the extension of the dataset with new data. In the next pages the details of each building are illustrated.

(1_SC) Spedale del Ceppo

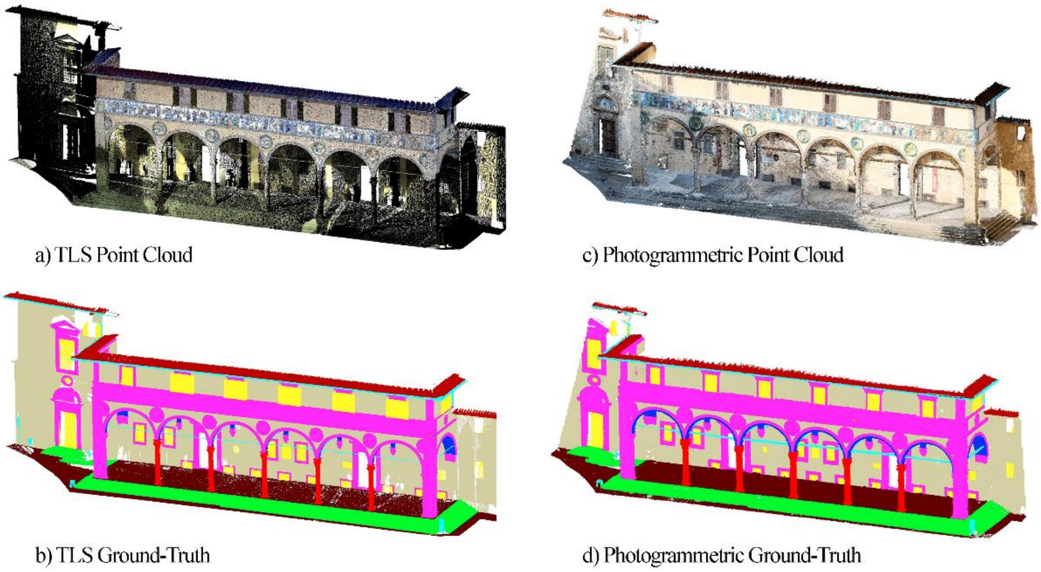

a) TLS Point Cloud
c) Photogrammetric Point Cloud
b) TLS Ground-Truth
d) Photogrammetric Ground-Truth

Figure 4.17 – (1_SC) Spedale del Ceppo Point Clouds: a) TLS PC, b) TLS ground-truth, c) photogrammetric PC, d) photogrammetric ground-truth.

DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	487056	4,9
1	COLUMN	205126	2,0
2	MOLDING	2099585	21,0
3	FLOOR	678706	6,8
4	DOOR/WINDOW	822276	8,2
5	WALL	2795178	27,9
6	STAIRS	586956	5,9
7	VAULT	1021416	10,2
8	ROOF	1067152	10,7
9	OTHER	254665	2,5
TOTAL POINTS		10018116	

DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	566459	9,1
1	COLUMN	208017	3,3
2	MOLDING	1036344	16,6
3	FLOOR	842980	13,5
4	DOOR/WINDOW	222478	3,6
5	WALL	1580062	25,4
6	STAIRS	388629	6,2
7	VAULT	848257	13,6
8	ROOF	378933	6,1
9	OTHER	152748	2,5
TOTAL POINTS		6224907	

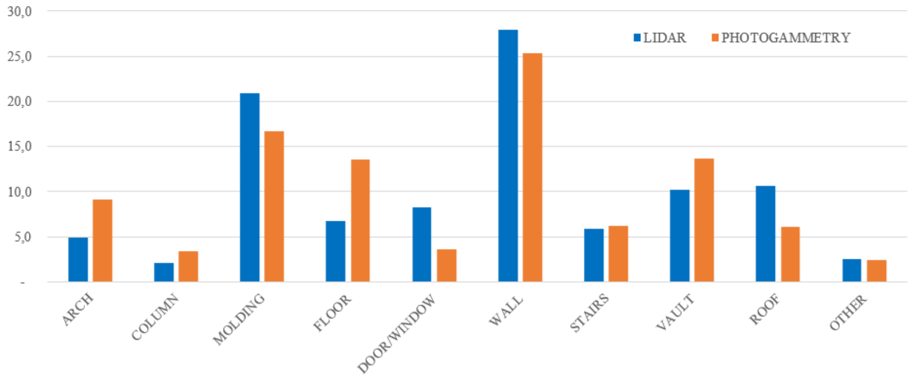
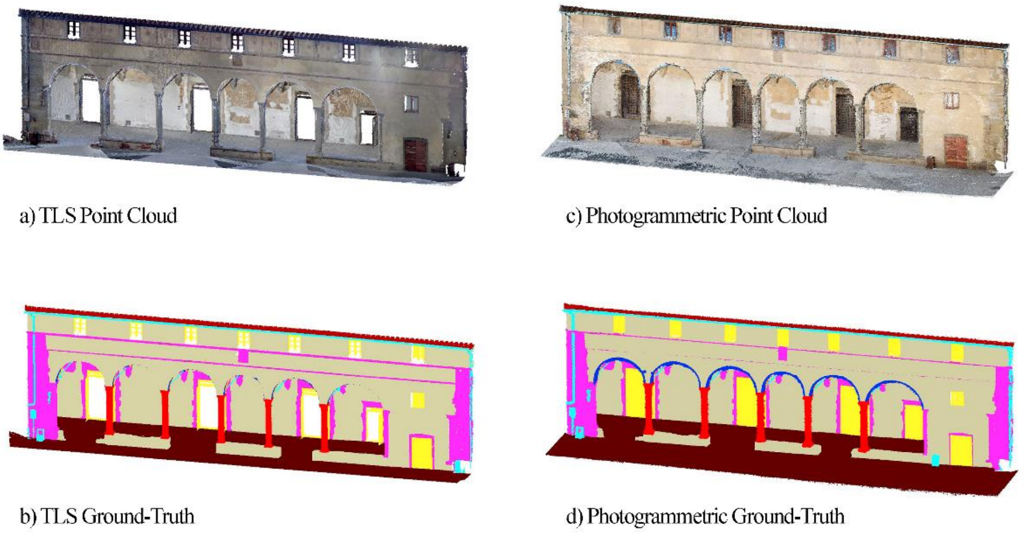


Figure 4.18 – (1_SC) Class distribution and percentage on the two clouds: TLS (blue) and Photogrammetric (Orange).

(2_OSA) Ospedale Sant'Antonio



a) TLS Point Cloud

c) Photogrammetric Point Cloud

b) TLS Ground-Truth

d) Photogrammetric Ground-Truth

Figure 4.19 – (2_OSA) Ospedale Sant'Antonio Point Clouds: a) TLS PC, b) TLS ground-truth, c) photogrammetric PC, d) photogrammetric ground-truth.

DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	293182	5,3
1	COLUMN	165146	3,0
2	MOLDING	378204	6,9
3	FLOOR	1229459	22,4
4	DOOR/WINDOW	344931	6,3
5	WALL	1737287	31,6
6	STAIRS	0	-
7	VAULT	1045995	19,0
8	ROOF	188913	3,4
9	OTHER	108863	2,0
TOTAL POINTS		5491980	

DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	485007	6,8
1	COLUMN	240692	3,4
2	MOLDING	348515	4,9
3	FLOOR	2170941	30,5
4	DOOR/WINDOW	595890	8,4
5	WALL	1956554	27,5
6	STAIRS	0	-
7	VAULT	898703	12,6
8	ROOF	239507	3,4
9	OTHER	187739	2,6
TOTAL POINTS		7123548	

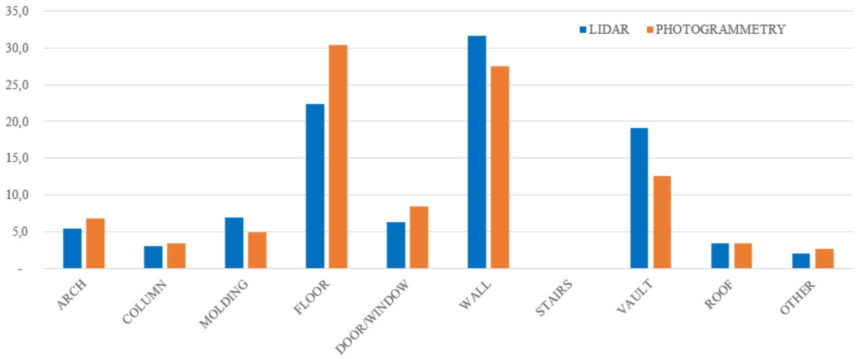
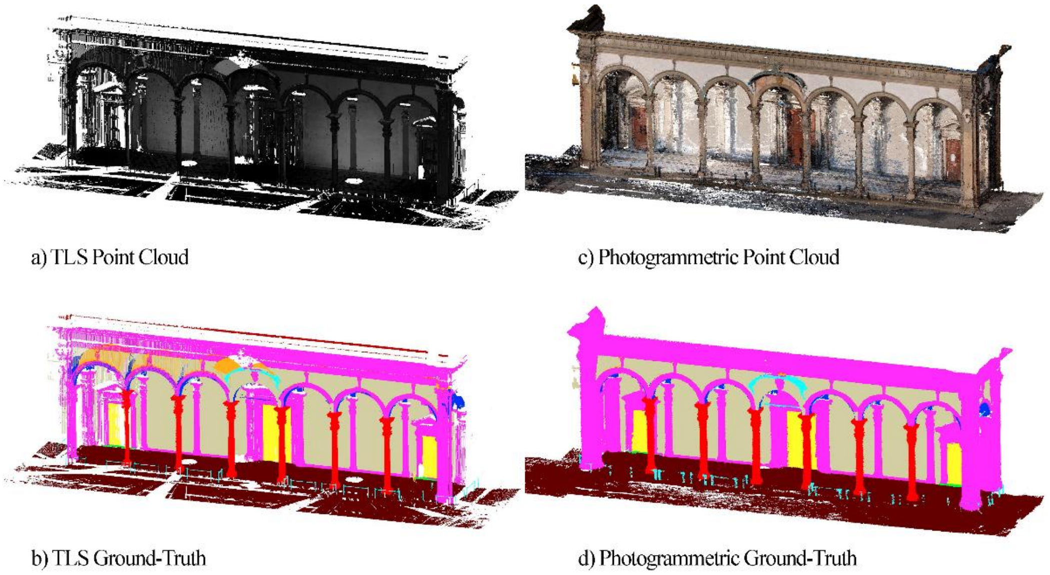


Figure 4.20 – (2_OSA) Class distribution and percentage on the two clouds: TLS (blue) and Photogrammetric (Orange).

(3_SSA) Basilica della Santissima Annunziata



a) TLS Point Cloud

c) Photogrammetric Point Cloud

b) TLS Ground-Truth

d) Photogrammetric Ground-Truth

Figure 4.21 – (3_SSA) Basilica Santissima Annunziata Point Clouds: a) TLS PC, b) TLS ground-truth, c) photogrammetric PC, d) photogrammetric ground-truth.

DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	411232	6,7
1	COLUMN	380121	6,2
2	MOLDING	1056396	17,1
3	FLOOR	1530102	24,8
4	DOOR/WINDOW	111974	1,8
5	WALL	1242500	20,2
6	STAIRS	5090	0,1
7	VAULT	1200098	19,5
8	ROOF	188913	3,1
9	OTHER	37419	0,6
TOTAL POINTS		**6163845**	

DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	720187	6,5
1	COLUMN	681656	6,2
2	MOLDING	3508295	31,8
3	FLOOR	2623182	23,8
4	DOOR/WINDOW	302887	2,7
5	WALL	1812068	16,4
6	STAIRS	12750	0,1
7	VAULT	1229262	11,1
8	ROOF	0	-
9	OTHER	149494	1,4
TOTAL POINTS		**11039781**	

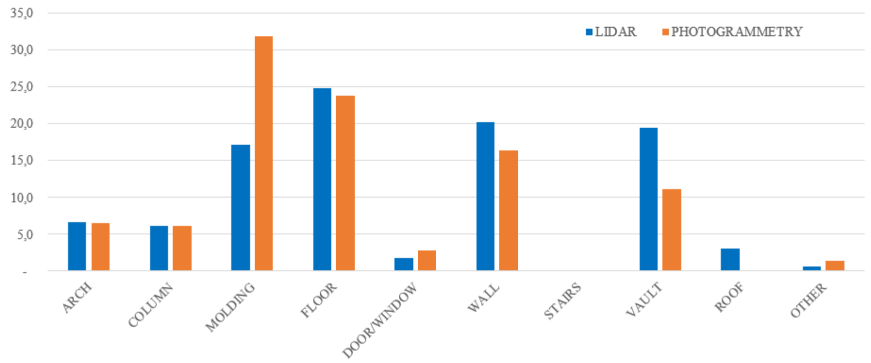


Figure 4.22 – (3_SS) Class distribution and percentage on the two clouds: TLS (blue) and Photogrammetric (Orange).

(4_CG) Certosa del Galluzzo

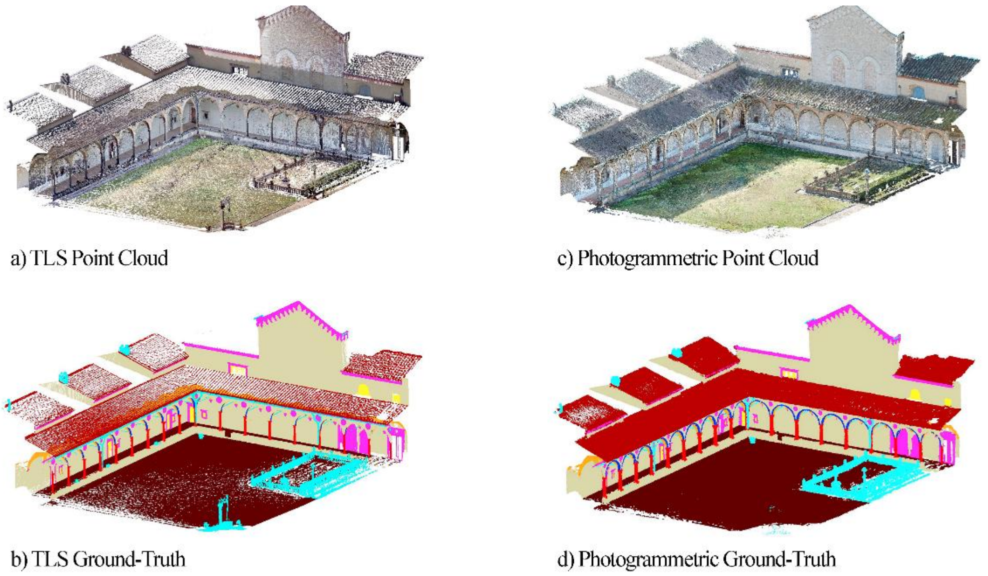
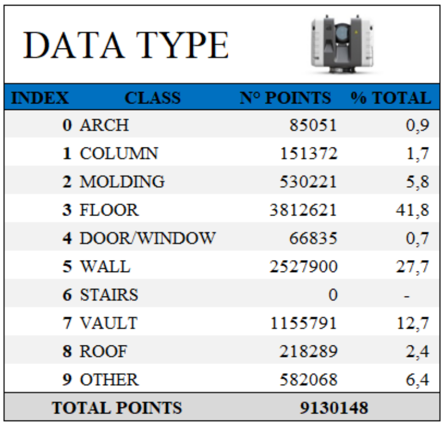
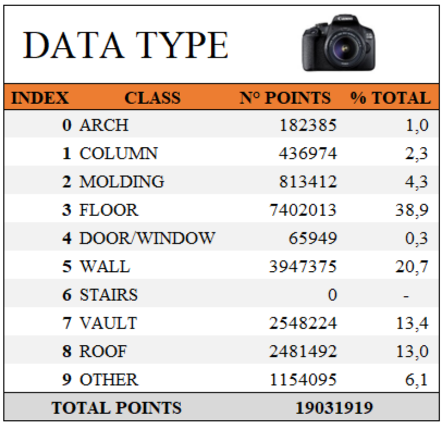

a) TLS Point Cloud
c) Photogrammetric Point Cloud
b) TLS Ground-Truth
d) Photogrammetric Ground-Truth

Figure 4.23 – (4_CG) Certosa del Galluzzo Point Clouds: a) TLS PC, b) TLS ground-truth, c) photogrammetric PC, d) photogrammetric ground-truth.



DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	85051	0,9
1	COLUMN	151372	1,7
2	MOLDING	530221	5,8
3	FLOOR	3812621	41,8
4	DOOR/WINDOW	66835	0,7
5	WALL	2527900	27,7
6	STAIRS	0	-
7	VAULT	1155791	12,7
8	ROOF	218289	2,4
9	OTHER	582068	6,4
TOTAL POINTS		9130148	



DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	182385	1,0
1	COLUMN	436974	2,3
2	MOLDING	813412	4,3
3	FLOOR	7402013	38,9
4	DOOR/WINDOW	65949	0,3
5	WALL	3947375	20,7
6	STAIRS	0	-
7	VAULT	2548224	13,4
8	ROOF	2481492	13,0
9	OTHER	1154095	6,1
TOTAL POINTS		19031919	

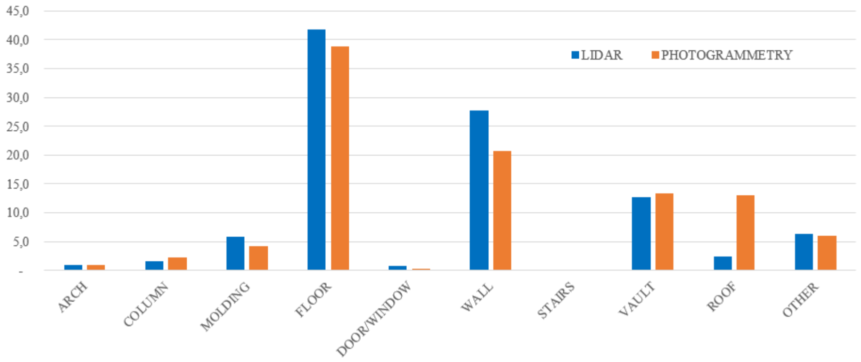
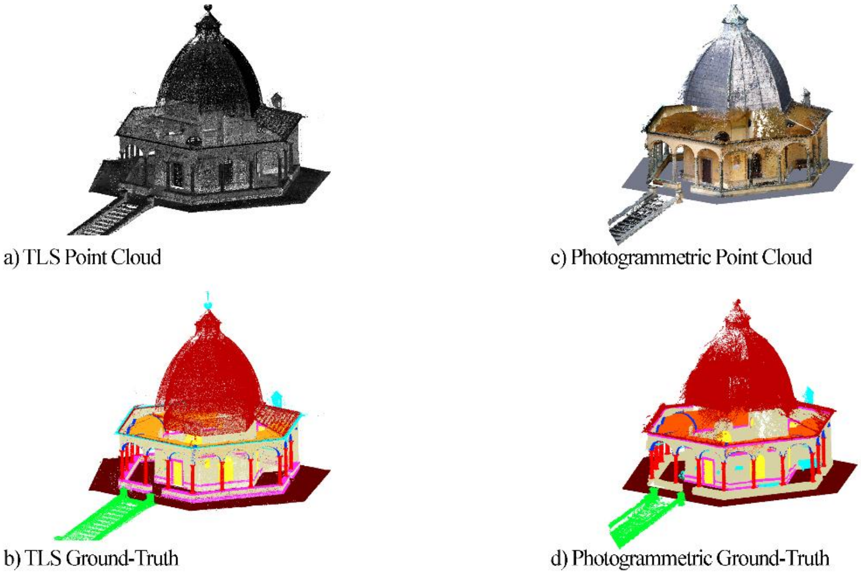


Figure 4.24 – (4_CG) Class distribution and percentage on the two clouds: TLS (blue) and Photogrammetric (Orange).

(5_CB) Cappella Buontalenti



a) TLS Point Cloud

c) Photogrammetric Point Cloud

b) TLS Ground-Truth

d) Photogrammetric Ground-Truth

Figure 4.25 – (5_CB) Cappella Buontalenti Point Clouds: a) TLS PC, b) TLS ground-truth, c) photogrammetric PC, d) photogrammetric ground-truth.

DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	40547	2,7
1	COLUMN	57916	3,8
2	MOLDING	85419	5,6
3	FLOOR	313715	20,6
4	DOOR/WINDOW	57679	3,8
5	WALL	341168	22,4
6	STAIRS	115132	7,5
7	VAULT	201880	13,2
8	ROOF	283288	18,6
9	OTHER	29615	1,9
TOTAL POINTS		**1526359**	

DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	216536	3,1
1	COLUMN	335246	4,8
2	MOLDING	296770	4,2
3	FLOOR	1323182	18,9
4	DOOR/WINDOW	182695	2,6
5	WALL	2160060	30,9
6	STAIRS	121070	1,7
7	VAULT	1305650	18,6
8	ROOF	899064	12,8
9	OTHER	160993	2,3
TOTAL POINTS		**7001266**	

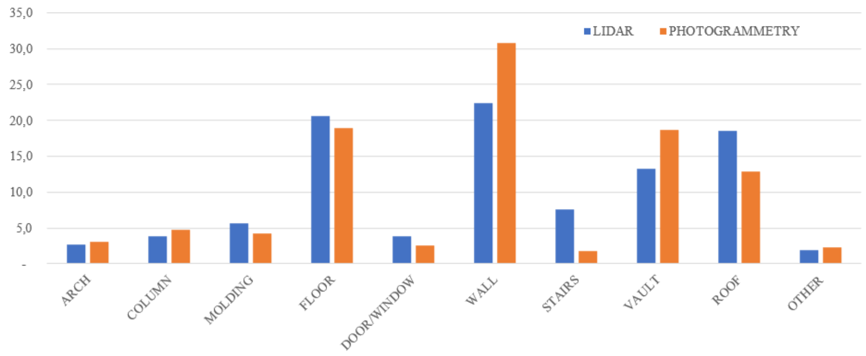


Figure 4.26 – (5_CB) Class distribution and percentage on the two clouds: TLS (blue) and Photogrammetric (Orange).

(1_SC) Spedale del Ceppo

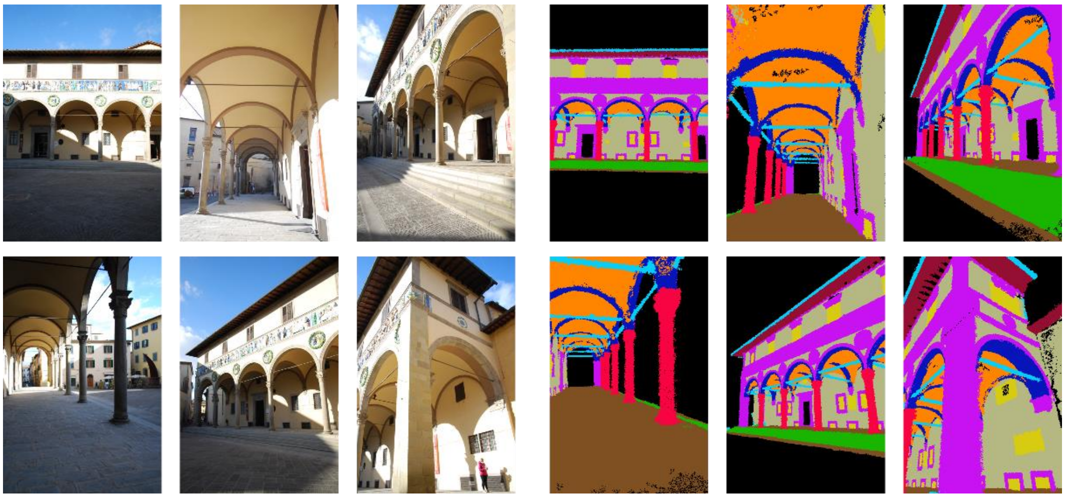


Figure 4.27 – (1_SC) Spedale del Ceppo: RGB images and generated ground-truth.

SETTINGS	
Label Transfering	
size_neighborhood	4
SubstituteWithMostPopular	
area	500
size_neighborhood	5
FillWithMostPopular	
area	500
size_neighborhood	5
Output	
resolution	2592x3872
format	*.png*
n° of images	**748**

DATA TYPE

INDEX	CLASS	N° PIXELS	% TOTAL
0	ARCH	14619387	4,9
1	COLUMN	8958437	3,0
2	MOLDING	35733288	11,9
3	FLOOR	33927627	11,3
4	DOOR/WINDOW	7304201	2,4
5	WALL	50042597	16,6
6	STAIRS	5986167	2,0
7	VAULT	36530717	12,1
8	ROOF	4717237	1,6
9	OTHER	7404198	2,5
10	NONE	95640444	31,8
TOTAL POINTS		**300864300**	

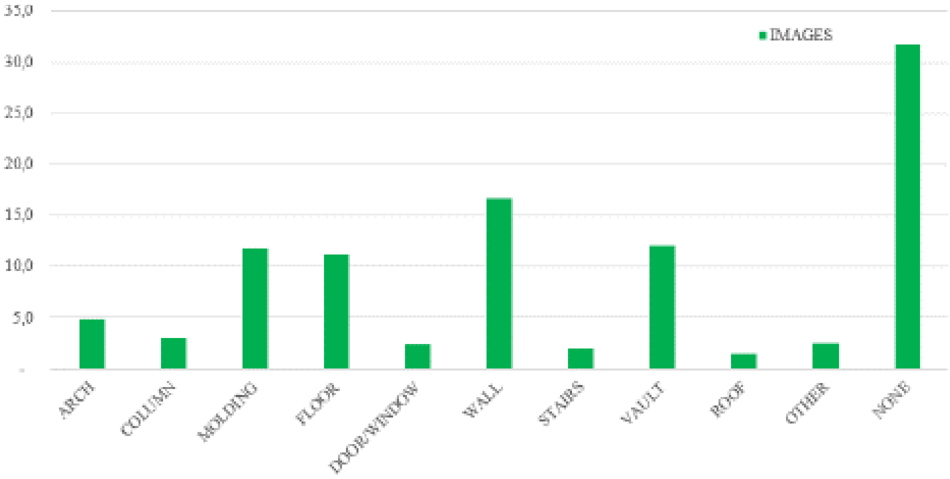


Figure 4.28 – (1_SC) Class distribution and percentage of the building images.

(2_OSA) Ospedale Sant'Antonio

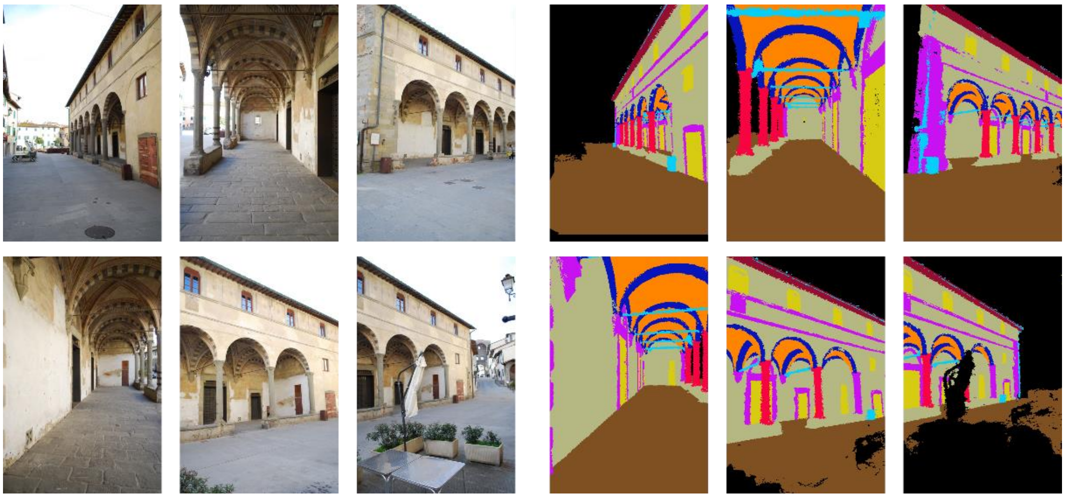


Figure 4.29 – (2_OSA) Ospedale del Ceppo: RGB images and generated ground-truth.

SETTINGS

Label Transfering	
size_neighborhood	3
SubstituteWithMostPopular	
area	500
size_neighborhood	5
FillWithMostPopular	
area	500
size_neighborhood	5
Output	
resolution	2592x3872
format	*.png*
n° of images	**755**

DATA TYPE

INDEX	CLASS	N° PIXELS	% TOTAL
0	ARCH	20267791	6,7
1	COLUMN	13215531	4,4
2	MOLDING	19020724	6,3
3	FLOOR	49009208	16,1
4	DOOR/WINDOW	14163012	4,7
5	WALL	76047543	25,0
6	STAIRS	0	-
7	VAULT	51154189	16,8
8	ROOF	2598455	0,9
9	OTHER	5340183	1,8
10	NONE	52863239	17,4
TOTAL POINTS		**303679875**	

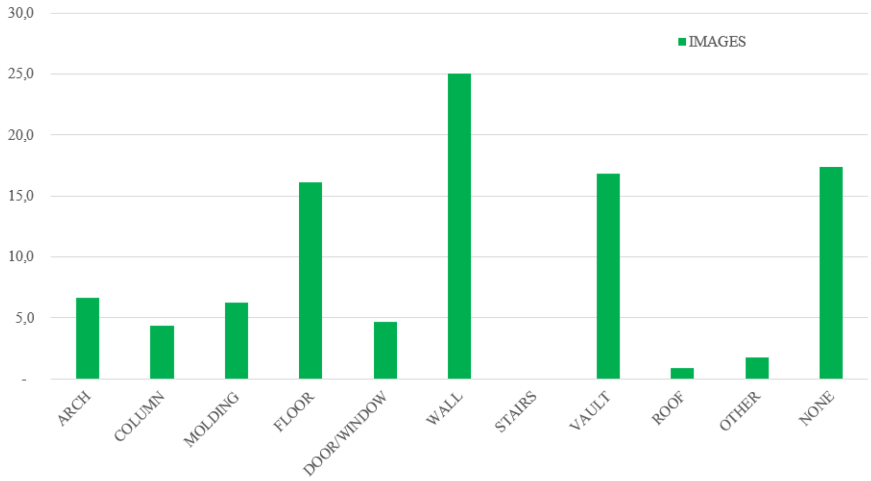


Figure 4.30 – (2_OSA) Class distribution and percentage of the building images.

(3_SSA) Basilica della Santissima Annunziata

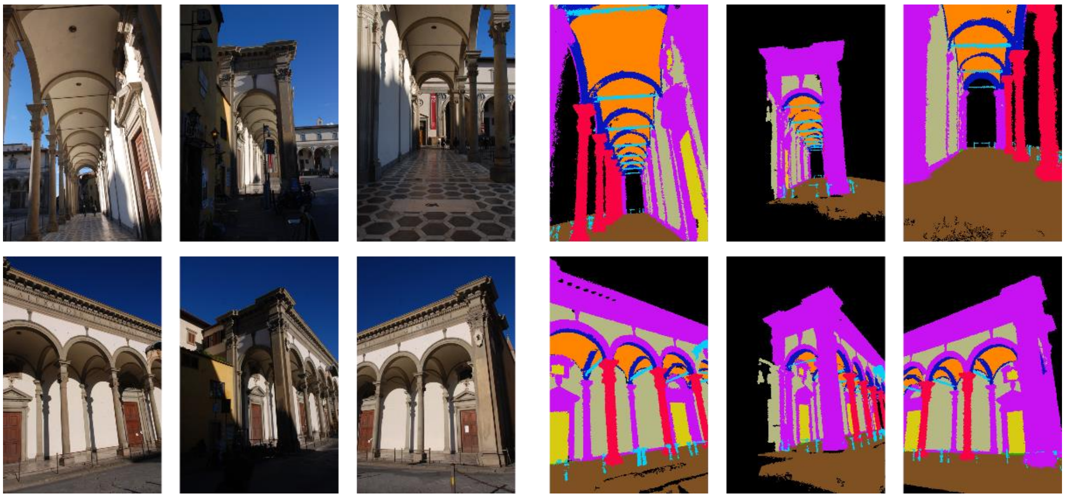


Figure 4.31 – (3_SSA) Santissima Annunziata: RGB images and generated ground-truth.

SETTINGS	
Label Transfering	
size_neighborhood	3
SubstituteWithMostPopular	
area	700
size_neighborhood	7
FillWithMostPopular	
area	700
size_neighborhood	7
Output	
resolution	2592x3872
format	*.png*
n° of images	**473**

DATA TYPE

INDEX	CLASS	N° PIXELS	% TOTAL
0	ARCH	6599073	3,5
1	COLUMN	10199003	5,4
2	MOLDING	42730902	22,5
3	FLOOR	22359052	11,8
4	DOOR/WINDOW	2810427	1,5
5	WALL	23980260	12,6
6	STAIRS	89777	0,0
7	VAULT	10787211	5,7
8	ROOF	0	-
9	OTHER	3158264	1,7
10	BACKGROUND	67538456	35,5
TOTAL POINTS		**190252425**	

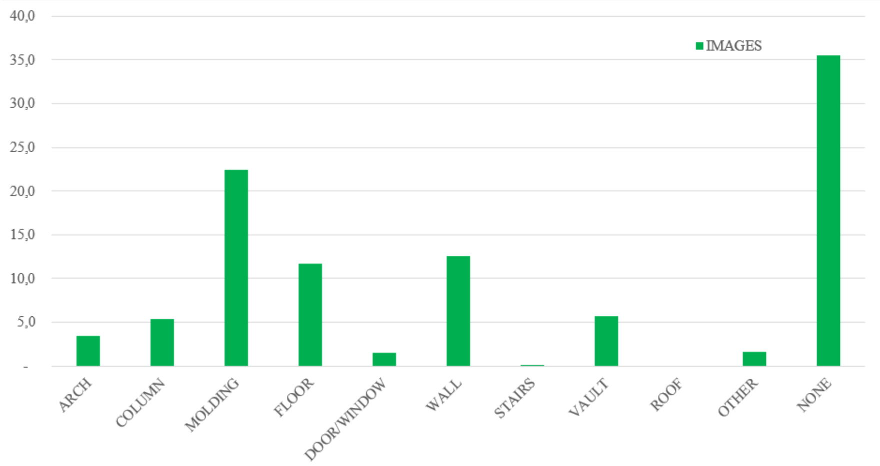


Figure 4.32 – (3_SS) Class distribution and percentage of the building images.

(4_CG) Certosa del Galluzzo

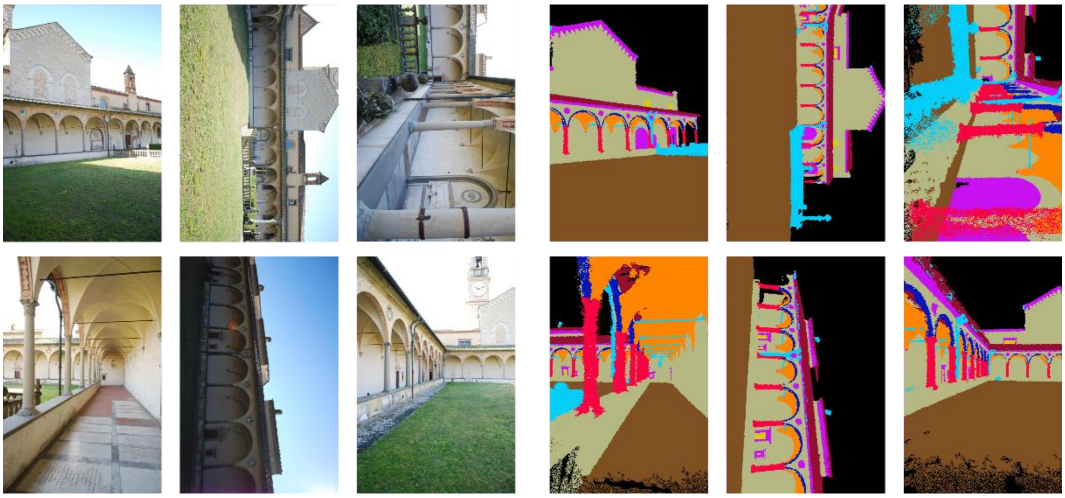


Figure 4.33 – (4_CG) Certosa del Galluzzo: RGB images and generated ground-truth.

SETTINGS

Label Transfering	
size_neighborhood	3
SubstituteWithMostPopular	
area	700
size_neighborhood	6
FillWithMostPopular	
area	700
size_neighborhood	6
Output	
resolution	2592x3872
format	*.png*
n° of images	**1102**

DATA TYPE

INDEX	CLASS	N° PIXELS	% TOTAL
0	ARCH	7565247	1,7
1	COLUMN	20922279	4,7
2	MOLDING	14172703	3,2
3	FLOOR	93981355	21,2
4	DOOR/WINDOW	1296411	0,3
5	WALL	112055824	25,3
6	STAIRS	0	-
7	VAULT	66737436	15,1
8	ROOF	11685087	2,6
9	OTHER	20318130	4,6
10	BACKGROUND	94517478	21,3
TOTAL PIXEL		**443251950**	

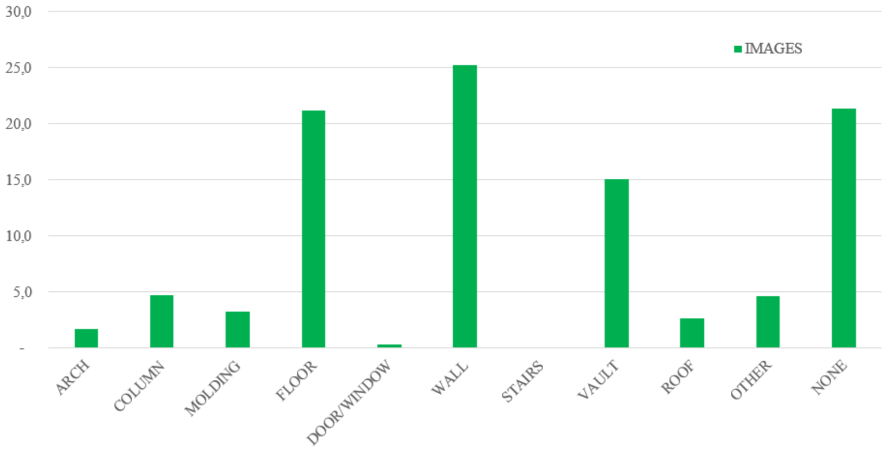


Figure 4.34 – (4_CG) Class distribution and percentage of the building images.

(5_CB) Cappella Buontalenti

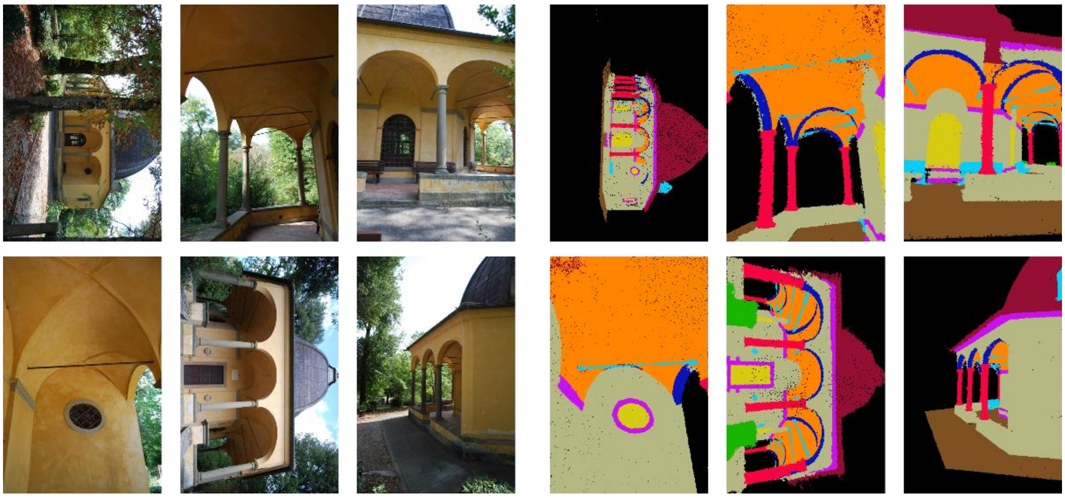
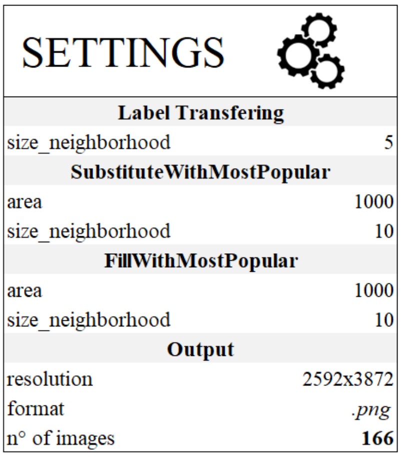
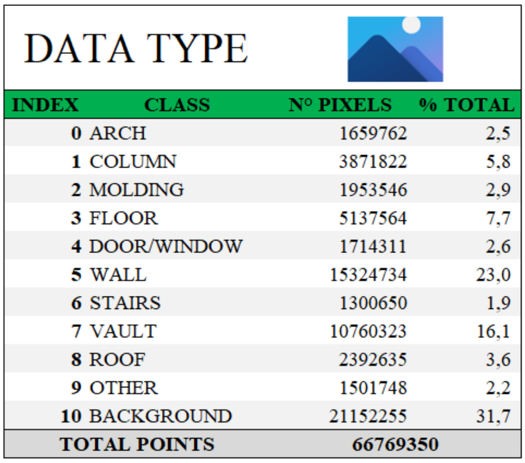


Figure 4.35 – (5_CG) Cappella Buontalenti: RGB images and generated ground-truth.



SETTINGS

Label Transfering	
size_neighborhood	5
SubstituteWithMostPopular	
area	1000
size_neighborhood	10
FillWithMostPopular	
area	1000
size_neighborhood	10
Output	
resolution	2592x3872
format	*.png*
n° of images	**166**



DATA TYPE

INDEX	CLASS	N° PIXELS	% TOTAL
0	ARCH	1659762	2,5
1	COLUMN	3871822	5,8
2	MOLDING	1953546	2,9
3	FLOOR	5137564	7,7
4	DOOR/WINDOW	1714311	2,6
5	WALL	15324734	23,0
6	STAIRS	1300650	1,9
7	VAULT	10760323	16,1
8	ROOF	2392635	3,6
9	OTHER	1501748	2,2
10	BACKGROUND	21152255	31,7
TOTAL POINTS		**66769350**	

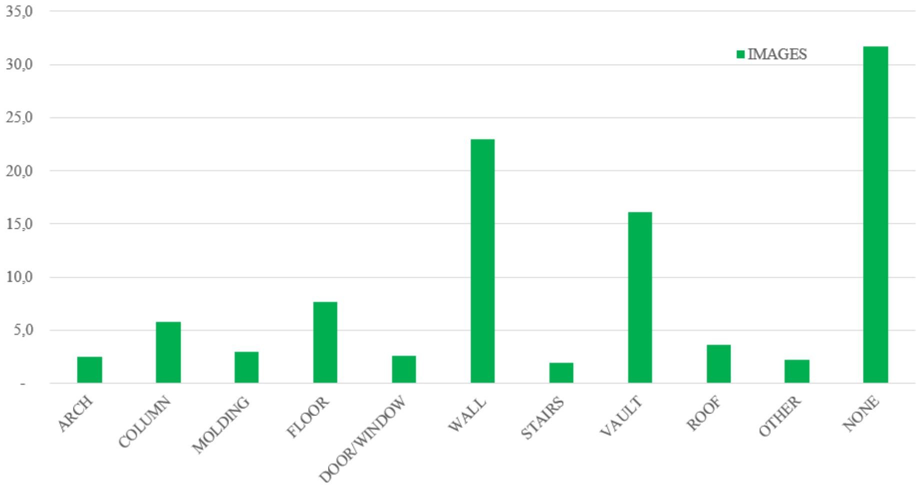
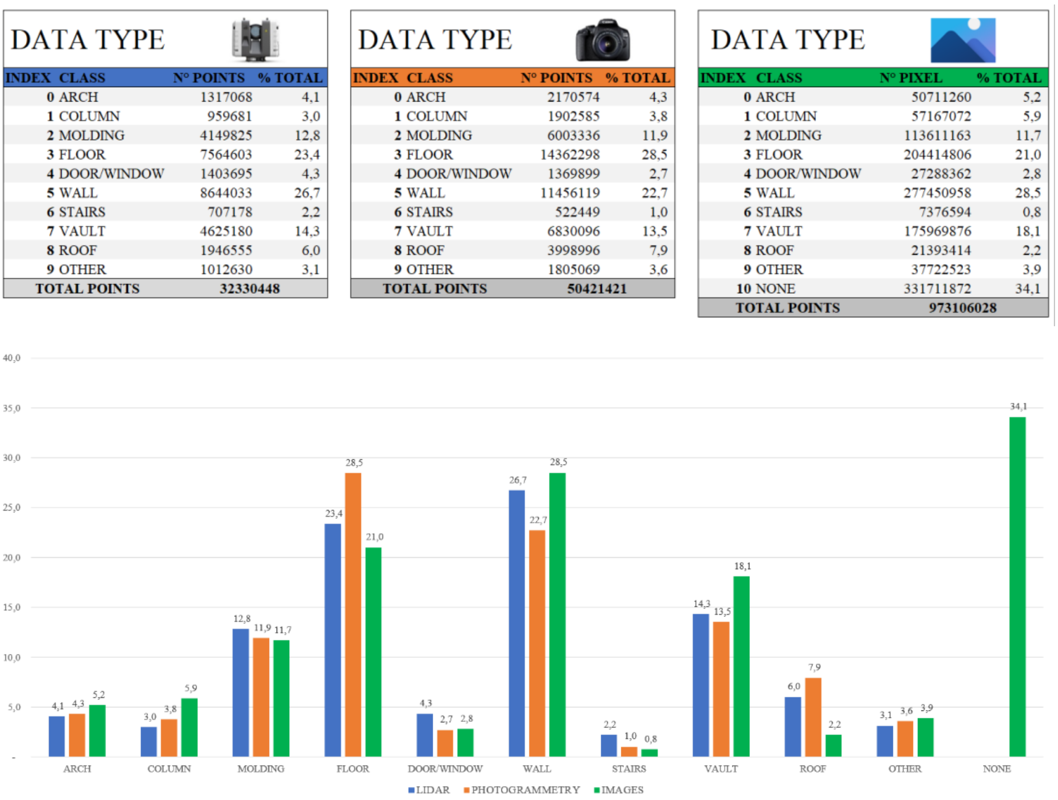


Figure 4.36 – (5_CB) Class distribution and percentage of the building images.

The five building point clouds composing the current dataset, their composition and characteristics were presented. The image dataset is currently composed by 3,244 images and the relative ground truth of five heritage buildings. Despite the large amount of data, which is numerically adequate to train a neural network, it is worth to underline that, as shown in the example images of the various buildings, and since the images are collected in the photogrammetric survey context, they represent thousands of different views of the same building, at different distance, angles and perspective. Hence, they are actually representative of a small range of building typologies. According to the first training results, future integration will be considered, in order to increase the capabilities of the dataset with new architectural styles, constructive elements, and object typologies enabling the networks to learn and generalize new scenes.



DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	1317068	4,1
1	COLUMN	959681	3,0
2	MOLDING	4149825	12,8
3	FLOOR	7564603	23,4
4	DOOR/WINDOW	1403695	4,3
5	WALL	8644033	26,7
6	STAIRS	707178	2,2
7	VAULT	4625180	14,3
8	ROOF	1946555	6,0
9	OTHER	1012630	3,1
TOTAL POINTS		32330448	

DATA TYPE

INDEX	CLASS	N° POINTS	% TOTAL
0	ARCH	2170574	4,3
1	COLUMN	1902585	3,8
2	MOLDING	6003336	11,9
3	FLOOR	14362298	28,5
4	DOOR/WINDOW	1369899	2,7
5	WALL	11456119	22,7
6	STAIRS	522449	1,0
7	VAULT	6830096	13,5
8	ROOF	3998996	7,9
9	OTHER	1805069	3,6
TOTAL POINTS		50421421	

DATA TYPE

INDEX	CLASS	N° PIXEL	% TOTAL
0	ARCH	50711260	5,2
1	COLUMN	57167072	5,9
2	MOLDING	113611163	11,7
3	FLOOR	204414806	21,0
4	DOOR/WINDOW	27288362	2,8
5	WALL	277450958	28,5
6	STAIRS	7376594	0,8
7	VAULT	175969876	18,1
8	ROOF	21393414	2,2
9	OTHER	37722523	3,9
10	NONE	331711872	34,1
TOTAL POINTS		973106028	

Figure 4.37 – Balancing of the classes on the final dataset: on the LiDAR set (blue), on the photogrammetric set (orange), and on the image set (green).

The tables and the histograms in Figure 4.37 show the final balancing of the dataset, considering all the five clouds and all the images. The Figure shows the percentage of the classes on: the LiDAR set (blue bars), the photogrammetric set (orange bars), the percentage referring to the images (green bars). The histogram shows an overall significant imbalance of the classes, with a remarkable predominance of the classes 'floor' and 'wall', and a small representation of the classes 'stair', 'door/window' and 'other'. Class imbalance is a common issue in several semantic segmentation datasets, and, if not properly handled, it can be detrimental to the learning process, biasing the results in favour of dominant classes. There are several techniques to face the unbalancing issues: on one hand, operating directly during the training phase, using for example class weighting or online data augmentation, and choosing appropriately the correct evaluation metrics. On the other hand, operating on the unbalanced dataset, before feeding the neural network. For instance, selecting a specific set of the images, reducing images of classes over-represented, hence reducing such strong class imbalance, or using offline data augmentation. Future integrations and extensions will be focused also on the acquisition of scenes to privilege buildings with the prevalence of low percentage classes. The histogram in Figure 4.37 also reveals a predominance of the pixels labelled as 'none' in the image dataset. Differently from the point clouds, this class is inevitably present, and it comprises a wide range of element typologies, in some cases similar to the building elements. Hence, it could bias the training, and it may negatively influence the network learning procedure. The reduction of this class with a specific image selection will be taken into account during training, and its effects will be discussed in detail in the next chapter. Another important aspect in a benchmark design is data splitting, which is the partition of the images in training, validation and test set. Most of the existing datasets are provided with a specific partition, in order to allow the comparison with various models and assess the performance improvements. Currently the dataset is still under construction, and since the partitioning strictly depends on the full size of the dataset, the splitting strategy will be designed in the future, depending on the first training results.

In conclusion, the main aim of this dataset is to offer the possibility to implement and compare multi-view approaches on heritage building scenarios and leverage on the existing 2D segmentation architectures to ease the development of new classification machine learning and deep learning techniques. For this reason, once that the dataset will have a definitive structure, it will be made freely available to the research community. On one hand, it can be useful to test and compare new algorithms, on the other hand, it allows to collaborate at the integration and at the extension of the dataset. In addition, TLS clouds and the photogrammetric clouds, both segmented following the same class definition used for the images, will be available in addition to the images for each building. These multiple-source data can be useful to perform comparisons and assessments such as: (i) compare the accuracy of point-based and multi-view based methods on the same dataset, (ii) compare the accuracy of multi-view based approach on heritage benchmark with that obtained on standard buildings, (ii) assess the accuracy of point-based networks on two types (TLS and photogrammetric) of point cloud data. Hence, the presented dataset can be (i) integrated with ARCHdataset, (ii) used to tailor existing network architectures on the cultural heritage building case, (iii) exploited to develop new hybrid networks that can leverage on both images and point clouds.

4.4 How to Improve Datasets?

Since the remarkable difficulties and challenges to structure a dataset, especially in the first phase, there are several strategies and techniques that can be used to improve the quality, the quantity, and the generalization of the dataset without requiring the acquisition of new real data, and to improve the performance of the models always using the same data source. In this section two of the main techniques are briefly introduced: (i) data augmentation, and (ii) synthetic data generation.

4.4.1 Data Augmentation

Data augmentation is a technique widely used during model training in many applications, especially in the image classification and segmentation workflow. It consists in generating new data artificially altering the existing ones by applying a set of transformations to them, in such a way to increase the number of training samples, and to improve the performance and results of deep models by generating new and diverse instances for the training dataset. The key concept of data augmentation is that CNNs are invariant to translation, viewpoint, size, or illumination, and they are able to classify objects in different orientations. The images composing the training dataset are captured in real-world under specific set of conditions, but they may exist in a large number of variations, such as varying orientations, locations, scales, colours, brightness, and so on. Manipulating the images by adding synthetic transformations and by simulating different conditions help the model to increase its generalization ability. By performing these operations it can also be possible to prevent the model from learning irrelevant patterns, essentially boosting the overall performance and improving its robustness. The most used augmentation method can be divided into two main groups: *position augmentation* and *colour augmentation*. The most common transformations of the first group are flipping, rotation, mirroring, scaling, cropping, and translation. The second group comprises the transformation without a position modification, including brightness, contrast, saturation, and so on. For a classification problem, the task of assigning one category to an image, the output label remains always the same after each type of augmentation. For a semantic segmentation problem, the pixel-level ground-truth do not change with colour augmentation, but it should be fixed according to the applied transformation if a position augmentation is applied. However, data augmentation has its challenges: (i) quality assurance of the augmented dataset is often time expensive, (ii) the inherent bias of original data may persist in augmented data, (iii) finding an optimal augmentation strategy for the data is non-trivial, (iv) it is not possible to use data augmentation in all the possible working condition, (v) increasing the number of images will increase computational time. The transformations can be applied directly on the data before training or on the mini-batch, just before feeding it to the machine learning model. The first method is known as *offline augmentation* and is preferred for smaller datasets, since the procedure ends up increasing the size of the dataset by a factor equal to the number of transformations performed. The second option is known *as online augmentation* and it is preferred for bigger datasets since it avoids the explosive increase in size and, consequently, in computing time.

4.4.2 Synthetic Dataset

The concept of synthetic data refers to artificial data that mimics real-world observations and it is used to train machine learning algorithms when actual data are difficult or expensive to collect. Binary, numerical, categorical, or unstructured data, such as images or videos, can be included in a synthetic dataset. There are many key reasons that promote the use of synthetic data, which allow data scientists to achieve several advantages:

- *Cost and time efficiency*. It may be cheaper to generate synthetic data than to collect from real world events if you lack a proper dataset. The same is valid for the time factor: collecting and processing real data might take weeks, months, or even years for some projects, while synthesizing might only take hours/days.

- *Exploring rare data*. There are cases in which data are rare to accumulate or are challenging to acquire. It may generate a lack in specific context or area of the dataset. With the use of synthetic data it is possible to simulate every type of conditions and situations, generating every type of image.

- *Privacy issues resolved.* When sensitive data must be processed or given to third parties to work with, privacy issues must be taken into consideration. Unlike anonymization, generating synthetic data removes any identity trace of the real data, creating a new valid dataset without compromising privacy.

- *Easy labelling and control.* Synthetic data makes labelling easy, fast, and accurate. In addition, it is possible to generate different types of annotations in every moment, such as pixel-wise, bounding boxes, or others. And fully synthesized data can be easily controlled and adjusted, both in the input and output data.

- *Data quality*. In addition to being difficult and expensive to collect, real-world data are often inaccurate or biased, which can affect the performance of a neural network. A high-quality, balanced, and varied dataset can be achieved by using synthetic data. By automatically filling in missing values and assigning labels to artificially generated data, more accurate predictions can be made.
- *Scalability*. Machine learning requires massive amounts of data. The availability of sufficient data at a sufficient scale is often a challenge for training and testing a predictive model. By supplementing real-world data with synthetic data, we can achieve a greater scale of inputs.

Regarding their composition, synthetic data generally falls into three main categories: (i) *fully synthetic* and (ii) *partially synthetic,* and *(iii) hybrid*. *Fully* synthetic retains nothing from the original data. Real-world characteristics of the data are usually identified by the data generating program, such as the feature density, to estimate realistic parameters. It then randomly generates data based on estimated feature densities or using generative methods. With this technique, no real data are used, so it offers robust privacy protection at the expense of data truthfulness. *Partially synthetic* data replaces some of the real data with synthetic values, while retaining some of the real data, or it permutes existing unstructured data. It is also useful for filling in gaps in the original data. In order to generate partially synthetic data, data scientists use methods such as model-based imputation. *Hybrid* data combines real and synthetic data. Hybrid synthetic data pairs random records from a real dataset with close synthetic records. As a hybrid of fully and partially synthetic data, it provides high utility as well as privacy protection. The disadvantage of this data type is that it requires more memory and processing time. Although synthetic data has many benefits, there are still cases when it would be better not to use it. Synthesizing data is faster and cheaper than collecting data, but it is still a complex process that requires experienced operators. Synthesizing data in a wrong way might not represent the events in the real world correctly, introducing a bias as well. If synthetic data are not sufficiently accurate, or they do not accurately represent the real-world data, they do not reflect the patterns crucial to test and train a machine learning system.

The generation of synthetic data needs a robust model that can recreate a real dataset based on the probabilities that some types of data occur in real world. Neural Networks are particularly suitable at learning data distribution and at generalizing them. There are different methods and models to generate synthetic data: the state-of-the-art techniques are reported in the following.

Variational Autoencoders (VAEs) are unsupervised generative models that can learn the underlying distribution of data and generate a complex model. This type of approach takes an original distribution, transforms it into a latent distribution, and then returns it to its original space (this is known as encoded-decoded). In this process, a "reconstruction error" functional is generated, and the model aims at minimizing it. VAEs are very useful for continuous data but less effective at categorical data. They are not capable of generating unstructured data, such as images or videos. More detail can be found in (Papadopoulos et al., 2023), (Dai et al., 2023).

Generative Adversarial Networks (GANs) were introduced by (Goodfellow et al., 2014) and they are supervised generative models that can be exploited to generate realistic and highly detailed data. In this method, two neural networks are trained, one for generating fake data points (a generator), and the other for distinguishing fake from real data points (a discriminator). As the generator is trained thousands of times, it becomes more and more adept at creating highly realistic fake data points that can "fool" the generator. A GAN is particularly effective at synthesizing images, videos, and other unstructured data. They have the disadvantage that they require specialized knowledge to construct and train, as well as that the model can "collapse" and produce a limited set of very similar fake data points.

Neural Radiance Fields (NeRFs) is a method of generating new views from a partially-known 3D scene. Using a set of images as input, the algorithm interpolates them and adds new perspectives to the same object. To predict the content of each voxel, a fully connected neural network is used to treat the static scene as a continuous 5-dimensional function. For each ray, it provides a predicted volume for one voxel, and so it fills in an entire missing picture in the scene. NeRF is a very useful way to generate realistic images from an existing image set. This technique has the disadvantages of being slow to train, slow to render, and generating images that may be low-quality or aliased. It is now possible to address these challenges using neural rendering algorithms. More detail can be found in (Mildenhall et al., 2020).

Simulated Data are a form of synthetic generation that uses a virtual camera to generate physics-based and photorealistic simulations. In order to produce realistic 3D data, simulated data include all the necessary annotations, dimensions and labels. Compared with the other methods, that typically are focused on a single task or scenario, this type of simulations allows a more flexible generation, and it is more suitable for complex scenarios. Simulated data allow to adjust light conditions, to modify textures, colours and layouts, to place several elements in the scenes, or to capture cases that rarely occur in the real world. Some recent applications can be found in (Mittal et al., 2022), (Huang et al., 2022), (Kerley et al., 2022).

There are several platforms or software that allow the creation of synthetic dataset, and support data generation. The most popular are Tonic.ai, GenRocket, MDClone, YData, Anyverse, DataGen, Neuromation.

4.5 Summary

In this chapter a new dataset for the image semantic segmentation of heritage buildings has been introduced. Its construction carried out from the lack of a specific benchmark for image segmentation in the context of heritage scenarios. The new dataset is particularly suited for training, validating, and testing machine learning and deep learning models, and it is mainly focused to support automation in three-dimensional and informative model generation via semantic segmentation. In the first paragraph (§4.1) the relevance of a benchmark for the training and the evaluation of a ML or NN model has been explained, and the challenges and the difficulties to define and to structure a dataset from scratch have been pointed out. In the second paragraph (§4.2) the most popular and interesting datasets for the semantic segmentation have been illustrated. The datasets have been divided into three categories according with their segmentation target: *2D image segmentation* (§4.2.1), *RGB-D image segmentation* (§4.2.2), and *3D point cloud segmentation* (§4.2.3). In addition, the heritage focused datasets have been examined (§4.2.4), and it turned out the missing of a specific dataset for heritage buildings image segmentation. In paragraph §4.3 the structure of the new dataset has been shown in detail, and it includes the buildings composing the current dataset (§4.3.1), the acquisitions (§4.3.2), the processing phase (§4.3.3), and the class definition (§4.3.4). To speed-up the image annotation process, a labelling projection procedure has been developed and illustrated (§4.3.5). It allows to annotate all the images of a photogrammetric survey starting from a manual segmentation of the related point cloud. The procedure has been tested and its performance assessed. Finally, the trend and the statistic of the dataset have been reported (§4.3.6). For all the five buildings the detail on the point clouds, including the number of points for each classes and their percentage have been reported. Moreover, the details on the generated image dataset, including the image size, the format, the resolution and the class indexing, have been shown. As discussed in the last paragraph the new dataset is promising, but it is still under construction, and it needs further improvements, in particular to increase its size and to enhance the variety among the scenes and the images. For completeness, two common techniques used to increase the size of a dataset have been introduced in paragraph §4.4: data augmentation, and synthetic data generation.

Chapter 5

Semantic Segmentation Tests and Results

In this chapter the overall results of the proposed semantic segmentation pipeline presented in (§3.3) are described and widely discussed. The first section (§5.1) is a brief introduction about the challenging in structuring a machine learning project from scratch. In paragraph §5.2 the details about the implementation of the neural networks are illustrated, together with the details on the three exploited neural network architectures, FCN (§5.2.1), SegNet (§5.2.2), and DeepLabv3+ (§5.2.3). The first part of the chapter is focused on image segmentation. In the paragraph §5.3 the training settings are shown, including the image processing (§5.3.1), the tests organization (§5.3.2), the hyperparameters tuning (§5.3.3), and the evaluation metrics (§5.3.4). Afterwards the result for each planned tests are illustrated in detail (§5.4) . Three main test typologies have been carried out: *Test A* (§5.4.1), *Test B* (§5.4.2), and *Test C* (§5.4.3). The second part of the chapter is focused on the features transfer from the images to the point cloud. At first, the settings and the parameters that rule the reprojection procedure will be analysed, and the various test structures will be explained (§5.2.2). Three test typologies have been conducted: *Test R.GT* (§5.6.1), *Test R.A* (§5.6.2), and *Test R.C* (§5.6.3). In the paragraph §5.6 the results for each building composing the dataset will be reported in detail. The results will then be extensively discussed (§5.7) both for image and point cloud segmentation. The chapter ends with a general summary (§5.8).

5.1 Introduction

Structuring a machine learning or deep learning project from scratch is a daunting task, and it requires multiple decision making, planning, and understanding. Training a deep learning model is a long process that needs lot of computing time, lot of efforts and attempts, hence the optimization of the available resources is a necessary requirement to obtain a successful project. Only in such way is possible to achieve a productive, reproducible, and understandable model. The lifecycle of a machine learning system is highly iterative, and each phase requires to reach a satisfactory level of performance before moving onto the next step. The main phases are illustrated in Figure 5.1 and are briefly described in the next sections.

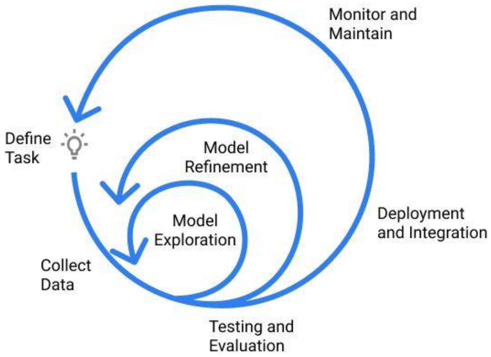


Figure 5.1 – ML projects lifecycle (from jeremyjordan.me)

Planning and project setup. This is the first step of the machine learning pipeline, and it includes defining the task, specifying requirements, and determining the feasibility. The problem should be clear and well defined to find the optimal strategy or approach to solve it.

Data collection and labelling. This phase was widely illustrated in Chapter 4, and it involves the definition of the ground truth, labelling the data if not available, and validating the quality of the data. It is one of the most time-consuming operations, in many cases the annotation is mainly manual, and it requires a careful labelling, since its quality has a large effect on the model performance.

Model exploration. It consists of setting up baselines on the problem, useful to establish an expected or a target performance. Baselines turn out from published similar tasks, from human-level performance, and testing various existing architectures or approaches to face similar problem. In this phase the most suitable models and strategies will be carried on.

Model refinement. It involves the fine setting-up of the model and the optimization of its performance, by tuning the hyperparameter, debugging iteratively the model as complexity is added, and performing error analysis to uncover common failure modes. In this phase the efforts should be focused on the distribution shift, the difference between the train set error and the validation set error, addressing the problem of underfitting and overfitting.

Testing and evaluation. It involves evaluating the performance of the model on test distribution, understanding differences and issues between train and test set performance. Test sets could change during time, and it is important that the model score does not degrade with new examples or new data.

Ongoing model maintenance. If used consistently over time*,* a machine learning project requires maintenance and a continuous updating since its performance could decline or input signals may change over time. In addition, the model can be improved over time, adding new training examples, or changing the network architecture with more performing model.

5.2 Algorithm implementation

The development and the training of the various models, as well as the implementation of supporting functions and additional scripts, was carried out using the coding language of MATLAB® with the support of various add-on Toolboxes. The Image Processing Toolbox™ provides a set of reference-standard algorithms and workflow for image analysis, visualization, manipulation and processing. The Computer Vision Toolbox™ provides functions and apps for designing computer vision systems and tasks, such as feature detection or matching, and allow the calibration workflows. Deep Learning Toolbox™ provides several tools and frameworks for implementing and designing deep neural networks, using various algorithms and pretrained models. It helps to design and train graphically, and to manage complex deep learning experiments, keeping track of training parameters, and analyzing the results.

As already mentioned in the previous paragraph (§3.3), the point cloud semantic segmentation pipeline involves firstly the segmentation of a set of images. As illustrated in the paragraph §3.2.1, there are several state-of-the-art models for image semantic segmentation, and depending on the specific tasks, some work better than others. In this study three neural network have been tested and implemented: *Fully Convolutional Network* (FCN), *SegNet*, and *DeepLabv3+*.

5.2.1 Fully Convolutional Network (FCN)

Fully Convolutional Network (FCN) (Long et al., 2015) is the first architecture trained end-to-end for pixel-wise prediction and from supervised pre-training. It allows to adapt standard classification networks such as VGG net, AlexNet or GoogLeNet into fully convolutional networks and transfer their learned representation by fine-tuning to segmentation task. The structure of the architecture is shown in Figure 5.2.

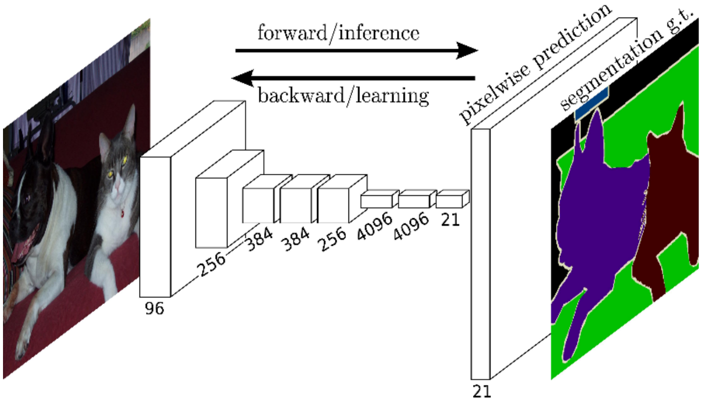


Figure 5.2 – Fully Convolutional Network architecture (Long et al., 2015)

The network is composed by a down-sampling part and an up-sampling part. The first part is a standard CNN composed by series of layers, in which the image features are extracted via convolution, followed by activation functions and pooling layers. At the end of the down-sampling network the number of channels is transformed into number of classes with a 1×1 convolutional layer. The up-sampling network transforms the height and width of the feature maps to those of the input image via *deconvolution* or *transposed convolution*. Just like standard convolution the layers are defined by the padding and stride, and they have a learnable kernel and activation functions. To refine the output by adding links that combine the final prediction layer with lower layers with finer strides. Depending on the type and the depth of the skip connections, three model types are available: FCN-32s, FCN-16s and FCN-8s. The function `fcnLayers()` in MATLAB, returns a fully convolutional network configured by default as FCN-8s, preinitialized using layers and weights from the VGG-16 based architecture (Simonyan & Zisserman, 2014) pretrained on the ImageNet database. With the function `type()`it is possible to specified other types of FCN model: FCN-32s, that up samples the final feature map by a factor of 32, FCN-16s that up samples the final feature map by a factor of 16 after fusing the feature map from the fourth pooling layer, FCN-8s, that up samples the final feature map by a factor of 8 after fusing the feature map from the third and fourth pooling layers. In this study FCN-8s has been used since it provided finer-grain segmentation at the cost of additional computational.

5.2.2 SegNet

SegNet (Badrinarayanan et al., 2017) is composed by an encoder network and a corresponding decoder network, followed by a final pixel-wise classification layer, as shown in Figure 5.3.

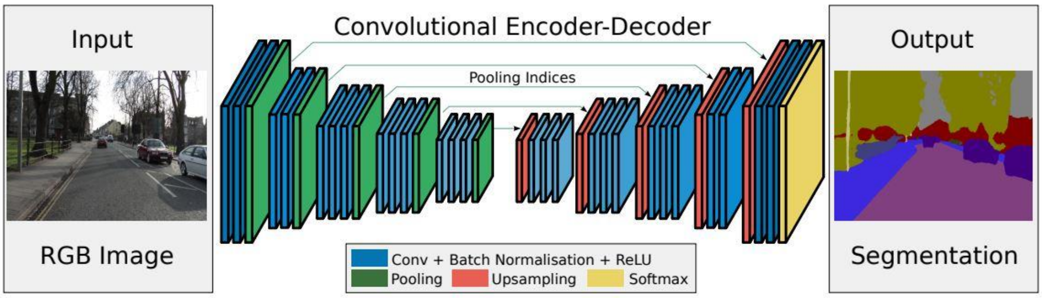


Figure 5.3 – SegNet architecture (Badrinarayanan et al., 2016)

The encoder network consists of 13 convolutional layers, and each encoder performs a convolution to produce a set of feature maps. The maps are then batch normalized and passed through an element-wise rectified linear unit $(\mathrm{ReLU})max(0, x)$. Following that, a 2×2 window with stride 2 max-pooling layer is applied, and the result sub-sampled by a factor of 2. To avoid loss of spatial resolution it is necessary to capture and store boundary information before max-pooling and sub-sampling. The decoder network has the same number of layers of the encoder, and it up samples the input feature maps using the memorized max-pooling indices. The SegNet decoding technique consists in convolving the feature maps with a trainable decoder filter bank to produce a dense feature maps that is then batch normalized. The final high dimensional feature output by the last decoder is fed to a trainable soft-max classifier. The output is a K channel image of probabilities where K is the number of classes. In MATLAB the function `segnetLayers()` returns the SegNet architecture. It requires the specification of the input image size, the number of categories and the choice of a base model. The available models are VGG-16 and VGG-19, with an encoder depth of 5, pretrained on ImageNet database. The results presented below in this study are carried out with VGG-19.

5.2.3 DeepLabv3+

Deeplabv3+ (L.-C. Chen et al., 2018) employs atrous convolution with upsampled filters to extract dense feature maps and to capture long range context. Atrous convolution allows to explicitly control how densely to compute the feature, and it allows to avoid signal decimation caused by stride and pooling. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, while the simple yet effective decoder module refines the segmentation results along object boundaries. The structure of the network is illustrated in Figure 5.4.

MATLAB allows the implementation of this network architecture with the function `deeplabv3plus()`, that requires three inputs: the image size, specified as a 2-element or 3-elements vector in the format `[height, width, 3]`, the number of the classes, specified as an integer greater than 1, and the base classification network. Several base architectures are available, and they have different characteristics mainly differing on precision, speed, and network dimension. The choice of the architecture is based on a compromise between these characteristics. In this study four based architectures have been tested: ResNet18, ResNet50, VGG-16, VGG-19. After several tests, it turned out that ResNet18 was the most suitable on the data, and the best compromise between speed and precision. The results that are going to be illustrated are the results obtained with ResNet18 (K. He et al., 2015), pretrained on the ImageNet database.

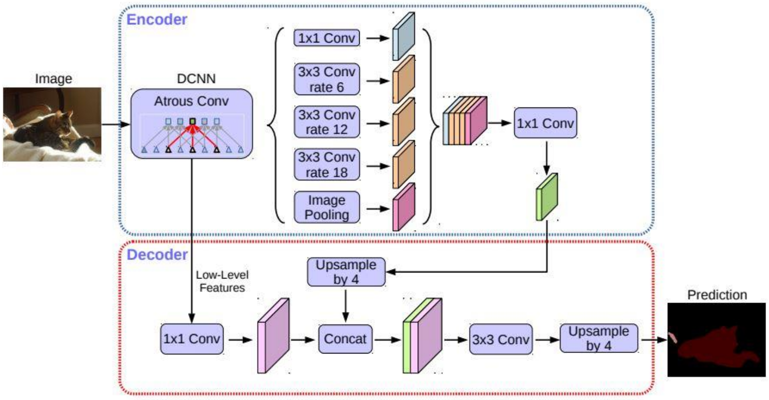


Figure 5.4 – DeepLabv3+ architecture (Chen et al., 2018)

5.3 Training settings

In this paragraph the setting used to evaluate the performance of the various models will be illustrated and explained, and they include the image processing and preparation (§5.4.1), the training tests (§5.4.2), the hyperparameter tuning (§5.4.3), and the evaluation metrics (§5.4.4).

5.3.1 Image processing and preparation

For all the tests, before starting the training procedure, the images generated by the labelling projection procedure (§4.3.5) have been processed and set up, to make them more homogeneous and suitable to feed the network. Each image has been processed by four steps, described below:

Resizing. To maintain the highest quality and accuracy, the ground-truth output by the labelling has been produced with the same dimension of the input images, and currently, the images composing the dataset have the dimension of 2592×3872 pixels. This input size is too large to train a deep network, and it would require long training time and high memory consumption. After a series of experiments, the images have been downsized to 720×1075 pixels. Furthermore, this operation allows to homogenize data of different size in case of an integration with new images captured with different cameras or sensors.

Keeping Verticality. The images of the photogrammetric survey could be acquired with different camera orientations, hence, in some cases, they do not respect the correct verticality of the scene. To help the network to learn more easily some features during the training, each image has been rotated to keep the correct verticality of the building or the scene on the images.

Cropping. Due to the rotation the images could have different aspect ratios between width and height, but the neural network needs the same input size for training. To avoid resizing and distortion the images have been cropped in a square format, producing two overlapping square tiles for each image. Hence the final size of the input is 720×720 pixels with a depth of 3 channels (RGB).

Image Discarding. From a visual examination of the generated ground truth, turned out the presence of some useless tiles, mainly caused by three factors: (i) too high percentage of background, (ii) the occurrence of too few classes, and (iii) the predominance of one class on the others. To avoid biases during training, the images could be filtered through a series of selecting rules, discarding the images with the mentioned issues. Several rules could be set and implemented, and the selecting rules that have been used in the following tests are reported below:

1) $\%\, pixels\ class\ \boldsymbol{background} < 30\%$

2) $number\ of\ classes > 5$

3) $\%\, pixels\ \boldsymbol{1^{st}\ class} < 2 \times (\%\, pixels\ \boldsymbol{2^{st}\ class})$

4) $\%\, pixels\ \boldsymbol{2^{st}\ class} < 2 \times (\%\, pixels\ \boldsymbol{3^{rd}\ class})$

5.3.2 Training tests

Data distribution and splitting are two key points to structure a machine learning project, and they have a remarkable effect on the model performance and usability. The main aim is to develop a wide-range model able to generalize as many scenes as possible, and this ability can be obtained by providing a large training set, with a wide range of scenes, buildings, constructive elements and structure typologies, and a validation/test set quite different and varied in relation to the training set. Nevertheless, the available dataset is still limited in building typologies, and, currently, it does not allow a good level of flexibility in data organization and splitting, and it makes challenging reaching a wide capability. However, in this study three test typologies have been carried out, and they are described in the following sections.

Test A. The first set of tests is the simplest and less challenging, and it consists in testing each building of the dataset one by one. The entire set of images of each building was randomly shuffled, and then partitioned in training set, validation set and test set, with the percentage respectively of 60%, 20%, and 20%. Since the images in the test set are similar to the images in the training set, the model should be able to generalize the solutions quite easily in this series of tests. Despite these tests do not provide a general model with a wide capability, they are helpful to set the hyperparameter of the networks, to compare the performance of the various architectures, to assess the quality and the correct functioning of the generated dataset, and, generally, to conduct easily preliminary evaluations. Figure 5.5 shows the structure of the test for the first building (1_SC) Spedale del Ceppo.

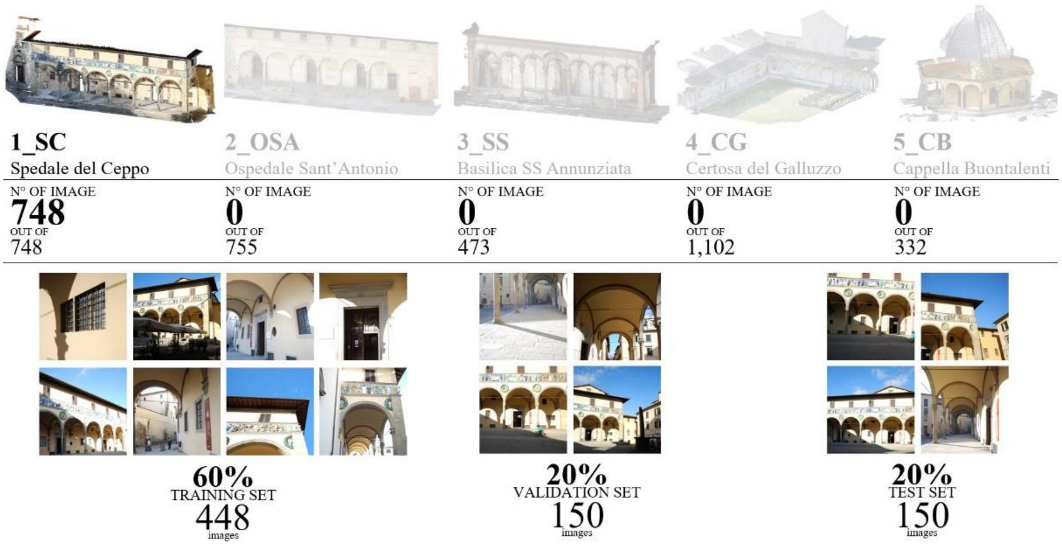


Figure 5.5 – Structure of test A for (1_SC) Spedale del Ceppo

Test B. In this test all the images of the five buildings were used. The images were randomly shuffled, and then partitioned in training set, validation set and test set, with the percentage respectively of 60%, 20%, and 20%. Although the presence of several building typologies, even this test is not particularly relevant to achieve a general model with a wide capability, since some images in the training set are analogous to some images in the test set. However, the test is helpful to assess the capability with several building typologies, to fine-tune the hyperparameters, and to evaluate the effect of transfer learning and data augmentation on the performance. Figure 5.6 shows the structure of Test B, including the number of used images, and the percentage of images used to train and test the model.

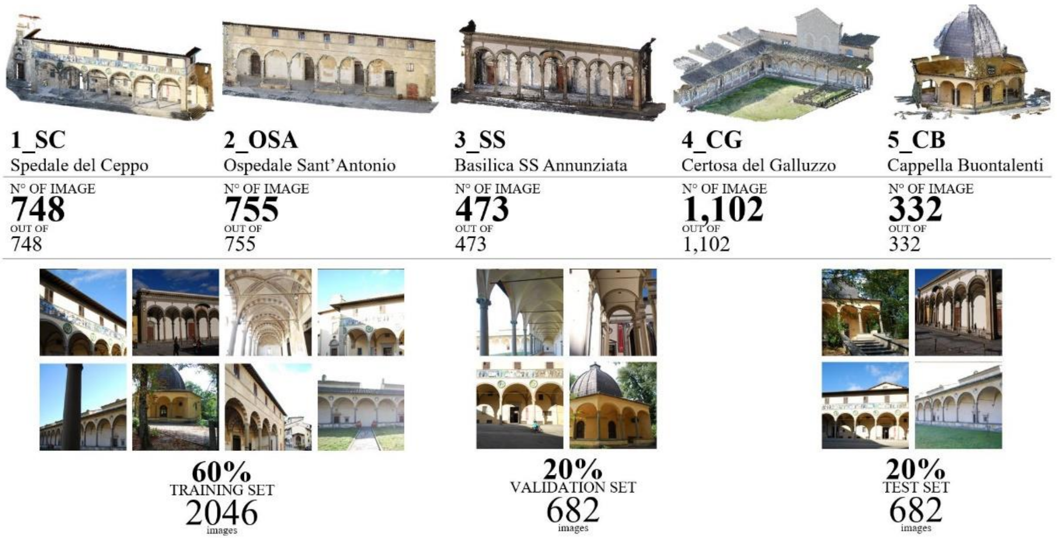


Figure 5.6 – Structure of test B

Test C. The last set of tests is the most challenging, and it represents the target task of a general semantic segmentation procedure. The tests consist of attempting the prediction of an unseen scenario. To perform these tests the images of four buildings were used for the training phase, splitting them in training set (60%), validation set (20%) and internal test set (20%), and the images of the remaining building were used to the external test of the model. Despite the number of images may seem large enough to perform this type of tests, the generalization of the solution will be a challenging task for the model, since the typologies of buildings to learn the features are limited. To obtain a comprehensive view of the performance, a cross-validation method was used, and each of the five buildings was used alternately as test set. Figure 5.7 shows one of the five cross validation test structures, in which (1_SC) Spedale del Ceppo has been used as test set, and the other four buildings to train and validate the neural network.

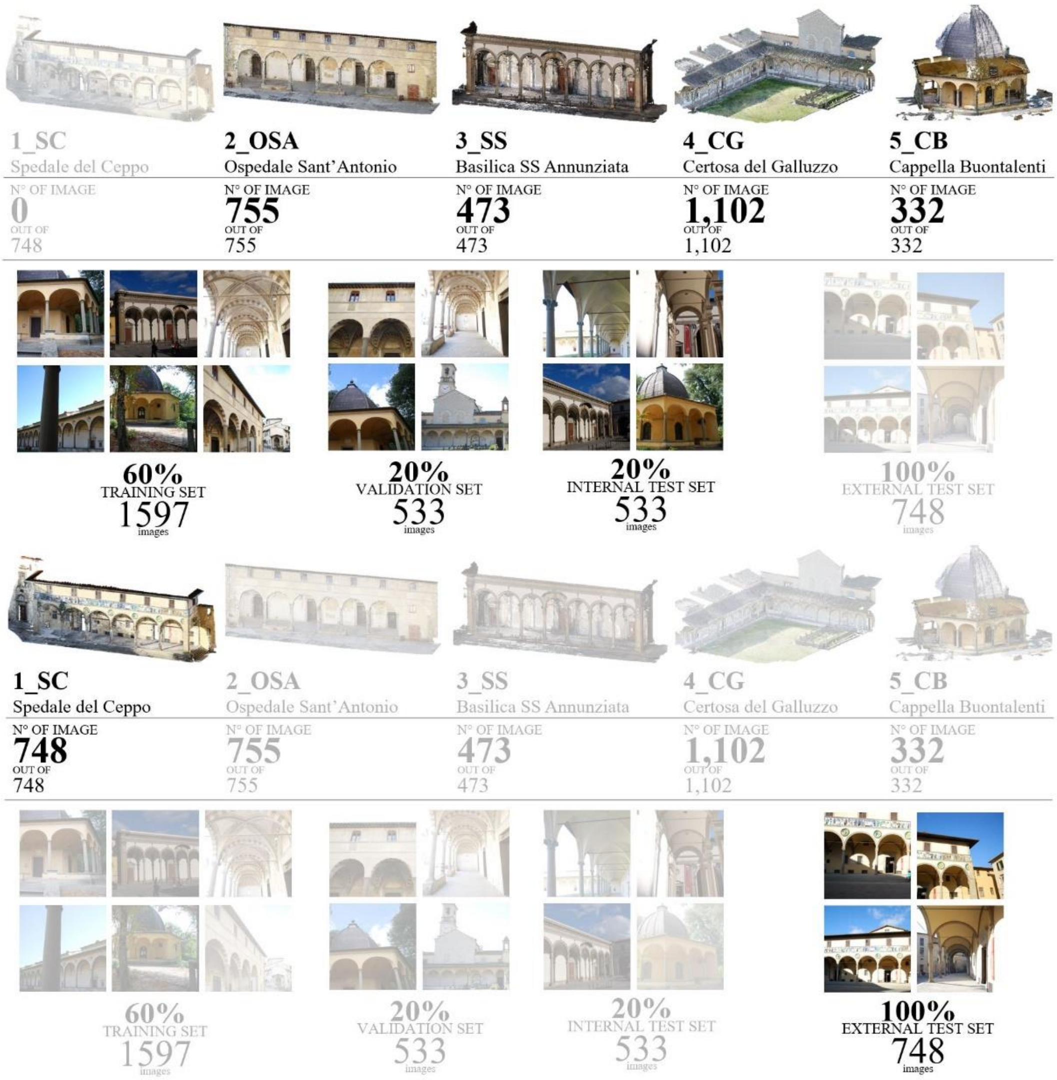


Figure 5.7 – Structure of test C

5.3.3 Hyperparameter tuning

In deep learning tasks, the choice of appropriate hyperparameters is important for an efficient training convergence, and for an optimal performance achievement. Hyperparameter tuning consists of finding a set of optimal hyperparameter values for a learning algorithm while applying this optimized algorithm to any data set. That combination of hyperparameters maximizes the model's performance, minimizing a predefined loss function to produce better results with fewer errors.

Learning Rate. It is an hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are update. Too small learning rate may result in long training, while too large rate may result in unstable training process. After a series of tests, the initial learning rate was set to α = 0.001 with a drop during training, updating the value every *5 epochs* with a factor of *0.3*.

Batch Size. It is the size of the mini-batch to use for each training iteration. A mini-batch is a subset of the training set that is used to evaluate the gradient of the loss function and update the weights. A large batch size allows a faster convergence but is more computationally expensive and lead to poor generalization. Depending on the number of images during training the size was set from *4* to *8*, as a compromise between GPU and fast convergence.

Loss Function. It is the function that maps onto a numerical value the difference between the predicted label $\hat{y}$ and the ground truth label y during the training. Various loss functions have been proposed in literature, and a detailed survey on existing loss function for semantic segmentation can be found in (Jadon, 2020). In the proposed tests the Cross-Entropy loss is used (Ma et al., 2004), and it is defined as a measure of the difference between two probability distributions for a given set of events. It is defined as follows:

$$L_{BCE}(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \tag{5.1}$$

Optimizer. The optimizer or solver is used to update the parameters at each iteration during training to minimize the loss function. There are many optimizers, and its choice is an important aspect to perform a good training. In this study three optimizers have been tested. The Stochastic Gradient Descent (SGD), the Root Mean Square Propagation (RMSProp) and the Adam. After a series of tests, the SGD with Momentum turned out to be the most suitable. The stochastic gradient descend (SGD) algorithm updates the weight and biases to minimize the loss function, by determining small steps at each iteration in the direction of the negative gradient of the loss, but it can oscillate along the path towards the optimum. The Stochastic Gradient Descent with Momentum (SGDM) reduces this oscillations adding an additional term. It is defined as follow:

$$\theta_{\ell+1} = \theta_{\ell} - \alpha \nabla E(\theta_{\ell}) + \gamma(\theta_{\ell} - \theta_{\ell-1}) \tag{5.2}$$

where ℓ is the iteration number, $\alpha > 0$ is the learning rate, θ is the parameter vector, $E(\theta)$ is the loss function, and γ is the momentum. More detailed information about the optimizers can be found in (D. Choi et al., 2019).

L$_2$ Regularization. In order to reduce the overfitting a regularization term for the weight to the loss function can be added, and the loss function takes the form:

$$E_R(\theta) = E(\theta) + \lambda \sum_{i=1}^{N} w_i^2 \tag{5.3}$$

Where w_i is i-th element of the the weight vector w, and λ is the regularization factor. After a series of tests, the regularization factor was set to $\lambda = 0{,}005$.

Class Weighting. As already shown in the previous chapter, the classes of the new dataset are not balanced, and to improve the performance when class imbalance is present, class weighting can be used. Class weights define the relative importance of each class to the training process. They are inversely proportional to the frequency of the respective classes therefore they increase the importance of less prevalent classes to the training process.

N° of Epochs. It is the maximum number of epochs during the training. One epoch is when an entire dataset is passed forward and backward through the neural network only once. As the number of epochs increases, a greater number of times the weights are changed in the neural network, and as it increases, the result goes from underfitting to optimal to overfitting. Experiments have shown that over 30 epochs there was no remarkable benefits in terms of loss, hence the maximum was set to *35 epochs*.

5.3.4 Evaluation metrics

In addition to a visual evaluations and assessments based on human perception, it is fundamental that semantic segmentation systems are evaluated rigorously, in order to compare the performance in a systematic way. This evaluation must also be conducted using standard and well-known metrics that allow fair comparisons with other existing methods. Several aspects can be evaluated to assert the usefulness and the validity of a model: *execution time*, *memory footprint*, and *accuracy*. Depending on the task, the purposes, or the context some metrics could be more important than others. In this study the attention will be focused on the accuracy, not having to deal for example with real time applications, that require a certain execution speed, or not needing a specific threshold of memory usage. Nevertheless, they are two important aspects to deal with, since they have a relevant impact during the training. Many evaluation criteria have been proposed to assess the accuracy of semantic segmentation models. The predicted labels are divided into true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Common evaluation metrics for the semantic segmentation are the overall accuracy, the precision, recall and the F$_1$ score. The overall accuracy is the ratio of the correct classified pixels to the total number of pixels, both correct and incorrect.

$$Overall\_Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.4}$$

Precision gives the percentage of correct predictions:

$$Precision = \frac{TP}{TP + FP} \tag{5.5}$$

while recall gives the percentage of the correctly predicted positives:

$$Recall = \frac{TP}{TP + FN} \tag{5.6}$$

And their harmonic mean is the so-called F1-measure:

$$F_1 = \frac{2(Precision \times Recall)}{Precision + Recall} \tag{5.7}$$

In the context of this study three main evaluation metrics will be used: *Overall Accuracy* or *Global Accuracy*, *mean Intersection Over Union*, and the *Confusion Matrix*. The *Global Accuracy* (GA) is the ratio of correctly classified pixels, regardless of class, to the total number of pixels. It is used to have a quick and computationally inexpensive estimation of the percentage of correctly classified pixels. The *Intersection Over Union* (IoU), also known as the Jaccard Index, is one of the most commonly used metrics for the task of semantic segmentation. In addition to being extremely effective, it is very straightforward to implement as well, and it is used mostly to penalize false positive. IoU is defined as the area of overlap between the predicted segmentation and the ground truth, divided by the area of union between the predicted segmentation and the ground truth. It ranges from 0-1 with 0 meaning no overlap, and 1 a perfect overlap. To evaluate multi-class segmentation problems, the mean IoU (mIoU) is employed, and it is calculated by averaging the IoU of each class. The *Confusion Matrix*, also known as error matrix, is a table layout that allows the visualization of the performance of the model. Each row represents the instances in the predicted class, while each column the instances in the true class. It allows a more in-depth analysis of the performance of the model, by comparing and identifying the error for each class. GA and mIoU are defined in the equation below:

$$GA = \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{5.8}$$

$$mIoU = \frac{1}{n_c} \sum_i \frac{n_{ii}}{(t_i + \sum_j n_{ji} - n_{ii})} \tag{5.9}$$

where n_c = number of classes included in ground truth

n_{ij} = number of pixels of class *i* predicted to belong class *j*

t_i = total number of pixels of class *i* in ground truth

5.4 Image segmentation results

In this paragraph the results obtained with the various semantic segmentation models on the images will be reported. The related results on the point cloud will be illustrated in section 5.7. For each of the proposed test will be reported the selected evaluation metrics, the *Global Accuracy*, the *mean Intersection over Union*, the *F_1 Score*, and the *Confusion Matrix*. In addition, some comparisons between the a) input, b) the ground truth, and c) the prediction output by the neural network on the test images will be shown. At first the results of Test A for each of the five building are reported. Each of the five tests have been performed with the 3 architectures: FCN, SegNet and DeepLabv3+. These series of tests were helpful to tune the hyperparameters, to set up the training options, and to evaluate the best working architecture on the data. Several trainings have been performed, and in the following paragraphs the best obtained performances will be reported. Secondly, the result of Test B will be shown. All the five buildings have been used to train and test the network, and the test was performed only with DeepLabv3+, that turned out to be the most suitable. Finally, the results of test C will be reported. This test is a cross validation between the five buildings of the dataset, hence five different results are available. For each test the performance on the internal and the external test set is compared, together with some examples of the image predictions on the external test set. For test C only DeepLabv3+ has been employed. A summary of the training tests is reported in Figure 5.8.

Figure 5.8 – Summary of the training test and experiments

TEST		TRAINING SET	TEST SET
A	**A.1**	(1_SC)	(1_SC)
	A.2	(2_OSA)	(2_OSA)
	A.3	(3_SS)	(3_SS)
	A.4	(3_SS)	(4_CG)
	A.5	(5_CB)	(5_CB)
B		(1_SC), (2_OSA), (3_SS), (4_CG), (5_CB)	(1_SC), (2_OSA), (3_SS), (4_CG), (5_CB)
C	**C.1**	(2_OSA), (3_SS), (4_CG), (5_CB)	(1_SC)
	C.2	(1_SC), (3_SS), (4_CG), (5_CB)	(2_OSA)
	C.3	(1_SC), (2_OSA), (4_CG), (5_CB)	(3_SS)
	C.4	(1_SC), (2_OSA), (3_SS), (5_CB)	(4_CG)
	C.5	(1_SC), (2_OSA), (3_SS), (4_CG)	(5_CB)

5.4.1 Test A

A.1 (1_SC) Spedale del Ceppo

Table 5.1 – (1_SC) Class metrics using Fully Convolutional Network (FCN).

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,67	0,77	0,48	0,78	0,60	0,68	0,76	0,84	0,90	0,78	0,43
IoU	0,49	0,33	0,35	0,57	0,28	0,47	0,45	0,50	0,45	0,46	0,40
F_1 score	0,62	0,34	0,44	0,35	0,31	0,47	0,36	0,58	0,44	0,64	0,31

Table 5.2 – (1_SC) Class metrics using SegNet.

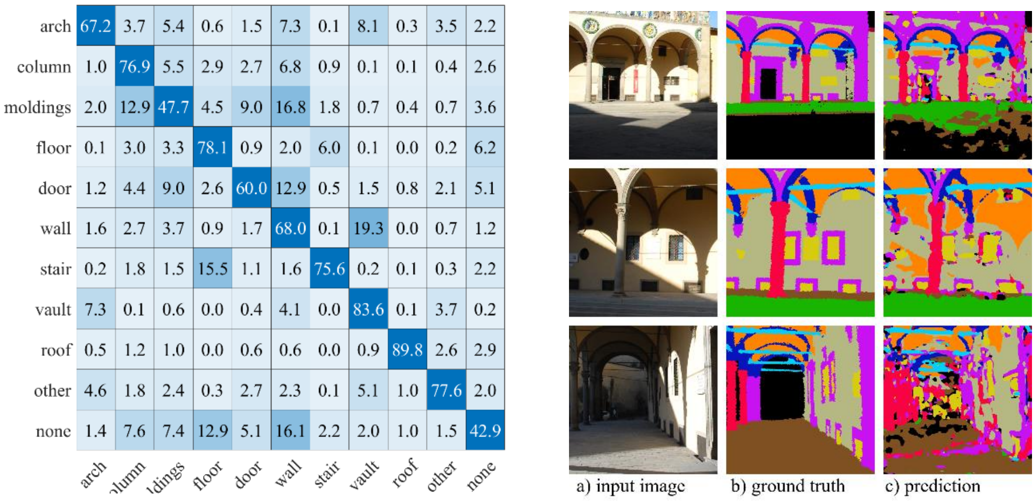
	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,89	0,93	0,83	0,94	0,78	0,87	0,91	0,88	0,94	0,89	0,85
IoU	0,67	0,73	0,70	0,88	0,55	0,78	0,75	0,81	0,70	0,55	0,82
F_1 score	0,77	0,71	0,70	0,73	0,59	0,71	0,68	0,76	0,62	0,74	0,55

Table 5.3 – (1_SC) Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,90	0,95	0,90	0,96	0,91	0,92	0,93	0,91	0,95	0,91	0,92
IoU	0,76	0,83	0,80	0,92	0,73	0,85	0,81	0,87	0,86	0,66	0,90
F_1 score	0,88	0,85	0,83	0,80	0,78	0,81	0,80	0,86	0,91	0,89	0,69

Table 5.4 – (1_SC) Dataset metrics.

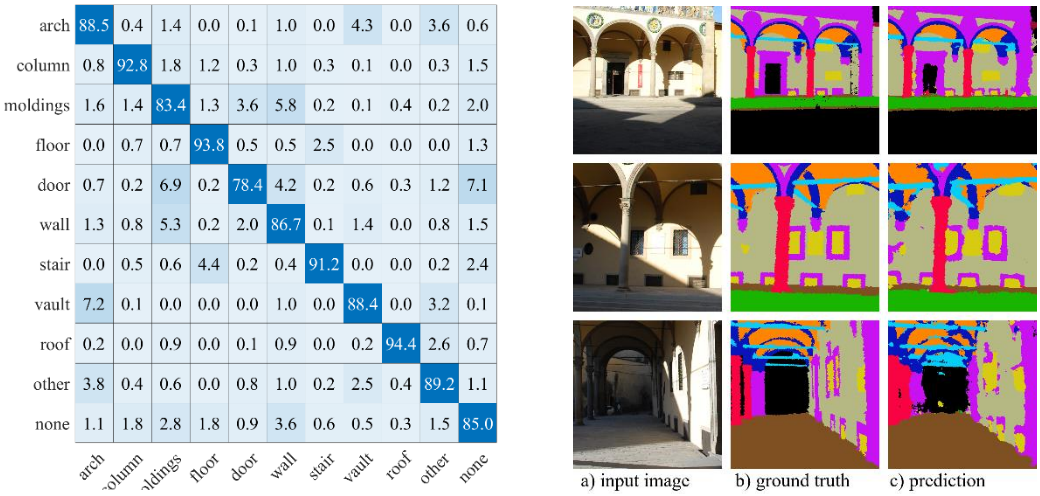
	Global Accuracy	mean IoU	mean F_1 score
FCN	0,60	0,43	0,43
SegNet	0,87	0,72	0,67
DeepLabv3+	**0,92**	**0,81**	**0,80**



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	67.2	3.7	5.4	0.6	1.5	7.3	0.1	8.1	0.3	3.5	2.2
column	1.0	76.9	5.5	2.9	2.7	6.8	0.9	0.1	0.1	0.4	2.6
moldings	2.0	12.9	47.7	4.5	9.0	16.8	1.8	0.7	0.4	0.7	3.6
floor	0.1	3.0	3.3	78.1	0.9	2.0	6.0	0.1	0.0	0.2	6.2
door	1.2	4.4	9.0	2.6	60.0	12.9	0.5	1.5	0.8	2.1	5.1
wall	1.6	2.7	3.7	0.9	1.7	68.0	0.1	19.3	0.0	0.7	1.2
stair	0.2	1.8	1.5	15.5	1.1	1.6	75.6	0.2	0.1	0.3	2.2
vault	7.3	0.1	0.6	0.0	0.4	4.1	0.0	83.6	0.1	3.7	0.2
roof	0.5	1.2	1.0	0.0	0.6	0.6	0.0	0.9	89.8	2.6	2.9
other	4.6	1.8	2.4	0.3	2.7	2.3	0.1	5.1	1.0	77.6	2.0
none	1.4	7.6	7.4	12.9	5.1	16.1	2.2	2.0	1.0	1.5	42.9

a) input image b) ground truth c) prediction

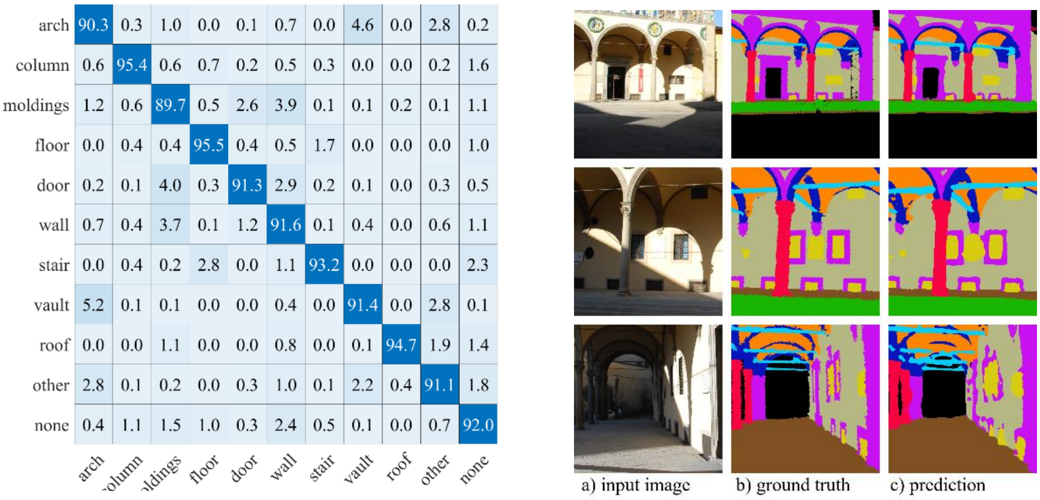
Figure 5.10 – (1_SC) Confusion matrix and image predictions with FCN.



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	88.5	0.4	1.4	0.0	0.1	1.0	0.0	4.3	0.0	3.6	0.6
column	0.8	92.8	1.8	1.2	0.3	1.0	0.3	0.1	0.0	0.3	1.5
moldings	1.6	1.4	83.4	1.3	3.6	5.8	0.2	0.1	0.4	0.2	2.0
floor	0.0	0.7	0.7	93.8	0.5	0.5	2.5	0.0	0.0	0.0	1.3
door	0.7	0.2	6.9	0.2	78.4	4.2	0.2	0.6	0.3	1.2	7.1
wall	1.3	0.8	5.3	0.2	2.0	86.7	0.1	1.4	0.0	0.8	1.5
stair	0.0	0.5	0.6	4.4	0.2	0.4	91.2	0.0	0.0	0.2	2.4
vault	7.2	0.1	0.0	0.0	0.0	1.0	0.0	88.4	0.0	3.2	0.1
roof	0.2	0.0	0.9	0.0	0.1	0.9	0.0	0.2	94.4	2.6	0.7
other	3.8	0.4	0.6	0.0	0.8	1.0	0.2	2.5	0.4	89.2	1.1
none	1.1	1.8	2.8	1.8	0.9	3.6	0.6	0.5	0.3	1.5	85.0

a) input image b) ground truth c) prediction

Figure 5.9 – (1_SC) Confusion matrix and image predictions with SegNet.



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	90.3	0.3	1.0	0.0	0.1	0.7	0.0	4.6	0.0	2.8	0.2
column	0.6	95.4	0.6	0.7	0.2	0.5	0.3	0.0	0.0	0.2	1.6
moldings	1.2	0.6	89.7	0.5	2.6	3.9	0.1	0.1	0.2	0.1	1.1
floor	0.0	0.4	0.4	95.5	0.4	0.5	1.7	0.0	0.0	0.0	1.0
door	0.2	0.1	4.0	0.3	91.3	2.9	0.2	0.1	0.0	0.3	0.5
wall	0.7	0.4	3.7	0.1	1.2	91.6	0.1	0.4	0.0	0.6	1.1
stair	0.0	0.4	0.2	2.8	0.0	1.1	93.2	0.0	0.0	0.0	2.3
vault	5.2	0.1	0.1	0.0	0.0	0.4	0.0	91.4	0.0	2.8	0.1
roof	0.0	0.0	1.1	0.0	0.0	0.8	0.0	0.1	94.7	1.9	1.4
other	2.8	0.1	0.2	0.0	0.3	1.0	0.1	2.2	0.4	91.1	1.8
none	0.4	1.1	1.5	1.0	0.3	2.4	0.5	0.1	0.0	0.7	92.0

a) input image b) ground truth c) prediction

Figure 5.11 – (1_SC) Confusion matrix and image predictions with DeepLabv3+.

A.2 (2_OSA) Ospedale Sant'Antonio

Table 5.5 – (2_OSA) Class metrics using Fully Convolutional Network (FCN).

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,52	0,53	0,67	0,64	0,09	0,26	-	0,24	0,54	0,69	0,70
IoU	0,20	0,30	0,23	0,60	0,09	0,23	-	0,22	0,17	0,08	0,38
F_1 score	0,37	0,33	0,30	0,37	0,16	0,21	-	0,24	0,24	0,20	0,23

Table 5.6 – (2_OSA) Class metrics using SegNet.

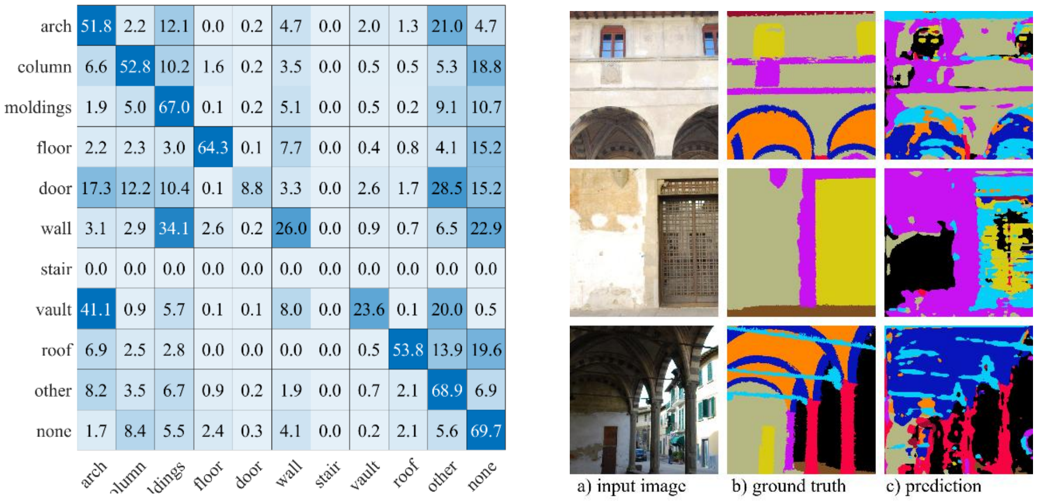
	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,84	0,85	0,75	0,95	0,95	0,82	-	0,84	0,95	0,81	0,87
IoU	0,62	0,64	0,53	0,86	0,85	0,77	-	0,79	0,80	0,35	0,84
F_1 score	0,71	0,59	0,54	0,61	0,74	0,55	-	0,65	0,85	0,46	0,52

Table 5.7 – (2_OSA) Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,90	0,93	0,87	0,97	0,98	0,92	-	0,91	0,99	0,82	0,94
IoU	0,73	0,80	0,73	0,94	0,91	0,90	-	0,87	0,84	0,57	0,91
F_1 score	0,85	0,84	0,79	0,81	0,89	0,78	-	0,84	0,88	0,79	0,75

Table 5.8 – (2_OSA) Dataset metrics.

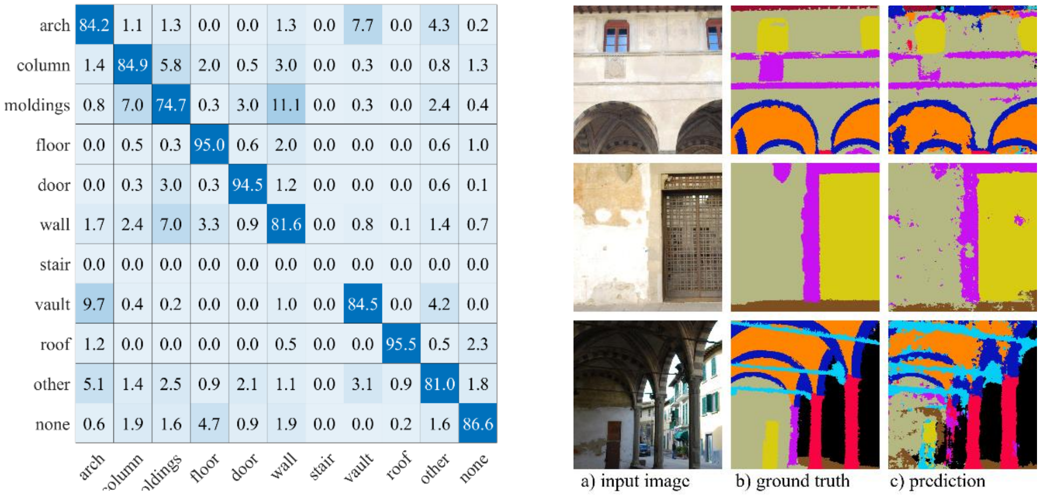
	Global Accuracy	mean IoU	mean F_1 score
FCN	0,44	0,22	0,25
SegNet	0,85	0,70	0,58
DeepLabv3+	**0,93**	**0,81**	**0,80**



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	51.8	2.2	12.1	0.0	0.2	4.7	0.0	2.0	1.3	21.0	4.7
column	6.6	52.8	10.2	1.6	0.2	3.5	0.0	0.5	0.5	5.3	18.8
moldings	1.9	5.0	67.0	0.1	0.2	5.1	0.0	0.5	0.2	9.1	10.7
floor	2.2	2.3	3.0	64.3	0.1	7.7	0.0	0.4	0.8	4.1	15.2
door	17.3	12.2	10.4	0.1	8.8	3.3	0.0	2.6	1.7	28.5	15.2
wall	3.1	2.9	34.1	2.6	0.2	26.0	0.0	0.9	0.7	6.5	22.9
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	41.1	0.9	5.7	0.1	0.1	8.0	0.0	23.6	0.1	20.0	0.5
roof	6.9	2.5	2.8	0.0	0.0	0.0	0.0	0.5	53.8	13.9	19.6
other	8.2	3.5	6.7	0.9	0.2	1.9	0.0	0.7	2.1	68.9	6.9
none	1.7	8.4	5.5	2.4	0.3	4.1	0.0	0.2	2.1	5.6	69.7

a) input image b) ground truth c) prediction

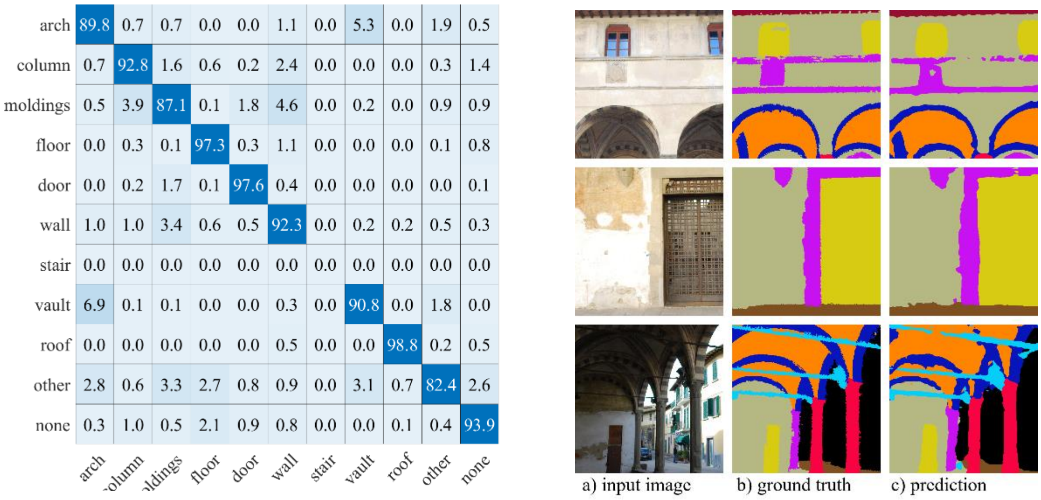
Figure 5.13 – (2_OSA) Confusion matrix and image predictions with FCN.



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	84.2	1.1	1.3	0.0	0.0	1.3	0.0	7.7	0.0	4.3	0.2
column	1.4	84.9	5.8	2.0	0.5	3.0	0.0	0.3	0.0	0.8	1.3
moldings	0.8	7.0	74.7	0.3	3.0	11.1	0.0	0.3	0.0	2.4	0.4
floor	0.0	0.5	0.3	95.0	0.6	2.0	0.0	0.0	0.0	0.6	1.0
door	0.0	0.3	3.0	0.3	94.5	1.2	0.0	0.0	0.0	0.6	0.1
wall	1.7	2.4	7.0	3.3	0.9	81.6	0.0	0.8	0.1	1.4	0.7
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	9.7	0.4	0.2	0.0	0.0	1.0	0.0	84.5	0.0	4.2	0.0
roof	1.2	0.0	0.0	0.0	0.0	0.5	0.0	0.0	95.5	0.5	2.3
other	5.1	1.4	2.5	0.9	2.1	1.1	0.0	3.1	0.9	81.0	1.8
none	0.6	1.9	1.6	4.7	0.9	1.9	0.0	0.0	0.2	1.6	86.6

a) input image b) ground truth c) prediction

Figure 5.12 – (2_OSA) Confusion matrix and image predictions with SegNet.



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	89.8	0.7	0.7	0.0	0.0	1.1	0.0	5.3	0.0	1.9	0.5
column	0.7	92.8	1.6	0.6	0.2	2.4	0.0	0.0	0.0	0.3	1.4
moldings	0.5	3.9	87.1	0.1	1.8	4.6	0.0	0.2	0.0	0.9	0.9
floor	0.0	0.3	0.1	97.3	0.3	1.1	0.0	0.0	0.0	0.1	0.8
door	0.0	0.2	1.7	0.1	97.6	0.4	0.0	0.0	0.0	0.0	0.1
wall	1.0	1.0	3.4	0.6	0.5	92.3	0.0	0.2	0.2	0.5	0.3
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	6.9	0.1	0.1	0.0	0.0	0.3	0.0	90.8	0.0	1.8	0.0
roof	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	98.8	0.2	0.5
other	2.8	0.6	3.3	2.7	0.8	0.9	0.0	3.1	0.7	82.4	2.6
none	0.3	1.0	0.5	2.1	0.9	0.8	0.0	0.0	0.1	0.4	93.9

a) input image b) ground truth c) prediction

Figure 5.14 – (2_OSA) Confusion matrix and image predictions with DeeLabv3+.

A.3 (3_SS) Basilica della Santissima Annunziata

Table 5.9 – (3_SS) Class metrics using Fully Convolutional Network (FCN).

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,86	0,88	0,73	0,91	0,96	0,82	0,87	0,86	-	0,86	0,81
IoU	0,58	0,59	0,67	0,68	0,62	0,74	0,08	0,79	-	0,28	0,77
F_1 score	0,79	0,56	0,67	0,55	0,65	0,75	0,31	0,82	-	0,49	0,58

Table 5.10 – (3_SS) Class metrics using SegNet.

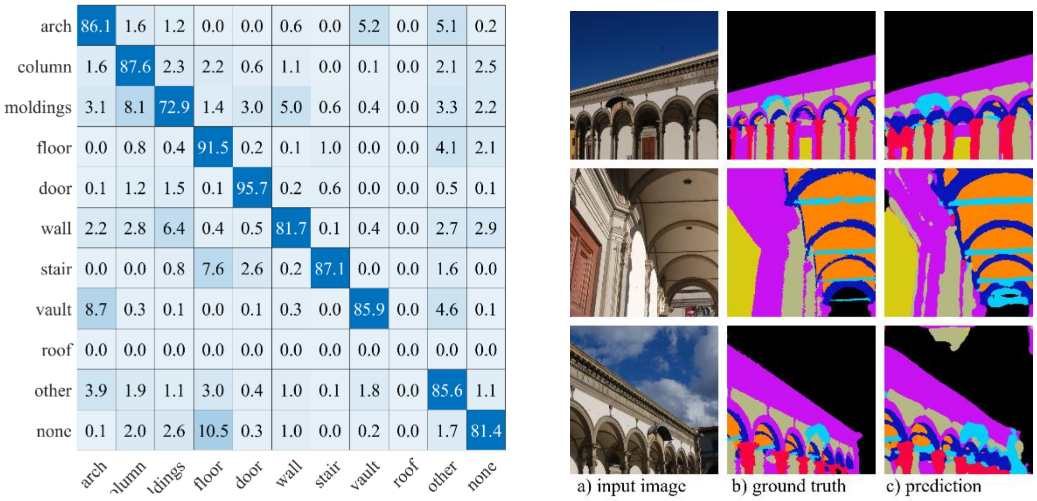
	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,89	0,92	0,87	0,91	0,96	0,91	0,86	0,90	-	0,82	0,90
IoU	0,68	0,82	0,81	0,80	0,74	0,83	0,32	0,84	-	0,44	0,87
F_1 score	0,85	0,88	0,83	0,65	0,76	0,85	0,65	0,90	-	0,64	0,70

Table 5.11 – (3_SS) Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,88	0,91	0,87	0,92	0,94	0,88	0,75	0,90	-	0,79	0,93
IoU	0,65	0,75	0,80	0,84	0,74	0,81	0,37	0,83	-	0,48	0,90
F_1 score	0,84	0,76	0,80	0,70	0,76	0,84	0,66	0,88	-	0,71	0,73

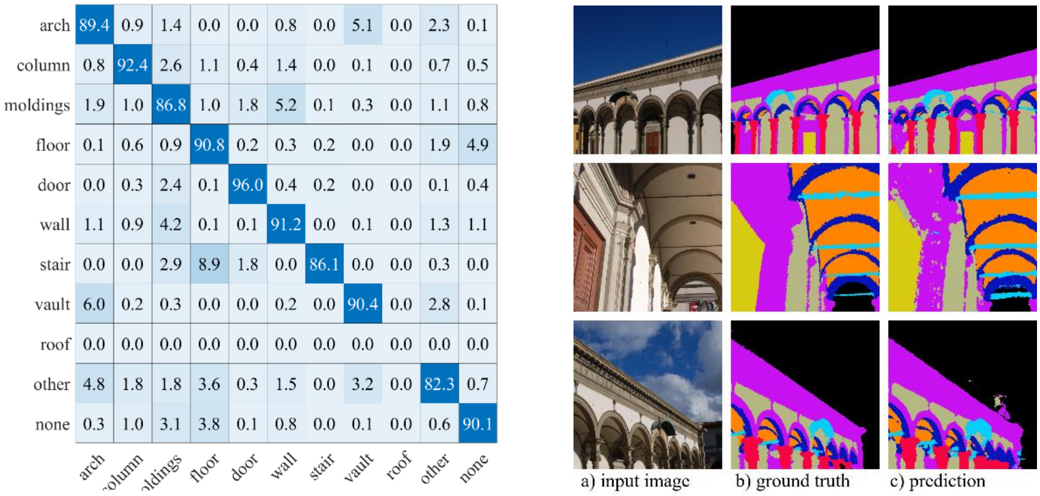
Table 5.12 – (3_SS) Dataset metrics.

	Global Accuracy	mean IoU	mean F_1 score
FCN	0,81	0,58	0,64
SegNet	0,89	0,70	0,76
DeepLabv3+	**0,89**	**0,71**	**0,77**



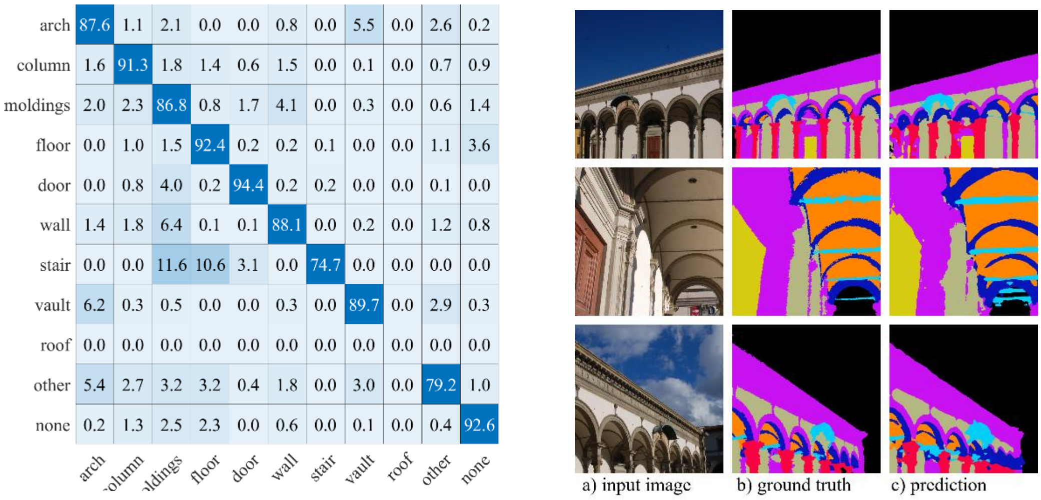
	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	86.1	1.6	1.2	0.0	0.0	0.6	0.0	5.2	0.0	5.1	0.2
column	1.6	87.6	2.3	2.2	0.6	1.1	0.0	0.1	0.0	2.1	2.5
moldings	3.1	8.1	72.9	1.4	3.0	5.0	0.6	0.4	0.0	3.3	2.2
floor	0.0	0.8	0.4	91.5	0.2	0.1	1.0	0.0	0.0	4.1	2.1
door	0.1	1.2	1.5	0.1	95.7	0.2	0.6	0.0	0.0	0.5	0.1
wall	2.2	2.8	6.4	0.4	0.5	81.7	0.1	0.4	0.0	2.7	2.9
stair	0.0	0.0	0.8	7.6	2.6	0.2	87.1	0.0	0.0	1.6	0.0
vault	8.7	0.3	0.1	0.0	0.1	0.3	0.0	85.9	0.0	4.6	0.1
roof	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
other	3.9	1.9	1.1	3.0	0.4	1.0	0.1	1.8	0.0	85.6	1.1
none	0.1	2.0	2.6	10.5	0.3	1.0	0.0	0.2	0.0	1.7	81.4

Figure 5.17 – (3_SS) Confusion matrix and image predictions with FCN.



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	89.4	0.9	1.4	0.0	0.0	0.8	0.0	5.1	0.0	2.3	0.1
column	0.8	92.4	2.6	1.1	0.4	1.4	0.0	0.1	0.0	0.7	0.5
moldings	1.9	1.0	86.8	1.0	1.8	5.2	0.1	0.3	0.0	1.1	0.8
floor	0.1	0.6	0.9	90.8	0.2	0.3	0.2	0.0	0.0	1.9	4.9
door	0.0	0.3	2.4	0.1	96.0	0.4	0.2	0.0	0.0	0.1	0.4
wall	1.1	0.9	4.2	0.1	0.1	91.2	0.0	0.1	0.0	1.3	1.1
stair	0.0	0.0	2.9	8.9	1.8	0.0	86.1	0.0	0.0	0.3	0.0
vault	6.0	0.2	0.3	0.0	0.0	0.2	0.0	90.4	0.0	2.8	0.1
roof	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
other	4.8	1.8	1.8	3.6	0.3	1.5	0.0	3.2	0.0	82.3	0.7
none	0.3	1.0	3.1	3.8	0.1	0.8	0.0	0.1	0.0	0.6	90.1

Figure 5.16 – (3_SS) Confusion matrix and image predictions with SegNet.



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	87.6	1.1	2.1	0.0	0.0	0.8	0.0	5.5	0.0	2.6	0.2
column	1.6	91.3	1.8	1.4	0.6	1.5	0.0	0.1	0.0	0.7	0.9
moldings	2.0	2.3	86.8	0.8	1.7	4.1	0.0	0.3	0.0	0.6	1.4
floor	0.0	1.0	1.5	92.4	0.2	0.2	0.1	0.0	0.0	1.1	3.6
door	0.0	0.8	4.0	0.2	94.4	0.2	0.2	0.0	0.0	0.1	0.0
wall	1.4	1.8	6.4	0.1	0.1	88.1	0.0	0.2	0.0	1.2	0.8
stair	0.0	0.0	11.6	10.6	3.1	0.0	74.7	0.0	0.0	0.0	0.0
vault	6.2	0.3	0.5	0.0	0.0	0.3	0.0	89.7	0.0	2.9	0.3
roof	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
other	5.4	2.7	3.2	3.2	0.4	1.8	0.0	3.0	0.0	79.2	1.0
none	0.2	1.3	2.5	2.3	0.0	0.6	0.0	0.1	0.0	0.4	92.6

Figure 5.15 – (3_SS) Confusion matrix and image predictions with DeepLabv3+.

A.4 (4_CG) Certosa del Galluzzo

Table 5.13 – (4_CG) Class metrics using Fully Convolutional Network (FCN).

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,93	0,85	0,77	0,91	0,93	0,72	-	0,80	0,83	0,80	0,68
IoU	0,35	0,54	0,42	0,79	0,19	0,67	-	0,68	0,67	0,49	0,63
F_1 score	0,51	0,55	0,55	0,53	0,36	0,50	-	0,57	0,73	0,50	0,40

Table 5.14 – (4_CG) Class metrics using SegNet.

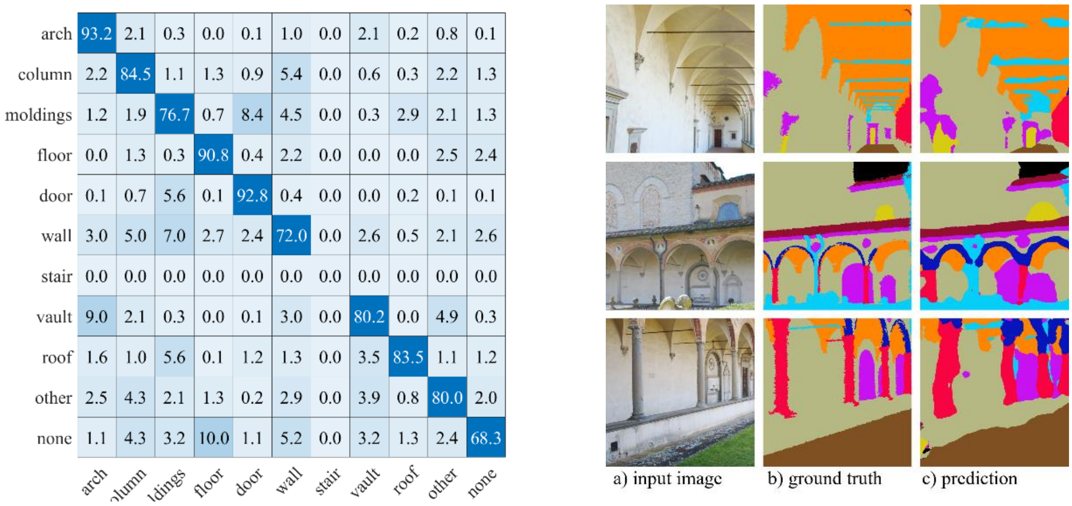
	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,89	0,84	0,79	0,92	0,86	0,71	-	0,84	0,84	0,74	0,62
IoU	0,34	0,50	0,41	0,80	0,32	0,64	-	0,67	0,66	0,47	0,57
F_1 score	0,52	0,46	0,54	0,57	0,49	0,51	-	0,60	0,67	0,48	0,41

Table 5.15 – (4_CG) Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,91	0,88	0,85	0,92	0,89	0,84	-	0,88	0,91	0,84	0,85
IoU	0,51	0,65	0,58	0,87	0,51	0,79	-	0,78	0,76	0,64	0,78
F_1 score	0,77	0,69	0,73	0,67	0,76	0,67	-	0,77	0,84	0,70	0,59

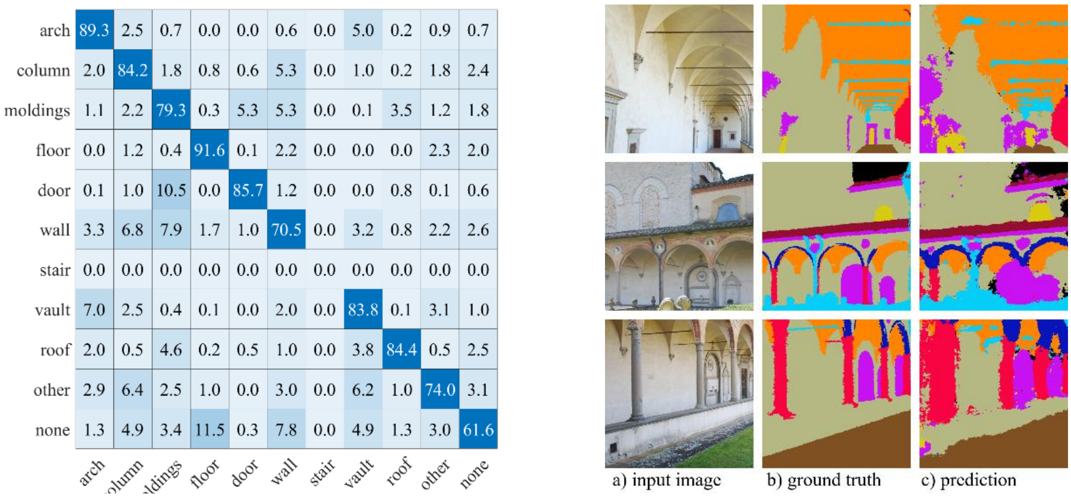
Table 5.16 – (4_CG) Dataset metrics.

	Global Accuracy	mean IoU	mean F_1 score
FCN	0,77	0,54	0,49
SegNet	0,76	0,53	0,50
DeepLabv3+	**0,86**	**0,68**	**0,68**



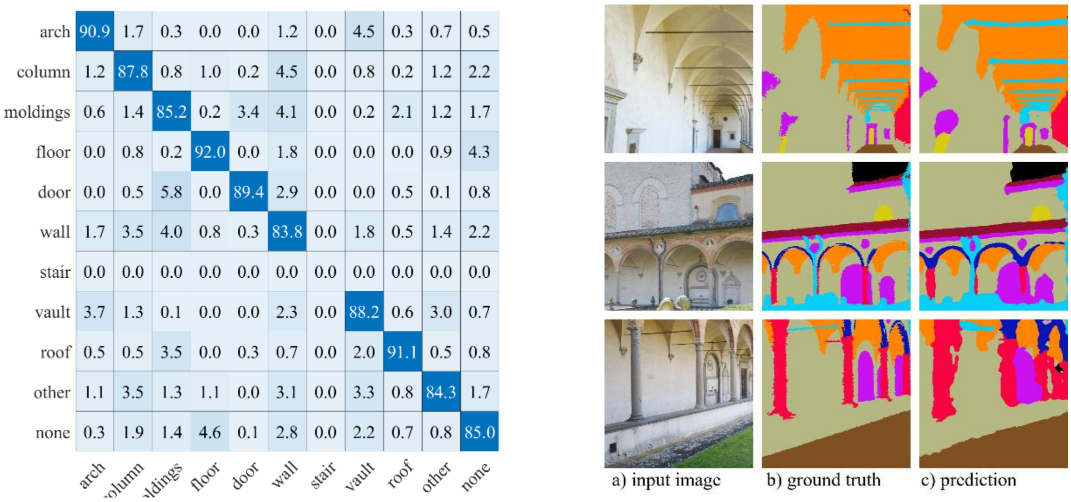
	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	93.2	2.1	0.3	0.0	0.1	1.0	0.0	2.1	0.2	0.8	0.1
column	2.2	84.5	1.1	1.3	0.9	5.4	0.0	0.6	0.3	2.2	1.3
moldings	1.2	1.9	76.7	0.7	8.4	4.5	0.0	0.3	2.9	2.1	1.3
floor	0.0	1.3	0.3	90.8	0.4	2.2	0.0	0.0	0.0	2.5	2.4
door	0.1	0.7	5.6	0.1	92.8	0.4	0.0	0.0	0.2	0.1	0.1
wall	3.0	5.0	7.0	2.7	2.4	72.0	0.0	2.6	0.5	2.1	2.6
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	9.0	2.1	0.3	0.0	0.1	3.0	0.0	80.2	0.0	4.9	0.3
roof	1.6	1.0	5.6	0.1	1.2	1.3	0.0	3.5	83.5	1.1	1.2
other	2.5	4.3	2.1	1.3	0.2	2.9	0.0	3.9	0.8	80.0	2.0
none	1.1	4.3	3.2	10.0	1.1	5.2	0.0	3.2	1.3	2.4	68.3

Figure 5.18 – (4_CG) Confusion matrix and image predictions with FCN.



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	89.3	2.5	0.7	0.0	0.0	0.6	0.0	5.0	0.2	0.9	0.7
column	2.0	84.2	1.8	0.8	0.6	5.3	0.0	1.0	0.2	1.8	2.4
moldings	1.1	2.2	79.3	0.3	5.3	5.3	0.0	0.1	3.5	1.2	1.8
floor	0.0	1.2	0.4	91.6	0.1	2.2	0.0	0.0	0.0	2.3	2.0
door	0.1	1.0	10.5	0.0	85.7	1.2	0.0	0.0	0.8	0.1	0.6
wall	3.3	6.8	7.9	1.7	1.0	70.5	0.0	3.2	0.8	2.2	2.6
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	7.0	2.5	0.4	0.1	0.0	2.0	0.0	83.8	0.1	3.1	1.0
roof	2.0	0.5	4.6	0.2	0.5	1.0	0.0	3.8	84.4	0.5	2.5
other	2.9	6.4	2.5	1.0	0.0	3.0	0.0	6.2	1.0	74.0	3.1
none	1.3	4.9	3.4	11.5	0.3	7.8	0.0	4.9	1.3	3.0	61.6

Figure 5.19 – (4_CG) Confusion matrix and image predictions with SegNet.



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	90.9	1.7	0.3	0.0	0.0	1.2	0.0	4.5	0.3	0.7	0.5
column	1.2	87.8	0.8	1.0	0.2	4.5	0.0	0.8	0.2	1.2	2.2
moldings	0.6	1.4	85.2	0.2	3.4	4.1	0.0	0.2	2.1	1.2	1.7
floor	0.0	0.8	0.2	92.0	0.0	1.8	0.0	0.0	0.0	0.9	4.3
door	0.0	0.5	5.8	0.0	89.4	2.9	0.0	0.0	0.5	0.1	0.8
wall	1.7	3.5	4.0	0.8	0.3	83.8	0.0	1.8	0.5	1.4	2.2
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	3.7	1.3	0.1	0.0	0.0	2.3	0.0	88.2	0.6	3.0	0.7
roof	0.5	0.5	3.5	0.0	0.3	0.7	0.0	2.0	91.1	0.5	0.8
other	1.1	3.5	1.3	1.1	0.0	3.1	0.0	3.3	0.8	84.3	1.7
none	0.3	1.9	1.4	4.6	0.1	2.8	0.0	2.2	0.7	0.8	85.0

Figure 5.20 – (4_CG) Confusion matrix and image predictions with DeepLabv3+.

A.5 (5_CB) Cappella Buontalenti

Table 5.17 – (5_CB) Class metrics using Fully Convolutional Network (FCN).

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,62	0,70	0,59	0,76	0,60	0,56	0,18	0,73	0,53	0,55	0,82
IoU	0,23	0,43	0,27	0,38	0,43	0,51	0,16	0,53	0,41	0,11	0,74
F_1 score	0,27	0,35	0,35	0,28	0,33	0,32	0,26	0,36	0,24	0,26	0,37

Table 5.18 – (5_CB) Class metrics using SegNet.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,78	0,88	0,76	0,73	0,79	0,61	0,00	0,76	0,61	0,74	0,86
IoU	0,28	0,63	0,31	0,41	0,59	0,57	0,00	0,61	0,49	0,15	0,80
F_1 score	0,35	0,39	0,36	0,33	0,43	0,37	0,00	0,44	0,34	0,26	0,37

Table 5.19 – (5_CB) Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,76	0,88	0,73	0,91	0,91	0,83	0,90	0,85	0,87	0,70	0,89
IoU	0,41	0,71	0,52	0,73	0,76	0,76	0,76	0,76	0,75	0,44	0,85
F_1 score	0,48	0,54	0,60	0,26	0,54	0,48	0,49	0,55	0,40	0,48	0,47

Table 5.20 – (5_CB) Dataset metrics.

	Global Accuracy	mean IoU	mean F_1 score
FCN	0,67	0,38	0,31
SegNet	0,72	0,43	0,35
DeepLabv3+	**0,86**	**0,67**	**0,49**

	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	62.2	4.6	2.4	0.5	0.4	3.4	0.2	18.0	0.4	4.0	3.9
column	4.0	70.2	3.2	1.7	2.4	5.1	0.2	0.5	0.4	4.0	8.3
moldings	0.6	3.4	59.1	5.0	3.6	14.4	0.3	1.8	1.0	6.1	4.6
floor	0.1	1.0	5.8	76.2	2.8	0.9	0.7	0.0	1.9	6.7	3.9
door	2.0	5.0	2.3	0.4	59.7	2.6	0.5	1.0	2.5	17.3	6.7
wall	3.4	5.7	9.3	4.0	2.1	55.8	0.2	13.6	0.4	3.3	2.3
stair	0.3	2.9	1.2	37.8	0.4	2.1	18.4	0.1	16.5	6.3	14.0
vault	11.6	1.1	1.7	0.1	0.7	4.6	0.2	72.5	0.1	5.9	1.4
roof	1.3	1.7	4.3	7.4	0.4	1.7	1.2	0.6	52.8	8.3	20.3
other	4.2	2.4	6.5	9.7	7.8	1.4	0.3	2.4	7.4	55.5	2.6
none	1.7	4.1	0.6	1.9	0.9	2.5	0.6	0.3	3.4	1.6	82.4

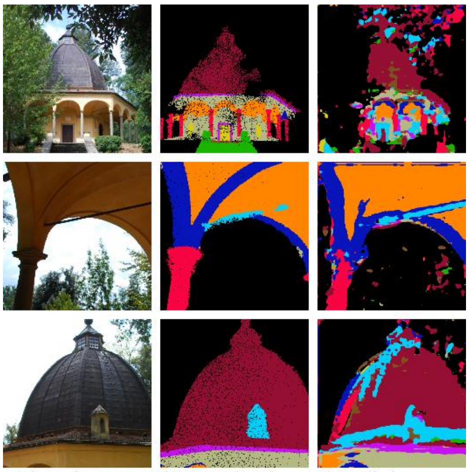


Figure 5.22 – (5_CB) Confusion matrix and image predictions with FCN.

	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	77.9	1.1	1.9	0.5	0.0	1.6	0.0	11.8	0.2	3.7	1.3
column	2.2	88.4	2.4	0.7	0.3	1.2	0.0	0.2	0.1	0.5	4.0
moldings	0.7	1.1	76.2	4.7	1.6	4.8	0.0	1.0	1.6	5.6	2.7
floor	0.4	1.7	5.2	73.5	0.8	1.1	0.0	0.0	1.5	11.2	4.6
door	0.6	0.4	2.7	0.6	78.8	2.1	0.0	0.1	5.5	9.0	0.2
wall	5.3	2.6	11.4	2.6	1.6	60.5	0.0	9.4	0.5	3.5	2.6
stair	0.4	2.0	9.0	41.2	1.5	2.4	0.0	0.0	17.5	13.3	12.6
vault	11.7	1.1	2.1	0.9	0.4	2.3	0.0	76.3	0.1	3.7	1.3
roof	0.3	0.0	3.5	0.1	5.3	2.0	0.0	0.2	60.9	16.5	11.3
other	4.7	1.4	3.8	3.3	3.5	0.7	0.0	1.8	6.1	73.9	0.7
none	1.6	3.8	0.8	1.8	0.7	2.1	0.0	0.2	1.8	1.1	86.1

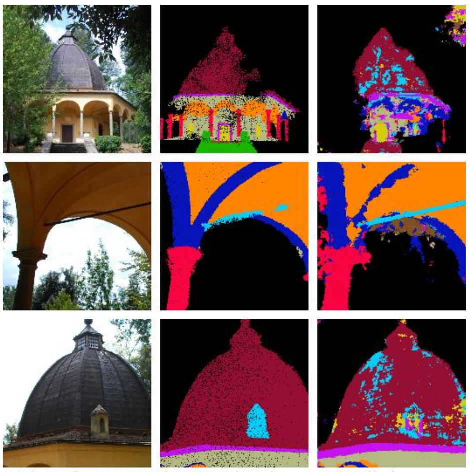


Figure 5.21 – (5_CB) Confusion matrix and image predictions with SegNet.

	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	76.1	2.1	0.5	0.3	0.0	2.5	0.0	16.3	0.3	1.0	0.8
column	0.6	88.2	0.6	0.4	0.4	3.8	0.3	0.7	0.0	0.6	4.4
moldings	0.3	0.6	73.5	1.7	1.9	12.6	0.3	1.6	4.7	1.1	1.6
floor	0.0	0.4	0.4	90.7	0.0	2.8	1.6	0.0	0.0	1.4	2.5
door	0.0	0.4	1.2	0.0	90.6	5.0	0.4	0.1	0.2	1.6	0.3
wall	2.0	1.4	3.1	1.3	1.7	83.4	0.7	3.5	0.4	0.6	1.9
stair	0.0	0.2	0.1	1.9	0.0	5.2	89.6	0.0	0.7	0.1	2.2
vault	7.1	1.4	0.9	0.1	0.3	2.3	0.0	85.5	0.0	1.6	0.8
roof	0.1	0.1	3.0	0.2	0.7	2.6	0.0	0.2	87.4	0.1	5.7
other	2.8	2.0	0.9	1.3	1.1	6.2	0.4	8.5	3.9	70.2	2.8
none	0.8	2.2	0.3	1.3	0.4	2.1	1.1	0.3	2.7	0.3	88.6

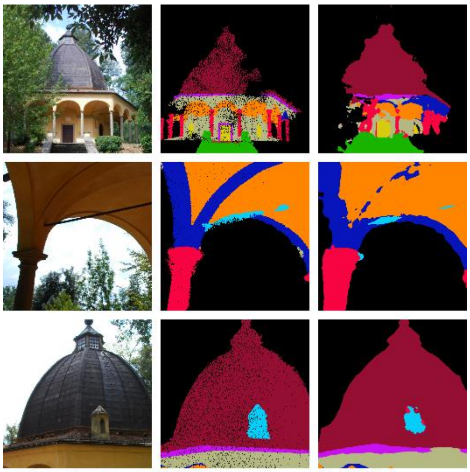


Figure 5.23 – (5_CB) Confusion matrix and image predictions with DeepLabv3+.

5.4.2 Test B

Table 5.21 – Class metrics using Fully Convolutional Network (FCN).

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,89	0,89	0,86	0,93	0,92	0,84	0,91	0,88	0,89	0,81	0,87
IoU	0,63	0,70	0,72	0,86	0,76	0,79	0,75	0,80	0,73	0,54	0,81
F_1 score	0,77	0,69	0,70	0,64	0,72	0,66	0,67	0,76	0,75	0,67	0,57

Table 5.22 – Class metrics using SegNet.

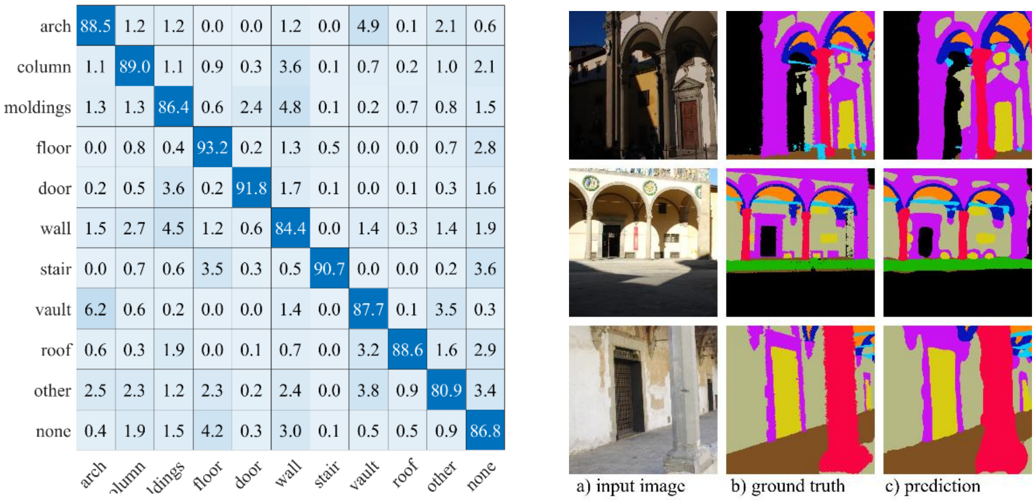
	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,82	0,84	0,74	0,88	0,86	0,68	0,82	0,77	0,84	0,72	0,67
IoU	0,45	0,55	0,51	0,77	0,58	0,60	0,56	0,67	0,55	0,29	0,60
F_1 score	0,56	0,48	0,48	0,49	0,44	0,44	0,39	0,55	0,49	0,38	0,34

Table 5.23 – Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
accuracy	0,90	0,91	0,90	0,95	0,96	0,87	0,93	0,90	0,93	0,87	0,87
IoU	0,68	0,75	0,76	0,88	0,76	0,83	0,76	0,84	0,76	0,61	0,84
F_1 score	0,83	0,78	0,77	0,68	0,74	0,73	0,70	0,81	0,78	0,75	0,61

Table 5.24 – Dataset metrics.

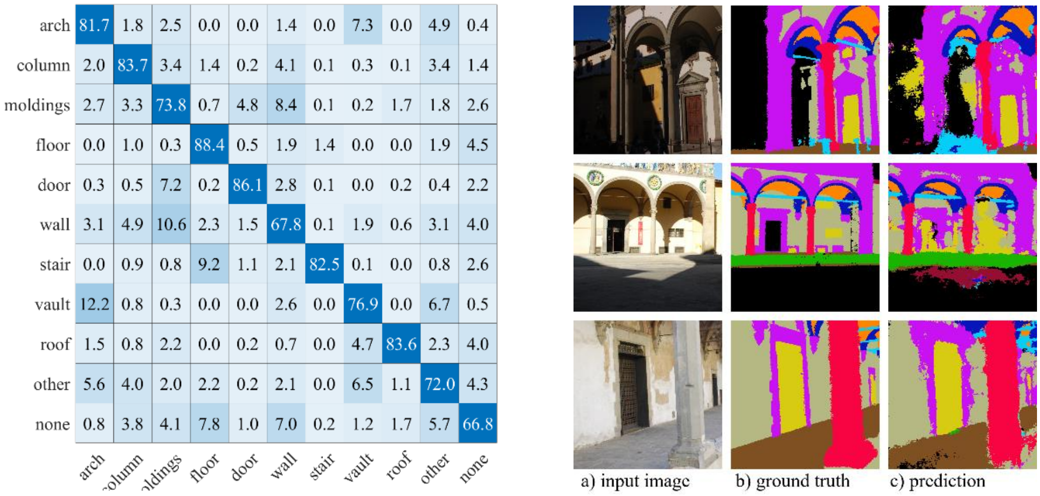
	Global Accuracy	mean IoU	mean F_1 score
FCN	0,87	0,73	0,67
SegNet	0,74	0,55	0,45
DeepLabv3+	**0,89**	**0,76**	**0,73**



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	88.5	1.2	1.2	0.0	0.0	1.2	0.0	4.9	0.1	2.1	0.6
column	1.1	89.0	1.1	0.9	0.3	3.6	0.1	0.7	0.2	1.0	2.1
moldings	1.3	1.3	86.4	0.6	2.4	4.8	0.1	0.2	0.7	0.8	1.5
floor	0.0	0.8	0.4	93.2	0.2	1.3	0.5	0.0	0.0	0.7	2.8
door	0.2	0.5	3.6	0.2	91.8	1.7	0.1	0.0	0.1	0.3	1.6
wall	1.5	2.7	4.5	1.2	0.6	84.4	0.0	1.4	0.3	1.4	1.9
stair	0.0	0.7	0.6	3.5	0.3	0.5	90.7	0.0	0.0	0.2	3.6
vault	6.2	0.6	0.2	0.0	0.0	1.4	0.0	87.7	0.1	3.5	0.3
roof	0.6	0.3	1.9	0.0	0.1	0.7	0.0	3.2	88.6	1.6	2.9
other	2.5	2.3	1.2	2.3	0.2	2.4	0.0	3.8	0.9	80.9	3.4
none	0.4	1.9	1.5	4.2	0.3	3.0	0.1	0.5	0.5	0.9	86.8

a) input image b) ground truth c) prediction

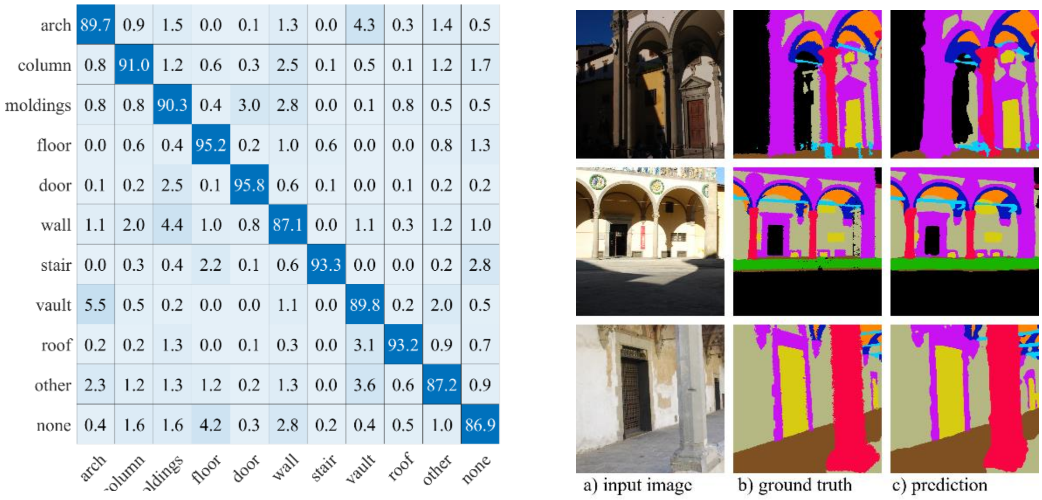
Figure 5.25 – Confusion matrix and image predictions with FCN.



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	81.7	1.8	2.5	0.0	0.0	1.4	0.0	7.3	0.0	4.9	0.4
column	2.0	83.7	3.4	1.4	0.2	4.1	0.1	0.3	0.1	3.4	1.4
moldings	2.7	3.3	73.8	0.7	4.8	8.4	0.1	0.2	1.7	1.8	2.6
floor	0.0	1.0	0.3	88.4	0.5	1.9	1.4	0.0	0.0	1.9	4.5
door	0.3	0.5	7.2	0.2	86.1	2.8	0.1	0.0	0.2	0.4	2.2
wall	3.1	4.9	10.6	2.3	1.5	67.8	0.1	1.9	0.6	3.1	4.0
stair	0.0	0.9	0.8	9.2	1.1	2.1	82.5	0.1	0.0	0.8	2.6
vault	12.2	0.8	0.3	0.0	0.0	2.6	0.0	76.9	0.0	6.7	0.5
roof	1.5	0.8	2.2	0.0	0.2	0.7	0.0	4.7	83.6	2.3	4.0
other	5.6	4.0	2.0	2.2	0.2	2.1	0.0	6.5	1.1	72.0	4.3
none	0.8	3.8	4.1	7.8	1.0	7.0	0.2	1.2	1.7	5.7	66.8

a) input image b) ground truth c) prediction

Figure 5.24 – Confusion matrix and image predictions with SegNet.



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	89.7	0.9	1.5	0.0	0.1	1.3	0.0	4.3	0.3	1.4	0.5
column	0.8	91.0	1.2	0.6	0.3	2.5	0.1	0.5	0.1	1.2	1.7
moldings	0.8	0.8	90.3	0.4	3.0	2.8	0.0	0.1	0.8	0.5	0.5
floor	0.0	0.6	0.4	95.2	0.2	1.0	0.6	0.0	0.0	0.8	1.3
door	0.1	0.2	2.5	0.1	95.8	0.6	0.1	0.0	0.1	0.2	0.2
wall	1.1	2.0	4.4	1.0	0.8	87.1	0.0	1.1	0.3	1.2	1.0
stair	0.0	0.3	0.4	2.2	0.1	0.6	93.3	0.0	0.0	0.2	2.8
vault	5.5	0.5	0.2	0.0	0.0	1.1	0.0	89.8	0.2	2.0	0.5
roof	0.2	0.2	1.3	0.0	0.1	0.3	0.0	3.1	93.2	0.9	0.7
other	2.3	1.2	1.3	1.2	0.2	1.3	0.0	3.6	0.6	87.2	0.9
none	0.4	1.6	1.6	4.2	0.3	2.8	0.2	0.4	0.5	1.0	86.9

a) input image b) ground truth c) prediction

Figure 5.26 – Confusion matrix and image predictions with DeepLabv3+.

5.4.3 Test C

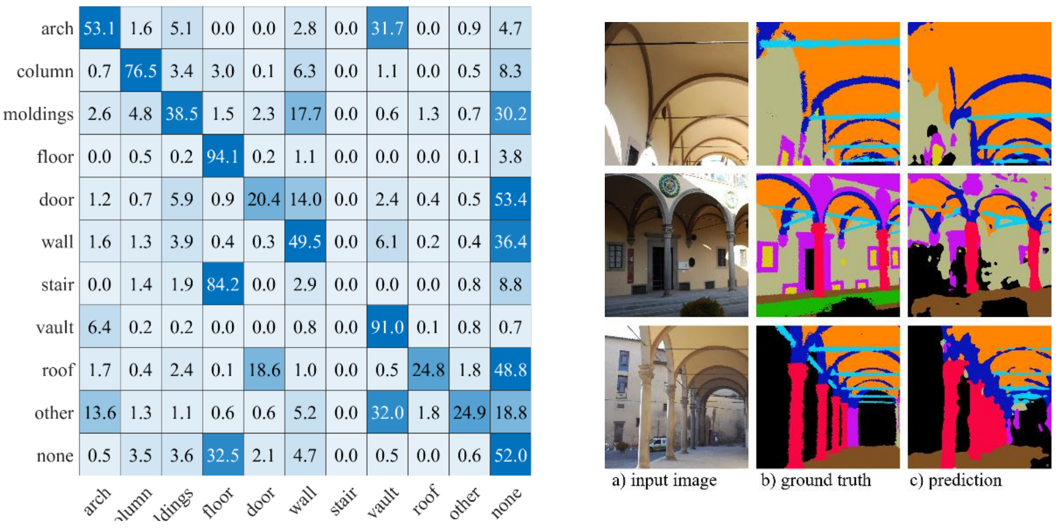
C.1 - (2_OSA, 3_SS, 4_CG, 5_CB) Training Set (1_SC) Test Set

Table 5.25 – Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
					internal test set						
accuracy	0,90	0,89	0,88	0,95	0,95	0,88	0,92	0,90	0,90	0,86	0,83
IoU	0,63	0,73	0,74	0,86	0,85	0,83	0,78	0,81	0,72	0,58	0,80
F$_1$ score	0,76	0,75	0,77	0,63	0,82	0,70	0,54	0,78	0,76	0,70	0,56
					external test set						
accuracy	**0,53**	**0,77**	**0,39**	**0,94**	**0,20**	**0,49**	**0,00**	**0,91**	**0,25**	**0,25**	**0,52**
IoU	**0,39**	**0,49**	**0,32**	**0,49**	**0,14**	**0,39**	**0,00**	**0,62**	**0,19**	**0,20**	**0,35**
F$_1$ score	**0,61**	**0,54**	**0,34**	**0,41**	**0,15**	**0,37**	**0,03**	**0,57**	**0,26**	**0,45**	**0,27**

Table 5.26 – Dataset metrics using DeepLabv3+.

	Global Accuracy	mean IoU	mean F$_1$ score
internal test set	0,88	0,75	0,70
external test set (1_SC)	**0,56**	**0,32**	**0,39**



	arch	lumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	53.1	1.6	5.1	0.0	0.0	2.8	0.0	31.7	0.0	0.9	4.7
column	0.7	76.5	3.4	3.0	0.1	6.3	0.0	1.1	0.0	0.5	8.3
moldings	2.6	4.8	38.5	1.5	2.3	17.7	0.0	0.6	1.3	0.7	30.2
floor	0.0	0.5	0.2	94.1	0.2	1.1	0.0	0.0	0.0	0.1	3.8
door	1.2	0.7	5.9	0.9	20.4	14.0	0.0	2.4	0.4	0.5	53.4
wall	1.6	1.3	3.9	0.4	0.3	49.5	0.0	6.1	0.2	0.4	36.4
stair	0.0	1.4	1.9	84.2	0.0	2.9	0.0	0.0	0.0	0.8	8.8
vault	6.4	0.2	0.2	0.0	0.0	0.8	0.0	91.0	0.1	0.8	0.7
roof	1.7	0.4	2.4	0.1	18.6	1.0	0.0	0.5	24.8	1.8	48.8
other	13.6	1.3	1.1	0.6	0.6	5.2	0.0	32.0	1.8	24.9	18.8
none	0.5	3.5	3.6	32.5	2.1	4.7	0.0	0.5	0.0	0.6	52.0

Figure 5.27 – (1_SC) Confusion matrix and image predictions with DeepLabv3+.

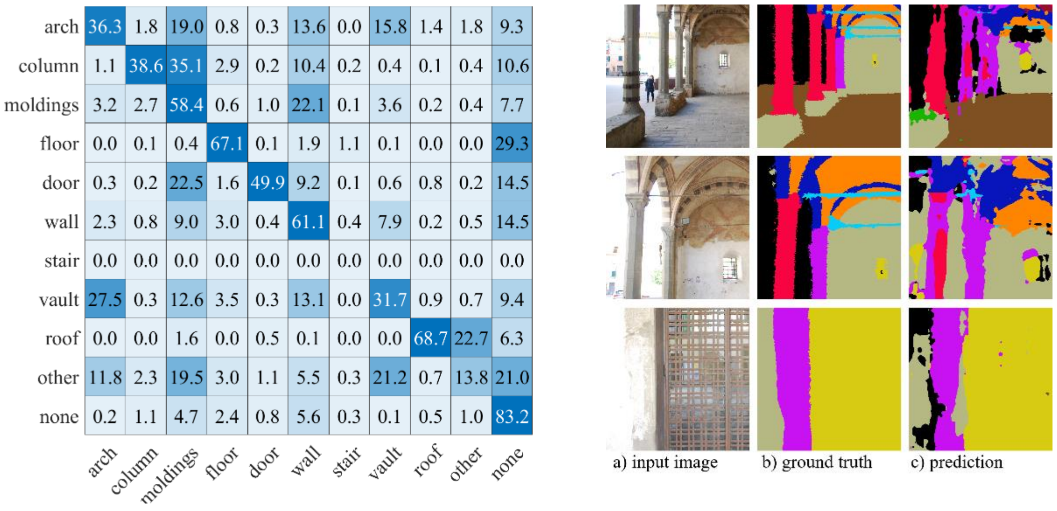
C.2 - (1_SC, 3_SS, 4_CG, 5_CB) Training Set (2_OSA) Test Set

Table 5.27 – Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
					internal test set						
accuracy	0,91	0,90	0,87	0,93	0,94	0,86	0,92	0,90	0,90	0,88	0,86
IoU	0,65	0,74	0,74	0,86	0,67	0,80	0,76	0,82	0,73	0,61	0,83
F_1 score	0,81	0,78	0,76	0,65	0,73	0,70	0,73	0,79	0,80	0,73	0,58
					external test set (2_OSA)						
accuracy	**0,36**	**0,39**	**0,58**	**0,67**	**0,50**	**0,61**	**-**	**0,32**	**0,69**	**0,14**	**0,83**
IoU	**0,20**	**0,34**	**0,25**	**0,60**	**0,47**	**0,50**	**-**	**0,24**	**0,34**	**0,10**	**0,47**
F_1 score	**0,40**	**0,35**	**0,31**	**0,43**	**0,32**	**0,33**	**-**	**0,24**	**0,47**	**0,28**	**0,28**

Table 5.28 – Dataset metrics using DeepLabv3+.

	Global Accuracy	mean IoU	mean F_1 score
internal test set	0,88	0,74	0,70
external test set (2_OSA)	**0,58**	**0,31**	**0,31**



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	36.3	1.8	19.0	0.8	0.3	13.6	0.0	15.8	1.4	1.8	9.3
column	1.1	38.6	35.1	2.9	0.2	10.4	0.2	0.4	0.1	0.4	10.6
moldings	3.2	2.7	58.4	0.6	1.0	22.1	0.1	3.6	0.2	0.4	7.7
floor	0.0	0.1	0.4	67.1	0.1	1.9	1.1	0.1	0.0	0.0	29.3
door	0.3	0.2	22.5	1.6	49.9	9.2	0.1	0.6	0.8	0.2	14.5
wall	2.3	0.8	9.0	3.0	0.4	61.1	0.4	7.9	0.2	0.5	14.5
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	27.5	0.3	12.6	3.5	0.3	13.1	0.0	31.7	0.9	0.7	9.4
roof	0.0	0.0	1.6	0.0	0.5	0.1	0.0	0.0	68.7	22.7	6.3
other	11.8	2.3	19.5	3.0	1.1	5.5	0.3	21.2	0.7	13.8	21.0
none	0.2	1.1	4.7	2.4	0.8	5.6	0.3	0.1	0.5	1.0	83.2

Figure 5.28 – (2_OSA) Confusion matrix and image predictions with DeepLabv3+.

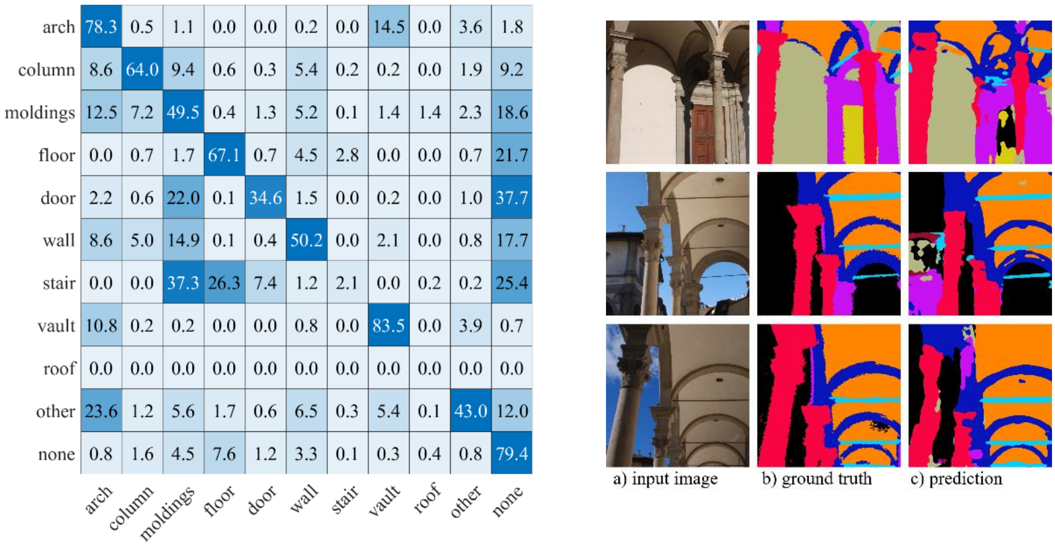
C.3 - (1_SC, 2_OSA, 4_CG, 5_CB) Training Set (3_SS) Test Set

Table 5.29 – Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
						internal test set					
accuracy	0,92	0,91	0,87	0,92	0,96	0,86	0,92	0,88	0,89	0,89	0,89
IoU	0,63	0,71	0,71	0,87	0,79	0,82	0,76	0,82	0,77	0,59	0,83
F_1 score	0,76	0,75	0,77	0,68	0,78	0,70	0,74	0,76	0,82	0,73	0,61
						external test set (3_SS)					
accuracy	**0,78**	**0,64**	**0,49**	**0,67**	**0,35**	**0,50**	**0,02**	**0,83**	**-**	**0,43**	**0,79**
IoU	**0,30**	**0,44**	**0,41**	**0,57**	**0,25**	**0,42**	**0,00**	**0,65**	**-**	**0,20**	**0,54**
F_1 score	**0,44**	**0,40**	**0,39**	**0,31**	**0,23**	**0,34**	**0,03**	**0,59**	**-**	**0,37**	**0,35**

Table 5.30 – Dataset metrics using DeepLabv3+.

	Global Accuracy	mean IoU	mean F_1 score
internal test set	0,88	0,75	0,71
external test set (3_SS)	**0,62**	**0,34**	**0,37**



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	78.3	0.5	1.1	0.0	0.0	0.2	0.0	14.5	0.0	3.6	1.8
column	8.6	64.0	9.4	0.6	0.3	5.4	0.2	0.2	0.0	1.9	9.2
moldings	12.5	7.2	49.5	0.4	1.3	5.2	0.1	1.4	1.4	2.3	18.6
floor	0.0	0.7	1.7	67.1	0.7	4.5	2.8	0.0	0.0	0.7	21.7
door	2.2	0.6	22.0	0.1	34.6	1.5	0.0	0.2	0.0	1.0	37.7
wall	8.6	5.0	14.9	0.1	0.4	50.2	0.0	2.1	0.0	0.8	17.7
stair	0.0	0.0	37.3	26.3	7.4	1.2	2.1	0.0	0.2	0.2	25.4
vault	10.8	0.2	0.2	0.0	0.0	0.8	0.0	83.5	0.0	3.9	0.7
roof	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
other	23.6	1.2	5.6	1.7	0.6	6.5	0.3	5.4	0.1	43.0	12.0
none	0.8	1.6	4.5	7.6	1.2	3.3	0.1	0.3	0.4	0.8	79.4

Figure 5.29 – (3_SS) Confusion matrix and image predictions with DeepLabv3+.

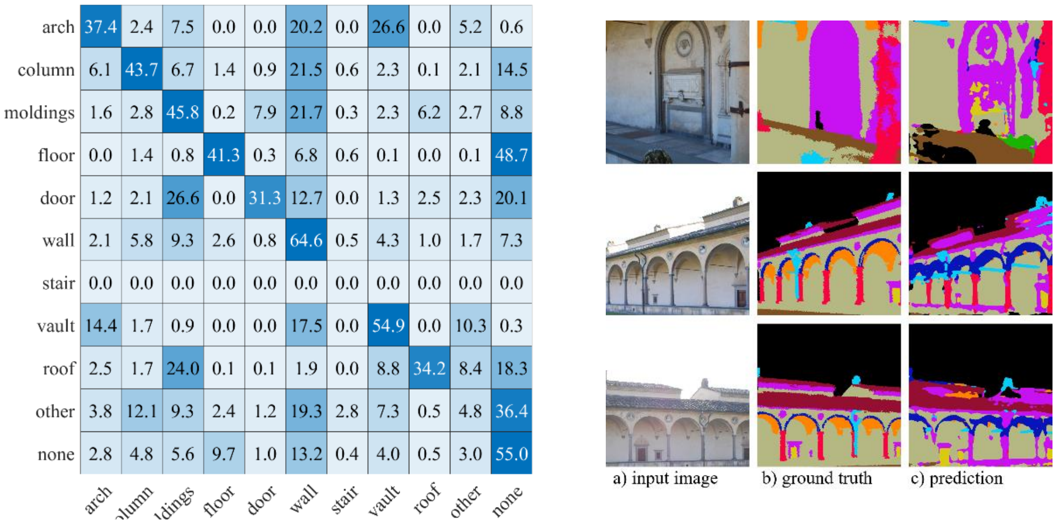
C.4 - (1_SC, 2_OSA, 3_SS, 5_CB) Training Set (4_CG) Test Set

Table 5.31 – Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
					internal test set						
accuracy	0,88	0,95	0,90	0,96	0,95	0,91	0,93	0,91	0,97	0,91	0,92
IoU	0,72	0,83	0,81	0,91	0,84	0,87	0,79	0,86	0,76	0,54	0,90
F$_1$ score	0,85	0,84	0,82	0,75	0,83	0,81	0,76	0,85	0,75	0,73	0,68
					external test set (4_CG)						
accuracy	**0,37**	**0,44**	**0,46**	**0,41**	**0,31**	**0,65**	**-**	**0,55**	**0,34**	**0,05**	**0,55**
IoU	**0,11**	**0,26**	**0,18**	**0,36**	**0,09**	**0,50**	**-**	**0,42**	**0,27**	**0,03**	**0,31**
F$_1$ score	**0,32**	**0,27**	**0,30**	**0,21**	**0,17**	**0,36**	**-**	**0,28**	**0,42**	**0,18**	**0,21**

Table 5.32 – Dataset metrics using DeepLabv3+.

	Global Accuracy	mean IoU	mean F$_1$ score
internal test set	0,92	0,80	0,78
external test set (4_CG)	**0,51**	**0,22**	**0,26**



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	37.4	2.4	7.5	0.0	0.0	20.2	0.0	26.6	0.0	5.2	0.6
column	6.1	43.7	6.7	1.4	0.9	21.5	0.6	2.3	0.1	2.1	14.5
moldings	1.6	2.8	45.8	0.2	7.9	21.7	0.3	2.3	6.2	2.7	8.8
floor	0.0	1.4	0.8	41.3	0.3	6.8	0.6	0.1	0.0	0.1	48.7
door	1.2	2.1	26.6	0.0	31.3	12.7	0.0	1.3	2.5	2.3	20.1
wall	2.1	5.8	9.3	2.6	0.8	64.6	0.5	4.3	1.0	1.7	7.3
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	14.4	1.7	0.9	0.0	0.0	17.5	0.0	54.9	0.0	10.3	0.3
roof	2.5	1.7	24.0	0.1	0.1	1.9	0.0	8.8	34.2	8.4	18.3
other	3.8	12.1	9.3	2.4	1.2	19.3	2.8	7.3	0.5	4.8	36.4
none	2.8	4.8	5.6	9.7	1.0	13.2	0.4	4.0	0.5	3.0	55.0

a) input image b) ground truth c) prediction

Figure 5.30 – (4_CG) Confusion matrix and image predictions with DeepLabv3+.

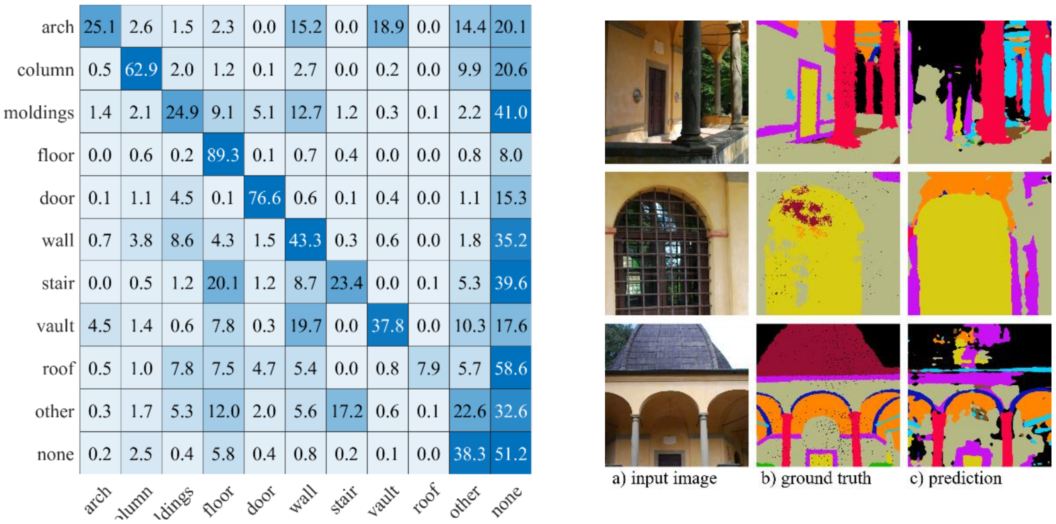
C.5 - (1_SC, 2_OSA, 3_SS, 4_CG) Training Set (5_CB) Test Set

Table 5.33 – Class metrics using DeepLabv3+.

	arch	colu.	moul.	floo.	wind.	wall	stai.	vaul.	roof	othe.	back.
					internal test set						
accuracy	0,89	0,92	0,90	0,95	0,94	0,89	0,94	0,90	0,89	0,89	0,88
IoU	0,69	0,76	0,78	0,89	0,82	0,84	0,75	0,84	0,77	0,60	0,85
F_1 score	0,84	0,79	0,80	0,69	0,81	0,76	0,69	0,82	0,85	0,74	0,64
					external test set (5_CB)						
accuracy	**0,25**	**0,63**	**0,25**	**0,89**	**0,77**	**0,43**	**0,23**	**0,38**	**0,08**	**0,23**	**0,51**
IoU	**0,17**	**0,46**	**0,12**	**0,51**	**0,58**	**0,38**	**0,17**	**0,36**	**0,08**	**0,02**	**0,31**
F_1 score	**0,36**	**0,46**	**0,18**	**0,28**	**0,36**	**0,25**	**0,10**	**0,33**	**0,18**	**0,13**	**0,29**

Table 5.34 – Dataset metrics using DeepLabv3+.

	Global Accuracy	mean IoU	mean F_1 score
internal test set	0,90	0,77	0,75
external test set (5_CB)	**0,46**	**0,28**	**0,27**



	arch	olumn	ldings	floor	door	wall	stair	vault	roof	other	none
arch	25.1	2.6	1.5	2.3	0.0	15.2	0.0	18.9	0.0	14.4	20.1
column	0.5	62.9	2.0	1.2	0.1	2.7	0.0	0.2	0.0	9.9	20.6
moldings	1.4	2.1	24.9	9.1	5.1	12.7	1.2	0.3	0.1	2.2	41.0
floor	0.0	0.6	0.2	89.3	0.1	0.7	0.4	0.0	0.0	0.8	8.0
door	0.1	1.1	4.5	0.1	76.6	0.6	0.1	0.4	0.0	1.1	15.3
wall	0.7	3.8	8.6	4.3	1.5	43.3	0.3	0.6	0.0	1.8	35.2
stair	0.0	0.5	1.2	20.1	1.2	8.7	23.4	0.0	0.1	5.3	39.6
vault	4.5	1.4	0.6	7.8	0.3	19.7	0.0	37.8	0.0	10.3	17.6
roof	0.5	1.0	7.8	7.5	4.7	5.4	0.0	0.8	7.9	5.7	58.6
other	0.3	1.7	5.3	12.0	2.0	5.6	17.2	0.6	0.1	22.6	32.6
none	0.2	2.5	0.4	5.8	0.4	0.8	0.2	0.1	0.0	38.3	51.2

Figure 5.31 – (5_CB) Confusion matrix and image predictions with DeepLabv3+.

5.5 Labelling projection on point cloud

As already mentioned in paragraph §3.3 the 3D point cloud segmentation pipeline is composed by two main steps: the segmentation of the related photogrammetric images, and the projection of the extracted 2D features on the 3D point cloud. The detail about the functioning of reprojection procedure has been already explained in paragraph (§3.3.5). In this paragraph the settings and the detail about the application of the procedure on the dataset buildings will be illustrated and discussed.

5.5.1 Settings and options

The implementation of the reprojection procedure was carried out using MATLAB coding, and it allows to set up and to control the reprojection process by means of a set of parameters and options. The choice of the settings is fundamental to achieve a good performance and to optimize the results obtained by the neural network on the images. However, the optimal set of parameters could depend on the single building, and in any cases, the quality and the accuracy on the initial image segmentation is essential to achieve a high final point cloud segmentation accuracy. In the following sections the parameters that control the reprojection process will be illustrated and explained.

Point Cloud Subsampling. In many cases, 3D dense point clouds are made up of millions of points, and the management of such huge data is often challenging and time consuming, and it requires a high memory consumption. These limitations could slow down and compromise the reprojection process, in particular during the test phases, in which several experiments were necessary, in order to improve the performance and to define the optimal range of parameters. For these reasons a subsampling factor has been introduced, and it allows to reduce the number of points and to speed up the procedure. As specified in paragraph §2.3.2 there are several methods to down-sample a point cloud. In the following tests and experiments a simple random sub-sampling has been used, and it is controlled by means of the percentage of points to reduce. Besides the improvement of the computing time, the subsampling factor helps to find out the optimal point cloud density or number of points to achieve the highest accuracy.

Image Reduction Factor. The maximum number of images that can be involved in the reprojection can vary and it depends on the number of images acquired and used during the photogrammetric pipeline. However, not necessarily a higher number of images involved in the labelling projection guarantee a higher performance or accuracy. Some views could be most representative than others and they should have more weight during the features transferring. At the same time, some images with inadequate angle views or occlusion problems could bias the label assignment. Several works have been proposed for multiview aggregation and they proposed various methods to merge 2D features from multiple view in a meaningful way, addressing most challenges, like the large number of images, the image scale, blur, exposure or obstructions. (Robert et al., 2022), (Waechter et al., 2014). The proposed reprojection procedure is based on a voting label selection (§3.3.5), and most of these problems are implicitly addressed by selecting the most popular label between all the images involved in the vote. However, using a large number of images is computationally expensive and often useless. A reduction factor has been introduced, and it allows to reduce the number of involved images, to speed up the procedure, especially in a test phase, and to assess the effect of the number of images on the performance and on the accuracy.

Pixel Enlarging Factor. The label assignment needs to take into account the point position with respect to the camera, hence the related occlusions and obstructions. The closest points to the camera should obstruct the points behind them, but due to the dimensionless nature of a point, a meaningful way to quantify the amount of obstruction needs to be verified. For this reason, the parameter *pixel_enlarging_factor* has been introduced, and it rules the amount of volume obstructed by a point. A line passing through the optical centre of the camera is associated with each pixel of the related image, and each 3D points on this line could be associated to that pixel. Specifically, the parameter rules the distance threshold from the line that determines the quantity of points to associate to the pixel. The factor should be expressed as an integer major than 1, and the higher its value, the higher the obstruction of the points behind the closest point during the label assignment. At the same time, the higher the value of the parameter, the lower the number of points labelled at the end of the procedure.

K_Thr. Once that the N points potentially linkable to the pixel are defined, its still remains to determine the N_1 points that must be labelled, and the N_2 points that are hidden by the closer ones. This check is done by computing the distance between the points and the camera. If the points are closer than a distance threshold, they are labelled and classified as obstructed otherwise. The threshold is computed by the sum of the minimum distance from the camera and the points, and the pixel dimension projected at that distance increasing by the factor *k_thr*. Hence, the larger *k_thr* is, the lower the probability that a point is hidden. By increasing *k_thr*, the percentage of classified pixels grows up, but simultaneously, the accuracy and the precision could decrease.

Min N Vote. A point should be represented at least by two pixels in two different images to be considered for the label assignment. Nevertheless, in most of the cases a point is represented by several pixels in more than two images, and the voting procedure assigns the most popular label at the selected points. However, when the accuracy of the image segmentation network is not high, or the probability of a certain class is low, several labels could be assigned to a pixel ideally representative of just one point, without a remarkable prevalence of a label. For this reason, a minimum number of votes can be set, and it is expressed as a minimum percentage of votes that agrees on a certain label. In this way the procedure become more stable and reliable, ensuring the labelling only when a high probability is detected.

Evaluation Metrics. As for the semantic segmentation of the images, it is fundamental to evaluate rigorously the performance of the reprojection, and to choose the appropriate evaluation metrics. To be able to compare the performance between images and point cloud during the reprojection process the same evaluation metrics have been used, the *Global Accuracy*, the *mean Intersection over Union* and the *Confusion Matrix*.

5.5.2 Tests

Due to the large number of parameters and settings, several experiments have been performed to find the optimal range of each single parameter, and the optimal combinations of settings. Three typologies of tests were carried out and are explained in the following sections.

Test R.GT. The first series of experiments that are going to be reported consist in a systematic research of the optimal tuning of the parameters, testing various combinations and assessing the effect of each single parameter on the accuracy and the performance. The investigation was performed just on one building of the dataset, specifically *(1_SC) Spedale del Ceppo*. To avoid biases caused by an incorrect image segmentation, the ground truth has been used for the reprojection. It was thus possible to assess the actual performance of the procedure, without ambiguous interpretation of the results. Such analysis allowed the correct choice of the parameters to be used in the following tests.

Test R.A. The second set of tests consists in the projection of the predicted label on the images by the classifier trained for the Test A (§5.5.1). The predicted labels are obtained using the model trained with Deeplabv3+ architecture, according to the illustrated results, achieved the best performance. For each of the five buildings in the dataset, the photogrammetric images were fed into the trained networks linked to the building, and the output labels were used for the projection process. This tests were performed with the optimal parameter combination turned out from the previous experiments. The tests were helpful to validate the procedure with predicted labels which, in contrast to the ground truth, could have lower accuracy and a higher disaccording (offset ?) between pixel labels. This series of tests allows to assess the performance decreasing of the neural network results when transferring the image features to the 3D point cloud.

Test R.C. The final set of tests consists in the projection of the predicted label on the images by the classifier trained for the Test C (§5.5.3). The results represent the final outcome of the entire 3D semantic segmentation pipeline of the point cloud, in the case of unseen scenario, and the accuracy of such tests can be considered as the current performance of the segmentation procedure. Due to the low precision obtained on image segmentation on Test C, a high final performance is not expected, since it could not overcome the image segmentation performance. Such as Test C for image segmentation, a cross validation between the five buildings of the dataset has been performed.

5.6 Labelling projection test results

In this paragraph the results of the reprojection on the various tests will be reported, and they will be widely discussed in further paragraph (§5.7). The outcome of the procedure is the input 3D point cloud with each points associated to a label according with the categories of the input images. The points that are not labelled at the end of the procedure will be marked as "unclassified". Two types of methods are used to evaluate and compare such procedure. At first, the GA and the mIoU considering all the points involved in the reprojection. Secondly, the GA and the mIoU considering only the classified points, discarding the "unclassified" points when computing the metrics. Therefore, it allows to assess the proper functioning of the procedure, and to distinguish misclassification from unclassification. For the tests R.GT, a series of table results will be reported, and the two evaluation methods will be compared. For tests R.A and R.C will be reported on one hand, the best results obtained on all points, and on the other hand, the best results considering only the classified points. The related image performance obtained by the neural network will be shown, together with the confusion matrices for the "only classified" points case, that allows a deep visual assessment of the reprojection on the various classes.

5.6.1 Test R.GT

Point Cloud Subsampling. As shown in Table 5.33 five values of subsampling have been tested. Decreasing the percentage of points, the percentage of unlabelled point during the reprojection decreases, and, at the same time, the GA and the mIoU improve considering all the points, and they have a little decline considering only the classifying points.

Table 5.35 – Reprojection results at different point cloud subsampling factor.

	Point Subsampling	**100%**	**75%**	**50%**	**25%**	**10%**
	Image Reduction	100%	100%	100%	100%	100%
	pxl_enlarging_factor	1	1	1	1	1
	k_thr	1	1	1	1	1
	n_votes	1	1	1	1	1
	RESULTS					
	Unclassified points (%)	10,3	6,3	2,6	0,3	0,0
ALL POINTS	**Global Accuracy (%)**	85,1	87,7	91,4	93,3	92,5
	mean IoU (%)	72,7	74,1	78	79,8	78,2
ONLY CLASSIFIED	**Global Accuracy (%)**	94,9	94,4	93,9	93,6	92,5
	mean IoU (%)	82,3	81,2	80,7	80,1	78,2

Image Reduction Factor. The number of images used during the reprojection has a remarkable impact on the performance. As shown in Table 5.36 reducing the number of images involved, the unlabelled points increase and both the GA and mIoU decrease. Considering only the classified points the performance remains almost the same, revealing that with less images most of the points are unlabelled rather that misclassified.

Table 5.36 - Reprojection results at different image subsampling factor.

	Point Subsampling	100%	100%	100%	100%	100%
	Image Reduction	**75%**	**50%**	**25%**	**10%**	**5%**
	pxl_enlarging_factor	1	1	1	1	1
	k_thr	1	1	1	1	1
	n_votes	1	1	1	1	1
	RESULTS					
	Unclassified points (%)	12,3	15,6	23,1	39,1	53,5
ALL POINTS	**Global Accuracy (%)**	82,2	80,3	73,1	57,8	44,1
	mean IoU (%)	70,3	68,6	62,5	49	38,8
ONLY CLASSIFIED	**Global Accuracy (%)**	94,9	95,1	95,2	94,9	94,5
	mean IoU (%)	82,1	82,4	82,6	82,2	82

Pixel Enlarging Factor. Five values of the parameters have been tested. As expected, the percentage of labelled points reduces with the increase of the factor, but simultaneously, enlarging the occlusions has a positive effect on the performance considering only the classified points. Depending on the required outcome, the factor could be set favouring the accuracy on one hand, or the number of classified points on the other hand (Table 5.37).

Table 5.37 - Reprojection results at different pixel enlarging factor.

Point Subsampling		25%	25%	25%	25%	25%
Image Reduction		50%	50%	50%	50%	50%
pxl_enlarging_factor		**1**	**2**	**3**	**5**	**7**
k_thr		1	1	1	1	1
n_votes		1	1	1	1	1
RESULTS						
Unclassified points (%)		5,3	14,7	33,9	58,7	24,6
ALL POINTS	**Global Accuracy (%)**	85,2	80,7	63,7	40,1	31,5
	mean IoU (%)	73,3	68,9	55,6	35,6	27,6
ONLY CLASSIFIED	**Global Accuracy (%)**	92,9	94,7	96,4	97,2	97,1
	mean IoU (%)	77,9	81,8	84,6	86,3	86,5

K_thr factor. As shown in Table 5.38 the k_thr factor has remarkable benefits both for the percentage of classified points and for the accuracy with an improvement considering all the points involved, and only the classified points.

Table 5.38 - Reprojection results at different k_thr factor.

Point Subsampling		25%	25%	25%	25%	25%
Image Reduction		50%	50%	50%	50%	50%
pxl_enlarging_factor		1	1	1	1	1
k_thr		**1**	**2**	**3**	**5**	**10**
n_votes		1	1	1	1	1
RESULTS						
Unclassified points (%)		5,3	0,7	0,5	0,3	0,1
ALL POINTS	**Global Accuracy (%)**	85,2	92,8	93,2	93,6	93,9
	mean IoU (%)	73,3	79,1	79,5	80	80,2
ONLY CLASSIFIED	**Global Accuracy (%)**	92,9	93,5	93,7	93,9	94,1
	mean IoU (%)	77,9	79,8	80	80,3	80,3

N min Votes. The minimum number of votes chosen for the reprojection has a remarkable impact on the percentage of labelled points and on the accuracy. Its choice is strictly related to the number of images using during the process, and it is not possible to choose a correct value beforehand. As expected, increasing the votes reduces the number of labelled points, and the overall accuracy considering only the classified increases. Its effect will be better valued in the next tests, in which the pixel predictions could have remarkable discrepancy.

Table 5.39 - Reprojection results at different number of voting factor.

Point Subsampling		100%	100%	100%	100%	100%
Image Reduction		100%	100%	100%	100%	100%
pxl_enlarging_factor		1	1	1	1	1
k_thr		1	1	1	1	1
n_votes		**1**	**2**	**3**	**7**	**10**
RESULTS						
Unclassified points (%)		10,3	17,1	23	40,8	49,9
ALL POINTS	**Global Accuracy (%)**	85,1	79,4	74,2	57,7	48,9
	mean IoU (%)	72,7	68,4	64,3	50,2	42,5
ONLY CLASSIFIED	**Global Accuracy (%)**	94,9	95,5	96,4	97,5	97,8
	mean IoU (%)	82,3	83,9	85	86,8	87,4

Despite the use of the ground truth images for the reprojection, the accuracy of the final point cloud has never achieved a GA of 100% and a MIoU of 100% as expected. Rather than a reprojection malfunctioning, it can be explained as the result of an imperfect ground truth, since it has been generated automatically from a labelled point cloud. The transition from point-to-image, and then from image-to-point, has inevitably led to a loss of accuracy. Depending on the choice of the parameters and their combinations, it is possible to favour the number of classified points with a loss of accuracy, or to favour accuracy with a decrease in classified points.

5.6.2 Test R.A

R.A.1 (1_SC) Spedale del Ceppo

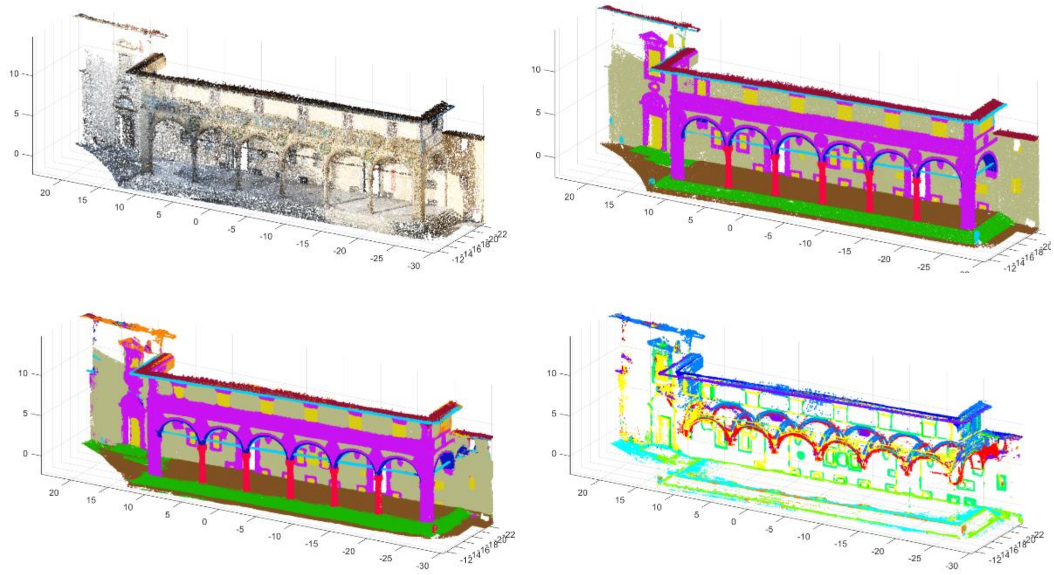
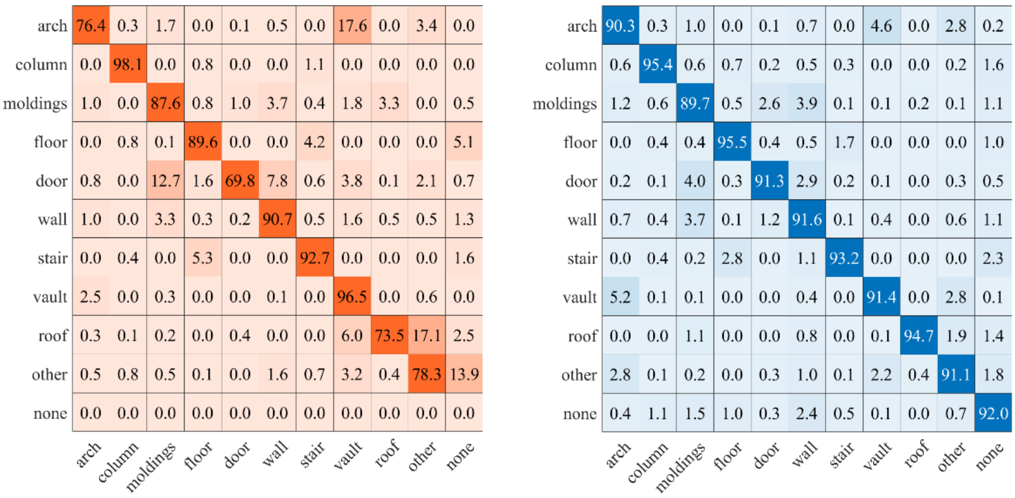


Figure 5.32 – (1_SC) RGB input point cloud (upper left), ground truth point cloud (upper right), predicted point cloud (down left), and misclassified points (down right).

Table 5.40 – (1_SC) Reprojection results of Test R.A.1.

RESULTS			
	All Points	**Only Classified**	**Images**
Unlabelled Points (%)	2,3	19,8	-
Global Accuracy (%)	87,9	91,4	92,1
Mean IoU (%)	68,4	76,3	81,7



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	76.4	0.3	1.7	0.0	0.1	0.5	0.0	17.6	0.0	3.4	0.0
column	0.0	98.1	0.0	0.8	0.0	0.0	1.1	0.0	0.0	0.0	0.0
moldings	1.0	0.0	87.6	0.8	1.0	3.7	0.4	1.8	3.3	0.0	0.5
floor	0.0	0.8	0.1	89.6	0.0	0.0	4.2	0.0	0.0	0.0	5.1
door	0.8	0.0	12.7	1.6	69.8	7.8	0.6	3.8	0.1	2.1	0.7
wall	1.0	0.0	3.3	0.3	0.2	90.7	0.5	1.6	0.5	0.5	1.3
stair	0.0	0.4	0.0	5.3	0.0	0.0	92.7	0.0	0.0	0.0	1.6
vault	2.5	0.0	0.3	0.0	0.0	0.1	0.0	96.5	0.0	0.6	0.0
roof	0.3	0.1	0.2	0.0	0.4	0.0	0.0	6.0	73.5	17.1	2.5
other	0.5	0.8	0.5	0.1	0.0	1.6	0.7	3.2	0.4	78.3	13.9
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	90.3	0.3	1.0	0.0	0.1	0.7	0.0	4.6	0.0	2.8	0.2
column	0.6	95.4	0.6	0.7	0.2	0.5	0.3	0.0	0.0	0.2	1.6
moldings	1.2	0.6	89.7	0.5	2.6	3.9	0.1	0.1	0.2	0.1	1.1
floor	0.0	0.4	0.4	95.5	0.4	0.5	1.7	0.0	0.0	0.0	1.0
door	0.2	0.1	4.0	0.3	91.3	2.9	0.2	0.1	0.0	0.3	0.5
wall	0.7	0.4	3.7	0.1	1.2	91.6	0.1	0.4	0.0	0.6	1.1
stair	0.0	0.4	0.2	2.8	0.0	1.1	93.2	0.0	0.0	0.0	2.3
vault	5.2	0.1	0.1	0.0	0.0	0.4	0.0	91.4	0.0	2.8	0.1
roof	0.0	0.0	1.1	0.0	0.0	0.8	0.0	0.1	94.7	1.9	1.4
other	2.8	0.1	0.2	0.0	0.3	1.0	0.1	2.2	0.4	91.1	1.8
none	0.4	1.1	1.5	1.0	0.3	2.4	0.5	0.1	0.0	0.7	92.0

Figure 5.33 – Test R.A.1 Point cloud confusion matrix (orange), related confusion matrix on neural network image prediction (blue).

R.A.2 (2_OSA) Ospedale Sant'Antonio

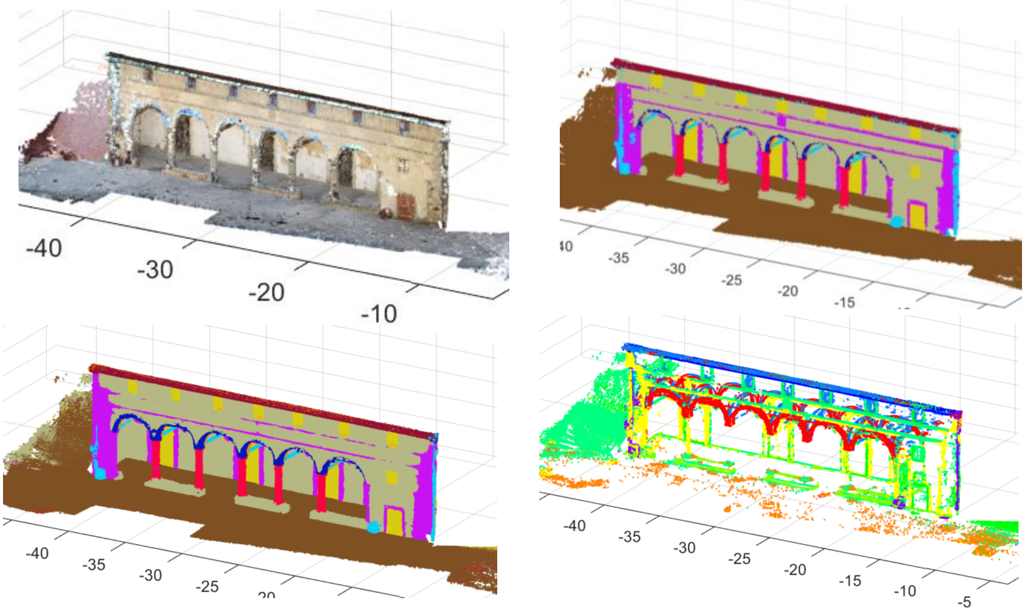
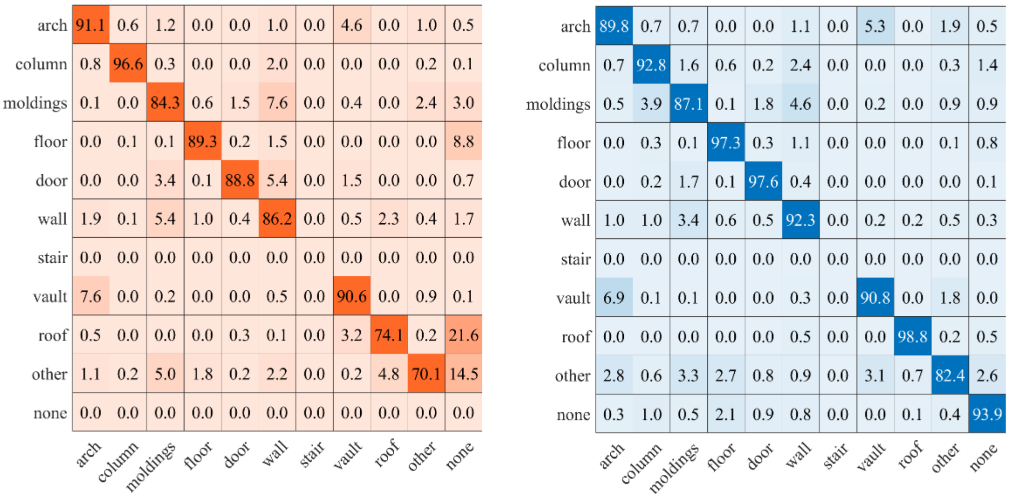


Figure 5.34 – (2_OSA) RGB input point cloud (upper left), ground truth point cloud (upper right), predicted point cloud (down left), and misclassified points (down right).

Table 5.41 – (2_OSA) Reprojection results of Test R.A.2.

RESULTS			
	All Points	**Only Classified**	**Images**
Unlabelled Points (%)	1,6	22,4	-
Global Accuracy (%)	87,8	92,4	93,1
Mean IoU (%)	68,8	75,3	81,9



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	91.1	0.6	1.2	0.0	0.0	1.0	0.0	4.6	0.0	1.0	0.5
column	0.8	96.6	0.3	0.0	0.0	2.0	0.0	0.0	0.0	0.2	0.1
moldings	0.1	0.0	84.3	0.6	1.5	7.6	0.0	0.4	0.0	2.4	3.0
floor	0.0	0.1	0.1	89.3	0.2	1.5	0.0	0.0	0.0	0.0	8.8
door	0.0	0.0	3.4	0.1	88.8	5.4	0.0	1.5	0.0	0.0	0.7
wall	1.9	0.1	5.4	1.0	0.4	86.2	0.0	0.5	2.3	0.4	1.7
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	7.6	0.0	0.2	0.0	0.0	0.5	0.0	90.6	0.0	0.9	0.1
roof	0.5	0.0	0.0	0.0	0.3	0.1	0.0	3.2	74.1	0.2	21.6
other	1.1	0.2	5.0	1.8	0.2	2.2	0.0	0.2	4.8	70.1	14.5
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	89.8	0.7	0.7	0.0	0.0	1.1	0.0	5.3	0.0	1.9	0.5
column	0.7	92.8	1.6	0.6	0.2	2.4	0.0	0.0	0.0	0.3	1.4
moldings	0.5	3.9	87.1	0.1	1.8	4.6	0.0	0.2	0.0	0.9	0.9
floor	0.0	0.3	0.1	97.3	0.3	1.1	0.0	0.0	0.0	0.1	0.8
door	0.0	0.2	1.7	0.1	97.6	0.4	0.0	0.0	0.0	0.0	0.1
wall	1.0	1.0	3.4	0.6	0.5	92.3	0.0	0.2	0.2	0.5	0.3
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	6.9	0.1	0.1	0.0	0.0	0.3	0.0	90.8	0.0	1.8	0.0
roof	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	98.8	0.2	0.5
other	2.8	0.6	3.3	2.7	0.8	0.9	0.0	3.1	0.7	82.4	2.6
none	0.3	1.0	0.5	2.1	0.9	0.8	0.0	0.0	0.1	0.4	93.9

Figure 5.35 – Test R.A.2 Point cloud confusion matrix (orange), related confusion matrix on neural network image prediction (blue).

R.A.3 (3_SS) Basilica della Santissima Annunziata

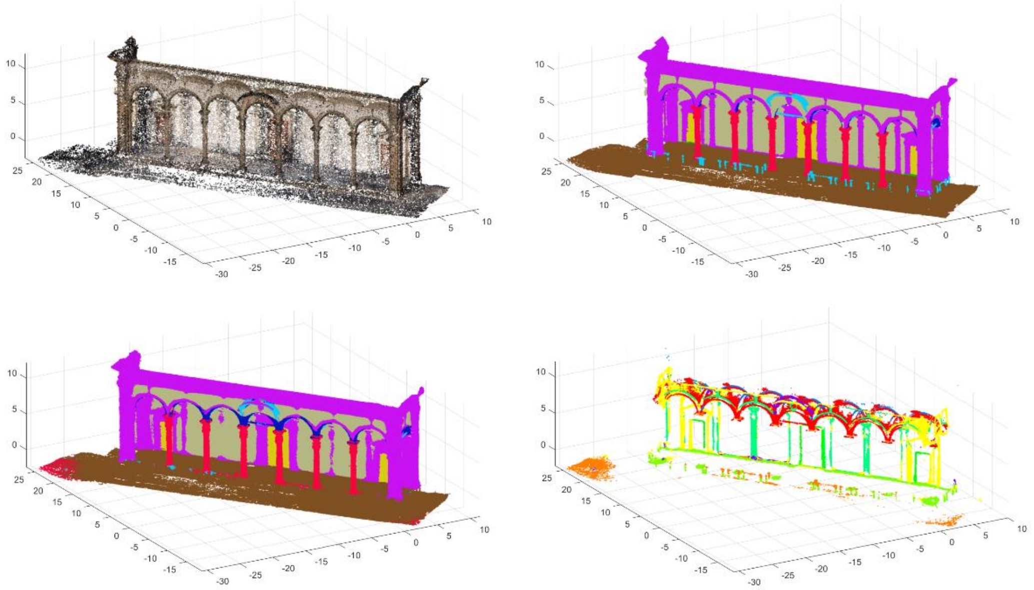
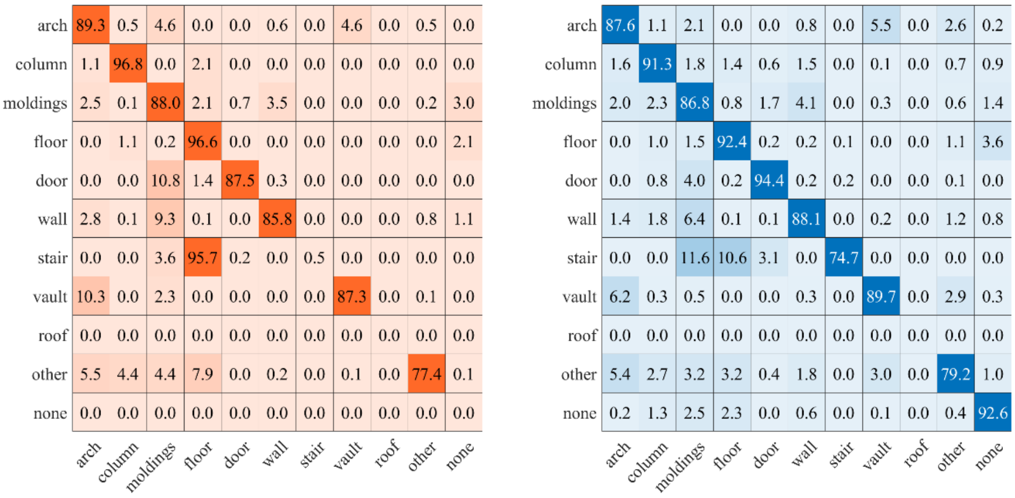


Figure 5.36 – (3_SS) RGB input point cloud (upper left), ground truth point cloud (upper right), predicted point cloud (down left), and misclassified points (down right).

Table 5.42 – (3_SS) Reprojection results of Test R.A.3.

RESULTS			
	All Points	**Only Classified**	**Images**
Unlabelled Points (%)	0,1	6,1	-
Global Accuracy (%)	89,9	92,2	89,7
Mean IoU (%)	64,2	68,8	71,6



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	89.3	0.5	4.6	0.0	0.0	0.6	0.0	4.6	0.0	0.5	0.0
column	1.1	96.8	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
moldings	2.5	0.1	88.0	2.1	0.7	3.5	0.0	0.0	0.0	0.2	3.0
floor	0.0	1.1	0.2	96.6	0.0	0.0	0.0	0.0	0.0	0.0	2.1
door	0.0	0.0	10.8	1.4	87.5	0.3	0.0	0.0	0.0	0.0	0.0
wall	2.8	0.1	9.3	0.1	0.0	85.8	0.0	0.0	0.0	0.8	1.1
stair	0.0	0.0	3.6	95.7	0.2	0.0	0.5	0.0	0.0	0.0	0.0
vault	10.3	0.0	2.3	0.0	0.0	0.0	0.0	87.3	0.0	0.1	0.0
roof	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
other	5.5	4.4	4.4	7.9	0.0	0.2	0.0	0.1	0.0	77.4	0.1
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	87.6	1.1	2.1	0.0	0.0	0.8	0.0	5.5	0.0	2.6	0.2
column	1.6	91.3	1.8	1.4	0.6	1.5	0.0	0.1	0.0	0.7	0.9
moldings	2.0	2.3	86.8	0.8	1.7	4.1	0.0	0.3	0.0	0.6	1.4
floor	0.0	1.0	1.5	92.4	0.2	0.2	0.1	0.0	0.0	1.1	3.6
door	0.0	0.8	4.0	0.2	94.4	0.2	0.2	0.0	0.0	0.1	0.0
wall	1.4	1.8	6.4	0.1	0.1	88.1	0.0	0.2	0.0	1.2	0.8
stair	0.0	0.0	11.6	10.6	3.1	0.0	74.7	0.0	0.0	0.0	0.0
vault	6.2	0.3	0.5	0.0	0.0	0.3	0.0	89.7	0.0	2.9	0.3
roof	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
other	5.4	2.7	3.2	3.2	0.4	1.8	0.0	3.0	0.0	79.2	1.0
none	0.2	1.3	2.5	2.3	0.0	0.6	0.0	0.1	0.0	0.4	92.6

Figure 5.37 – Test R.A.3 Point cloud confusion matrix (orange), related confusion matrix on neural network image prediction (blue).

R.A.4 (4_CG) Certosa del Galluzzo

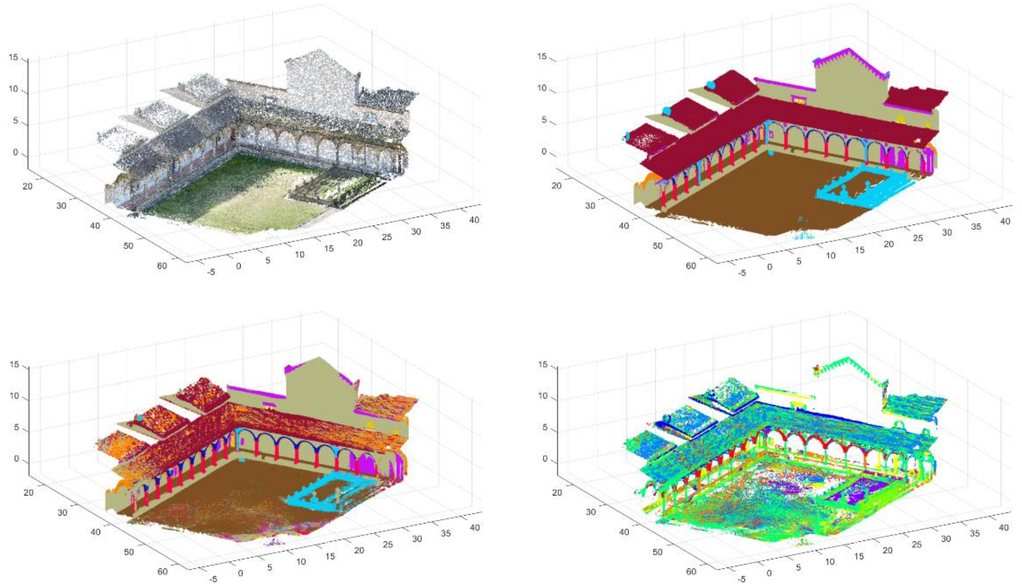
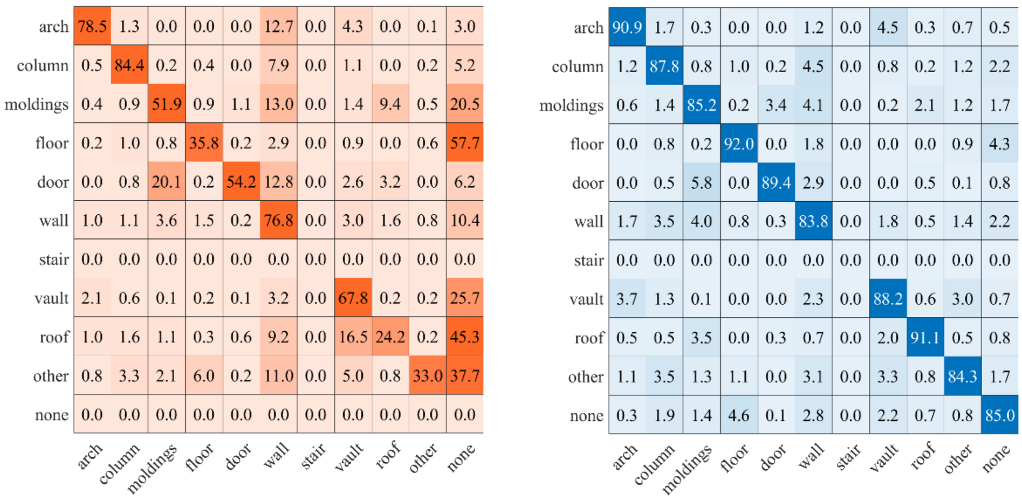


Figure 5.38 – (4_CG) RGB input point cloud (upper left), ground truth point cloud (upper right), predicted point cloud (down left), and misclassified points (down right).

Table 5.43 – (4_CG) Reprojection results of Test R.A.4.

RESULTS			
	All Points	**Only Classified**	**Images**
Unlabelled Points (%)	9,3	35,3	-
Global Accuracy (%)	67,3	83,2	86,8
Mean IoU (%)	35,4	51,4	68,7



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	78.5	1.3	0.0	0.0	0.0	12.7	0.0	4.3	0.0	0.1	3.0
column	0.5	84.4	0.2	0.4	0.0	7.9	0.0	1.1	0.0	0.2	5.2
moldings	0.4	0.9	51.9	0.9	1.1	13.0	0.0	1.4	9.4	0.5	20.5
floor	0.2	1.0	0.8	35.8	0.2	2.9	0.0	0.9	0.0	0.6	57.7
door	0.0	0.8	20.1	0.2	54.2	12.8	0.0	2.6	3.2	0.0	6.2
wall	1.0	1.1	3.6	1.5	0.2	76.8	0.0	3.0	1.6	0.8	10.4
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	2.1	0.6	0.1	0.2	0.1	3.2	0.0	67.8	0.2	0.2	25.7
roof	1.0	1.6	1.1	0.3	0.6	9.2	0.0	16.5	24.2	0.2	45.3
other	0.8	3.3	2.1	6.0	0.2	11.0	0.0	5.0	0.8	33.0	37.7
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	90.9	1.7	0.3	0.0	0.0	1.2	0.0	4.5	0.3	0.7	0.5
column	1.2	87.8	0.8	1.0	0.2	4.5	0.0	0.8	0.2	1.2	2.2
moldings	0.6	1.4	85.2	0.2	3.4	4.1	0.0	0.2	2.1	1.2	1.7
floor	0.0	0.8	0.2	92.0	0.0	1.8	0.0	0.0	0.0	0.9	4.3
door	0.0	0.5	5.8	0.0	89.4	2.9	0.0	0.0	0.5	0.1	0.8
wall	1.7	3.5	4.0	0.8	0.3	83.8	0.0	1.8	0.5	1.4	2.2
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	3.7	1.3	0.1	0.0	0.0	2.3	0.0	88.2	0.6	3.0	0.7
roof	0.5	0.5	3.5	0.0	0.3	0.7	0.0	2.0	91.1	0.5	0.8
other	1.1	3.5	1.3	1.1	0.0	3.1	0.0	3.3	0.8	84.3	1.7
none	0.3	1.9	1.4	4.6	0.1	2.8	0.0	2.2	0.7	0.8	85.0

Figure 5.39 – Test R.A.4 Point cloud confusion matrix (orange), related confusion matrix on neural network image prediction (blue).

R.A.5 (5_CB) Cappella Buontalenti

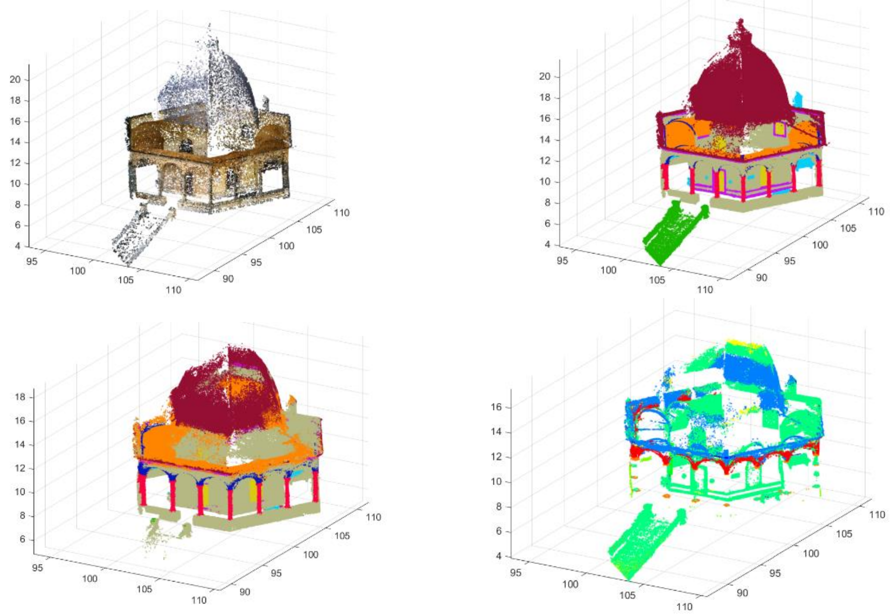
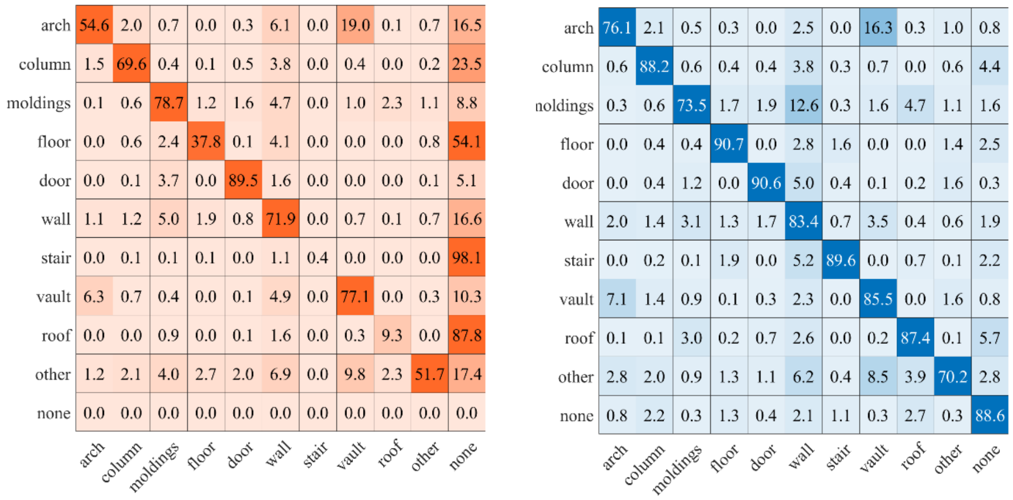


Figure 5.40 – (5_CB) RGB input point cloud (upper left), ground truth point cloud (upper right), predicted point cloud (down left), and misclassified points (down right).

Table 5.44 – (5_CB) Reprojection results of Test R.A.5.

RESULTS			
	All Points	**Only Classified**	**Images**
Unlabelled Points (%)	0,6	8,2	-
Global Accuracy (%)	73,9	79,1	86,1
Mean IoU (%)	38,3	44,3	67,5



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	54.6	2.0	0.7	0.0	0.3	6.1	0.0	19.0	0.1	0.7	16.5
column	1.5	69.6	0.4	0.1	0.5	3.8	0.0	0.4	0.0	0.2	23.5
moldings	0.1	0.6	78.7	1.2	1.6	4.7	0.0	1.0	2.3	1.1	8.8
floor	0.0	0.6	2.4	37.8	0.1	4.1	0.0	0.0	0.0	0.8	54.1
door	0.0	0.1	3.7	0.0	89.5	1.6	0.0	0.0	0.0	0.1	5.1
wall	1.1	1.2	5.0	1.9	0.8	71.9	0.0	0.7	0.1	0.7	16.6
stair	0.0	0.1	0.1	0.1	0.0	1.1	0.4	0.0	0.0	0.0	98.1
vault	6.3	0.7	0.4	0.0	0.1	4.9	0.0	77.1	0.0	0.3	10.3
roof	0.0	0.0	0.9	0.0	0.1	1.6	0.0	0.3	9.3	0.0	87.8
other	1.2	2.1	4.0	2.7	2.0	6.9	0.0	9.8	2.3	51.7	17.4
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	76.1	2.1	0.5	0.3	0.0	2.5	0.0	16.3	0.3	1.0	0.8
column	0.6	88.2	0.6	0.4	0.4	3.8	0.3	0.7	0.0	0.6	4.4
noldings	0.3	0.6	73.5	1.7	1.9	12.6	0.3	1.6	4.7	1.1	1.6
floor	0.0	0.4	0.4	90.7	0.0	2.8	1.6	0.0	0.0	1.4	2.5
door	0.0	0.4	1.2	0.0	90.6	5.0	0.4	0.1	0.2	1.6	0.3
wall	2.0	1.4	3.1	1.3	1.7	83.4	0.7	3.5	0.4	0.6	1.9
stair	0.0	0.2	0.1	1.9	0.0	5.2	89.6	0.0	0.7	0.1	2.2
vault	7.1	1.4	0.9	0.1	0.3	2.3	0.0	85.5	0.0	1.6	0.8
roof	0.1	0.1	3.0	0.2	0.7	2.6	0.0	0.2	87.4	0.1	5.7
other	2.8	2.0	0.9	1.3	1.1	6.2	0.4	8.5	3.9	70.2	2.8
none	0.8	2.2	0.3	1.3	0.4	2.1	1.1	0.3	2.7	0.3	88.6

Figure 5.41 – Test R.A.4 Point cloud confusion matrix (orange), related confusion matrix on neural network image prediction (blue).

5.6.3 Test R.C

R.C.1 (1_SC) Spedale del Ceppo

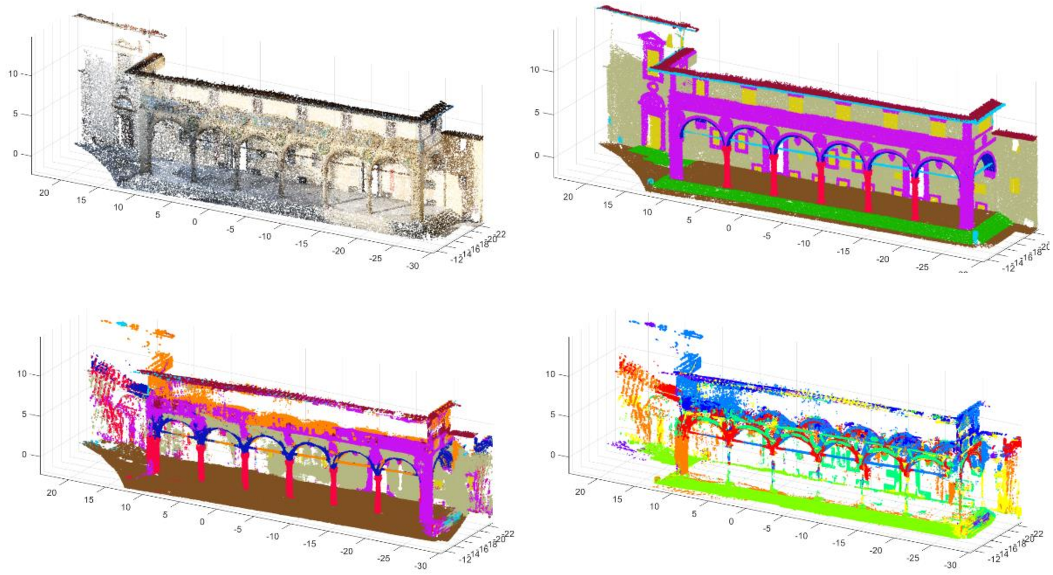
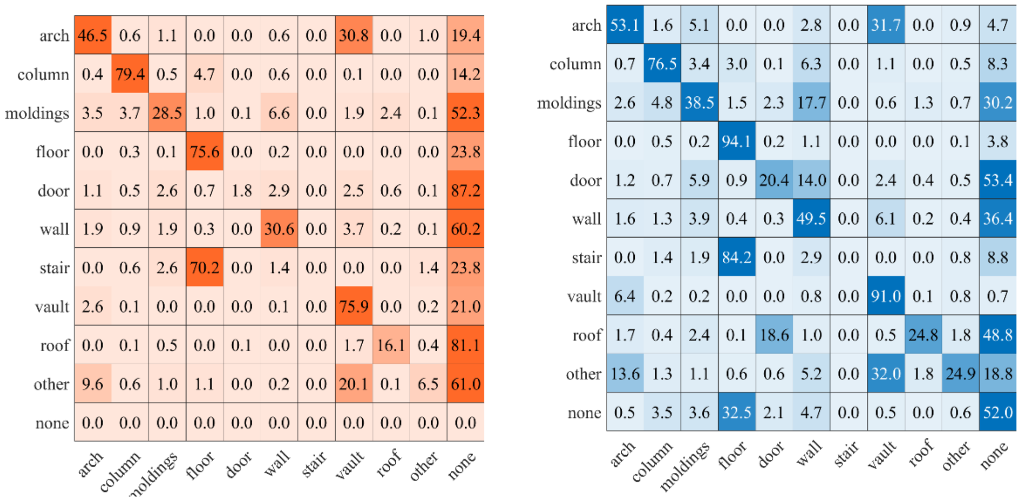


Figure 5.42 – (1_SC) RGB input point cloud (upper left), ground truth point cloud (upper right), predicted point cloud (down left), and misclassified points (down right).

Table 5.45 – (1_SC) Reprojection results of Test R.C.1.

RESULTS			
	All Points	**Only Classified**	**Images**
Unlabelled Points (%)	14,7	43,2	-
Global Accuracy (%)	50,4	52,0	56,1
Mean IoU (%)	30,6	32,6	32,5



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	46.5	0.6	1.1	0.0	0.0	0.6	0.0	30.8	0.0	1.0	19.4
column	0.4	79.4	0.5	4.7	0.0	0.6	0.0	0.1	0.0	0.0	14.2
moldings	3.5	3.7	28.5	1.0	0.1	6.6	0.0	1.9	2.4	0.1	52.3
floor	0.0	0.3	0.1	75.6	0.0	0.2	0.0	0.0	0.0	0.0	23.8
door	1.1	0.5	2.6	0.7	1.8	2.9	0.0	2.5	0.6	0.1	87.2
wall	1.9	0.9	1.9	0.3	0.0	30.6	0.0	3.7	0.2	0.1	60.2
stair	0.0	0.6	2.6	70.2	0.0	1.4	0.0	0.0	0.0	1.4	23.8
vault	2.6	0.1	0.0	0.0	0.0	0.1	0.0	75.9	0.0	0.2	21.0
roof	0.0	0.1	0.5	0.0	0.1	0.0	0.0	1.7	16.1	0.4	81.1
other	9.6	0.6	1.0	1.1	0.0	0.2	0.0	20.1	0.1	6.5	61.0
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	53.1	1.6	5.1	0.0	0.0	2.8	0.0	31.7	0.0	0.9	4.7
column	0.7	76.5	3.4	3.0	0.1	6.3	0.0	1.1	0.0	0.5	8.3
moldings	2.6	4.8	38.5	1.5	2.3	17.7	0.0	0.6	1.3	0.7	30.2
floor	0.0	0.5	0.2	94.1	0.2	1.1	0.0	0.0	0.0	0.1	3.8
door	1.2	0.7	5.9	0.9	20.4	14.0	0.0	2.4	0.4	0.5	53.4
wall	1.6	1.3	3.9	0.4	0.3	49.5	0.0	6.1	0.2	0.4	36.4
stair	0.0	1.4	1.9	84.2	0.0	2.9	0.0	0.0	0.0	0.8	8.8
vault	6.4	0.2	0.2	0.0	0.0	0.8	0.0	91.0	0.1	0.8	0.7
roof	1.7	0.4	2.4	0.1	18.6	1.0	0.0	0.5	24.8	1.8	48.8
other	13.6	1.3	1.1	0.6	0.6	5.2	0.0	32.0	1.8	24.9	18.8
none	0.5	3.5	3.6	32.5	2.1	4.7	0.0	0.5	0.0	0.6	52.0

Figure 5.43 – Test R.C.1 Point cloud confusion matrix (orange), related confusion matrix on neural network image prediction (blue).

R.C.2 (2_OSA) Ospedale Sant'Antonio

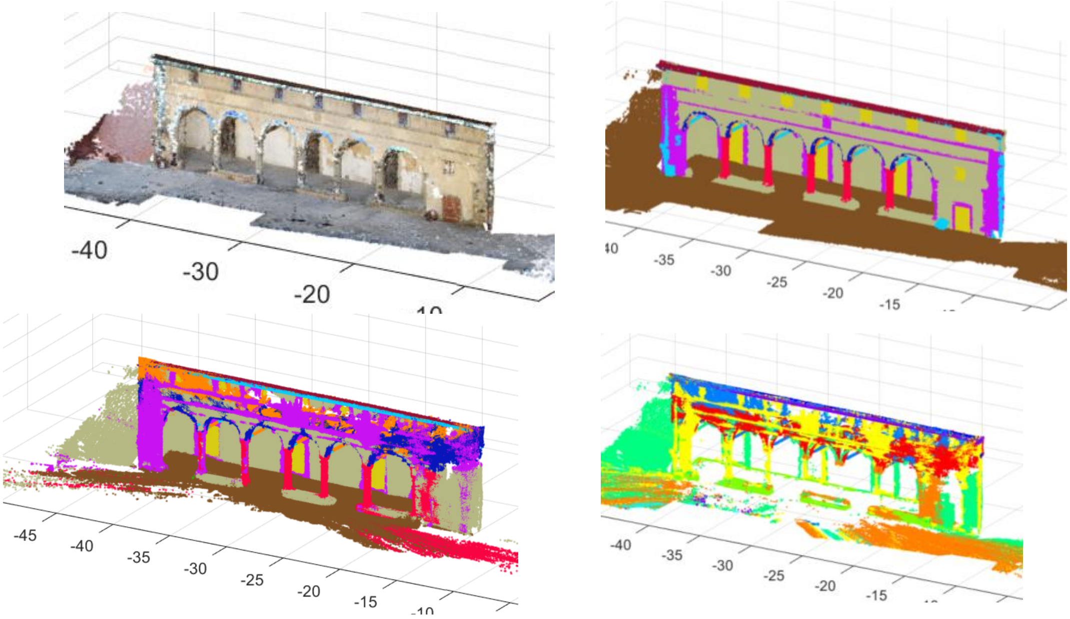
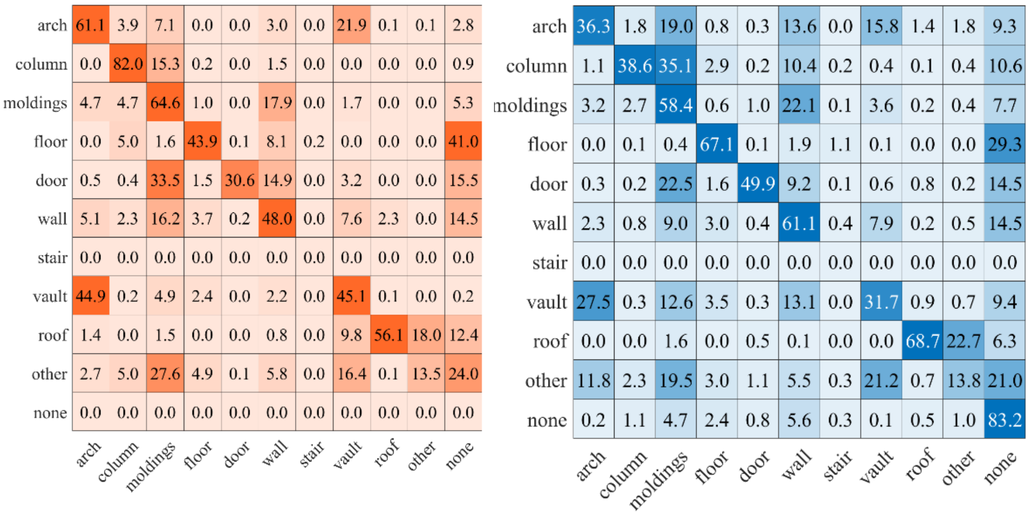


Figure 5.44 – (2_OSA) RGB input point cloud (upper left), ground truth point cloud (upper right), predicted point cloud (down left), and misclassified points (down right).

Table 5.46 – (2_OSA) Reprojection results of Test R.C.2.

RESULTS			
	All Points	**Only Classified**	**Images**
Unlabelled Points (%)	2,6	26,0	-
Global Accuracy (%)	48,1	54,8	58,4
Mean IoU (%)	27,2	31,2	31,8



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	61.1	3.9	7.1	0.0	0.0	3.0	0.0	21.9	0.1	0.1	2.8
column	0.0	82.0	15.3	0.2	0.0	1.5	0.0	0.0	0.0	0.0	0.9
moldings	4.7	4.7	64.6	1.0	0.0	17.9	0.0	1.7	0.0	0.0	5.3
floor	0.0	5.0	1.6	43.9	0.1	8.1	0.2	0.0	0.0	0.0	41.0
door	0.5	0.4	33.5	1.5	30.6	14.9	0.0	3.2	0.0	0.0	15.5
wall	5.1	2.3	16.2	3.7	0.2	48.0	0.0	7.6	2.3	0.0	14.5
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	44.9	0.2	4.9	2.4	0.0	2.2	0.0	45.1	0.1	0.0	0.2
roof	1.4	0.0	1.5	0.0	0.0	0.8	0.0	9.8	56.1	18.0	12.4
other	2.7	5.0	27.6	4.9	0.1	5.8	0.0	16.4	0.1	13.5	24.0
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	36.3	1.8	19.0	0.8	0.3	13.6	0.0	15.8	1.4	1.8	9.3
column	1.1	38.6	35.1	2.9	0.2	10.4	0.2	0.4	0.1	0.4	10.6
moldings	3.2	2.7	58.4	0.6	1.0	22.1	0.1	3.6	0.2	0.4	7.7
floor	0.0	0.1	0.4	67.1	0.1	1.9	1.1	0.1	0.0	0.0	29.3
door	0.3	0.2	22.5	1.6	49.9	9.2	0.1	0.6	0.8	0.2	14.5
wall	2.3	0.8	9.0	3.0	0.4	61.1	0.4	7.9	0.2	0.5	14.5
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	27.5	0.3	12.6	3.5	0.3	13.1	0.0	31.7	0.9	0.7	9.4
roof	0.0	0.0	1.6	0.0	0.5	0.1	0.0	0.0	68.7	22.7	6.3
other	11.8	2.3	19.5	3.0	1.1	5.5	0.3	21.2	0.7	13.8	21.0
none	0.2	1.1	4.7	2.4	0.8	5.6	0.3	0.1	0.5	1.0	83.2

Figure 5.45 – Test R.C.2 Point cloud confusion matrix (orange), related confusion matrix on neural network image prediction (blue).

R.C.3 (3_SS) Basilica della Santissima Annunziata

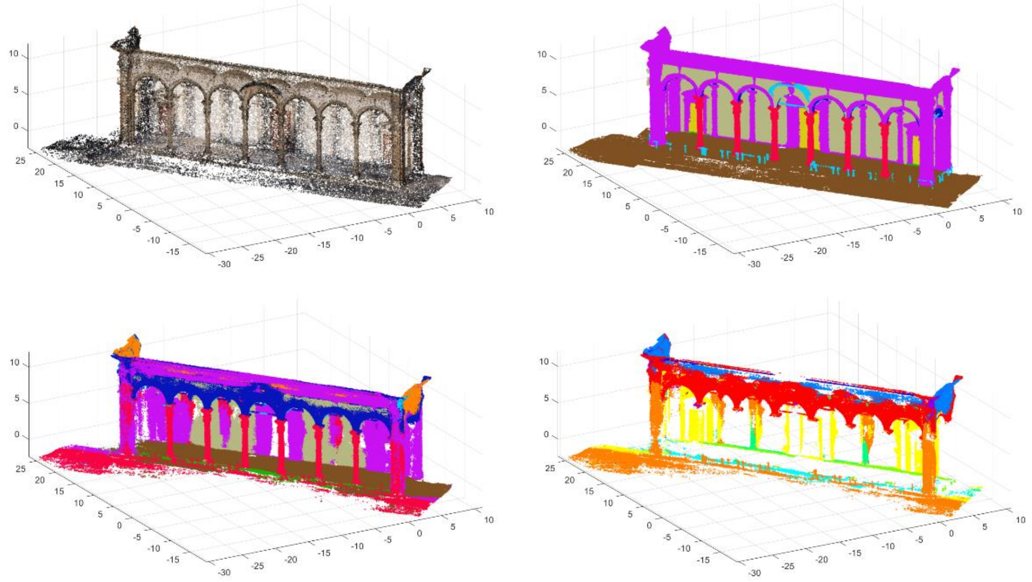
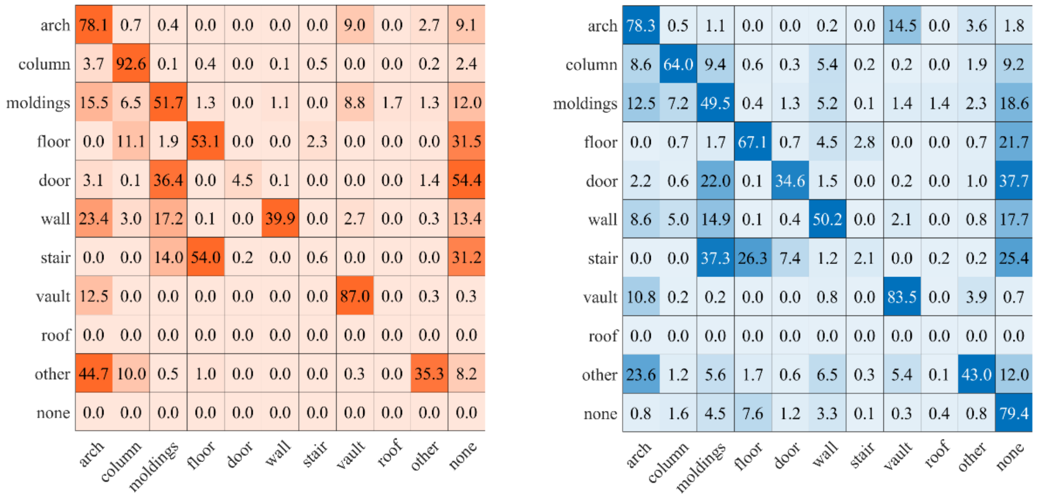


Figure 5.46 – (3_SS) RGB input point cloud (upper left), ground truth point cloud (upper right), predicted point cloud (down left), and misclassified points (down right).

Table 5.47 – (3_SS) Reprojection results of Test R.C.3.

RESULTS			
	All Points	**Only Classified**	**Images**
Unlabelled Points (%)	0,0	13,9	-
Global Accuracy (%)	56,7	61,3	62,9
Mean IoU (%)	27,9	32,2	34,4



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	78.1	0.7	0.4	0.0	0.0	0.0	0.0	9.0	0.0	2.7	9.1
column	3.7	92.6	0.1	0.4	0.0	0.1	0.5	0.0	0.0	0.2	2.4
moldings	15.5	6.5	51.7	1.3	0.0	1.1	0.0	8.8	1.7	1.3	12.0
floor	0.0	11.1	1.9	53.1	0.0	0.0	2.3	0.0	0.0	0.0	31.5
door	3.1	0.1	36.4	0.0	4.5	0.1	0.0	0.0	0.0	1.4	54.4
wall	23.4	3.0	17.2	0.1	0.0	39.9	0.0	2.7	0.0	0.3	13.4
stair	0.0	0.0	14.0	54.0	0.2	0.0	0.6	0.0	0.0	0.0	31.2
vault	12.5	0.0	0.0	0.0	0.0	0.0	0.0	87.0	0.0	0.3	0.3
roof	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
other	44.7	10.0	0.5	1.0	0.0	0.0	0.0	0.3	0.0	35.3	8.2
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	78.3	0.5	1.1	0.0	0.0	0.2	0.0	14.5	0.0	3.6	1.8
column	8.6	64.0	9.4	0.6	0.3	5.4	0.2	0.2	0.0	1.9	9.2
moldings	12.5	7.2	49.5	0.4	1.3	5.2	0.1	1.4	1.4	2.3	18.6
floor	0.0	0.7	1.7	67.1	0.7	4.5	2.8	0.0	0.0	0.7	21.7
door	2.2	0.6	22.0	0.1	34.6	1.5	0.0	0.2	0.0	1.0	37.7
wall	8.6	5.0	14.9	0.1	0.4	50.2	0.0	2.1	0.0	0.8	17.7
stair	0.0	0.0	37.3	26.3	7.4	1.2	2.1	0.0	0.2	0.2	25.4
vault	10.8	0.2	0.2	0.0	0.0	0.8	0.0	83.5	0.0	3.9	0.7
roof	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
other	23.6	1.2	5.6	1.7	0.6	6.5	0.3	5.4	0.1	43.0	12.0
none	0.8	1.6	4.5	7.6	1.2	3.3	0.1	0.3	0.4	0.8	79.4

Figure 5.47 – Test R.C.3 Point cloud confusion matrix (orange), related confusion matrix on neural network image prediction (blue).

R.C.4 (4_CG) Certosa del Galluzzo

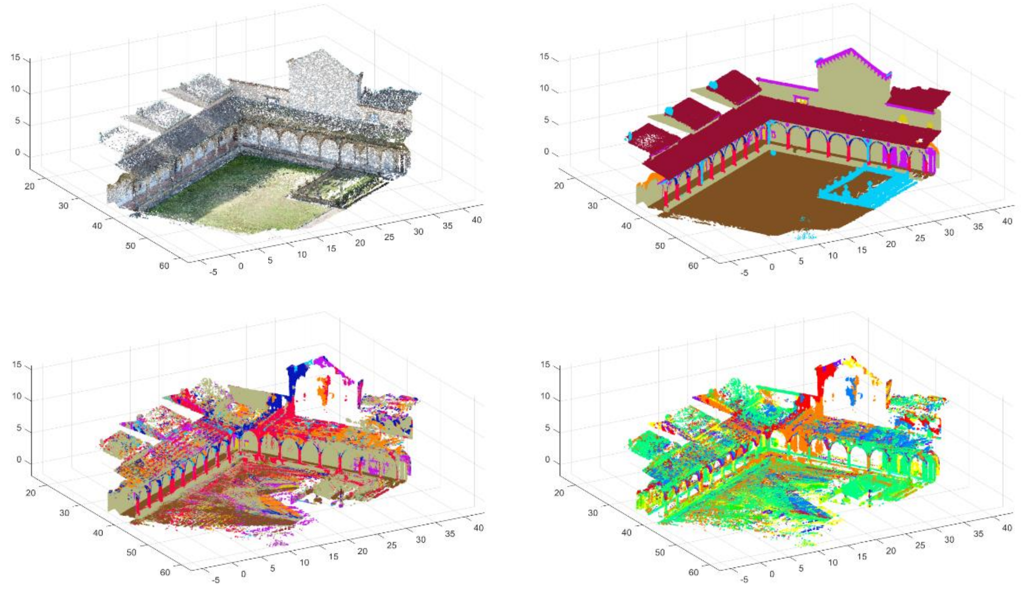
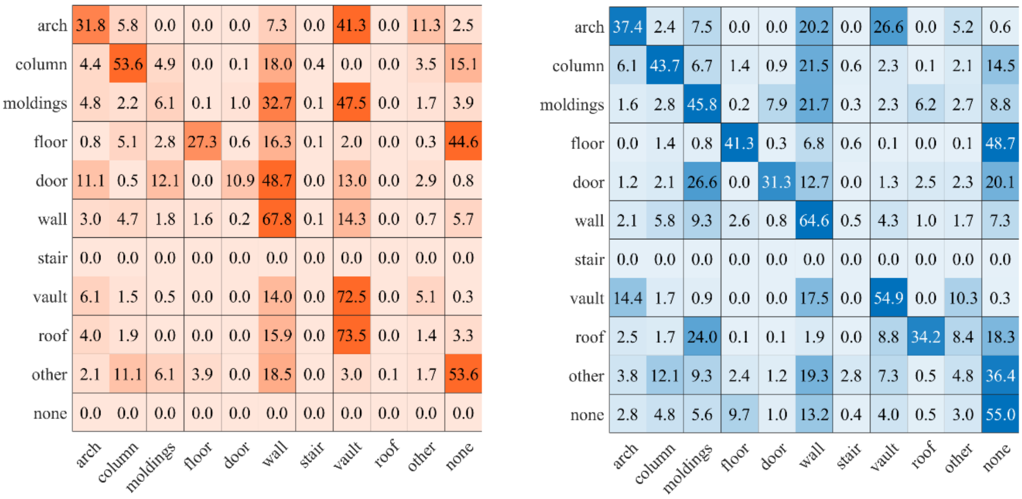


Figure 5.48 – (4_CG) RGB input point cloud (upper left), ground truth point cloud (upper right), predicted point cloud (down left), and misclassified points (down right).

Table 5.48 – (4_CG) Reprojection results of Test R.C.4.

RESULTS			
	All Points	**Only Classified**	**Images**
Unlabelled Points (%)	7,6	22,3	-
Global Accuracy (%)	39,3	53,4	51,1
Mean IoU (%)	13,1	19,7	22,9



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	31.8	5.8	0.0	0.0	0.0	7.3	0.0	41.3	0.0	11.3	2.5
column	4.4	53.6	4.9	0.0	0.1	18.0	0.4	0.0	0.0	3.5	15.1
moldings	4.8	2.2	6.1	0.1	1.0	32.7	0.1	47.5	0.0	1.7	3.9
floor	0.8	5.1	2.8	27.3	0.6	16.3	0.1	2.0	0.0	0.3	44.6
door	11.1	0.5	12.1	0.0	10.9	48.7	0.0	13.0	0.0	2.9	0.8
wall	3.0	4.7	1.8	1.6	0.2	67.8	0.1	14.3	0.0	0.7	5.7
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	6.1	1.5	0.5	0.0	0.0	14.0	0.0	72.5	0.0	5.1	0.3
roof	4.0	1.9	0.0	0.0	0.0	15.9	0.0	73.5	0.0	1.4	3.3
other	2.1	11.1	6.1	3.9	0.0	18.5	0.0	3.0	0.1	1.7	53.6
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	37.4	2.4	7.5	0.0	0.0	20.2	0.0	26.6	0.0	5.2	0.6
column	6.1	43.7	6.7	1.4	0.9	21.5	0.6	2.3	0.1	2.1	14.5
moldings	1.6	2.8	45.8	0.2	7.9	21.7	0.3	2.3	6.2	2.7	8.8
floor	0.0	1.4	0.8	41.3	0.3	6.8	0.6	0.1	0.0	0.1	48.7
door	1.2	2.1	26.6	0.0	31.3	12.7	0.0	1.3	2.5	2.3	20.1
wall	2.1	5.8	9.3	2.6	0.8	64.6	0.5	4.3	1.0	1.7	7.3
stair	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
vault	14.4	1.7	0.9	0.0	0.0	17.5	0.0	54.9	0.0	10.3	0.3
roof	2.5	1.7	24.0	0.1	0.1	1.9	0.0	8.8	34.2	8.4	18.3
other	3.8	12.1	9.3	2.4	1.2	19.3	2.8	7.3	0.5	4.8	36.4
none	2.8	4.8	5.6	9.7	1.0	13.2	0.4	4.0	0.5	3.0	55.0

Figure 5.49 – Test R.C.4 Point cloud confusion matrix (orange), related confusion matrix on neural network image prediction (blue).

R.C.5 (5_CB) Cappella Buontalenti

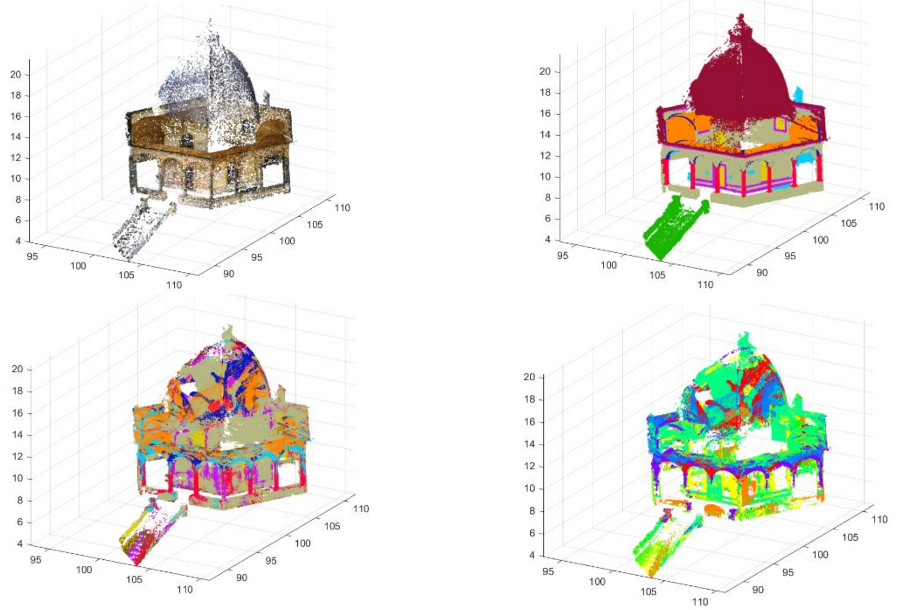
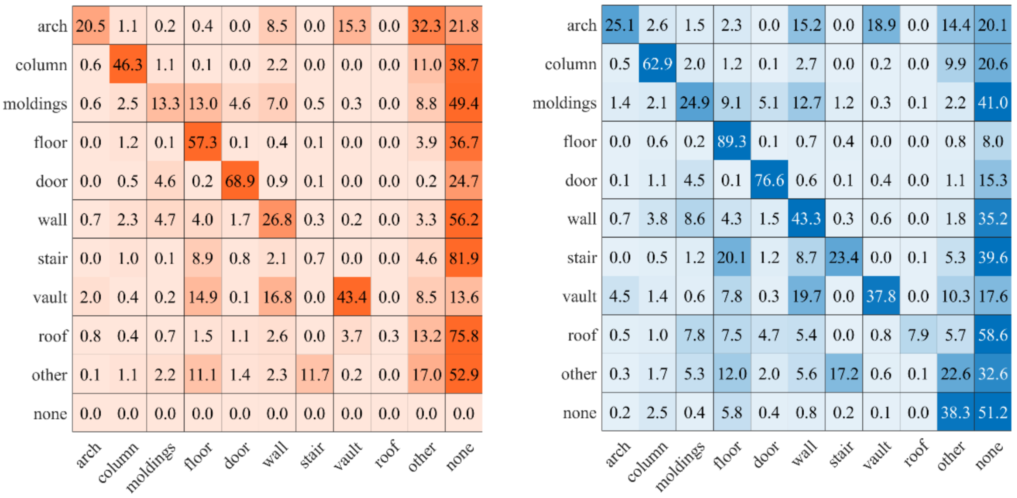


Figure 5.50 – (5_CB) RGB input point cloud (upper left), ground truth point cloud (upper right), predicted point cloud (down left), and misclassified points (down right).

Table 5.49 – (5_CB) Reprojection results of Test R.C.5.

RESULTS			
	All Points	**Only Classified**	**Images**
Unlabelled Points (%)	0,0	32,7	-
Global Accuracy (%)	35,2	42,3	46,9
Mean IoU (%)	14,9	18,8	28,7



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	20.5	1.1	0.2	0.4	0.0	8.5	0.0	15.3	0.0	32.3	21.8
column	0.6	46.3	1.1	0.1	0.0	2.2	0.0	0.0	0.0	11.0	38.7
moldings	0.6	2.5	13.3	13.0	4.6	7.0	0.5	0.3	0.0	8.8	49.4
floor	0.0	1.2	0.1	57.3	0.1	0.4	0.1	0.0	0.0	3.9	36.7
door	0.0	0.5	4.6	0.2	68.9	0.9	0.1	0.0	0.0	0.2	24.7
wall	0.7	2.3	4.7	4.0	1.7	26.8	0.3	0.2	0.0	3.3	56.2
stair	0.0	1.0	0.1	8.9	0.8	2.1	0.7	0.0	0.0	4.6	81.9
vault	2.0	0.4	0.2	14.9	0.1	16.8	0.0	43.4	0.0	8.5	13.6
roof	0.8	0.4	0.7	1.5	1.1	2.6	0.0	3.7	0.3	13.2	75.8
other	0.1	1.1	2.2	11.1	1.4	2.3	11.7	0.2	0.0	17.0	52.9
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	25.1	2.6	1.5	2.3	0.0	15.2	0.0	18.9	0.0	14.4	20.1
column	0.5	62.9	2.0	1.2	0.1	2.7	0.0	0.2	0.0	9.9	20.6
moldings	1.4	2.1	24.9	9.1	5.1	12.7	1.2	0.3	0.1	2.2	41.0
floor	0.0	0.6	0.2	89.3	0.1	0.7	0.4	0.0	0.0	0.8	8.0
door	0.1	1.1	4.5	0.1	76.6	0.6	0.1	0.4	0.0	1.1	15.3
wall	0.7	3.8	8.6	4.3	1.5	43.3	0.3	0.6	0.0	1.8	35.2
stair	0.0	0.5	1.2	20.1	1.2	8.7	23.4	0.0	0.1	5.3	39.6
vault	4.5	1.4	0.6	7.8	0.3	19.7	0.0	37.8	0.0	10.3	17.6
roof	0.5	1.0	7.8	7.5	4.7	5.4	0.0	0.8	7.9	5.7	58.6
other	0.3	1.7	5.3	12.0	2.0	5.6	17.2	0.6	0.1	22.6	32.6
none	0.2	2.5	0.4	5.8	0.4	0.8	0.2	0.1	0.0	38.3	51.2

Figure 5.51 – Test R.C.5 Point cloud confusion matrix (orange), related confusion matrix on neural network image prediction (blue).

5.7 Discussion

In this chapter, the results of the 3D point cloud semantic segmentation procedure have been reported. As mentioned, the procedure is composed by two main blocks: the semantic segmentation of the images, and the reprojection of the features on the point cloud. In this paragraph the performance and the accuracy of each block are assessed and widely discussed, trying to figure out the main issues and challenges, as well as the strengths and the positive results. Altough the final outcome of the procedure is the segmented 3D point cloud, and the image segmentation is only an intermediate phase of the segmentation, both blocks are analysed and discussed separately. This method allows a better understanding of the overall performance, and it allows to figure out and localize the major issues and limitation of the procedure.

5.7.1 Image semantic segmentation

Test A. As already explained, this is the simplest set of tests, and just one building at once has been used for the training and the evaluation phase. This tests have been mainly used to tune the hyperparameters of the neural networks, and to evaluate the best performing model. In general, the performances of the three models are satisfactory for each of the five buildings. The accuracy of the three models is quite similar in terms of GA and mIoU, but though slightly, the best performances have been achieved in all the three cases by Deeplabv3+ (Figure 5.52).

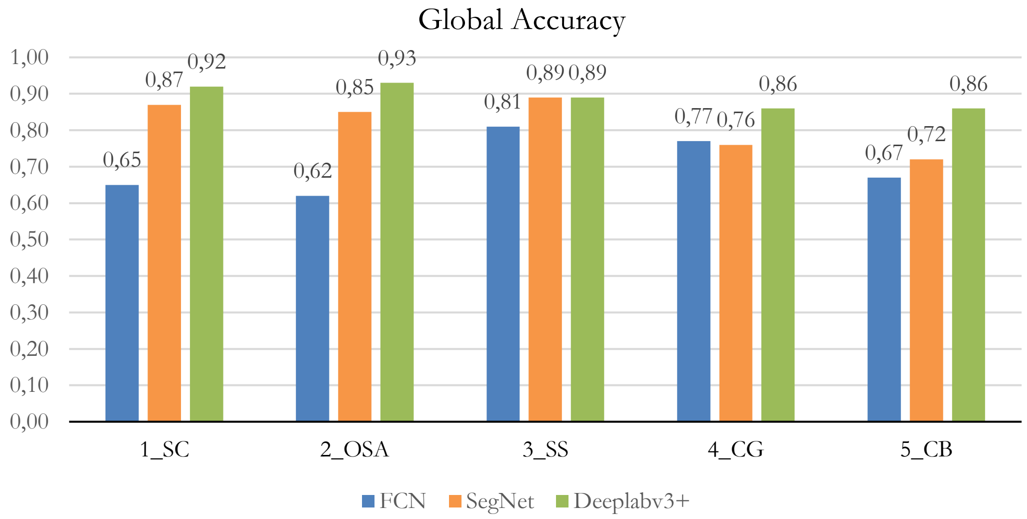


Figure 5.52 – GA of Test A for each building with FCN, SegNet and Deeplabv3+

This architecture turned out to be the most efficient also in terms of computational memory required, and training time. To achieve the highest performance FCN and

SegNet have been trained for 100 epochs, with a stabilization of the loss value after 80-90 epochs. For the same results, Deeplabv3+ has been trained for 50 epochs, and it achieved the best performance and the stabilization of the loss value after 30-40 epochs. This can be explained by the classification architecture used as based for the three models: FCN and SegNet use VGG-16, with a depth of 16 layers and 138 millions of learnable parameters. ResNet18, used by Deeplabv3+, has a depth of 18 layers, but just 11,7 millions of learnable parameters, that makes the training faster. Despite the lower number of parameters, ResNet based classification architectures are more efficient and ones of the best feature extractors for semantic segmentation. As shown in Figure 5.52, the first three buildings of the dataset, (1_SC), (2_OSA) and (3_SS) achieved a similar performance, with a GA around 90%. Each class is well predicted, without significant errors. From the confusion matrices turned out a general minor trend to confuse the class "arch" with the class "vault", and the class "stair" with the class "floor". It is not surprising since the similar nature of these class typologies. The last two buildings of the dataset (4_CG) and (5_CB) showed a lower overall performance compared to the first ones, and they showed a sparser matrix without a precise scheme of errors. It can be the result of a general lower precision of the ground truth caused by some issues in the 3D scene. In (4_CG) just a portion of the 3D scene has been reconstructed, and the remaining part has been classified like "background" in the images. Therefore, during the testing phase, the background is sometimes confused with other categories. In (5_CB) the large presence of vegetation around the buildings led to a less accurate point cloud, hence to a less accurate ground-truth caused by occlusion problems. These problems can be addressed and overcome by directly improving the accuracy of the initial point cloud. In conclusion, Test A allowed to tune the various networks and identify the most suitable and efficient one. As expected, the overall performance is good for each building, but it is not significant in terms of the generalization and capabilities of the model, since the images used to train the model are similar to the images used to test it, without relevant differences among the same class.

Test B. This test allowed to evaluate the performance of the three network architectures with more than one building in the training set and in the test set, and to assess which one is more able to generalize among different building typologies. Moreover, these tests were helpful to fine-tune the hyperparameters and to find the optimal training settings. The performances of all the three cases are quite good, and once again, Deeplabv3+ turned out to be the most accurate and efficient, with a GA of 89% and a mIoU of 76% obtained with a training of 50 epochs. Similar results were obtained with FCN with a 100 epochs training, while SegNet achieved the worst performance with a GA of 74% of and a mIoU of 55%. The confusion matrices show the absence of remarkable prediction errors, with the same little exchange of Test A between the classes “arch” and “vault”. As the previous set of tests, these results are not significant in terms of generalization of the model since similar images of the same building were used both for training and test the network. Due to the higher performance and efficiency of Deeplabv3+ compared to the other networks, the following tests have been performed using only this network architectures.

Test C. The final set of tests was the most challenging, since the main aim was the prediction of an unseen scenario. A cross validation between the five buildings has been performed, using as training set four buildings, and as test set the remaining one. Considering the efficiency compared to the other networks, just Deeplabv3+ was used to carry on Test C. The general performance of the cross-validation test for all the five buildings is still unsatisfactory. The average GA of the cross-validation test is 55% and the average mIoU is 30%.

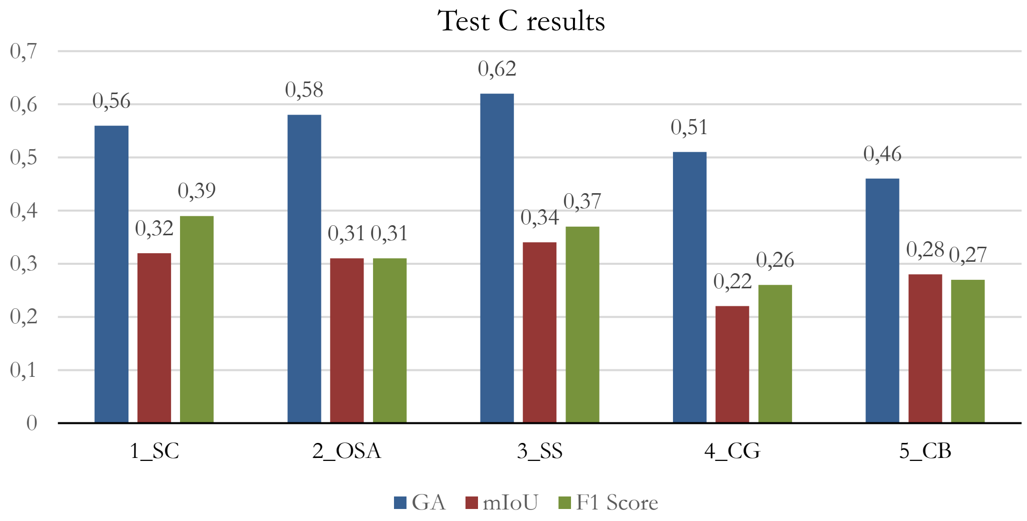
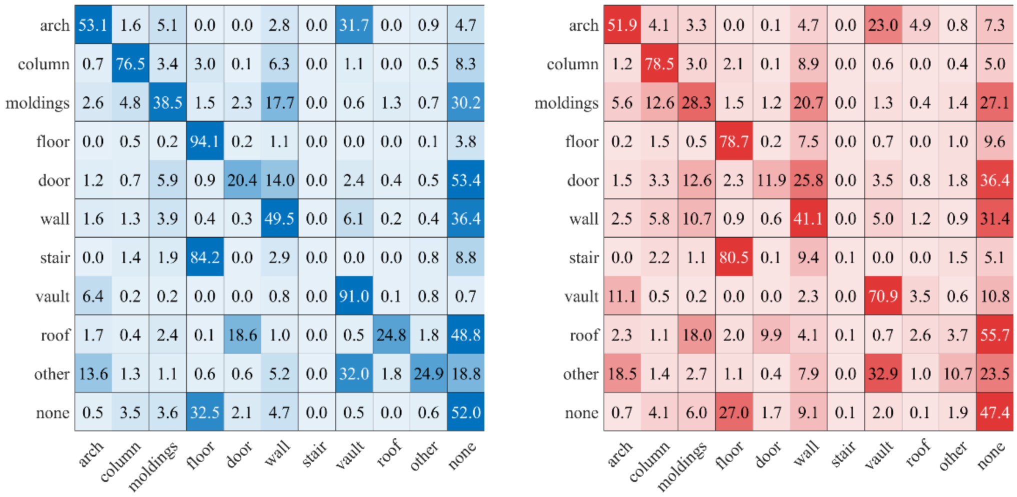


Figure 5.53 – Performance on Test C for the five buildings

At the same time, the GA and the mIoU on the related internal test set composed by an image subset of the training buildings are respectively 89% and 78%. This kind of performance reduction represents a clear case of overfitting. Overfitting occurs when the machine model gives accurate prediction for training data but not for new data, and when the model cannot generalize and fits too closely to the training dataset instead. Generally, it happens due to several reasons: (i) the training data size is too small and does not contain enough relevant data samples, (ii) the training data contains large amount of irrelevant or noisy information, (iii) the model trains for too long on a single sample of data, (iv) the model complexity is high, so it learns the noise within the training data. However, it is not surprising that the models show overfitting problems, since as already mentioned, the dataset still presents limitation in number of samples and relevant data. Despite the number of images could be enough to train such models, the images in the training set represents just four buildings, hence they are not various and relevant enough to generalize different class typologies. Several methods have been used to address the overfitting problem during training: regularization, class weighting, dropout, early stopping. These methods showed no remarkable improvements, which implies the issue is mostly data related. In that case, data augmentation can be an effective strategy, but also in this case no improvements have been achieved. As example, the results of Test C.1 using data augmentation are reported. Online augmentation has been used, and it consists of applying the transformation directly on the minibatch during training with a random step. Various types of transformations can be used, and in this example vertical mirroring, cropping, blurring, noising, and colour filtering have been tested. The performance on the test set with data augmentation decrease from 56% to 47% for the GA and from 32% to 24% for the mIoU. Figure 5.54 shows the confusion matrices obtained without data augmentation (blue) and with data augmentation (red), and it shows a remarkable decrease of the accuracy for each class.

According with the performed tests and the related results, the other strategy to avoid overfitting and improve the overall performance of the model is to increase the number of building typologies in the dataset. Despite the evident weak results achieved in the case of unseen scenarios for all the five buildings, by looking deeper at the results in the confusion matrices some positive observations can be made. At first, generally the matrices maintain a good diagonality, and the prediction errors is mostly



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	53.1	1.6	5.1	0.0	0.0	2.8	0.0	31.7	0.0	0.9	4.7
column	0.7	76.5	3.4	3.0	0.1	6.3	0.0	1.1	0.0	0.5	8.3
moldings	2.6	4.8	38.5	1.5	2.3	17.7	0.0	0.6	1.3	0.7	30.2
floor	0.0	0.5	0.2	94.1	0.2	1.1	0.0	0.0	0.0	0.1	3.8
door	1.2	0.7	5.9	0.9	20.4	14.0	0.0	2.4	0.4	0.5	53.4
wall	1.6	1.3	3.9	0.4	0.3	49.5	0.0	6.1	0.2	0.4	36.4
stair	0.0	1.4	1.9	84.2	0.0	2.9	0.0	0.0	0.0	0.8	8.8
vault	6.4	0.2	0.2	0.0	0.0	0.8	0.0	91.0	0.1	0.8	0.7
roof	1.7	0.4	2.4	0.1	18.6	1.0	0.0	0.5	24.8	1.8	48.8
other	13.6	1.3	1.1	0.6	0.6	5.2	0.0	32.0	1.8	24.9	18.8
none	0.5	3.5	3.6	32.5	2.1	4.7	0.0	0.5	0.0	0.6	52.0

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	51.9	4.1	3.3	0.0	0.1	4.7	0.0	23.0	4.9	0.8	7.3
column	1.2	78.5	3.0	2.1	0.1	8.9	0.0	0.6	0.0	0.4	5.0
moldings	5.6	12.6	28.3	1.5	1.2	20.7	0.0	1.3	0.4	1.4	27.1
floor	0.2	1.5	0.5	78.7	0.2	7.5	0.0	0.7	0.0	1.0	9.6
door	1.5	3.3	12.6	2.3	11.9	25.8	0.0	3.5	0.8	1.8	36.4
wall	2.5	5.8	10.7	0.9	0.6	41.1	0.0	5.0	1.2	0.9	31.4
stair	0.0	2.2	1.1	80.5	0.1	9.4	0.1	0.0	0.0	1.5	5.1
vault	11.1	0.5	0.2	0.0	0.0	2.3	0.0	70.9	3.5	0.6	10.8
roof	2.3	1.1	18.0	2.0	9.9	4.1	0.1	0.7	2.6	3.7	55.7
other	18.5	1.4	2.7	1.1	0.4	7.9	0.0	32.9	1.0	10.7	23.5
none	0.7	4.1	6.0	27.0	1.7	9.1	0.1	2.0	0.1	1.9	47.4

Figure 5.54 – Comparison of confusion matrix with (blue) and without (red) data augmentation.

concentrated in the last column of the matrix. Hence most of the incorrect pixels are classified like "none", that can be considered as a non-classification rather than a misclassification. Generally, some classes always have a better accuracy compared to others. The best accuracies are always achieved on "column", "floor", "vault" and "none" that are the most prevalent classes in the scene. The worst accuracies are achieved on "stairs", "roof" and "other" that are the lower percentage classes. The resulting mIoU is negatively affected by these errors equally, without considering the percentage proportion in the scene. The class "wall" is mostly well predicted with a good accuracy, but generally it is over segmented, and other classes are sometimes confused with it. Such the other tests, the class "arch" and "vault" are sometimes interchanged as well as "stair" and "floor". Finally, the limitation of the class "none" or "background" during training is to underline. As shown in Figure 5.55 the labels in the ground truth are assigned only to the main building, while the remaining pixels are automatically labelled as "none" (in black). These pixels comprise several objects such as sky, vegetation, streets, etc., but often, they comprise other buildings in the background, moreover with elements similar to the main building. Figure 5.55 is a clear example of this issue, in which the windows, the wall and the roof of the background building are analogous to the related objects in the main building. This issue creates a bias during training, and it prevents the model to learn and generalize appropriately the various categories. Since the class "none" is the most represented in the dataset, it has a relevant weight during training, hence it partially explains the high percentage of "none" pixels in the predicted images. Moreover, this issue makes the correct evaluation of the model more challenging to interpret, since when the model correctly predicts a class in the background building, for instance a window or a wall, it is computed as an error. Two main strategies can be used to address the problem. Firstly,

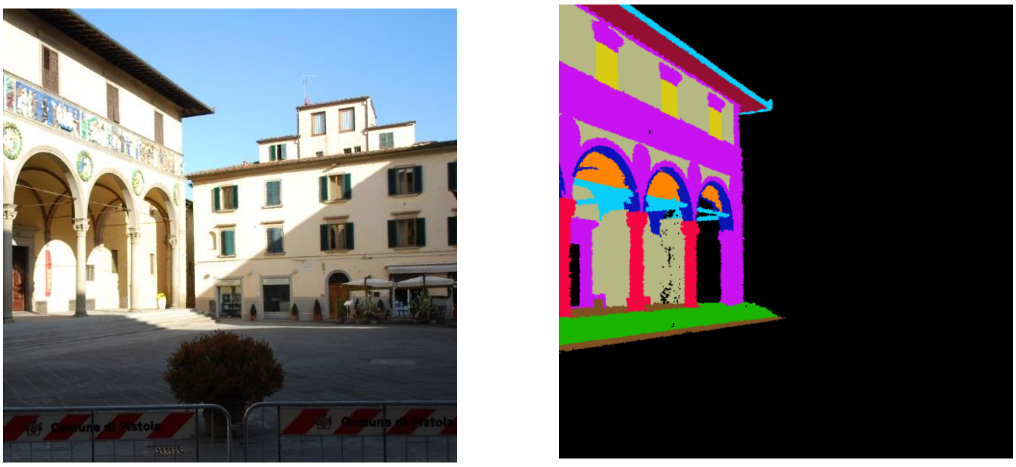


Figure 5.55 – An example of background bias of (1_SC)

by giving to the class "none" a small weight during training. It is a tricky and fast solution to overcome the issue, but some preliminary tests have shown that together with the improvements of "none" predictions, in general it involves a decrease of the overall performance. The second solution is to integrate and label the background buildings in the scene, hence, to correctly label as much as pixels as possible. Despite it requires longer annotation time, it would have lot of benefits in terms of generalization, class balancing and obstruction problems. The annotation of the dataset will be considered in further improvements of the dataset, as well as further expansion and integration.

Overall Conclusion. Image segmentation is the first block of the segmentation procedure, and a fundamental step in obtaining a final good point cloud performance. The first two tests (A and B) have shown a good overall performance. Despite they are not significant in terms of generalization, they proved the effectiveness of deep neural network for semantic segmentation of heritage scenes when a relevant training dataset is provided. The last test (C) did not achieve sufficient results, but it was expected since the low number of building typologies in the dataset. Several measures and expedients have been used to overcome the dataset limitation during training, but they were not adequate, underlying a strong relevance of the weakness of the training data during the learning phase. As already mentioned, two strategies can be carried on: the expansion of the dataset with new buildings or with other existing datasets, and the improvement of the image ground truth by labelling the background buildings.

5.7.2 Labelling projection on point cloud

Test R.GT. This series of tests aimed to assess and evaluate the effect of the projection parameters on the final point cloud, and to find the optimal parameters combination. To simplify the valuation of the correct settings, the ground truth images have been used. Generally, the procedure works well, but considering a 100% GA and a 100% mIoU of the ground truth, a decreasing of the performance on the images is always present. Considering all the points of the cloud in the metrics computation, the best test achieved a GA of 86% and a mIoU of 73%, with a percentage of unclassified points of 10%. Taking out the unclassified points from the metrics computation, the best result achieved a GA of 97% and a mIoU of 86%. Hence, in best of the cases, a reduction of 3% of the GA and 14% of the mIoU was observed. Depending on the choice of the parameters, it is possible to privilege a high accuracy at the cost of a higher percentage of unclassified points and vice versa.

Test R.A. This set of tests consisted in the projection of the extracted features by the classifier of Test A on the related point cloud. The survey images of each building were input in the trained classifier, and then used to transfer the label to the related point cloud. Generally, the results are promising, and the accuracy obtained on the images is maintained on the point cloud with no remarkable decreasing. In the figures below (Figure 5.56 and Figure 5.57) are reported the metrics for each of the five buildings, both for the images (blue) and for the point cloud (orange).

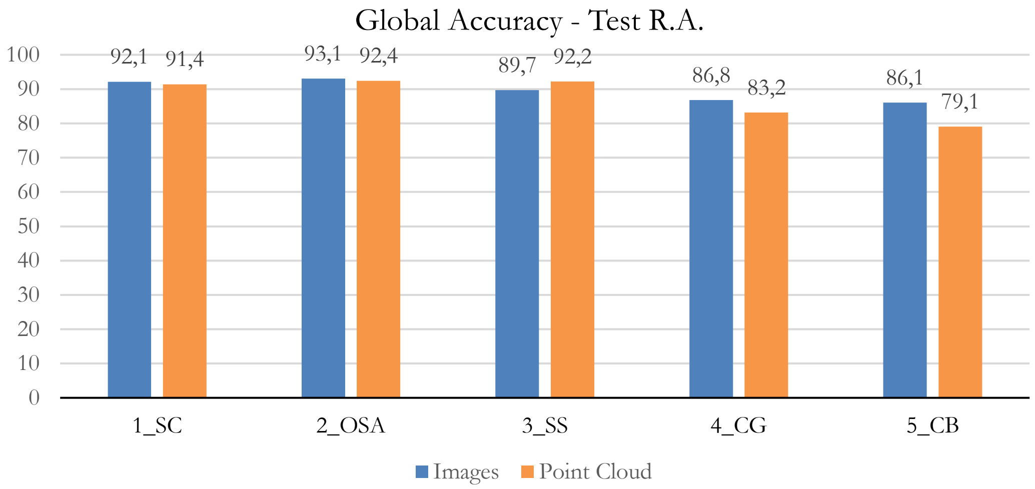


Figure 5.56 – Reprojection results for the five buildings on Test R.A (GA)

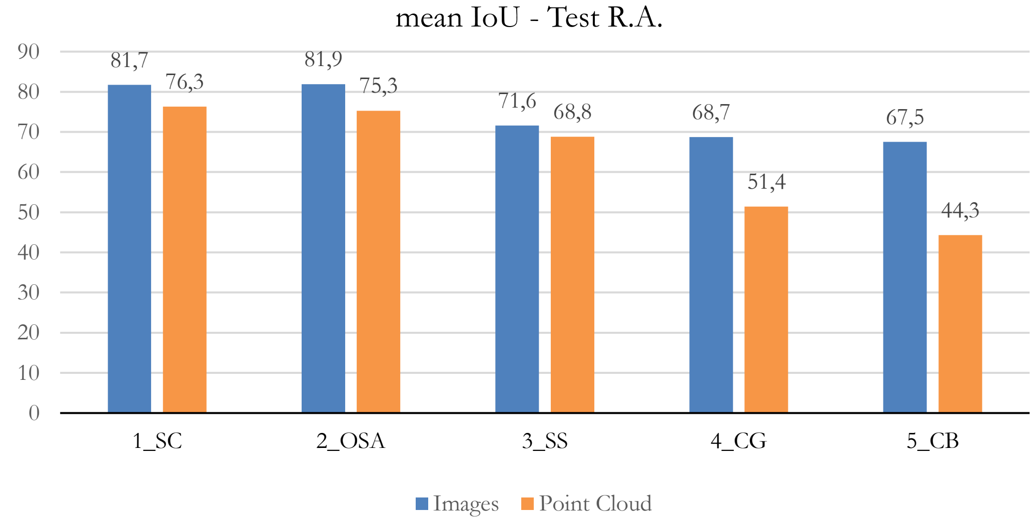


Figure 5.57- Reprojection results for the five buildings on Test R.A (mIoU)

As shown in the histograms, all the five tests reported a little decreasing of the performance both for the GA and the mIoU, except the GA of (3_SS) in which a little increasing is reported. Such as in the previous tests, the results on the first three buildings of the dataset turned out to be the most accurate. These cases reported a decrease of the GA of 1-2% and of the mIoU of 5-6%. The last two buildings have a general worst performance and a GA and mIoU decreasing after the projection of 5-7% and 10-15% respectively. Looking deeper at the confusion matrices turned out a good performance for each class without errors during the label transfer. The wrong predictions are generally the same as the images, and they are propagated to the point cloud with the same percentages. As reported in the images, the final segmented point clouds show a good visual quality and accuracy, and they show a good overlapping with the ground truth. As shown in the misclassified point clouds, the wrong predicted or unclassified points are mainly located on the connection regions between different categories. In conclusion, the tests proved a proper functioning of the projection procedure, with a robust and stable performance even with predicted images with different level of accuracy.

Test R.C. This set of tests consisted in the projection of the extracted features by the classifier of Test C on the external test building. The evident poor performances of this series of tests are not surprising, since the results are strongly related to the image segmentation performance of Test C discussed in the previous paragraph. The starting image segmentation accuracy was low, and the quality of the images was insufficient to expect good results. In the figures below (Figure 5.58 and Figure 5.59) are reported the GA and the mIoU obtained on the images (blue) and on the point cloud (orange).

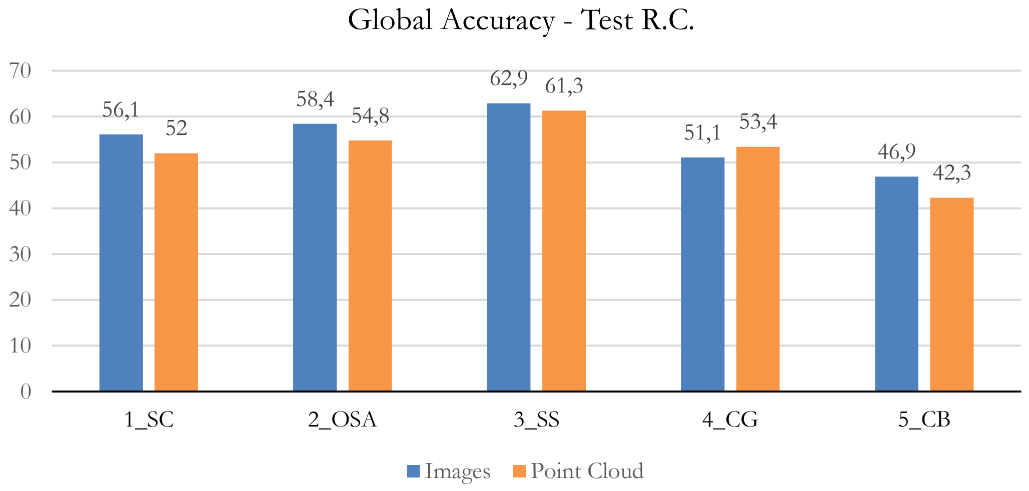


Figure 5.58 – Reprojection results for the five buildings on Test R.C (GA)

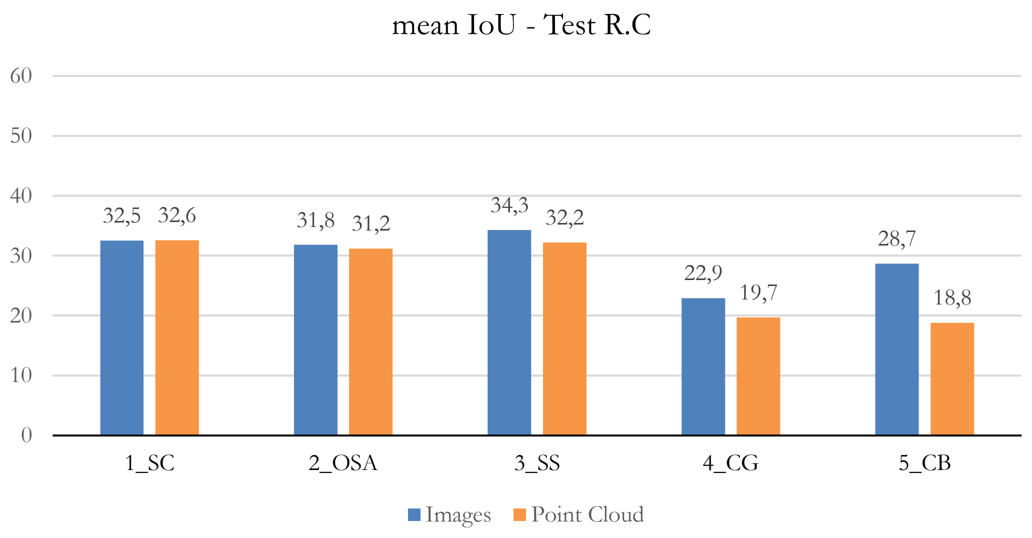


Figure 5.59 – Reprojection results for the five buildings on Test R.C (mIoU)

Despite the overall performance is clearly unsatisfactory, the stability and the robustness of the projection procedure is validated even in this series of tests. There is no remarkable reduction of the accuracy achieved on the images, and there is no remarkable error propagation between the various classes. To make a comparison, in the figure below (Figure 5.60) are reported the results achieved with a masking-based methodology introduced in (Murtiyoso et al., 2022) and reported in (Pellis et al., 2022).

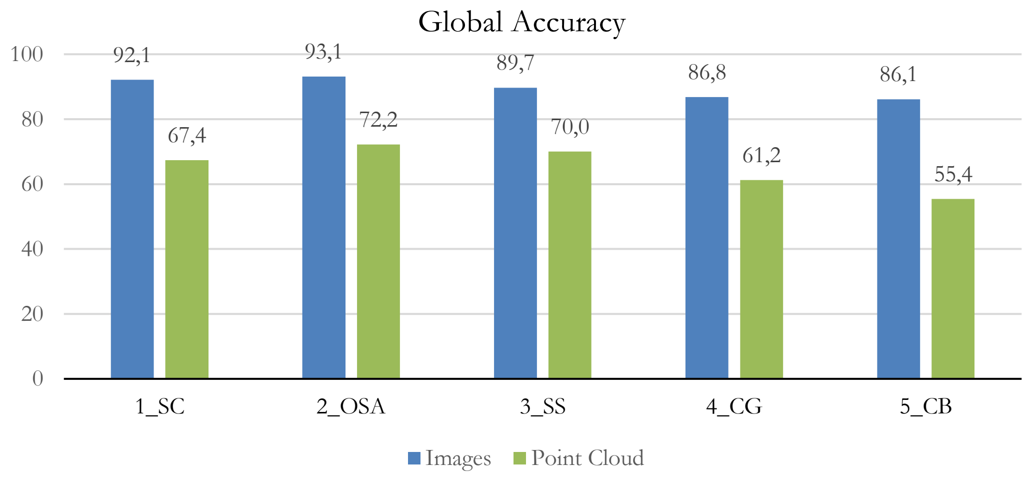


Figure 5.60 – Masking-based results for the five buildings on Test A (GA)

In green are reported the accuracy obtained on the reconstructed point clouds using as masks the prediction output by the classifier trained in test A. As shown in the histogram, the performance on the images (blue) remarkable decreased with the reconstruction. The proposed projection procedure overcomes the performance of this methodology.

Overall Conclusion. Labels projection on the point cloud is the second block of the segmentation procedure, and it allows to transfer the features extracted by the neural network from the images to the points cloud. The results have shown a good stability of the procedure, even in the case of low-quality images, with no notable loss of accuracy of the input image performances. However, the procedure leads to a good accuracy only if the input labelled images have an equally good accuracy. Unfortunately, when the accuracy of the images is low, the projection procedure is not still able to improve the performance during the transferring of the labels to the cloud. Future improvements could be done in order to allow the overcoming of the input accuracy. For instance, input labelled images selection or weighting depending on the view, angle or distance from the target building could be carried on.

5.8 Summary

In this chapter the results of the 3D semantic segmentation procedure have been illustrated and discussed in detail both for the image segmentation and the label transfer to the point cloud. In the first part of the chapter, the details about the implementation of the neural networks have been shown, including FCN (§5.2.1), SegNet (§5.2.2), and Deeplabv3+ (§5.2.3). Paragraph §5.3 reported the detail about the training phase, including the image processing (§5.3.1), the test structures (§5.3.2), the hyperparameter tuning (§5.3.3), and the metrics used to evaluate the models (§5.3.4). Finally, paragraph §5.4 reported in detail all the results achieved by the three neural networks on the three Tests A, B, and C. The second part of the chapter showed the results of the label projection from the image to the cloud. At first, paragraph §5.5 explained the functioning of the procedure, illustrating the various parameters and options that rules the reprojection, and their effect on the final outcome. Secondly, paragraph §5.6 reported in detail all the results achieved on the point clouds, organized in three tests: Test R.GT, Test R.A and Test R.C. At the end of the chapter the results have been exhaustive discussed both for image segmentation (§5.7.1) and for labelling projection (§5.7.2).

Chapter 6

Conclusions and Future Developments

This final chapter concludes the work presented in this thesis by first presenting a summary of the material discussed previously (§6.1), and then by a discussion on the conclusion of this dissertation (§6.2). Remarks and future developments are also presented and discussed in (§6.3).

6.1 Summary

This dissertation arises in the wide process of the digitization of heritage buildings, largely promoted and supported in recent years by European Union. This thesis is focused particularly on the 3D point cloud processing that leads to the creation of 3D informative models, through the process commonly known as Scan-to-BIM. The main goal of this thesis was to develop a new deep learning workflow for the semantic segmentation of 3D point clouds, aiming at supporting and speeding up the modeling phase of the Scan-to-BIM, currently the most tedious and time-consuming operations of the entire process. The first part of the thesis (§2, §3) provided a comprehensive background about the research topic, and the relative literature review. Chapter 2 introduced the concept of H-BIM and the Scan-to-BIM, and it provided an exhaustive review about the algorithmic approaches to address each phase of the Scan-to-BIM. Point cloud processing has been widely investigated, starting from point cloud acquisition, up to their manipulation including down sampling, registration, segmentation and the BIM modeling from point cloud. Chapter 3 deepened the problem of semantic segmentation of point cloud, one of the most challenging steps in the Scan-to-BIM process, and the main problem faced in this dissertation. The main algorithms and methods existing in the literature have been widely analysed and discussed, focusing in particular on the deep learning approaches. They are categorized into two main groups: projection-based methods, that leverage on an intermediate representation of the cloud to extract features, and point-based methods, that work directly with the raw point cloud. At the end of Chapter 3 the developed procedure has been introduced. It is based on a multiview approach, a projection-based method in which the segmentation of the point cloud is carried out at first on images, and then the features are transferred to the point cloud. This method is especially suited for the photogrammetric point clouds, since they are generated starting from images. The method could be integrated in the photogrammetric pipeline, aiming at generating a directly segmented point cloud. The second part of the thesis (§4, §5) provided the practical development and application of the proposed procedure. Chapter 4 introduced the new dataset, specifically created for training and testing the procedure. The benchmark deployment has been illustrated in detail, including the acquisition and the processing phase, the choice of the categories, the labelling procedure, and the final statistic and trend. The benchmark contains several data typologies, and currently it is composed of five heritage scenes, each one including the TLS and photogrammetric point cloud labelled according to the ARCHdataset, and the photogrammetric images with their related labelling. Chapter 5 illustrated in detail the results of the segmentation procedure on the new dataset, both for image segmentation and labelling projection on the point cloud. Several experiments have been proposed, up to training and developing a model potentially able to predict an unseen scenario. Finally, the results have been widely discussed, and the weaknesses and the strengths of the procedure have been pointed out.

6.2 Conclusions and remarks

As mentioned in paragraph (§1.3) the overall goal of this dissertation was to improve the automation in the digitization of cultural heritage, providing an efficient strategy to speed up the transition from point cloud data to 3D digital model in the context of the Scan-to-BIM. Since the huge and complex structure of the Scan-to-BIM, the first research questions (§1.3.2) were focused on identifying the main issues that usually make challenging the shifting from point data to 3D models, and which could be the main instruments and approaches to face them. Chapter 2 and Chapter 3 widely analysed these two aspects: point cloud semantic segmentation turned out to be the most challenging step in Scan-to-BIM and a key point to reduce manual intervention and time-consuming operations. It is worth to underline that semantic segmentation is only a small part of the extensive Scan-to-BIM process, and this research work set its confines in deepening just this limited aspect. The presented approach and the related results are focused on improving just the performance of point cloud semantic segmentation, excluding at the moment the integration of the segmented clouds in the BIM platforms, and the transformation of segmented clouds in parametric elements. However, the improvement of results in cloud segmentation would lead to a significant breakthrough, and a remarkable step ahead in the Scan-to-BIM process applied to heritage buildings. Due to the recent and impressive progress in artificial intelligence in several fields, the second research question asked whether AI could be used effectively for the digitization of cultural heritage. Chapter 3 largely analysed all the semantic segmentation approaches, and the AI branches of machine learning and deep learning turned out to be the most efficient and promising techniques to address the problem of semantic segmentation of point clouds. Several approaches have been developed in literature, and they are grouped into two main categories: point-based and projection-based methods. As specified in the objectives (§1.3.2) this thesis aimed at proposing an innovative segmentation pipeline particularly suited for the heritage building point clouds. The resulting pipeline is a multiview based approach that leverages on image features extraction, and on the projection of the features on the point cloud. This approach turned out to be very suitable for heritage point clouds, and it showed several advantages reported in the following.

- Higher performance on image semantic segmentation compared to point cloud segmentation, obtained in particular by means of CNNs.
- Large availability of existing image segmentation datasets to pretrain the model or exploit transfer learning.
- Large availability of high-resolution images acquired during the survey, remarkably relevant to capture geometrical details, articulated textures, or complex constructive elements of heritage buildings.
- Possible integration of the segmentation approach in the photogrammetric pipeline, in order to obtain a directly segmented point cloud.
- Larger availability and easier acquisition of images to increase and expand the dataset compared to point clouds.

The last research question was focused on the applicability and the generalization of the proposed approach in a large-scale context, and on the capabilities of the model to generalize among several building typologies, multiple constructive elements, complex and non-standard elements. This question was largely faced in Chapter 4 and Chapter 5, at first by introducing a new specific benchmark, and secondly by testing and assessing the procedure with several experiments.

The aim of the benchmark was to train the convolutional neural network at the core of feature extraction block. Chapter 4 has shown in detail the procedures that led at the creation of an exhaustive dataset composed both by point clouds and images. A remarkable contribution was the development of the semiautomatic labelling procedure of the images starting from a manual segmentation of the related point cloud. The procedure turned out to be very effective in terms of time saving and accuracy. It could be applied to any point cloud aligned with a set of photogrammetric images, and it could be useful to extend the dataset in the future and increase its generalization and capabilities. The dataset turned out to be valuable and well-structured, and it gives the opportunity to develop and test various segmentation strategies since the availability of both images and point clouds. However, it has three main limitations. At first, the generation of a dataset always requires lot of time, and currently it is composed by just five buildings. The current number of images is more than 3000, but they are representative of too few building typologies. Secondly, the current statistic shows a remarkable class imbalance, that could negatively affect the training, and bias the correct evaluation of the model. This issue could be faced on one hand directly during training (class weighting or data augmentation), and on the other hand by new targeted acquisitions. Finally, the current point cloud scenes did not consider the background, and as explained in (§5.7.1), it caused biases during the training phase. Future integrations and improvements will consider the background annotation.

Concerning the achieved results, in the general case of unseen scenario, currently the performance is totally unsatisfactory: the average of the cross-validation test on the five buildings of the dataset achieved a GA of 54%. It proved the model to be still unable to generalize among several building typologies. As shown previously, this poor overall performance is caused just by the performance of the neural network on the image segmentation rather than the label projection step. However, these results are not surprising. As pointed out by several researches, deep learning model are data hungry, and image segmentation models particularly require thousands of relevant images to develop a highly capable network. Despite the number of images in the dataset is considerable, they are representative of only five buildings, hence they are not relevant enough. This is an inadequate number of architectural cases to enable the model to generalize among several buildings. To have a reference, the number of building typologies should be at least an order of magnitude higher: indicatively 50-100 buildings could be a reasonable number to obtain a good capable model.

Nevertheless, despite this limitation, several methods have been used to improve the performance: data augmentation, transfer learning, class balancing, hyperparameter tuning and so on. The failure of these methods proved and confirmed the lack of an adequate number of data to train the models. However, despite the current limitation of the size of the dataset, simpler tests have shown the potentiality of the proposed approach.

6.3 Future developments

The overall results of the proposed method are still unsatisfactory, and the semantic segmentation procedure still needs further developments to improve the performance in the case if unseen scenario. The main limitations and bottlenecks of the procedure have been well identified, and in this last paragraph some possible future advancements are proposed for the dataset, for the image segmentation, and for the labels transfer to the point cloud.

Dataset. As repeated several times, increasing the number of the building of the dataset should be the first priority. This can be achieved by several methods. The acquisition and the processing of new building scenes according with the mentioned procedure is a direct way. Despite its simplicity, it requires several manual steps and time-consuming operations, including the photogrammetric acquisition, the point cloud generation, the point cloud processing and the final manual segmentation. However, it allows to increase the dataset with specific target buildings that could improve the class balance, or that could improve the performance on some specific categories that require a better accuracy. The second method could be the integration with existing datasets that share common features. Unfortunately, at this time, the ARCHdataset is the only dataset that would allow an integration. The freely available point cloud scenes are already labelled, and together with some related images, they could be easily used by the labelling projection procedure to generate new image ground truth. The third method to increase the images in the dataset could be the use of synthetic data generation. It has been discussed in paragraph (§5.7.1), and two strategies could be effective. The first is the use of Generative Adversarial Networks (GANs) using as training set the available dataset, or secondly, the generation of simulated images. As instance, in this second case the use of rendered images created from a 3D or a BIM model could be an efficient strategy. Once created, the model allows to generate various rendered images, with different lights or weather condition, with several materials or colours, changing systematically the main constructive elements. Therefore, the images could be different enough, providing a good generalization and variety among the scenes. However, their proper functioning in a real-world application should be carefully evaluated. The second issue concerning the dataset is the strong presence of the class background in the images. As it reported previously, it biased the training phase, and it made difficult the correct evaluation of the model. It could be addressed by adding the background buildings and elements at the already labelled point cloud. Subsequently, the background should be annotated according with the chosen categories, and a new ground should be generated. Nevertheless, any new acquisition and integration should consider the background and the building context.

Image Segmentation. The image segmentation is currently the issue of the procedure, and as shown in previous paragraph is the key point to obtain a good accuracy. Despite the main problem is data related, some future improvements could be done directly on the neural network developments. Currently, the segmentation block uses 3-channels RGB images as input. To improve the segmentation performance, adding an additional depth channel could be an effective strategy. The depth information could be easily available, since the input images are the result of a photogrammetric survey, and they are used to generate the point cloud. It could provide useful spatial and geometrical information during training, enabling the network to learn more complex dimensional relationships. Several RGB-D neural networks are available in literature, and they could be easily applied to the new dataset with a quite simple integration of the depth in the existing images. Alternatively, depth information could be used at the end of the segmentation pipeline to improve and fix the output segmentation map (Hoyer et al., 2021). In addition, several post-processing modules are available to improve the output map at the end of the segmentation pipeline. For instance, the authors in (Chopin et al., 2022) proposed a graph-based structural knowledge method to learn more complex relationship. The authors in (Dhawan et al., 2019) proposed the use of Conditional Random Fields (CRF) to achieve better clarity in segmented images, or the authors (X. Cheng & Liu, 2020) proposed a novel post-processing enhancement framework and a weighted composite filter to improve the segmentation mask. However, several methods have been proposed in the literature to post-process the segmentation mask, and they could be tested and applied to the output of the segmentation block, before using the output labels in the projection process. Concerning the semantic segmentation models, Deeplabv3+ is currently the most performing, but any new architecture could be tested in future developments. In addition, an interesting step forward could be the shift to instance segmentation, a higher level of segmentation in which each instance of the same category is labelled separately and detected with a bounding box. For instance, Mask-CNN is the most popular instance segmentation architecture. To enable the use of this network, the dataset should be integrated with a new level of annotation, and the bounding boxes should be added. An improvement of the labelling transfer procedure could easily automate this process. Finally, this dissertation did not consider the point-based approaches for the segmentation of the point cloud. However, it could be interesting to evaluate the performance of such approaches on the proposed dataset, since it includes the point clouds along with the images. In addition, such double data availability could allow the use of the technique of *ensembling*. It involves combining the output of multiple models to produce a more reliable final segmentation map. In this case an image-based and point-based model could be fused together to reduce errors and increase the overall robustness of the system.

Labelling projection. The labelling projection from images to point cloud showed a good performance, and a good robustness even in the case of low accuracy images. It is able to maintain the accuracy achieved on the images after the projection on the cloud. Therefore, it does not need remarkable improvement or developments. However, it is not still able to overcome the performance of the images, and future developments could be focused to address this aspect. As instance, some improvements could be done in the voting procedure, assigning a weight to each vote depending on several factors, such as the distance from the target point cloud, the angle of view, the quality of the segmentation maps of each single image, the point of view of the images, or the light and the exposure condition. In that way, the most probable labels could have more weight during the assignment, removing noise and reducing errors. However, as already mentioned previously, the performance is strictly related to the accuracy of the images, and future advances should be done in that section.

Finally, going a step ahead in the context of the digitization of heritage buildings and in the Scan-to-BIM, the required advancements of this work are the developments of well-structured workflows to proper manage the segmented point cloud in the CAD or BIM environment, and to use it to support the 3D model generation. Several works have been proposed to face this issue, and several tools are available in order to assist and help the modeling procedure. However, they still lack a good level of automation and accuracy, and currently a specific workflow for heritage building point clouds is still missing.

The main contributions and the contents of this thesis are mainly based on the material published in the following conference proceedings and journal articles:

Pellis, E., Murtiyoso, A., Masiero, A., Tucci, G., Betti, M., and Grussenmeyer, P. (2022). *2D to 3D Label Propagation for the Semantic Segmentation of Heritage Building Point Clouds*. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLIII-B2-2022, 861–867. https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-861-2022.

Pellis, E., Murtiyoso, A., Masiero, A., Tucci, G., Betti, M., and Grussenmeyer, P. (2022). *An image-based Deep Learning Workflow for 3D Heritage Point Cloud Semantic Segmentation*. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVI-2/W1-2022, 429–434. https://doi.org/10.5194/isprs-archives-XLVI-2-W1-2022-429-2022.

Murtiyoso, A., **Pellis, E.**, Grussenmeyer, P., Landes, T., Masiero, A. (2022). *Towards Semantic Photogrammetry: Generating Semantically Rich Point Clouds from Architectural Close-Range Photogrammetry*. Sensors. 22(3):966. https://doi.org/10.3390/s22030966.

Pellis, E., Masiero, A., Tucci, G., Betti, M., and Grussenmeyer, P. (2021). *Assembling an Image and Point Cloud Dataset for heritage Building Semantic Segmentation*. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVI-M-1-2021, 539–546, https://doi.org/10.5194/isprs-archives-XLVI-M-1-2021-539-2021.

Pellis, E., Masiero, A., Tucci, G., Betti, M., and Grussenmeyer, P. (2021) Towards an Integrated Design methodology for H-BIM. ARQUEOLOGICA 2.0 – 9th International Congress & 3rd GEORES. 10.4995/arqueologica9.2021.12158.

Bibliography

Adam, A., Chatzilari, E., Nikolopoulos, S., & Kompatsiaris, I. (2018). H-RANSAC: A Hybrid Point Cloud Segmentation Combining 2D and 3D Data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*(2), 1–8. https://doi.org/10.5194/isprs-annals-IV-2-1-2018

Adan, A., & Huber, D. (2011). 3D reconstruction of interior wall surfaces under occlusion and clutter. *Proceedings - 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT 2011*, 275–281. https://doi.org/10.1109/3DIMPVT.2011.42

AIA. (2007). *Integrated Project Delivery: A Guide*. onilne on: http://info.aia.org/siteobjects/files/ipd_guide_2007.pdf

Al-Durgham, M. M. (2019). *The Registration and Segmentation of Heterogeneous Laser Scanning Data*.

Alexandru Rosu, R., Schütt, P., Quenzel, J., & Behnke, S. (2020). LatticeNet: Fast Point Cloud Segmentation Using Permutohedral Lattices. *ArXiv*. https://doi.org/10.15607/rss.2020.xvi.006

Al-Rawabdeh, A., He, F., & Habib, A. (2020). Automated feature-based down-sampling approaches for fine registration of irregular point clouds. *Remote Sensing*, *12*(7). https://doi.org/10.3390/rs12071224

Andriasyan, M., Moyano, J., Nieto-Julián, J. E., & Antón, D. (2020). From point cloud data to Building Information Modelling: An automatic parametric workflow for heritage. *Remote Sensing*, *12*(7). https://doi.org/10.3390/rs12071094

Antón, D., Medjdoub, B., Shrahily, R., & Moyano, J. (2018). Accuracy evaluation of the semi-automatic 3D modeling for historical building information models. *International Journal of Architectural Heritage*, *12*(5), 790–805. https://doi.org/10.1080/15583058.2017.1415391

Antonello, M., Wolf, D., Prankl, J., Ghidoni, S., Menegatti, E., & Vincze, M. (2018). Multi-View 3D Entangled Forest for Semantic Segmentation and

Mapping. *Proceedings - IEEE International Conference on Robotics and Automation*, 1855–1862. https://doi.org/10.1109/ICRA.2018.8460837

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016). 3D Semantic Parsing of Large-Scale Indoor Spaces. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1534–1543. http://buildingparser.stanford.edu/

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615

Baik, A., Alitany, A., Boehm, J., & Robson, S. (2014). Jeddah historical building information modeling "JHBIM"-object library. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*(5), 41–47. https://doi.org/10.5194/isprsannals-II-5-41-2014

Bamler, R., Eineder, M., Adam, N., Zhu, X., & Gernhardt, S. (2009). Interferometric Potential of High Resolution Spaceborne SAR. *Photogrammetrie • Fernerkundung • Geoinformation*, *5*, 407–419. https://doi.org/10.1127/1432-8364/2009/00291

Bhanu, B., Lee, S. K., Ho, C. C., Henderson, T. C., & Bhanu, B. (1985). Range Data Processing: Representation of Surfaces by Edges. .

Błaszczak-Bąk, W. (2016). New optimum dataset method in LiDAR processing. *Acta Geodynamica et Geomaterialia*, *13*(4), 381–388. https://doi.org/10.13168/AGG.2016.0020

Boulch, A., Guerry, J., Le Saux, B., & Audebert, N. (2017). SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers and Graphics (Pergamon)*, *71*, 189–198. https://doi.org/10.1016/j.cag.2017.11.010

Boulch, A., Guerry, J., Le Saux, B., & Audebert, N. (2018). SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers and Graphics (Pergamon)*, *71*, 189–198. https://doi.org/10.1016/j.cag.2017.11.010

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(11), 1222–1239. https://doi.org/doi: 10.1109/34.969114.

Brostow, G. J., Fauqueur, J., & Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, *30*(2), 88–97. https://doi.org/10.1016/j.patrec.2008.04.005

Bruno, S., De Fino, M., & Fatiguso, F. (2018). Historic Building Information Modelling: performance assessment for diagnosis-aided information modelling and management. In *Automation in Construction* (Vol. 86, pp. 256–276). Elsevier B.V. https://doi.org/10.1016/j.autcon.2017.11.009

Castellano-Román, M., & Pinto-Puerto, F. (2019). Dimensions and Levels of Knowledge in Heritage Building Information Modelling, HBIM: The model of the Charterhouse of Jerez (Cádiz, Spain). *Digital Applications in Archaeology and Cultural Heritage*, *14*. https://doi.org/10.1016/j.daach.2019.e00110

Censi, A. (2008). An ICP variant using a point-to-line metric. *IEEE International Conference on Robotics and Automation*, 19–25. https://doi.org/doi: 10.1109/ROBOT.2008.4543181

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., & Yu, F. (2015). *ShapeNet: An Information-Rich 3D Model Repository*. http://arxiv.org/abs/1512.03012

Chen, D., Zhang, L., Mathiopoulos, P. T., & Huang, X. (2014). A methodology for automated segmentation and reconstruction of urban 3-D buildings from ALS point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(10), 4199–4217. https://doi.org/10.1109/JSTARS.2014.2349003

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *ArXiv*. http://arxiv.org/abs/1412.7062

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *ArXiv*. http://arxiv.org/abs/1606.00915

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation. *ArXiv*. http://arxiv.org/abs/1706.05587

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018, February 7). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. http://arxiv.org/abs/1802.02611

Chen, M., Hu, Q., Yu, Z., Thomas, H., Feng, A., Hou, Y., McCullough, K., Ren, F., & Soibelman, L. (2022). *STPLS3D: A Large-Scale Synthetic and Real Aerial Photogrammetry 3D Point Cloud Dataset*. http://arxiv.org/abs/2203.09065

Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2016). Multi-View 3D Object Detection Network for Autonomous Driving. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6526–6534. http://arxiv.org/abs/1611.07759

Chen, X. W., & Lin, X. (2014). Big data deep learning: Challenges and perspectives. In *IEEE Access* (Vol. 2, pp. 514–525). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2014.2325029

Chen, Y., & Medioni, G. (1991). Object modelling by registration of multiple range images. *1991 IEEE International Conference on Robotics and Automation*, 2724–2729. https://doi.org/doi: 10.1109/ROBOT.1991.132043.

Cheng, L., Chen, S., Liu, X., Xu, H., Wu, Y., Li, M., & Chen, Y. (2018). Registration of laser scanning point clouds: A review. In *Sensors (Switzerland)* (Vol. 18, Issue 5). MDPI AG. https://doi.org/10.3390/s18051641

Cheng, X., & Liu, H. (2020). A novel post-processing method based on a weighted composite filter for enhancing semantic segmentation results. *Sensors (Switzerland)*, *20*(19), 1–13. https://doi.org/10.3390/s20195500

Chiabrando, F., Lo Turco, M., & Rinaudo, F. (2017). Modeling the decay in an hbim starting from 3d point clouds. A followed approach for cultural heritage knowledge. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2W5), 605–612. https://doi.org/10.5194/isprs-archives-XLII-2-W5-605-2017

Choi, D., Shallue, C. J., Nado, Z., Lee, J., Maddison, C. J., & Dahl, G. E. (2019). On Empirical Comparisons of Optimizers for Deep Learning. *ArXiv*. http://arxiv.org/abs/1910.05446

Choi, S., Kim, T., & Yu, W. (2009). Performance evaluation of RANSAC family. *British Machine Vision Conference, BMVC 2009 - Proceedings*. https://doi.org/10.5244/C.23.81

Choy, C., Dong, W., & Koltun, V. (2020). Deep Global Registration. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2511–2520. http://arxiv.org/abs/2004.11540

Choy, C., Park, J., & Vladlen Koltun, P. (2019). Fully Convolutional Geometric Features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 8957–8965.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. *ArXiv*. http://arxiv.org/abs/1606.06650

Conti, A., Fiorini, L., Massaro, R., Santoni, C., & Tucci, G. (2022). HBIM for the preservation of a historic infrastructure: the Carlo III bridge of the Carolino Aqueduct. *Applied Geomatics*, *14*, 41–51. https://doi.org/10.1007/s12518-020-00335-2

Costantino, D., Pepe, M., & Restuccia, A. G. (2021). Scan-to-HBIM for conservation and preservation of Cultural Heritage building: the case study of San Nicola in Montedoro church (Italy). *Applied Geomatics*. https://doi.org/10.1007/s12518-021-00359-2

Croce, V., Caroti, G., De Luca, L., Jacquot, K., Piemonte, A., & Véron, P. (2021). From the Semantic Point Cloud to Heritage-Building Information Modeling: A Semiautomatic Approach Exploiting Machine Learning. *Remote Sensing*, *13*(3), 461. https://doi.org/10.3390/rs

Cursi, S., Martinelli, L., Paraciani, N., Calcerano, F., & Gigliarelli, E. (2022). Linking external knowledge to heritage BIM. In *Automation in Construction* (Vol. 141). Elsevier B.V. https://doi.org/10.1016/j.autcon.2022.104444

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., & Nießner, M. (2017). ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. *ArXiv*. http://arxiv.org/abs/1702.04405

Dai, A., Nießner, M., Dai, A., & Nießner, M. (2018). 3DMV: Joint 3D-Multi-View Prediction for 3D Semantic Scene Segmentation. *European Conference on Computer Vision*. https://github.com/angeladai/3DMV

Daniel Maturana, S. S. (2015). VoxNet: A 3D Convolutional Neural Network for Real-Time Object Detection. *IEEE/RSJ International Conference on Intelligent Robots and Sustems (IROS)*, 922–928. http://www.thepositiveencourager.global/the-mentoring-approach/

De Deuge, M., Quadros, A., Hung, C., & Douillard, B. (2013). Unsupervised Feature Learning for Classification of Outdoor 3D Scans. *Australasian Conference on Robotics and Automation*. http://www.acfr.usyd.edu.au/

De Luca, L. (2006). *Relevé et multi-représentations du patrimoine architectural Définition d'une approche hybride pour la reconstruction 3D d'édifices*. https://www.researchgate.net/publication/32228351

Decther R. (1986). Learning While Searching in Constrain-Satisfaction-Problems. *National Conference on Artificial Intelligence*. Vol 1.

Dekker, R. (2006). The importance of having data-sets. *Proceedings of the IATUL Conference*, *12*, 0. https://docs.lib.purdue.edu/iatul/2006/papers/16

D'Emilia, G., & Di Gasbarro, D. (2017). Review of techniques for 2D camera calibration suitable for industrial vision systems. *Journal of Physics: Conference Series*, *841*(1). https://doi.org/10.1088/1742-6596/841/1/012030

Deng, H., Birdal, T., & Ilic, S. (2018a). PPF-FoldNet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors. *ArXiv*. http://arxiv.org/abs/1808.10322

Deng, H., Birdal, T., & Ilic, S. (2018b). PPFNet: Global Context Aware Local Features for Robust 3D Point Matching. *ArXiv*. http://arxiv.org/abs/1802.02669

Deschaud, J.-E., & Goulette, F. (2010). A Fast and Accurate Plane Detection Algorithm for Large Noisy Point Clouds Using Filtered Normals and Voxel Growing. *3DPVT*. https://hal-mines-paristech.archives-ouvertes.fr/hal-01097361

Dhanachandra, N., Manglem, K., & Chanu, Y. J. (2015). Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm. *Procedia Computer Science*, *54*, 764–771. https://doi.org/10.1016/j.procs.2015.06.090

Dhawan, A., Bodani, P., & Garg, V. (2019). Post Processing of Image Segmentation using Conditional Random Fields. *6th International Conference on Computing for Sustainable Global Development,* 729–734.

Dore, C., & Murphy, M. (2013). Semi-automatic modelling of Building Facade with Shape Grammar using Historical Building Information Modelling. *International Archives of the Photogrammetry, Remote Sensing and Spatial*

Information Sciences. https://doi.org/10.5194/isprsarchives-XL-5-W1-57-2013

El-Sayed, E., Abdel-Kader, R. F., Nashaat, H., & Marei, M. (2018). Plane detection in 3D point cloud using octree-balanced density down-sampling and iterative adaptive plane extraction. *IET Image Processing*, *12*(9), 1595–1605. https://doi.org/10.1049/iet-ipr.2017.1076

Eppich, R., & Chabbi, A. (2007). *Recording, Documentation, and Information Management for the Conservation of Heritage Places: Illustrated Examples*. http://hdl.handle.net/10020/gci_pubs/recordim_vol2

European Commission. (2011). Commission Recommendation of 27 October 2011 on the digitisation and online accessibility of cultural material and digital preservation. *Official Journal of the European Union*, *L 283*(39).

European Commission. (2019). Basic principles and tips for 3D digitisation of tangible cultural heritage for cultural heritage professionals and institutions and other custodians of cultural heritage. *Official Journal of the European Union*.

European Commission. (2021). Commission Recommendation of 10 November 2021 on a common European data space for cultural heritage. *Official Journal of the European Union*, *L 401*(5). https://digital-strategy.ec.europa.eu/en/news/eu-member-states-sign-cooperate-digitising-cultural-heritage

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

Fai, S., Graham, K., Duckworth, T., Wood, N., & Attar, R. (2011). Building Information Modeling and Heritage Documentation. . http://www.210king.com/

Fiorucci, M., Khoroshiltseva, M., Pontil, M., Traviglia, A., Del Bue, A., & James, S. (2020). Machine Learning for Cultural Heritage: A Survey. *Pattern Recognition Letters*, *133*, 102–108. https://doi.org/10.1016/j.patrec.2020.02.017

Fontanelli, D., Ricciato, L., & Soatto, S. (2007). A Fast RANSAC-Based Registration Algorithm for Accurate Localization in Unknown Environments using LIDAR Measurements. *2007 IEEE International Conference on*

Automation Science and Engineering, 597–602. https://doi.org/10.1109/COASE.2007.4341827

Fu, K., Liu, S., Luo, X., & Wang, M. (2021). Robust Point Cloud Registration Framework Based on Deep Graph Matching. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)*, 8889–8898. http://arxiv.org/abs/2103.04256

Furukawa, Y., & Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(8), 1362–1376. https://doi.org/10.1109/TPAMI.2009.161

Gählert, N., Jourdan, N., Cordts, M., Franke, U., & Denzler, J. (2020). Cityscapes 3D: Dataset and Benchmark for 9 DoF Vehicle Detection. *ArXiv*. http://arxiv.org/abs/2006.07864

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets Robotics: The KITTI Dataset. *The International Journal of Robotic Research*, *32*(11). https://doi.org/https://doi.org/10.1177/0278364913491297

Genova, K., Yin, X., Kundu, A., Pantofaru, C., Cole, F., Sud, A., Brewington, B., Shucker, B., & Funkhouser, T. (2021, October 21). Learning 3D Semantic Segmentation with only 2D Image Supervision. *2021 International VìConference on 3D Vision (3DV)*. https://doi.org/10.1109/3dv53792.2021.00046

Gerdzhev, M., Razani, R., Taghavi, E., & Bingbing, L. (2021). TORNADO-Net: mulTiview tOtal vaRiatioN semAntic segmentation with Diamond inceptiOn module. *Proceedings - IEEE International Conference on Robotics and Automation*, *2021-May*, 9543–9549. https://doi.org/10.1109/ICRA48506.2021.9562041

Giel, B., & Issa, R. R. A. (2016). Framework for Evaluating the BIM Competencies of Facility Owners. *Journal of Management in Engineering*, *32*(1). https://doi.org/10.1061/(asce)me.1943-5479.0000378

Grant, D., Bethel, J., & Crawford, M. (2012). Point-to-plane registration of terrestrial laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, *72*, 16–26. https://doi.org/10.1016/j.isprsjprs.2012.05.007

Gregory, K., Groth, P., Cousijn, H., Scharnhorst, A., & Wyatt, S. (2019). Searching Data: A Review of Observational Data Retrieval Practices in Selected Disciplines. In *Journal of the Association for Information Science and*

Technology (Vol. 70, Issue 5, pp. 419–432). John Wiley and Sons Inc. https://doi.org/10.1002/asi.24165

Gressin, A., Mallet, C., Demantké, J. Ô., & David, N. (2013). Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. *ISPRS Journal of Photogrammetry and Remote Sensing*, *79,* 240–251. https://doi.org/10.1016/j.isprsjprs.2013.02.019

Grilli, E., & Remondino, F. (2020). Machine learning generalisation across different 3D architectural heritage. *ISPRS International Journal of Geo-Information*, *9*(6). https://doi.org/10.3390/ijgi9060379

Groß, J., Osep, A., & Leibe, B. (2019). AlignNet-3D: Fast Point Cloud Registration of Partially Observed Objects. *2019 International Conference on 3D Vision (3DV),* 623–632. http://arxiv.org/abs/1910.04668

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2019). Deep Learning for 3D Point Clouds: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43,* 4338–4364. http://arxiv.org/abs/1912.12033

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., & Pollefeys, M. (2017). Semantic3D.net: A new Large-scale Point Cloud Classification Benchmark. *ArXiv*. http://arxiv.org/abs/1704.03847

Hajian, H., Astani, S., & Becerik-Gerber, B. (2009). A Research Outlook for Real-time Project Information Management by Integrating Advanced Field Data Acquisition Systems and Building Information Modeling. .

Han, J., Shao, L., Xu, D., & Shotton, J. (2013). Enhanced computer vision with Microsoft Kinect sensor: A review. *IEEE Transactions on Cybernetics*, *43*(5), 1318–1334. https://doi.org/10.1109/TCYB.2013.2265378

Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., & Cipolla, R. (2015). SceneNet: Understanding Real World Indoor Scenes With Synthetic Data. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4077–4085. http://arxiv.org/abs/1511.07041

Hazirbas, C., Ma, L., Domokos, C., & Cremers, D. (2017). FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10111 LNCS*. https://doi.org/10.1007/978-3-319-54181-5_14

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Computer Science*. http://arxiv.org/abs/1703.06870

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. http://arxiv.org/abs/1512.03385

He, Y., Liang, B., Yang, J., Li, S., & He, J. (2017). An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features. *Sensors (Switzerland)*, *17*(8). https://doi.org/10.3390/s17081862

Hermans, A., Floros, G., & Leibe, B. (2014a). Dense 3D Semantic Mapping of Indoor Scenes from RGB-D Images. *IEEE International Conferenceon Robotics and Automation (ICRA)*.

Hermans, A., Floros, G., & Leibe, B. (2014b). Dense 3D Semantic Mapping of Indoor Scenes from RGB-D Images. *IEEE International Conferenceon Robotics and Automation (ICRA)*.

Hirschmüller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutua information. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, II*, 807–814. https://doi.org/10.1109/CVPR.2005.56

Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(2), 328–341. https://doi.org/10.1109/TPAMI.2007.1166

Hough, P. (1962). *Method and Means for Recognizing Complex Patterns*.

Hoyer, L., Zurich, E., Dai, D., Chen, Y., Köring, A., & Saha, S. (2021). Three Ways to Improve Semantic Segmentation with Self-Supervised Depth Estimation. *CVPR* . https://github.com/lhoyer/improving_segmentation_

Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K., & Zurich, E. (2021). PREDATOR: Registration of 3D Point Clouds with Low Overlap. *CVRP 2021*.

Hulik, R., Spanel, M., Smrz, P., & Materna, Z. (2014). Continuous plane detection in point-cloud data based on 3D Hough Transform. *Journal of Visual Communication and Image Representation*, *25*(1), 86–97. https://doi.org/10.1016/j.jvcir.2013.04.001

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *ArXiv*. http://arxiv.org/abs/1602.07360

Jadon, S. (2020). A survey of loss functions for semantic segmentation. *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 1–7. https://doi.org/10.1109/CIBCB48159.2020.9277638

Jain, A. N., & Nicholls, A. (2008). Recommendations for evaluation of computational methods. *Journal of Computer-Aided Molecular Design*, *22*(3–4), 133–139. https://doi.org/10.1007/s10822-008-9196-5

Jaritz, M., Gu, J., Su, H., Valeo, I. /, & Diego, S. (2019). Multi-view PointNet for 3D Scene Understanding. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 3995–4003. https://doi.org/10.1109/ICCVW.2019.00494

José López, F., Martin Lerones, P., Llamas, J., Gómez-García-Bermejo, J., & Zalama, E. (2017). Semi-automatic generation of bim models for cultural heritage. *International Journal of Heritage Architecture: Studies, Repairs and Maintence*, *2*(2), 293–302. https://doi.org/10.2495/ha-v2-n2-293-302

Kass, M., & Witkin, A. (1988). Snakes: Active Contour Models. In *International Journal of Computer Vision*. KIuwer Academic Publishers.

Kemp, A. (2014). *Why BIM Is So Important to Our Industry*. www.atkinsglobal.com/en-GB/angles/opinion/why-bim-is-so-important-to-our-industry

Kleene, S.C. (1956). Representation of Events in Nerve Nets and Finite Automata. *Annals of Mathematics Studies. No. 34. Princeton University Press*. pp. 3–41. Retrieved 17 June 2017.

Koesten, L., Simperl, E., Blount, T., Kacprzak, E., & Tennison, J. (2020a). Everything you always wanted to know about a dataset: Studies in data summarisation. *International Journal of Human Computer Studies*, *135*. https://doi.org/10.1016/j.ijhcs.2019.10.004

Koesten, L., Simperl, E., Blount, T., Kacprzak, E., & Tennison, J. (2020b). Everything you always wanted to know about a dataset: Studies in data summarisation. *International Journal of Human Computer Studies*, *135*. https://doi.org/10.1016/j.ijhcs.2019.10.004

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, *25*(2). https://doi.org/10.1145/3065386

Kundu, A., Yin, X., Fathi, A., Ross, D., Brewington, B., Funkhouser, T., & Pantofaru, C. (2020, July 26). Virtual Multi-view Fusion for 3D Semantic Segmentation. *European Conference on Computer Vision*. http://arxiv.org/abs/2007.13138

Kwok, T. H. (2019). DNSS: Dual-Normal-Space Sampling for 3-D ICP Registration. *IEEE Transactions on Automation Science and Engineering*, *16*(1), 241–252. https://doi.org/10.1109/TASE.2018.2802725

Labussiere, M., Laconte, J., & Pomerleau, F. (2018). Geometry Preserving Sampling Method based on Spectral Decomposition for 3D Registration. *ArXiv*. http://arxiv.org/abs/1810.01666

Lachat, E., Macher, H., Mittet, M. A., Landes, T., & Grussenmeyer, P. (2015). First experiences with kinect V2 sensor for close range 3D modelling. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *40*(5W4), 93–100. https://doi.org/10.5194/isprsarchives-XL-5-W4-93-2015

Lai, K., Bo, L., Ren, X., & Fox, D. (2011). A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. *2011 IEEE International Conference on Robotics and Automation*, 1817–1824. https://doi.org/10.1109/ICRA.2011.5980382.

Lamas, A., Tabik, S., Cruz, P., Montes, R., Martínez-Sevilla, Á., Cruz, T., & Herrera, F. (2021). MonuMAI: Dataset, deep learning pipeline and citizen science based app for monumental heritage taxonomy and classification. *Neurocomputing*, *420*, 266–280. https://doi.org/10.1016/j.neucom.2020.09.041

Lawin, F. J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F. S., & Felsberg, M. (2017a). *Deep Projective 3D Semantic Segmentation*. http://arxiv.org/abs/1705.03428

Lawin, F. J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F. S., & Felsberg, M. (2017b, May 9). Deep Projective 3D Semantic Segmentation. *International Conference on Computer Analysis of Images and Patterns*. http://arxiv.org/abs/1705.03428

LeCun, Y., Boser, B., Denker, J. S., & Henderson, D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. .

Lertniphonphan, K., Satoshi, K., Kazuyuki, T., & Hiromasa, Y. (2018). 2D to 3D Label Propagation for Object Detection in Point Cloud. *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 1–6.

Li, P., Wang, R., Wang, Y., & Tao, W. (2020). Evaluation of the ICP Algorithm in 3D Point Cloud Registration. *IEEE Access*, *8*, 68030–68048. https://doi.org/10.1109/ACCESS.2020.2986470

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2016). Feature Pyramid Networks for Object Detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944. http://arxiv.org/abs/1612.03144

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2014). *Microsoft COCO: Common Objects in Context*. http://arxiv.org/abs/1405.0312

Lin, Y. J., Benziger, R. R., & Habib, A. (2016). Planar-based adaptive down-sampling of point clouds. *Photogrammetric Engineering and Remote Sensing*, *82*(12), 955–966. https://doi.org/10.14358/PERS.82.12.955

Linning, C. (2014). *BIM – An Information Resource for Future Generations*. http://brisbim.com/ wp-content/uploads/2014/12/BrisBIMx-Sydney-Opera-House-Presentation-FINAL.pdf.

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 IEEE International Conference on Computer Vision*, *2017-Octob*, 5679–5688. https://doi.org/10.1109/ICCV.2017.605

Liu, W., Rabinovich, A., & Berg, A. C. (2015). ParseNet: Looking Wider to See Better. *ArXiv*. http://arxiv.org/abs/1506.04579

Liu, A., Hu, N., Song, D., Guo, F., Zhou, H., Hao, T. (2019). Multi-View Hierarchical Fusion Network for 3D Object Retrieval and Classification. *IEEE Access,* pp. (99):1 – 1. 10.1109/ACCESS.2019.2947245

Llamas, J., Lerones, P. M., Medina, R., Zalama, E., & Gómez-García-Bermejo, J. (2017). Classification of architectural heritage images using deep learning techniques. *Applied Sciences (Switzerland)*, *7*(10). https://doi.org/10.3390/app7100992

Long, J., Shelhamer, E., & Darrell, T. (2014). Fully Convolutional Networks for Semantic Segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431–3440. http://arxiv.org/abs/1411.4038

Ma, Y. De, Liu, Q., & Qian, Z. B. (2004). Automated image segmentation using improved PCNN model based on cross-entropy. *2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, ISIMP 2004*, 743–746. https://doi.org/10.1109/isimp.2004.1434171

Macher, H., Boudhaim, M., Grussenmeyer, P., Siroux, M., & Landes, T. (2019). Combination of Thermal and Geometric Information for BIM Enrichment. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2/W15), 719–725. https://doi.org/10.5194/isprs-archives-XLII-2-W15-719-2019

Macher, H., Landes, T., & Grussenmeyer, P. (2015). Point clouds segmentation as base for as-built BIM creation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*(5W3), 191–197. https://doi.org/10.5194/isprsannals-II-5-W3-191-2015

Maietti, F., Di Giulio, R., Balzani, M., Piaia, E., Medici, M., & Ferrari, F. (2018). 3D Data Acquisition and Modelling of Complex Heritage Buildings. *Digital Cultural Heritage*, 1–13. https://doi.org/10.1007/978-3-319-75826-8_1

Makovetskii, A., Voronin, S., Kober, V., & Tihonkih, D. (2017). Affine registration of point clouds based on point-to-plane approach. *Procedia Engineering*, *201*, 322–330. https://doi.org/10.1016/j.proeng.2017.09.635

Malinverni, E. S., Pierdicca, R., Paolanti, M., Martini, M., Morbidoni, C., Matrone, F., & Lingua, A. (2019). Deep Learning for Semantic Segmentation of 3D Point Cloud. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2/W15), 735–742. https://doi.org/10.5194/isprs-archives-XLII-2-W15-735-2019

Mascaro, R., Teixeira, P., Teixeira, L., & Chli, M. (2021). Diffuser: Multi-View 2D-to-3D Label Diffusion for Semantic Scene Segmentation. *IEEE International Conference OnRobotics and Automation (ICRA)*. https://doi.org/10.3929/ethz-b-000484229

Matrone, F., Lingua, A., Pierdicca, R., Malinverni, E. S., Paolanti, M., Grilli, E., Remondino, F., Murtiyoso, A., & Landes, T. (2020). A Benchmark for Large-Scale Heritage Point Cloud Semanti Segmentation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences -*

ISPRS Archives, *43*(B2), 1419–1426. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1419-2020

Mattoccia, S., & Poggi, M. (2015). A passive RGBD sensor for accurate and real-time depth sensing self-contained into an FPGA. *ACM International Conference Proceeding Series*, *08-11-Sep-2015*, 146–151. https://doi.org/10.1145/2789116.2789148

McCormac, J., Handa, A., Davison, A., & Leutenegger, S. (2016a, September 16). SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. *IEEE International Conferenceon Robotics and Automation (ICRA)*. http://arxiv.org/abs/1609.05130

McCormac, J., Handa, A., Davison, A., & Leutenegger, S. (2016b, September 16). SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. *IEEE International Conferenceon Robotics and Automation (ICRA)*. http://arxiv.org/abs/1609.05130

McCulloch, W., Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. Bullettin of Mathematical Biophysics, 5(4), 115-133. 10.1007/BF02478259

Milioto, A., Vizzo, I., Behley, J., & Stachniss, C. (2019). RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. *IEEE International Conference on Intelligent Robots and Systems*, *September 2020*, 4213–4220. https://doi.org/10.1109/IROS40897.2019.8967762

Milletari, F., Navab, N., & Ahmadi, S. A. (2016). V-Net: Fully convolutional neural networks for volumetric medical image segmentation. *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, 565–571. https://doi.org/10.1109/3DV.2016.79

Murphy, M., Mcgovern, E., & Pavia, S. (2009). Historic building information modelling (HBIM). *Structural Survey*, *27*(4), 311–327. https://doi.org/10.1108/02630800910985108

Murtiyoso, A., Lhenry, C., Landes, T., Grussenmeyer, P., & Alby, E. (2021). Semantic segmentation for building façade 3D point cloud from 2D orthophoto images using transfer learning. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *43*(B2-2021), 201–206. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-201-2021

Murtiyoso, A., Pellis, E., Grussenmeyer, P., Landes, T., & Masiero, A. (2022). Towards Semantic Photogrammetry: Generating Semantically Rich Point Clouds from Architectural Close-Range Photogrammetry. *Sensors*, *22*(3). https://doi.org/10.3390/s22030966

Nex, F., & Remondino, F. (2014). UAV for 3D mapping applications: A review. In *Applied Geomatics* (Vol. 6, Issue 1, pp. 1–15). Springer Verlag. https://doi.org/10.1007/s12518-013-0120-x

Nieto, J. E., Moyano, J. J., Rico Delgado, F., & Antón García, D. (2016). Management of built heritage via HBIM Project: A case of study of flooring and tiling. *Virtual Archaeology Review*, *7*(14), 1. https://doi.org/10.4995/var.2016.4349

Ning, X., Zhang, X., Wang, Y., & Jaeger, M. (2009). Segmentationof Architecture Shape Information from 3D Point Cloud. *VRCAI 2009*, 374.

Nock, R., & Nielsen, F. (2004). Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(11), 1452–1458. https://doi.org/10.1109/TPAMI.2004.110

Noh, H., Hong, S., & Han, B. (2015). Learning Deconvolution Network for Semantic Segmentation. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1520–1528. http://arxiv.org/abs/1505.04366

Oreni, D., Brumana, R., Della Torre, S., Banfi, F., Barazzetti, L., & Previtali, M. (2014). Survey turned into HBIM: The restoration and the work involved concerning the Basilica di Collemaggio after the earthquake (L'Aquila). *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*(5), 267–273. https://doi.org/10.5194/isprsannals-II-5-267-2014

Osello, A., Lucibello, G., & Morgagni, F. (2018). HBIM and virtual tools: A new chance to preserve architectural heritage. *Buildings*, *8*(1). https://doi.org/10.3390/buildings8010012

Pan, R., & Taubin, G. (2016). Automatic segmentation of point clouds from multi-view reconstruction using graph-cut. *Visual Computer*, *32*(5), 601–609. https://doi.org/10.1007/s00371-015-1076-0

Pang, G., & Neumann, U. (2016). 3D point cloud object detection with multi-view convolutional neural network. *Proceedings - International Conference on Pattern Recognition*, *0*, 585–590. https://doi.org/10.1109/ICPR.2016.7899697

Papadakis, P. (2017). A Use-Case Study on Multi-View Hypothesis Fusion for 3D Object Classification. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2446–2452. https://doi.org/10.1109/ICCVW.2017.288.

Pepe, M., Costantino, D., & Garofalo, A. R. (2020). An efficient pipeline to obtain 3D model for HBIM and structural analysis purposes from 3D point clouds. *Applied Sciences (Switzerland)*, *10*(4). https://doi.org/10.3390/app10041235

Pətrəucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I., & Haas, C. (2015). State of research in automatic as-built modelling. *Advanced Engineering Informatics*, *29*(2), 162–171. https://doi.org/10.1016/j.aei.2015.01.001

Plath, N., Toussaint, M., & Nakajima, S. (2009). Multi-class image segmentation using Conditional Random Fields and Global Classification. *INternational Conference of Machine Learning*.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 77–85. https://doi.org/10.1109/CVPR.2017.16

Qi, C. R., Su, H., Niessner, M., Dai, A., Yan, M., & Guibas, L. J. (2016). Volumetric and Multi-View CNNs for Object Classification on 3D Data. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5648–5656. https://doi.org/10.1109/CVPR.2016.609

Qin, N., Hu, X., Dai, H., & Hu, X. (2018). Deep fusion of multi-view and multimodal representation of ALS point cloud for 3D terrain scene recognition. *ISPRS Journal of Photogrammetry and Remote Sensing*, *143*, 205–212. https://doi.org/10.1016/j.isprsjprs.2018.03.011

Qin, R., Tian, J., & Reinartz, P. (2016). 3D change detection – Approaches and applications. In *ISPRS Journal of Photogrammetry and Remote Sensing* (Vol. 122, pp. 41–56). Elsevier B.V. https://doi.org/10.1016/j.isprsjprs.2016.09.013

Quan, L., Wang, J., Tan, P., & Yuan, L. (2007). Image-based modeling by joint segmentation. *International Journal of Computer Vision*, *75*(1), 135–150. https://doi.org/10.1007/s11263-007-0044-1

Quattrini, R., Malinverni, E. S., Clini, P., Nespeca, R., & Orlietti, E. (2015). From tls to hbim. high quality semantically-aware 3d modeling of complex architecture. *International Archives of the Photogrammetry, Remote Sensing* *and Spatial Information Sciences - ISPRS Archives*, *40*(5W4), 367–374. https://doi.org/10.5194/isprsarchives-XL-5-W4-367-2015

Rabbani, T., Van Den Heuvel, F. A., & Vosselman, G. (2006). *Segmentation of Point Clouds using Smoothness Constrant*.

Raguram, R., Frahm, J.-M., & Pollefeys, M. (2008). A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. *European Conference on Computer Vision*, 500–513. https://doi.org/doi.org/10.1007/978-3-540-88688-4_37

Rashdi, R., Martínez-Sánchez, J., Arias, P., & Qiu, Z. (2022). Scanning Technologies to Building Information Modelling: A Review. In *Infrastructures* (Vol. 7, Issue 4). MDPI. https://doi.org/10.3390/infrastructures7040049

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*, 1137–1149. http://arxiv.org/abs/1506.01497

Riegler, G., Ulusoy, A. O., & Geiger, A. (2017). OctNet: Learning deep 3D representations at high resolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 6620–6629. https://doi.org/10.1109/CVPR.2017.701

Riemenschneider, H., Bódis-Szomorú, A., Weissenberg, J., & Van Gool, L. (2014). Learning where to classify in multi-view semantic segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8693 LNCS*(PART 5), 516–532. https://doi.org/10.1007/978-3-319-10602-1_34

Robert, D., Vallet, B., & Landrieu, L. (2022). Learning Multi-View Aggregation In the Wild for Large-Scale 3D Semantic Segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5565–5574. http://arxiv.org/abs/2204.07548

Rocha, Mateus, Fernández, & Ferreira. (2020). A Scan-to-BIM Methodology Applied to Heritage Buildings. *Heritage*, *3*(1), 47–67. https://doi.org/10.3390/heritage3010004

Rolin, R., Antaluca, E., Batoz, J. L., Lamarque, F., & Lejeune, M. (2019). From point cloud data to structural analysis through a geometrical hBIM-oriented model. *Journal on Computing and Cultural Heritage*, *12*(2). https://doi.org/10.1145/3242901

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *ArXiv*. http://arxiv.org/abs/1505.04597

Rosenblatt, F. (1957). The Perceptron - a perceiving and recognizing automaton. *Report 85-460-1*. Cornell Aeronautical Laboratory.

Roynard, X., Deschaud, J.-E., & Goulette, F. (2017). *Paris-Lille-3D: a large and high-quality ground truth urban point cloud dataset for automatic segmentation and classification*. http://arxiv.org/abs/1712.00032

Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*, 145–152. https://doi.org/10.1109/IM.2001.924423

Saleh Al-amri, S., & Kalyankar, N. (2010). Image Segmentation by Using Thershod Techniques. *ArXiv*, *2*. https://doi.org/https://doi.org/10.48550/arXiv.1005.4020

Salvi, J., Armanguã, X., & Batlle, J. (2002). A comparative review of camera calibrating methods with accuracy evaluation. In *Pattern Recognition* (Vol. 35). www.elsevier.com/locate/patcog

Sampath, A., & Shan, J. (2010). Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, *48*(3 PART2), 1554–1567. https://doi.org/10.1109/TGRS.2009.2030180

Sappa, A. D., & Devy, M. (2001). Fast range image segmentation by an edge detection strategy. *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*, *2001-January*, 292–299. https://doi.org/10.1109/IM.2001.924460

Sarode, V., Li, X., Goforth, H., Aoki, Y., Srivatsan, R. A., Lucey, S., & Choset, H. (2019). PCRNet: Point Cloud Registration Network using PointNet Encoding. *ArXiv*. http://arxiv.org/abs/1908.07906

Schmitt, M., Shahzad, M., & Zhu, X. X. (2015). Reconstruction of individual trees from multi-aspect TomoSAR data. *Remote Sensing of Environment*, *165*, 175–185. https://doi.org/10.1016/j.rse.2015.05.012

Segal, A. V, Haehnel, D., & Thrun, S. (2009a). *Generalized-ICP*.

Segal, A. V, Haehnel, D., & Thrun, S. (2009b). Generalized-ICP. *Robotics: Science and Systems 2009*. https://doi.org/10.15607/RSS.2009.V.021

Shahzad, M., Schmitt, M., & Zhu, X. X. (2015). Segmentation and crown parameter extraction of individual trees in an airborne TomoSAR point cloud. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *40*(3W2), 205–209. https://doi.org/10.5194/isprsarchives-XL-3-W2-205-2015

Shahzad, M., Zhu, X. X., & Bamler, R. (2012). Façade structure reconstruction using spaceborne TomoSAR point clouds. *International Geoscience and Remote Sensing Symposium (IGARSS),* 467–470. https://doi.org/10.1109/IGARSS.2012.6351385

Shi, Y., Zhu, X. X., & Bamler, R. (2019). Nonlocal Compressive Sensing-Based SAR Tomography. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(5), 3015–3024. https://doi.org/10.1109/TGRS.2018.2879382

Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2013). Indoor Segmentation and Support Inference from RGBD Images. *European Conference on Computer Vision*, 746–760. https://doi.org/doi.org/10.1007/978-3-642-33715-4_54

Simeone, D., Cursi, S., & Acierno, M. (2019). BIM semantic-enrichment for built heritage representation. *Automation in Construction*, *97*, 122–137. https://doi.org/10.1016/j.autcon.2018.11.004

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR Abs/1409.1556*. https://doi.org/https://doi.org/10.48550/arXiv.1409.1556

Snavely, N., Seitz, S. M., & Szeliski, R. (2008). Modeling the world from Internet photo collections. *International Journal of Computer Vision*, *80*(2), 189–210. https://doi.org/10.1007/s11263-007-0107-3

Snavely, N., Seitz, S. M., Szeliski, R., & Research, M. (2006). Photo Tourism: Exploring Photo Collections in 3D. *SIGGRAPH 2006*, 835–846. www.cc.gatech.edu/4d-cities

Song, S., Lichtenberg, S. P., & Xiao, J. (2015). *SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite*. http://rgbd.cs.princeton.edu

Starck, J. L., Elad, M., & Donoho, D. L. (2005). Image decomposition via the combination of sparse representations and a variational approach. *IEEE*

Transactions on Image Processing, *14*(10), 1570–1582. https://doi.org/10.1109/TIP.2005.852206

Stathopoulou, E. K., & Remondino, F. (2019). Semantic Photogrammetry - Boosting Image-based 3D Reconstruction with Semantic Labeling. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *42*(2/W9), 685–690. https://doi.org/10.5194/isprs-archives-XLII-2-W9-685-2019

Stober, D., Žarnić, R., Penava, D., Turkalj Podmanicki, M., & Virgej-Đurašević, R. (2018). Application of HBIM as a Research Tool for Historical Building Assessment. *Civil Engineering Journal*, *4*(7), 1565. https://doi.org/10.28991/cej-0309195

Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M. H., & Kautz, J. (2018). SPLATNet: Sparse Lattice Networks for Point Cloud Processing. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2530–2539. https://doi.org/10.1109/CVPR.2018.00268

Sztwiertnia, D., Ochałek, A., Tama, A., & Lewińska, P. (2021). HBIM (heritage Building Information Modell) of the Wang Stave Church in Karpacz–Case Study. *International Journal of Architectural Heritage*, *15*(5), 713–727. https://doi.org/10.1080/15583058.2019.1645238

Tang, P., Huber, D., Akinci, B., Lipman, R., & Lytle, A. (2010). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. In *Automation in Construction* (Vol. 19, Issue 7, pp. 829–843). Elsevier B.V. https://doi.org/10.1016/j.autcon.2010.06.007

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., Hough-Transform, P. G., Ransac, E., Tarsha-Kurdi, F., Landes, T., & Grussenmeyer, P. (2007). *Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data. ISPRS Workshop on Laser Scanning*. https://shs.hal.science/halshs-00264843

Thomson, C., & Boehm, J. (2015). Automatic geometry generation from point clouds for BIM. *Remote Sensing*, *7*(9), 11753–11775. https://doi.org/10.3390/rs70911753

Tóvári, D., & Pfeifer, N. (2005). Segmentation Based Robust Interpolation - A New Approach to Laser Data Filtering. .

Tyleček, R., & Radimšára, R. R. (2013). Spatial Pattern Templates for Recognition of Objects with Regular Structure. *Computer Science*, *8142*. https://doi.org/doi.org/10.1007/978-3-642-40602-7_39

Ulku, I., & Akagunduz, E. (2019). *A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D images*. http://arxiv.org/abs/1912.10230

Vineet, V., Miksik, O., Lidegaard, M., Nießner, M., Golodetz, S., Prisacariu, V. A., Kähler, O., Murray, D. W., Izadi, S., Pérez, P., & Torr, P. H. S. (2015). Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. *Proceedings - IEEE International Conference on Robotics and Automation*, *2015-June*(June), 75–82. https://doi.org/10.1109/ICRA.2015.7138983

Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings - Literature review and future needs. In *Automation in Construction* (Vol. 38, pp. 109–127). https://doi.org/10.1016/j.autcon.2013.10.023

Weinmann, M., Urban, S., Hinz, S., Jutzi, B., Mallet, C. (2015). Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. *Computer & Graphics*, 49, 47-57. https://doi.org/10.1016/j.cag.2015.01.006

Weinmann, M., Jutzi, B., Mallet, C. (2014). Semantic 3d scene interpretation: a framework combining optimal neighbourhood size selection with relevant features. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* II-3, pp. 181–188.

Waechter, M., Moehrle, N., & Goesele, M. (2014). Let There Be Color! Large-Scale Texturing of 3D Reconstructions. *European Conference on Computer Vision*. https://doi.org/10.1007/978-3-319-10602-1_54

Wang, B. H., Chao, W. L., Wang, Y., Hariharan, B., Weinberger, K. Q., & Campbell, M. (2019). LDLS: 3-D Object Segmentation Through Label Diffusion from 2-D Images. *IEEE Robotics and Automation Letters*, *4*(3), 2902–2909. https://doi.org/10.1109/LRA.2019.2922582

Wang, C., Pelillo, M., & Siddiqi, K. (2019). Dominant Set Clustering and Pooling for Multi-View 3D Object Recognition. *ArXiv*. https://doi.org/https://doi.org/10.48550/arXiv.1906.01592

Wang, Q., Fu, L., & Liu, Z. (2010). Review on camera calibration. *2010 Chinese Control and Decision Conference, CCDC 2010*, 3354–3358. https://doi.org/10.1109/CCDC.2010.5498574

Wang, R., Hammoudi, K., Pylvänäinen, T., Roimela, K., Vedantham, R., Itäranta, J., Wang NAVTEQ, R., & Grzeszczuk, R. (2010). Automatic Alignment and Multi-View Segmentation of Street View Data using 3D Shape Priors. *Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*.

Wang, W., Zhang, M., Chen, G., Jagadish, H. V, Ooi, B. C., & Tan, K.-L. (2016). Database Meets Deep Learning: Challenges and Opportunities. *SIGMOD Record*, *45*(2).

Wang, Y., Ji, R., & Chang, S. F. (2013). Label propagation from imagenet to 3D point clouds. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3135–3142. https://doi.org/10.1109/CVPR.2013.403

Wani, M. A. (2003). Parallel Edge-Region-Based Segmentation Algorithm Targeted at Reconfigurable MultiRing Network. In *The Journal of Supercomputing* (Vol. 25).

Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012). "Structure-from-Motion" photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, *179*, 300–314. https://doi.org/10.1016/j.geomorph.2012.08.021

Xiao, J., Owens, A., & Torralba, A. (2013a). SUN3D: A database of big spaces reconstructed using SfM and object labels. *Proceedings of the IEEE International Conference on Computer Vision*, 1625–1632. https://doi.org/10.1109/ICCV.2013.458

Xiao, J., Owens, A., & Torralba, A. (2013b). SUN3D: A database of big spaces reconstructed using SfM and object labels. *Proceedings of the IEEE International Conference on Computer Vision*, 1625–1632. https://doi.org/10.1109/ICCV.2013.458

Xiao, J., & Quan, L. (2009). Multiple View Semantic Segmentation for Street View Images. *2009 IEEE 12th International Conference on Computer Vision*, 686–693. https://doi.org/10.1109/ICCV.2009.5459249

Xiao, J., Wang, J., Tan, P., & Quan, L. (2007). Joint Affinity Propagation for Multiple View Segmentation. *IEEE 11th International Conference on Computer Vision, 2007 ICCV 2007 ; 14-21 Oct. 2007, Rio de Janeiro, Brazil*.

Xiao, J., Zhang, J., Adler, B., Zhang, H., & Zhang, J. (2013). Three-dimensional point cloud plane segmentation in both structured and unstructured environments. *Robotics and Autonomous Systems*, *61*(12), 1641–1652. https://doi.org/10.1016/j.robot.2013.07.001

Xie, Y., Tian, J., & Zhu, X. X. (2020). Linking Points with Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geoscience and Remote Sensing Magazine, 8*(4), 38–59. https://doi.org/10.1109/MGRS.2019.2937630

Yang, J., Li, H., Campbell, D., & Jia, Y. (2016). Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *ArXiv*. https://doi.org/10.1109/TPAMI.2015.2513405

Yu, C., & Ju, D. Y. (2018). A maximum feasible subsystem for globally optimal 3D point cloud registration. *Sensors (Switzerland)*, *18*(2). https://doi.org/10.3390/s18020544

Yu, Q., Yang, C., Fan, H., & Wei, H. (2020). Latent-MVCNN: 3D Shape Recognition Using Multiple Views from Pre-defined or Random Viewpoints. *Neural Processing Letters*, *52*(1), 581–602. https://doi.org/10.1007/s11063-020-10268-x

Yu, T., Meng, J., & Yuan, J. (2018). Multi-view Harmonized Bilinear Network for 3D Object Recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 186–194. https://doi.org/doi: 10.1109/CVPR.2018.00027.

Yuan, W., Eckart, B., Kim, K., Jampani, V., Fox, D., & Kautz, J. (2020). DeepGMR: Learning Latent Gaussian Mixture Models for Registration. *ArXiv*. http://arxiv.org/abs/2008.09088

Zhang, J., Zhao, X., Chen, Z., & Lu, Z. (2019). A Review of Deep Learning-Based Semantic Segmentation for Point Cloud. *IEEE Access*, *7*, 179118–179133. https://doi.org/10.1109/ACCESS.2019.2958671

Zhang, R., Li, G., Li, M., & Wang, L. (2018). Fusion of images and point clouds for the semantic segmentation of large-scale 3D scenes based on deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, *143*, 85–96. https://doi.org/10.1016/j.isprsjprs.2018.04.022

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid Scene Parsing Network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 6230–6239. http://arxiv.org/abs/1612.01105

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., & Torralba, A. (2016). Semantic Understanding of Scenes through the ADE20K Dataset. *ArXiv*. http://arxiv.org/abs/1608.05442

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). *UNet++: A Nested U-Net Architecture for Medical Image Segmentation*. http://arxiv.org/abs/1807.10165

Zhu, X. X., & Bamler, R. (2010). Very high resolution spaceborne SAR tomography in urban environment. *IEEE Transactions on Geoscience and Remote Sensing*, *48*(12), 4296–4308. https://doi.org/10.1109/TGRS.2010.2050487

Zhu, X. X., & Shahzad, M. (2014a). Facade reconstruction using multiview spaceborne TomoSAR point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, *52*(6), 3541–3552. https://doi.org/10.1109/TGRS.2013.2273619

Zhu, X. X., & Shahzad, M. (2014b). Facade reconstruction using multiview spaceborne TomoSAR point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, *52*(6), 3541–3552. https://doi.org/10.1109/TGRS.2013.2273619

Appendix

Résumé en français

Segmentation sémantique des nuages de points du patrimoine bâti : une approche multi-vues

1. Introduction

1.1 Numérisation du patrimoine culturel

Au cours des dernières années, la numérisation assistée par ordinateur s'est imposée comme une technologie puissante pour améliorer la documentation et la préservation du patrimoine culturel, en produisant de nouvelles formes de connaissance et des niveaux de compréhension plus profonds. Les technologies numériques ouvrent de nouvelles perspectives à la société, en offrant au public davantage de moyens d'accéder, de découvrir, d'explorer et d'apprécier les biens culturels, et de possibilités de réutiliser les biens culturels pour des services et des produits innovants et créatifs dans divers secteurs. De nos jours, le développement de technologies numériques avancées, telles que la modélisation 3D, l'intelligence artificielle, l'informatique en nuage, la réalité virtuelle et augmentée, a ouvert de nouvelles perspectives en matière de numérisation, d'accès en ligne et de conservation numérique. Considérant les perspectives offertes par la numérisation, la Commission européenne a publié une série de recommandations sur un espace européen commun de données pour le patrimoine culturel. L'objectif est d'accélérer la numérisation de tous les monuments et sites, objets et artefacts du patrimoine culturel pour les générations futures, de protéger et de préserver ceux qui sont en danger, et de stimuler leur réutilisation dans des domaines tels que l'éducation, le tourisme durable et les secteurs culturels créatifs (Commission européenne, 2011). En outre, la Commission encourage les États membres de l'UE à numériser d'ici 2030 tous les monuments et sites qui risquent de se dégrader et la moitié de ceux qui sont très fréquentés par les touristes (Commission européenne, 2019). Cette thèse s'inscrit dans le contexte de la numérisation du patrimoine culturel et du besoin émergent et croissant de définir des normes, des procédures et des flux de travail pour contribuer de manière opérationnelle à la conservation, à la protection et à la diffusion du patrimoine culturel dans le monde par le biais des technologies numériques.

1.2 Motivation et défis

Ces dernières années, la modélisation des données du bâtiment (BIM) a commencé à jouer un rôle important dans la gestion et la documentation du patrimoine culturel, et un nouveau paradigme de méthodologie de conception a été établi, la *modélisation des données du patrimoine bâti* (H-BIM). Il s'agit d'une représentation numérique d'un bâtiment existant à l'heure actuelle (tel que construit), qui comprend un large éventail d'informations telles que la géométrie, les matériaux, les systèmes technologiques, les quantités, les performances, la documentation, les informations sur la maintenance et bien d'autres encore. Plusieurs travaux récents ont montré que cette méthodologie de conception s'est avérée être un outil très puissant pour la numérisation des bâtiments patrimoniaux, en prouvant l'efficacité de cette procédure de modélisation dans un large éventail d'applications conformes aux principes des lignes directrices européennes. La chaine de traitement pour la création d'un modèle conforme à l'exécution est appelé Scan-to-BIM. Ce processus global comprend toutes les étapes depuis l'acquisition jusqu'à la phase de modélisation. Cependant, la reconstruction virtuelle de modèles conformes à l'exécution est une question ouverte dans le monde réel et les applications à grande échelle, et elle présente encore un certain nombre de problèmes et de défis. Actuellement, l'un des principaux problèmes du processus Scan-to-BIM est la gestion des données à grande échelle résultant de la campagne d'acquisition. Le haut niveau de détail et d'automatisation atteint par les dernières technologies d'acquisition, comme le scanner laser 3D ou la photogrammétrie, permet de collecter une grande quantité de données en peu de temps avec une précision impressionnante, mais le traitement correct de ces données reste une procédure difficile.

Parmi les diverses opérations de traitement Scan-to-BIM, *la segmentation sémantique des nuages de points* est l'une des opérations les plus difficiles, et son automatisation présenterait de nombreux avantages. Elle consiste à diviser les données du nuage de points en segments plus petits et significatifs, et à attribuer à chaque segment une étiquette ou une catégorie en fonction des objets présents dans la scène. Cela permet une compréhension détaillée de l'environnement 3D et permet à la machine d'appréhender la scène dans son ensemble. Au cours des dernières années, les progrès récents de l'intelligence artificielle, de l'apprentissage automatique et de l'apprentissage profond ont ouvert une nouvelle ère, caractérisée par la disponibilité d'algorithmes puissants pour la segmentation sémantique, qui ont déjà donné des résultats remarquables dans plusieurs applications, telles que la conduite autonome, la robotique et le diagnostic médical. Ces méthodes récemment développées ne sont pas encore pleinement exploitées pour la segmentation sémantique du patrimoine bâti, et à l'heure actuelle, peu de travaux de recherche ont exploré le potentiel de l'intelligence artificielle dans ce domaine. L'objectif principal de cette recherche est donc d'étudier l'efficacité de l'intelligence artificielle, et plus particulièrement de l'apprentissage profond, sur le problème de la segmentation sémantique des nuages de points des bâtiments patrimoniaux.

1.3 Objectifs

Pour répondre aux questions résultant de l'objectif principal de cette thèse, les objectifs suivants ont été définis :

Objectif 1. Identifier les principaux problèmes et défis du processus Scan-to-BIM, en analysant chaque étape qui mène à la création de modèles numériques 3D, et en fournissant une revue exhaustive de la littérature sur les approches algorithmiques de pointe pour aborder chaque phase du flux de travail Scan-to-BIM.

Objectif 2. Explorer comment les récentes avancées en matière d'apprentissage automatique et d'apprentissage profond peuvent être exploitées pour soutenir la génération de modèles 3D dans le processus Scan-to-BIM. À cette fin, les principales techniques de segmentation sémantique par apprentissage automatique et apprentissage profond devraient être examinées et comparées, en soulignant les forces et les faiblesses de chaque méthode et en identifiant les meilleures stratégies applicables au domaine du patrimoine bâti.

Objectif 3. Proposer une procédure de segmentation sémantique efficace adaptée aux nuages de points du patrimoine bâti. L'objectif principal est de créer une approche applicable à un large éventail de scénarios du monde réel avec des conditions différentes qui exploitent pleinement les données résultant des technologies d'acquisition avancées.

Objectif 4. Créer un ensemble de données spécifique pour développer et tester la procédure de segmentation de l'objectif 3. Le nouveau jeu de données doit être composé de plusieurs bâtiments pertinents pour le domaine du patrimoine et garantir un niveau de généralisation approprié. Il doit permettre l'intégration de jeux de données similaires existants et être facilement extensible avec de nouvelles données. Pour garantir les développements et améliorations futurs, il doit être disponible gratuitement, convivial et facilement accessible par la communauté des chercheurs.

Objectif 5. Tester la procédure proposée dans le cadre de l'objectif 3 sur le nouvel ensemble de données, en évaluant et en optimisant les performances dans le cas de scénarios inédits. Les aspects critiques et les limites des approches doivent être soulignés, et les performances doivent être comparées aux méthodes existantes ou à d'autres approches. La procédure proposée devrait dépasser les performances de l'état de l'art.

2. Modélisation sémantique du patrimoine bâti

2.1 Modélisation des données du patrimoine bâti (H-BIM)

L'importance de la préservation et de la protection du patrimoine bâti devient rapidement évidente et, à ce titre, le besoin de techniques de modélisation efficaces se fait de plus en plus sentir. La *modélisation sémantique* apporte une solution à ce besoin, car elle est capable de représenter les caractéristiques physiques et abstraites du patrimoine bâti d'une manière significative. Ces dernières années, l'utilisation de la modélisation sémantique dans le domaine de l'architecture, et en particulier pour étudier et analyser le patrimoine bâti, est devenue de plus en plus populaire, et l'un des outils les plus populaires qui a complètement changé l'approche de la représentation et de la gestion du patrimoine culturel est l'utilisation de *la modélisation des données du patrimoine bâti* (H-BIM). Le terme de modélisation des données du bâtiment (BIM) a été introduit à la fin de la dernière décennie, lorsque la BIM a remplacé la modélisation numérique en 3D et la conception assistée par ordinateur (CAO) en tant qu'expression généralement utilisée pour décrire l'utilisation des technologies de l'information et de la communication (TIC) pour la conception de l'environnement bâti moderne. *La modélisation des données du patrimoine bâti* (H-BIM) est l'extension de la modélisation de l'information sur les bâtiments dans l'environnement patrimonial ou historique. Le BIM est une approche numérique qui permet de créer, d'analyser et de gérer plus rapidement et plus efficacement des bâtiments en 3D, ainsi que de soutenir la prise de décision en matière de conservation des infrastructures existantes. Avec les modèles H-BIM, les bâtiments historiques sont représentés comme un jumeau numérique, ce qui permet des simulations virtuelles et l'analyse des performances du bâtiment tout au long de son cycle de vie. Au cours des dernières décennies, l'application du BIM dans les contextes patrimoniaux s'est accrue, car les avantages d'une approche numérique sont de plus en plus reconnus.

2.2 Le processus Scan-to-BIM

Depuis longtemps, les modèles paramétriques ont été utilisés avec succès dans la conception de nouveaux bâtiments, mais pour les maquettes BIM de l'existant, il est plus difficile d'atteindre un bon niveau de connaissance et de modéliser les informations en détail. Afin d'obtenir autant d'informations géométriques détaillées que possible, on utilise généralement un nuage de points comme référence et on modélise les caractéristiques et les éléments du bâtiment. Ce processus est généralement appelé Scan-To-BIM. Il se compose de cinq phases principales : la collecte des données, le traitement des données, l'organisation des données, la modélisation BIM et l'extraction des informations. Ces dernières années, les outils et les technologies utilisés pour créer des ensembles de données 3D ou des nuages de points se sont remarquablement améliorés et permettent l'acquisition et l'extraction rapides d'informations géométriques 3D à des résolutions plus élevées. Cependant, le processus d'interprétation des nuages de points pour obtenir des modèles paramétriques est généralement effectué manuellement, ce qui est une tâche très coûteuse et chronophage. Les principales étapes du flux de travail Scan-to-BIM qui impliquent l'utilisation et le traitement des nuages de points sont au nombre de cinq : *l'acquisition de données de nuages de points, le sous-échantillonnage de nuages de points, l'enregistrement de nuages de points, la segmentation de nuages de points et la modélisation BIM à partir de nuages de points*. Quatre méthodes principales sont utilisées pour acquérir des nuages de points dans les domaines de l'imagerie : les méthodes à base d'images (photogrammétrie), les systèmes à balayage laser (LiDAR), les caméras RGB-D (Red Green Blue Depth) et les systèmes de radar à synthèse d'ouverture (SAR). Les nuages de points générés par photogrammétrie ou par balayage laser ont tendance à être chronophages et difficiles à traiter en raison de leur taille, et l'échantillonnage est souvent utilisé comme étape de prétraitement, parallèlement à d'autres méthodes telles que les filtres et l'élimination des valeurs aberrantes. La consolidation est le processus d'assemblage ou d'ajustement d'un nuage de points ou d'un ensemble de données, et cet assemblage se fait généralement par rapport à une grille locale, à un autre nuage de points ou à une grille globale. La segmentation des nuages de points 3D est une étape essentielle du traitement des nuages de points et du processus Scan-to-BIM (Rashdi et al., 2022). L'objectif du processus de segmentation est de diviser les points qui partagent des caractéristiques communes ou qui respectent des classes prédéfinies en régions homogènes. Ces régions isolées doivent être suffisamment significatives pour être utiles lors de l'analyse de la scène de différentes manières, par exemple pour localiser et reconnaître des objets, les classer et extraire des caractéristiques. Dans un premier temps, il s'agit de modéliser la géométrie 3D du composant ou de l'élément, puis d'attribuer des propriétés à l'objet, telles que la catégorie, la famille, les caractéristiques matérielles, etc. et enfin d'établir des relations entre les différents composants et éléments.

3. Algorithmes de segmentation sémantique

L'apprentissage profond a récemment été utilisé avec succès sur plusieurs problèmes de vision 2D, en particulier sur la segmentation sémantique, et il est devenu de plus en plus populaire au cours des cinq dernières années, après l'introduction des réseaux neuronaux convolutifs (CNN) (He et al., 2016). Récemment, elle a été utilisée avec des résultats remarquables pour la segmentation de nuages de points en 3D. Les données tridimensionnelles fournissent des informations spatiales et géométriques plus riches que les données bidimensionnelles et pourraient mieux caractériser les scènes complexes. Cependant, l'utilisation de méthodes d'apprentissage profond sur les nuages de points reste confrontée à plusieurs défis importants, dus par exemple à : (i) la grande taille des données, qui implique un long temps de calcul, (ii) la nature non structurée des nuages de points en 3D, qui complique l'utilisation des architectures de réseau couramment utilisées pour les données en 2D, (iii) l'indisponibilité de grands ensembles de données partagées, qui rend les résultats du processus d'apprentissage difficilement exportables à des scénarios différents de celui qui a motivé la réalisation du réseau.

3.1 Algorithmes de segmentation sémantiques existants

Selon la littérature, les méthodes de segmentation sémantique pour les nuages de points 3D peuvent être divisées en deux groupes : (i) les méthodes basées sur la projection et (ii) les méthodes basées sur les nuages de points (J. Zhang et al., 2019). Pour répondre à la nature non structurée des nuages de points, les méthodes basées sur la projection appliquent d'abord une transformation pour convertir les nuages de points 3D sur une donnée avec une structure régulière, puis elles effectuent la tâche de segmentation sémantique en appliquant des approches standards, et enfin elles reprojettent les caractéristiques extraites sur la forme ou le nuage de points de départ (Lawin et al., 2017a). Selon le type de représentation utilisé, il est possible de distinguer quatre catégories parmi ces méthodes : a) multi-vues, b) volumétriques, c) sphériques d) en grille. Les méthodes basées sur les points, ou méthodes directes, travaillent directement avec des nuages de points et n'introduisent pas de perte d'information explicite avec des représentations intermédiaires. Cette approche directe s'appuie sur la pleine utilisation des caractéristiques des données brutes des nuages de points et prend en compte toutes les informations géométriques et spatiales. Ces méthodes peuvent être divisées en quatre catégories : a) les méthodes MLP ponctuelles, b) les méthodes de convolution, c) les méthodes basées sur les RNN d) les méthodes basées sur les graphes.

3.2 Méthodologie proposée

La procédure de segmentation proposée est basée sur une approche multi-vues d'apprentissage profond, dans laquelle la segmentation est d'abord effectuée sur une représentation intermédiaire du nuage, puis les étiquettes extraites sont projetées sur le nuage de points de départ. Bien que le fait de travailler directement avec le nuage de points 3D puisse permettre une meilleure compréhension des informations spatiales et géométriques, le choix de s'appuyer sur une approche multi-vues est une stratégie efficace. D'une part, cela permet d'exploiter les modèles et réseaux existants pour la segmentation d'images, en particulier les réseaux neuronaux à convolution (CNN), qui ont obtenu des résultats remarquables ces dernières années. D'autre part, la procédure proposée pourrait être intégrée dans la chaîne de traitement photogrammétrique standard, puisqu'elle utilise un ensemble d'images comme entrée pour la création d'un nuage de points dense. Elle permet donc de développer un flux de travail automatique pour la création d'un nuage directement segmenté à partir des images photogrammétriques. En outre, à ce jour, une approche multi-vues sur des données patrimoniales n'a jamais été testée, et il est intéressant d'explorer cette approche. Le processus de segmentation est composé de deux parties, qui partent toutes deux de l'étude photogrammétrique du bâtiment à traiter. Une partie permet la construction du nuage de points dense du bâtiment, au moyen de la chaîne de traitement photogrammétrique standard : la caméra est d'abord calibrée, les paramètres internes et externes de la caméra sont estimés, et le nuage dense est généré. L'autre partie permet la segmentation du nuage de points généré au moyen de deux opérations principales : la segmentation de toutes les images photogrammétriques respectives à l'aide d'un CNN, puis la projection des cartes de segmentation d'image sur le nuage de points en s'appuyant sur les paramètres internes et externes de la caméra déjà calculés.

Trois contributions principales peuvent être identifiées pour développer cette procédure. Tout d'abord, un nouvel ensemble de données « image-point » pour la segmentation sémantique du patrimoine bâti a été produit. Il est composé d'un jeu de données constitués de nuages de points de cinq bâtiments (sites historiques) et des images photogrammétriques correspondantes, toutes deux accompagnées de leur segmentation de vérité terrain respective. Toutes les phases de la génération du jeu de données sont illustrées en détail, y compris l'acquisition, le traitement, les normes d'annotation et la procédure d'étiquetage dans la thèse. Ensuite, trois architectures de segmentation d'images, à savoir Fully Convolutional Network, SegNet et Deeplabv3+, ont été entraînées, testées et comparées sur le nouvel ensemble de données. Enfin, une procédure de projection de l'étiquetage, basée sur le principe du vote majoritaire, a été développée et testée. Elle s'appuie sur les paramètres d'orientation interne et externe des caméras calculés au cours du traitement photogrammétrique afin de transférer les étiquettes produites par le réseau profond au nuage de points, produisant ainsi une scène segmentée en 3D.

4. Le jeu de données

Des jeux de données sont actuellement disponibles et accessibles, et pourraient être utilisés pour différentes tâches et à différentes fins dans de nombreux systèmes d'apprentissage automatique. Toutefois, il manque actuellement un jeu de données précises pour la segmentation sémantique des images du patrimoine (Fiorucci et al., 2020). Cette raison a conduit à la création d'un jeu de données sur mesure et personnalisé. L'objectif principal de la création du jeu de données est de concevoir une référence basée sur l'image pour la segmentation sémantique des images de bâtiments patrimoniaux. Le jeu de données sera utilisé pour développer et entraîner un modèle de réseau neuronal profond conçu pour être incorporé dans un flux de travail plus large de segmentation de nuages de points. Actuellement, l'ensemble de données est composé de cinq bâtiments, de périodes historiques et de styles architecturaux différents, principalement situés près de Florence. Pour chaque bâtiment composant l'ensemble de données, trois types de données sont disponibles : le nuage de points TLS annoté manuellement, le nuage photogrammétrique annoté obtenu avec une procédure de transfert d'annotations, et les images du relevé annotées avec une procédure de projection d'étiquetage, qui permet de projeter automatiquement les étiquettes définies sur un nuage de points sur les images photogrammétriques relatives. La structure finale de l'ensemble de données est organisée selon les normes des principaux jeux de données de segmentation sémantique. Un ensemble d'images RGB est fourni, avec l'ensemble correspondant d'images étiquetées de la même taille, toutes deux dans un format de fichier .png avec une taille de 2592x3872 pixels. Pour permettre des comparaisons ou des intégrations futures, les étiquettes du fichier de vérification sur le terrain sont compatibles avec celles de l'ensemble de données ARCH (Matrone et al., 2020), une référence pour la segmentation sémantique des nuages de points. Ainsi, les images sont annotées en 10 classes selon le format de fichier IFC, les normes ATT et CityGML 3/4. Il s'agit des classes suivantes : arc, colonne, moulure, plancher, porte/fenêtre, mur, escalier, voûte, toit, autre. Contrairement aux nuages de points, l'arrière-plan est toujours présent dans les images, c'est pourquoi une nouvelle classe a été introduite : elle comprend tous les pixels qui ne peuvent pas être classés dans les classes définies précédemment. Cette classe est conventionnellement appelée "arrière-plan". La procédure de segmentation proposée s'appuie sur un réseau entraîné sur des images. Cependant, l'ensemble de données proposé est un ensemble de données à sources multiples composé d'images et de nuages de points, et il peut être utile d'effectuer des comparaisons et des évaluations telles que : (i) comparer la précision des méthodes basées sur les points et sur les vues multiples sur le même jeu de données, (ii) comparer la précision de l'approche basée sur les vues multiples sur des références patrimoniales avec celle obtenue sur des bâtiments standards, (ii) évaluer la précision des réseaux basés sur les points sur deux types (TLS et photogrammétrique) de données de nuages de points. Par conséquent, le jeu de données présenté peut être (i) intégré au jeu de données ARCH, (ii) utilisé pour adapter les architectures de réseau existantes au cas du bâtiment CH, (iii) exploité pour développer de nouveaux réseaux hybrides qui peuvent tirer parti à la fois des images et des nuages de points.

5. Tests de segmentation sémantique et résultats

5.1 Segmentation des images

Plusieurs tests ont été effectués afin d'évaluer le fonctionnement de la procédure proposée. L'objectif principal est de développer un modèle de segmentation d'images à large spectre capable de généraliser autant de scènes que possibles, et cette capacité peut être obtenue en fournissant un grand ensemble d'entraînement, avec un large éventail de scènes, de bâtiments, d'éléments constructifs et de typologies de structures, et un jeu de validation/test assez différent et varié par rapport au jeu d'entraînement. Néanmoins, le jeu de données disponible est encore limité en termes de typologies de bâtiments et, actuellement, il ne permet pas un bon niveau de flexibilité dans l'organisation et la division des données, ce qui rend difficile l'atteinte d'une large capacité. Cependant, dans cette étude, trois typologies de tests ont été réalisées. La première série de tests est la plus simple et la moins difficile, et consiste à tester chaque bâtiment du jeu de données un par un. L'ensemble des images de chaque bâtiment a été mélangé de manière aléatoire, puis divisé en un jeu d'apprentissage, un jeu de validation et un jeu de test, avec des pourcentages respectifs de 60 %, 20 % et 20 %. Étant donné que les images du jeu de test sont similaires à celles du jeu d’apprentissage, le modèle devrait pouvoir généraliser les solutions assez facilement dans cette série de tests. Bien que ces tests ne permettent pas d'obtenir un modèle général doté d'une grande capacité, ils sont utiles pour définir l'hyperparamètre des réseaux, pour comparer les performances des différentes architectures et pour évaluer la qualité et le bon fonctionnement du jeu de données généré. Dans ce test, toutes les images des cinq bâtiments ont été utilisées. Dans le deuxième test, les images ont été mélangées de manière aléatoire, puis divisées en un jeu d’apprentissage, un jeu de validation et un jeu de test, avec des pourcentages respectifs de 60 %, 20 % et 20 %. Malgré la présence de plusieurs typologies de bâtiments, ce test n'est pas particulièrement pertinent pour obtenir un modèle général à large capacité, puisque certaines images du jeu d'apprentissage sont analogues à certaines images du jeu de test. Cependant, le test est utile pour évaluer la capacité avec plusieurs typologies de bâtiments, pour affiner les hyperparamètres et pour évaluer l'effet de l'apprentissage par transfert et de l'augmentation des données sur la performance. La dernière série de tests est la plus difficile et représente la tâche cible d'une procédure générale de segmentation sémantique. Les tests consistent à tenter de prédire un scénario inédit. Pour effectuer ces tests, les images de quatre bâtiments ont été utilisées pour la phase d’apprentissage, en les divisant en un jeu d’apprentissage (60 %), un jeu de validation (20 %) et un jeu de test interne (20 %), et les images du bâtiment restant ont été utilisées pour le test externe du modèle. Bien que le nombre d'images semble suffisamment important pour effectuer ce type de tests, la généralisation de la solution sera une tâche difficile pour le modèle, étant donné que les typologies de bâtiments pour apprendre les caractéristiques sont limitées. Pour obtenir une vue d'ensemble des performances, une méthode de validation croisée a été utilisée, et chacun des cinq bâtiments a été utilisé alternativement comme jeu de test.

5.2 Projection des données étiquetées

La mise en œuvre de la procédure de reprojection permet de configurer et de contrôler le processus de reprojection au moyen d'un ensemble de paramètres et d'options. Le choix des paramètres est fondamental pour obtenir de bonnes performances et optimiser les résultats obtenus par le réseau neuronal sur les images. Étant donné le grand nombre de paramètres et de réglages, plusieurs expériences ont été réalisées pour trouver la plage optimale de chaque paramètre et les combinaisons optimales de réglages. Trois types de tests ont été réalisés et sont expliqués dans les sections suivantes. La première série d'expériences qui va être rapportée consiste en une recherche systématique du réglage optimal des paramètres, en testant diverses combinaisons et en évaluant l'effet de chaque paramètre sur la précision et la performance. L'étude a été réalisée sur un seul bâtiment de l'ensemble de données, à savoir *(1_SC) Spedale del Ceppo*. La deuxième série de tests consiste à projeter l'étiquette prédite sur les images par le classificateur entraîné pour le test A. Les données étiquetées prédites sont obtenues en utilisant le modèle entraîné avec l'architecture Deeplabv3+, selon les résultats illustrés, a atteint la meilleure performance. Pour chacun des cinq bâtiments de l'ensemble de données, les images photogrammétriques ont été introduites dans les réseaux entraînés liés au bâtiment, et les étiquettes de sortie ont été utilisées pour le processus de projection. Ces tests ont été effectués avec la combinaison optimale de paramètres obtenue lors des expériences précédentes. La dernière série de tests consiste en la projection de l'étiquette prédite sur les images par le classificateur entraîné pour le test C. Les résultats représentent le résultat final de la chaine de traitement de segmentation sémantique 3D du nuage de points, dans le cas d'un scénario inédit, et la précision de ces tests peut être considérée comme la performance actuelle de la procédure de segmentation.

5.3 Résultats et discussion

La segmentation de l'image est la première étape de la procédure de segmentation et une étape fondamentale pour obtenir une bonne performance finale du nuage de points. Les deux premiers tests (A et B) ont montré une bonne performance globale. Bien qu'ils ne soient pas significatifs en termes de généralisation, ils ont prouvé l'efficacité du réseau neuronal profond pour la segmentation sémantique des scènes patrimoniales lorsqu'un ensemble de données d'entraînement pertinent est fourni. Le dernier test (C) n'a pas donné de résultats suffisants, mais cela était prévisible étant donné le faible nombre de typologies de bâtiments dans l'ensemble de données. Plusieurs mesures et moyens ont été utilisés pour surmonter la limitation du jeu de données d’entrainement, mais ils n'ont pas été suffisants, ce qui met en évidence la pertinence de la faiblesse des données d’entrainement pendant la phase d'apprentissage. Comme nous l'avons déjà mentionné, deux stratégies peuvent être poursuivies : l'extension de l'ensemble de données avec de nouveaux bâtiments ou avec d'autres ensembles de données existants, et l'amélioration de la vérité de base de l'image en étiquetant les bâtiments en arrière-plan.

La projection des étiquettes sur le nuage de points est la deuxième étape de la procédure de segmentation et permet de transférer les caractéristiques extraites par le réseau neuronal des images vers le nuage de points. Les résultats ont montré une bonne stabilité de la procédure, même dans le cas d'images de faible qualité, sans perte notable de précision des performances de l'image d'entrée. Toutefois, la procédure n’est précise que si les images étiquetées en entrée ont des résolutions similaires. Malheureusement, lorsque la résolution des images est faible, la procédure de projection n'est pas encore en mesure d'améliorer les performances lors du transfert des étiquettes vers le nuage. Des améliorations pourraient être apportées à l'avenir pour permettre de surmonter le problème de la précision des données d'entrée. Par exemple, on pourrait sélectionner ou pondérer les images étiquetées en fonction de la vue, de l'angle ou de la distance par rapport au bâtiment cible.

6. Conclusion and développements futurs

L'approche proposée s'est avérée très adaptée aux nuages de points du patrimoine et présente plusieurs avantages : (i) des performances supérieures en matière de segmentation d'images par rapport à la segmentation de nuages de points, (ii) une grande disponibilité d'ensembles de données d'images existants pour pré-entraîner le modèle ou exploiter l'apprentissage par transfert, (iii) une grande disponibilité d'images à haute résolution acquises au cours de l'enquête, pertinentes pour capturer les détails géométriques ou les éléments constructifs complexes, (iv) l'intégration possible de la procédure de segmentation dans la chaine de traitement photogrammétrique, (v) une plus grande disponibilité et une acquisition plus facile des images pour augmenter et élargir le jeu de données par rapport aux nuages de points. En ce qui concerne les résultats obtenus, dans le cas général d'un scénario inédit, les performances sont actuellement totalement insatisfaisantes : la moyenne du test de validation croisée sur les cinq bâtiments de l'ensemble de données a atteint un GA de 54 %. Cela prouve que le modèle est toujours incapable de se généraliser parmi plusieurs typologies de bâtiments. Toutefois, ces résultats ne sont pas surprenants. Comme l'ont souligné plusieurs recherches, les modèles d'apprentissage profond sont gourmands en données, et les modèles de segmentation d'images nécessitent en particulier des milliers d'images pertinentes pour développer un réseau hautement performant. Bien que le nombre d'images dans notre jeu de données soit conséquent, il ne correspond qu'à cinq bâtiments et n'est donc pas suffisamment pertinent.

Les principales limites de la procédure ont été bien identifiés, et certaines avancées futures possibles ont pu être définies. Tout d'abord, l'augmentation du nombre de bâtiments de l'ensemble de données (i) avec de nouvelles acquisitions, (ii) avec une intégration avec des ensembles de données existants, ou (iii) avec la génération d'images synthétiques. Deuxièmement, l'amélioration des performances de la segmentation des images (i) en exploitant les informations de profondeur au début ou à la fin du bloc de segmentation, (ii) en testant la segmentation par instance, (iii) ou en testant l'opération de post-traitement pour améliorer la qualité des cartes de segmentation. Enfin, l'amélioration de la procédure de projection de l'étiquetage afin d'améliorer la précision de la segmentation de l'image.

Université de Strasbourg

Eugenio PELLIS

École doctorale Mathématiques, sciences de l'information et de l'ingénieur | ED 269 Université de Strasbourg

A multiview approach for the semantic segmentation of heritage building point clouds

Résumé

Cette thèse aborde la numérisation du patrimoine culturel en utilisant les méthodes de modélisation des données de bâtiments (BIM) et de modélisation des données du patrimoine bâti (H-BIM) comme un puissant outil pour la conservation et la préservation. Elle se concentre sur le processus « Scan-to-BIM », qui rencontre des défis dans la gestion des données à grande échelle issues des technologies d'acquisition modernes comme le balayage laser 3D et la photogrammétrie. La thèse vise à automatiser certaines étapes du processus « Scan-to-BIM », en mettant l'accent sur la segmentation sémantique des nuages de points. Les recherches s'appuient sur les avancées de l'intelligence artificielle et de l'apprentissage profond pour améliorer le processus de segmentation du patrimoine bâti en utilisant une approche multi-vues. Trois contributions principales sont mises en avant dans la thèse. La création d'un jeu de données image-point3D pour la segmentation sémantique du patrimoine bâti, comprenant des scènes de nuages de points de bâtiments et des images photogrammétriques avec une segmentation de référence. Ensuite, l'entrainement, l'expérimentation et la comparaison de trois architectures de segmentation d'images reconnus (Fully Convolutional Network, SegNet et Deeplabv3+) sur le nouveau jeu de données. Enfin, le développement et le test d'une procédure de projection de données annotées, basée sur le principe du vote majoritaire, pour transférer les étiquettes générées par le réseau profond vers le nuage de points, aboutissant à une scène segmentée en 3D. Malgré le nombre limité de typologies de bâtiments dans le jeu de données, les résultats sont prometteurs, indiquant une amélioration de l'automatisation et de la fonctionnalité dans la préservation et la gestion du patrimoine bâti grâce à des modèles 3D.

Résumé en anglais

This dissertation addresses the need for digitizing cultural heritage using Building Information Modeling (BIM) and Heritage Building Information Modeling (H-BIM) as a powerful tool for conservation and preservation. It focuses on the Scan-to-BIM process, which faces challenges in handling large-scale data from modern acquisition technologies like 3D laser scanning and photogrammetry. The study aims to improve automation in the Scan-to-BIM pipeline, particularly in semantic segmentation, which involves categorizing raw point cloud data for machine understanding. The research leverages advancements in artificial intelligence and deep learning to enhance the segmentation process for heritage buildings leveraging on a multiview approach. Three main contributions are highlighted in the dissertation. The creation of a novel image-3Dpoint dataset for heritage building semantic segmentation, including building point cloud scenes and photogrammetric images with ground truth segmentation. Secondly, the training, testing, and comparison of three state-of-the-art image segmentation architectures (Fully Convolutional Network, SegNet, and Deeplabv3+) on the new dataset. Finally, the development and testing of a labeling projection procedure, based on the majority vote principle, to transfer deep network-generated labels to the point cloud, resulting in a 3D segmented scene. Despite the limited number of building typologies in the dataset, the results show promise, indicating improved automation and functionality in the preservation and management of heritage buildings through 3D models.