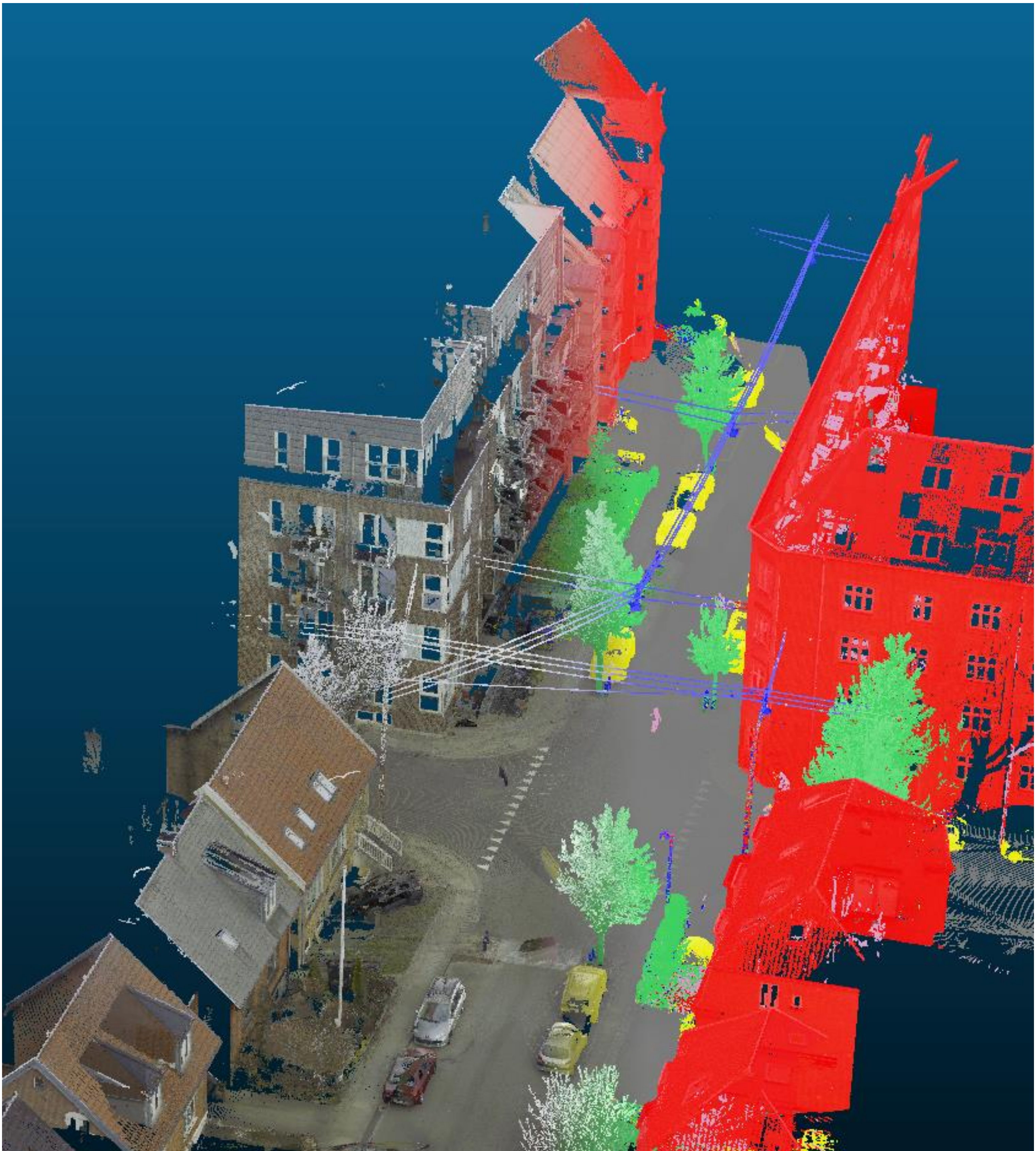

Deep Learning in land surveying

- Semantic Segmentation of terrestrial point clouds

By Mads Westergaard and Mikkel Knudsen

Master's thesis in Surveying and Mapping

Year 2021

AALBORG UNIVERSITY
STUDENT REPORT

Aalborg University
Technical Faculty of IT and Design
Surveying, Planning and Land Management
Rendsburggade 14
9000 Aalborg
http://www.tech.aau.dk

Title:
Deep learning in land surveying

Subtitle:
Semantic segmentation of terrestrial point clouds

Study Programme:
Masters programme in Surveying, Planning and Land Management - Surveying & Mapping

Theme:
Master's thesis

Project period:
February 1st 2021 to June 4th 2021

Project group:
Group 1

Authors:
Mikkel Knudsen
Mads Westergaard

Supervisor:
Peter Cederholm

Page count: 104
Standard pages: 47
Appendices: 3

Deadline:
June 4th 2021

Synopsis:

This master's thesis concerns the use of deep learning algorithms via semantic segmentation on point clouds in a land surveying perspective. The thesis is split up into two parts: An initial problem that examines the general theory and practice behind semantic segmentation using deep learning and a primary problem that concerns the potentials of using semantic segmentation on point clouds in a land surveying perspective. Through the answering of the primary problem typical point cloud assignments in land surveying are categorized, a segmentation model designed for use on outdoor point clouds measured with terrestrial laser scanners is trained and evaluated and a discussion of the potentials and challenges that semantic segmentation has in land surveying. The trained segmentation model reached the specified requirement of an overall accuracy of 96 %, meaning that the segmentation model guesses the semantic class of 96 % of the total point correctly. With this accuracy, the main potential in using the segmentation model is the ability to remove all data in irrelevant classes, which returns a cleaner, less noisy, and smaller (in terms of file size) point cloud focused only around specific semantic groups.

Resumé

Indeværende afgangsprojekt handler om semantisk segmentering af terrestriske punktskyer ved hjælp af deep learning, og hvordan det kan bruges i landmålings sammenhænge. Projektet tager udgangspunkt i en præudviklet Deep Learning metode kaldet PointNet++ til segmentering af disse punktskyer. Formålet med projektet er at undersøge, hvordan dataevalueringen af punktskysdata kan gøres mere effektiv.

Projektets indledende del undersøger hvordan PointNet++ kan implementeres på et "benchmark datasæt", men også på egne datasæt for at opnå en grundlæggende forståelse for hvordan teknologien virker.

Den primære problemstilling i projektet er centreret omkring, hvordan håndtering af punktskydata kan gøres mere effektiv ved hjælp af segmentering med deep learning. Undersøgelsen af problemstillingen er delt op i 3 dele, som hver især vil bidrage til den samlet besvarelse. Første led i besvarelsen omhandler hvilke typer af opgaver, der forekommer i landinspektørsammenhænge, hvor segmentering kan bidrage til en mere effektiv datahåndtering. Næste led tager fat i én af de netop beskrevne opgavetype og undersøger, hvordan semantisk segmentering kan implementeres for denne type opgave. Denne del vil teste forskellige metoder til at implementere Semantisk Segmentering, men også teste resultatet heraf. Sidste og tredje del af undersøgelsen vil undersøge, hvilke potentialer og udfordringer der er forbundet med Semantisk Segmentering i et landmålingsperspektiv.

I undersøgelsen konkluderes det, at der findes en række af opgavetyper, hvor segmentering potentielt kunne forbedre den nuværende håndtering og efterbehandling af punktskysdata. Dernæst implementeres segmenterings-teknologien for én af de beskrevne opgavetyper, i projektets tilfælde er det punktskyer af urbane miljøer. Implementering af semantisk segmentering kræver, at der trænes en målrettet model til segmentering af den valgte punktsky, hvorefter den segmenteres. Kvaliteten af den segmenterede punktsky evalueres ved at sammenholde den mod en ”ground truth” punktsky, hvor alle punkter manuelt er blevet opdelt i 7 fordefinerede semantiske klasser. Herved dokumenteres det, hvor og hvor meget segmenteringen fejler. Det konkluderes ved PointNet++ metoden at der kan opnås en overordnet nøjagtighed på næsten 96 % for klassificeret urbane punktskyer, hvilket vil sige, at den opdeler og klassificerer 96 % af punkterne korrekt ift. punkternes ”ground truth” semantiske klasse. Den tredje del og sidste delspørgsmål er overvejende en undersøgelse af hvilke potentialer og udfordringer, der vil være forbundet med at bruge denne teknologi i landmålingspraksis. Praktiske erfaringer opnået under implementering af semantisk segmentering kombineres med udtalelser og betragtninger om semantisk segmentering fra landmålingsvirksomheder. Det konkluderes, at det primære potentiale ved semantisk segmentering er effektivisering af datahåndteringen ved at kunne fjerne

støj eller uønskede elementer fra punktskyer. Modsat er den primære udfordring at implementere segmenteringen i landmålingspraksis, hvilket kunne imødekommes ved at udforme et program der gjorde implementeringen mere intuitiv.

Problemformulering besvares samlet i en konklusion for rapporten, heri konkluderes det at Semantisk Segmentering kan forbedre den nuværende håndtering af punktskyer. Konkret kan dele af den manuelle sortering af punktskyer fjernes ved at implementere Semantisk Segmentering. Dermed bliver den samlede datahåndteringstid skåret ned, hvilket potentielt gør punktskyer til et mere konkurrencedygtigt produkt i landinspektørsammenhænge.

Preface

This master's thesis is written by Mads Westergaard and Mikkel Knudsen on the 4^{th} semester of the master's program Surveying and Mapping. The project period is from February 1^{st} to June 4^{th}.

The thesis is written in a land surveying perspective, meaning that the report presupposes some understanding of surveying and its principles and that the target audience is readers with a surveying background.

A large thank you to Malte Holm-Christiansen from LE34, Morten Hellemann from GeoPartner, Peter Hastrup Jensen and Henrik Vad Jensen from Aakjaer Landinspektører for their inputs and comments regarding the subject of the thesis. Thanks to LE34 for lending out scanning equipment during the project period and for lending out point clouds for use in the thesis - and likewise thanks to GeoPartner for lending out a point cloud for use in the thesis.

Furthermore, a very large thank you to Peter Cederholm for his competent supervision and guidance during the project period.

Reading guide

Literature in this project is referenced using the Harvard method as (author year) and if page number(s) is given it will be given as (author year, page). n.d (no date) is given instead of a year when the source has no year. A list of references can be found in the back of the report. When references are placed before a period/full stop the reference is for that sentence only, whereas references placed after a period/full stop is for every previous sentence in that paragraph. English decimal separation is used, which means that periods (.) are used to indicate decimals, while commas (,) are used to separate thousands.

Figures, tables, and equations are numbered **x.y** where x is the chapter that the element in question is found in and y is the number of that type of element that it is in that chapter. A list of figures and tables can be found in the back of the project. If figures are not cited they are made by the authors themselves.

There are 3 appendices placed in the back of the report. Appendix A is a documentation of the contents of the zip file that is handed in along with the thesis. When elements contained in the zip file are referenced in the report, a reference to appendix A and reference to the ref.no. that the given file has as shown in appendix A.

The point cloud data collected in the project (downsampled and in ASCII-format) can be downloaded from the following link for the rest of June 2021:

`https://tinyurl.com/SM4MadsMikkel`

Contents

CHAPTER 1

Introduction

In land surveying point clouds are becoming a more and more common product because it is becoming easier and faster to collect precise and covering data that can represent all kinds of areas using, for example, UAVs or laser scanners. Point clouds can be a great data foundation, as point clouds usually contain spatial data about almost everything found in the scene measured. The problem with point clouds is that they can be very hard to manage and properly understand and therefore use to their full potential. This is because point clouds just are a large number of discrete points given in a coordinate system with no other attributes than XYZ data for each point and in some cases intensity data and/or RGB data for each point. Many people would not know what to do with a raw point cloud exported directly from a scanner or calculated from UAV data. Sure, the point cloud describes the scene well, but what are they to do with it aside from getting an overview of the area and performing a couple of measurements in it?

Consequently, point clouds require processing before they can be used properly. The kinds of processing required largely depend on the type of point cloud and the type of assignment that they are used for. In many cases, the processing includes operations such as vectorization of certain elements, generating terrain models, and removing noise/unnecessary data. These operations require human interaction and usually take more time than gathering the data in the first place. This is to some degree because processing point cloud requires a lot of decision-making since the point clouds themselves do not add any information as to what each point describes. Again, this is because point clouds in most cases have no other attributes than the position and intensity/color of each point.

Parallel with the developments in surveying technologies towards the use of point cloud data, developments within Artificial Intelligence (AI) and specifically Deep Learning have also been made toward handling of point clouds. Artificial Intelligence refers to when a machine or a program mimics cognitive human functions which can be when a machine is programmed to behave in a specific way in a given situation like *if-else* statements in programming. Machine Learning (ML) is a sub-category of the AI but where it is programmed to analyse the given data, learn from it and then make decisions based on the experience obtained. In ML training data must be given, so the algorithms can achieve experience from this training data. In principle, the more data is given the better the "decisions" get. Deep Learning is a sub-category in ML whereas the algorithms in deep learning are based on artificial neural networks.

Deep Learning also requires training data to solve tasks. The primary difference between ML and Deep Learning is that Deep Learning uses these neural networks which are multi-layered structures of algorithms. (Oppermann 2019). The relation between AI, ML, and Deep Learning is briefly illustrated in figure 1.1.

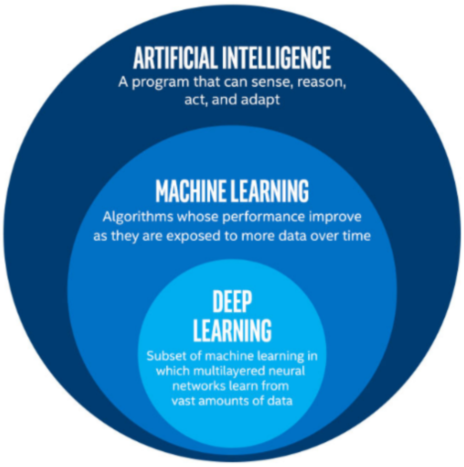

Figure 1.1: Artificial intelligence vs. Machine Learning vs. Deep Learning cf. (Oppermann 2019)

Using deep learning, some scientists have succeeded in training a computer to automatically examine point clouds to categorize each point based on their semantic class and then segmenting the point cloud on this basis. The advantage of classifying points is that the later processing of the point cloud is less time-consuming since whole objects and classes can be detected and processed automatically. This deep learning technology has been developed for use in many fields, see for example (Marr 2018). Semantic segmentation as a branch of deep learning was initially developed for image segmentation in computer vision, while the 3D use cases of semantic segmentation seem to be focused on use in autonomous vehicles and medical science. For autonomous vehicles, this allows the computer in said autonomous vehicles to deduce what and where elements like other cars, pedestrians, signs, and trees are in relation to itself by using the built-in LiDAR scanner, radars, cameras, etc. (Grigorescu et al. 2020). In medical science it is used to automatically classify, segment, detect, and locate different properties in medical images and data (Singh et al. 2020).

At the time of writing, deep learning and semantic segmentation are not known to be implemented in the business of any danish land surveyors. This is despite there being examples of deep learning performed on point clouds that might as well have been collected by a surveyor with a terrestrial, or even a mobile, laser scanner. One such example is the *Semantic3D* dataset, which is a benchmark dataset for training and evaluating deep learning algorithms, giving the different algorithms a common basis of comparison. An example of a semantically segmented point cloud from Semantic3D can be seen in figure 1.2. It appears how each point has a label (a color) that describes the semantic class of the point.

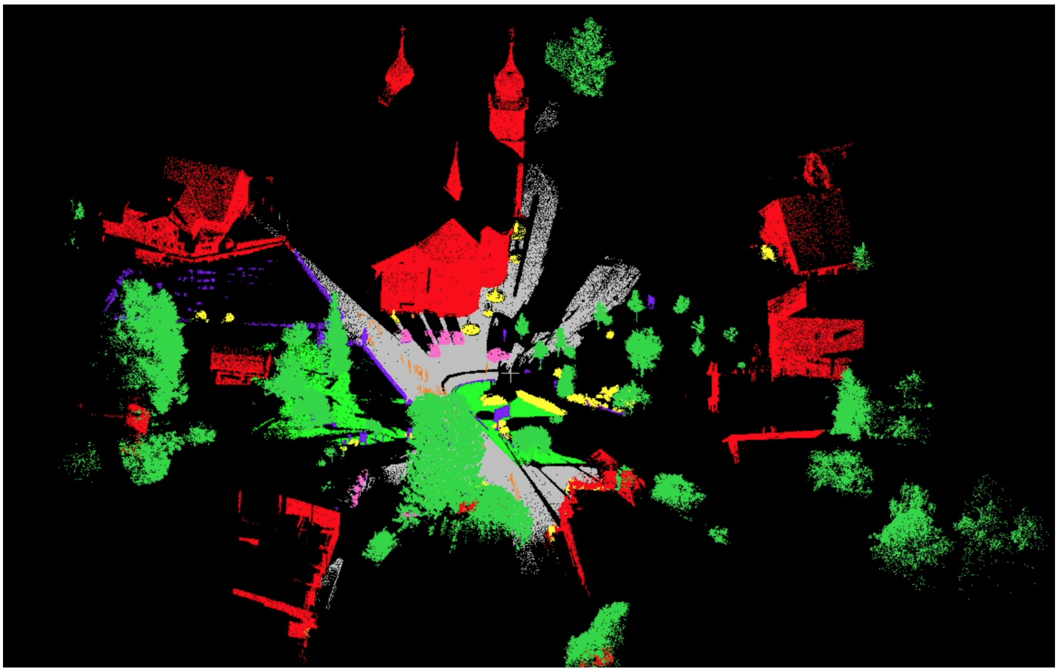

Figure 1.2: Example of a segmented point cloud data from Semantic3D's website. No legend is given. (Hackel et al. 2017)

The Semantic3D dataset was mainly made because of its potential in robotics, augmented reality, and urban planning (Hackel et al. 2017). However, there is no reason to believe that semantic segmentation as shown in figure 1.2 cannot be used in a land surveying perspective. As such, it should be possible to implement deep learning algorithms in the workflow of land surveyors. Being able to "augment" a point cloud with data regarding the semantic class of each point has the potential to make some of the post-processing of point clouds easier and might even open new doors for how point clouds can be processed. For example, if vegetation is not needed in a point cloud for a certain assignment, then semantic segmentation will make it easy to just remove that class in the point cloud. Or maybe the assignment is focused on building facades. Then removing everything that is not classified as buildings is easy, and allows for easier handling of the point cloud because of the lower file size. It may even be possible to train a segmentation model[1] towards being able to distinguish between other semantic classes, such as windows, manhole covers, drains, etc. And these are only examples for outdoor scans - likewise, it should be possible to use semantic segmentation algorithms on indoor scans, which may, for example, allow for removing all furniture in a point cloud. It may even be applicable on point clouds created from UAV data.

Point cloud segmentation is not a new technology in itself. Classic methods of segmenting point clouds include manual segmentation, segmentation by color, elevation, scans, filters, and specific object detection (such as road and trees detection) (Lohani and Ghosh 2017).

[1]A "model" is a term used to define experience achieved through training through the deep learning algorithm

Some of these functions are included in commercial software or external scripts. However, none of these functions are known to use semantic segmentation via deep learning. It would be very interesting and relevant to introduce semantic segmentation with deep learning approaches in land surveying companies to examine if this could affect the workflow with point clouds positively.

As such, it is clear that, while segmentation of point clouds already exists, it has not been researched if deep learning segmentation can improve the data evaluation of point clouds from a land surveying perspective. For this reason, this project will study if these methods can be implemented in data evaluation of point clouds and test if the methods can improve/streamline the workflow and processing of point clouds in regular surveying companies.

1.1 Initial problem

Before a problem statement can be formulated there are a few elements that need to be investigated. For one, it is necessary to find out how a semantic segmentation model is trained and how it is evaluated, as this is an important factor for using it. Moreover, it is necessary to examine *how* an automatic semantic segmentation is performed in practice, as the software needed to perform this operation is non-standard and mostly made with other software/data specialists in mind as the target audience.

> How is it generally possible to semantically segment point clouds using deep learning and what are the principles behind the method?

As such, semantic segmentation using deep learning will be the focus of this project. *For the rest of this report "semantic segmentation" or just "segmentation" will refer to automatic semantic segmentation using deep learning if nothing else is stated.*

The answering of the initial problem is documented in the following chapter 2.

1.2 Initial method and approach

The above initial problem consists of two parts - an investigation of the principles behind semantic segmentation and a more practical investigation of how to implement semantic segmentation on point clouds. Thus, the initial problem is focused on a theoretical and a practical part.

The theoretical part will focus on the principles and operations that are needed to start segmenting point clouds and evaluating the segmentation. This does not include the underlying principles in deep learning, as this is a whole other professional field than this report is written in. Therefore, the focus is on the principle of training a deep learning model, how to use this model and how to evaluate the performance of the model.

The practical part focuses on implementing the above principles in practice. To do this, specific software made to segment point clouds will have to be used. Many of the developers of deep learning algorithms made to semantically segment point clouds share their work in the form of code/software on the platform GitHub. Using one of the available approaches the idea is to train a segmentation model using benchmark data from Semantic3D. The performance of this model can then be evaluated using Semantic3D's evaluation data to see how it performs. However, one of the main purposes of the practical part is to find out how custom point clouds that are not affiliated with Semantic3D are processed using the trained segmentation model, as this is a crucial step if the method is to be used in practice.

Answering the two parts of the initial problem will lay a foundation for understanding the terminology used in semantic segmentation and the process of using semantic segmentation in practice, and will allow for a more focused problem statement wherein a more specific problem will be examined.

Semantic segmentation of point clouds

CHAPTER 2

This chapter will answer the initial problem stated in the introduction, section 1.1. The present section is divided into two parts where the first part describes the principles behind using deep learning to automatically segment point clouds and the second part briefly documents a pilot study meant to find out how to use deep learning to segment point clouds in practice.

Deep learning is a very specialized and complex subject within AI and machine learning, which in terms of profession is very different from the surveying perspective that this report is written in. Therefore, it will be difficult to make substantiated adjustments to the process and for that reason, default values are preferred. Consequently, this chapter and the report as a whole will not focus on the inner workings and processes behind deep learning used on point clouds, as this would require an immense amount of work to understand. Instead, the focus will be on the practical implementation of deep learning used to segment point clouds. As such, only the very general principles will be described here. The principles described will be focused around *training* and *evaluation* of deep learning "experience-files", or models, as they will be referred to as in this report.

2.1 General principles

As described above, deep learning is a very complex subject to explore with the professional background that this report is written with. Therefore this section will focus on describing the training of a model and after that the evaluation of a model, as these steps are critical factors when using deep learning algorithms.

2.1.1 Training

To use deep learning algorithms it is necessary to train them to perform the given task. In this context, the aim is to train the model to effectively and accurately segment point clouds into semantic groups without human supervision. By feeding the model large amounts of point cloud data that has been labeled in semantic groups, the computer can start examining the point cloud to find connections between every point present to infer what each individual point represents in the scene. As such, the model is able to guess if a point is a part of, for example, a tree or a building based on the geometry and the color of the point itself and its neighbouring points. The computer learns how to infer, or *predict*, this by iterating over training data

again and again while trying to correctly label each point. By continually guessing at what each point may be representing and then comparing this to the given label for each point the computer gradually learns to distinguish what each of the points may be representing. Each time the model tries to predict the label of every point and evaluates whether or not it guessed correctly. It uses this evaluation to slightly alter how it will predict the next epoch[1]. If all goes well, the model will have learned to segment point clouds into semantic groups with some accuracy after multiple epochs.
The above description of the process is a gross simplification of what is actually happening, but in this context, it is assessed sufficient for understanding the idea of deep learning. The above is based on a description of training a simple deep learning classification model in 2D from (David 2018, pp. 26-27). In the case of segmenting 3D point clouds, the process gets multiple times more complex, but the principle for the training is in this context considered to be the same.

The level of abstraction that the model is able to output is entirely dependent on the data that it has been trained on. If the data that it has been trained on includes many different semantic classes it will also be able to identify this amount of classes. In theory, it is possible to train a model that can identify and differentiate between, for example, windows, vents, drain pipes, roofs and walls of buildings, trees, bushes, cars, trucks, busses, bikes, and so on. It is relevant to consider which of these classes are necessary for the given task, as it rarely is all of them that are relevant. Thus, it is necessary to consider the level of abstraction. For example, it may in some cases only be relevant to segment the road into one class, whereas in other assignments that are more directed towards roads it may be relevant to segment the road into multiple sub-classes, like asphalt, drains, road markings, etc. The amount of classes that are given to the algorithm during training is the number of classes that the model will be able to output when the training is performed. Therefore, it is important to direct the training of the model towards the tasks that the model is intended to be able to solve.

When a model has been trained to semantically segment point clouds, it is possible to feed this model any other point cloud in the same data format, which it should then be able to segment by predicting labels for each of the points. This process is generally referred to as prediction or inference. If a point cloud is pre-labeled[2], then it is possible to use it for evaluation of the segmentation model, as it is then possible to check the predicted labels against the true labels. Point clouds used for evaluation should not be part of the dataset used for training, as this creates a bias. It makes more sense to evaluate the model using independent point clouds as this results in a more representative evaluation. The metrics used to evaluate semantic segmentation models are described below.

[1] An epoch is one iteration of guesses through the training data.
[2] labelling refers to giving each point a value that references the semantic class

2.1.2 Evaluation

When training a deep learning model to do semantic segmentation it is relevant to evaluate the model's performance. Typically, the metrics used for evaluating segmentation models are:

- Confusion matrix
- Accuracy
- Intersection over union (IoU or Jaccard Index).

Confusion matrix

In its basic form, the confusion matrix shows how the predictions that the segmentation model has made for each class are distributed between correct and incorrect guesses for each class. This gives an overview of the model's performance in raw numbers. An example of a confusion matrix for a hypothetical segmentation model can be seen in table 2.1.

Confusion Matrix	Predicted labels				Sum
	Classes	Trees	Buildings	Cars	
True labels	Trees	957	30	13	1000
	Buildings	35	873	92	1000
	Cars	18	184	798	1000
Sum		1010	1087	903	$\frac{2628}{3000}$

Table 2.1: Example of a confusion matrix based around segmentation of trees, buildings and cars in a point cloud with 1000 points in each class.

The diagonal highlighted in grey in table 2.1 shows how many of the 1000 points in each class that the model has been able to predict correctly while off-diagonal elements refer to the number of incorrect guesses on that class [3]. The last diagonal element shows the sum of points that has been predicted correctly across all classes divided by the total number of points. The sum of points in each row equals the total number of points in the class, while the sum of points in each column equals the number of guesses that the model has made on the class in the respective column. For better comprehensibility, the confusion matrix can be normalized with respect to the total number of points - exemplified in table 2.2.

[3] the row is the true label, the column is the predicted label. E.g, the 30 in row 1, column 2 refers to 30 points predicted as a building that was actually belonging to a tree.

Normalized CM	Predicted labels				True share	Accuracy
	Classes	Trees	Buildings	Cars		
True labels	Trees	31.9	1	0.4	33.3	95.7
	Buildings	1.2	29.1	3.1	33.3	87.3
	Cars	0.6	6.1	26.6	33.3	79.8
Predicted share		33.7	37.2	30.1	Overall accuracy	
Precision		94.8	80.3	88.4	87.6	

Table 2.2: The confusion matrix from table 2.1 normalized with respect to the total number of points, so that the numbers now represent the percentage of the total number of points that are in each box.

Normalizing the confusion matrix can in many cases make the results of the segmentation more comprehensible, as the number of points easily exceeds multiple millions of points. Normalized confusion matrices are not generally used as a standard way of describing the distribution of correct and wrong predictions. However, it will be used in this report because of its enhanced comprehensibility. Additionally, normalized confusion matrices in this report include the parameters accuracy and precision, which are described below.

Accuracy and precision

Accuracy describes the percentage of points that the model has predicted correctly in relation to the number of points in the given class. The overall accuracy (OA) is the total number of correctly predicted points in relation to the total number of points. Conveniently, the accuracy for each class and the overall accuracy can be seen in the normalized confusion matrix in table 2.2. In many of the benchmark 3D deep learning training datasets, overall accuracy is used as one of the main evaluation metrics for the trained model. The overall accuracy is however not always a great metric for evaluation, as it can become misleading in situations where the number of points in the different classes are imbalanced. For example, if there had been only 100 points representing cars in the example in table 2.1 and the model had predicted 10 of them correctly, then the accuracy for that class is 10 %. But the overall accuracy is still $\frac{957+873+10}{2100} = 87.6\%$, which does not give any indication of the very poor accuracy that it was able to predict cars with. For that reason, overall accuracy is not always a useful metric for evaluation by itself.

Precision refers to the relation between the number of predictions that are correct within each class in comparison to the total number of points the model has predicted to be in that class. As such, the precision can only be described separately for each class. It is a good idea to use both the precision and the accuracy for each class as this gives a better insight into how the prediction of each class went.

Intersection over Union

IoU, also known as the Jaccard index, is a more "robust" evaluation metric, as it can be used in both balanced and imbalanced datasets. The IoU can be described simply as the relation between the area of intersection and the area of union - commonly illustrated using figure 2.1.

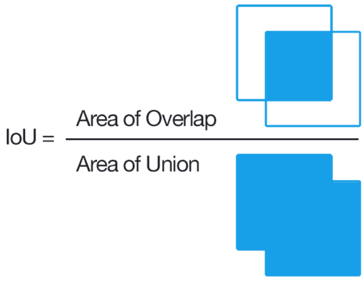

Figure 2.1: The standard illustration of the IoU. (Uniqtech 2020)

IoU as illustrated in figure 2.1 is not directly applicable to evaluate point clouds, but the principle behind using it on point clouds is similar. Mathematically the IoU for each semantic class is calculated as shown in equation 2.1

$$IoU = \frac{A \cap B}{A + B - A \cap B} \tag{2.1}$$

Cf. (Uniqtech 2020)

Where A is the ground truth labels for a class and B is the predicted labels for the same class.
As such, the numerator in equation 2.1 is the number of points that are the same between the true labels and the predicted labels. This is the intersection. If trees from the example in figure 2.1 are used as an example, then the numerator is 957. The denominator is the total number of points labeled as one class in both the true labels and the predicted labels, but with the intersection subtracted. This is the union. In the case of trees, the union is (1000+1010)-957=1053. As such, the IoU for the trees is $\frac{957}{1000+1010-957}$, which is 90.9 %. The IoU is calculated for each of the classes, and the mean IoU is then calculated. This is a more representative metric for evaluation of segmentation models, as each class weighs the same, even if the number of points is imbalanced. Continuing the example from table 2.1, the IoU for buildings is $\frac{873}{1000+1087-873} = 71.9\%$ and the IoU for cars is $\frac{798}{1000+903-798} = 72.2\%$. Consequently, the mean IoU (mIoU) in this example is 78.3 %. Thus, the mean IoU is slightly lower than the 87.6 % OA.

If the example with only 100 points indicating cars is used and the confusion matrix is as shown in table 2.3, then the IoU and OA are suddenly quite different.

Confusion Matrix	Predicted labels				Sum
	Classes	Trees	Buildings	Cars	
True labels	Trees	957	30	13	1000
	Buildings	35	873	92	1000
	Cars	18	72	10	100
Sum		1010	975	115	$\frac{1840}{2100}$

Table 2.3

The OA is yet again 87.6 %, but calculations of the IoU results in an IoU of $\frac{957}{1000+1010-957} = 90.9\%$ for the trees, an IoU of $\frac{873}{1000+975-873} = 79.2\%$ for the buildings and an IoU of $\frac{10}{100+115-10} = 4.9\%$ for the cars, resulting in a mean IoU of 58.3 %. This time the IoU is a lot lower than OA, as the value of the IoU decreases significantly when one or more of the classes are segmented badly - even if the number of points in that class is relatively low.

As such IoU is also a relevant parameter to consider when evaluating deep learning segmentation models, as it allows for an evaluation that is more likely to show if any of the classes are predicted poorly. In combination with the overall accuracy and the normalized confusion matrix, these metrics provide a good overview of the performance of a trained model.

The principles in semantic segmentation and how to evaluate the segmentation have been introduced. Accordingly, the following will describe how to implement these principles for specific point clouds.

2.2 Implementation of Semantic Segmentation

In the above section, the principles are introduced while this section will outline how semantic segmentation can be implemented in practice. The present section will describe which segmentation approach is chosen and how it is implemented. Hereafter a few examples will be examined to research the chosen segmentation approach actually works.

Segmentation approach
In the "segmentation world" there exists multiple methods to segment point cloud with deep learning approaches. Accordingly, this subsection will describe and argue for the chosen approach. The state-of-the-art methods are documented cf. (Liu et al. 2019, chapter 3) in a figure. Most of these segmentation methods can be downloaded from GitHub and are open source.

The frame of this project does not have the professional competencies to assess which segmentation method is preferable or the best and the segmentation method is therefore chosen from a point of view of how to implement segmentation on land surveying data. The specific approach Open3D-PointNet2-Semantic3D (PointNet++) (Lao 2019) is chosen to be the method for segmenting point clouds. Open3D in the title refers to a library that handles point clouds, PointNet2, or Pointnet++, refers to the semantic segmentation method (Deep learning on point clouds) and Semantic3D refers to the used benchmark data. The PointNet++ approach is chosen since it has the best description of how to implement it locally and the best descriptions of potential errors in the implementation.

As standard, the data in this PointNet++ approach is from Semantic3D, which provides prelabeled point clouds for training, validation, and prediction. The data in Semantic3D is similar to data used in land surveying companies which also substantiates the choice of this approach.

The method for implementing this approach is described and documented in appendix B. Herein the hardware and the software are described in the interest of reproducibility. The following will go through two examples of segmented point clouds, one with Semantic3D data and the other example with a "custom" point cloud data from Aalborg City. Both examples are segmented with a model trained from the Semantic3D benchmark dataset through 100 epochs.

Example of a segmented point cloud:
This subsection will illustrate an example of a segmented point cloud, the example is built on the dataset *Sg27_station9_intensity_rgb* downloaded from Semantic3D (Hackel et al. 2017). The data set comes from a terrestrial laser scanner and the data format is; x, y, z, intensity, red, green, and blue stored in an ASCII file. A label file describing the true semantic class for each point is also included. The label file is an ASCII file with a long column of integers from 1 to 8. The index in this column refers to the semantic class of the point with the corresponding index in the point cloud file, i.e. the first index in the label file refers to the first point in the point cloud file, the second index refers to point number 2 and so forth. It is worth noting that this point cloud data is from Semantic3D which also provides the training data, and as such, this dataset is very similar to the training data (the scenes are similar and are possibly from the same scanner). The point cloud is segmented by the procedure described in appendix B.

Since the segmented point cloud is a product of the trained model, it is worth understanding how the training data is made. The training is performed on 9 different point clouds which have all been labeled manually by the people behind Semantic3D. The categories in these manually labeled point clouds are documented in table 2.4. These point clouds are then used to train a model which is crucial to segment the point cloud. The training is one of the adjustable parameters when segmenting a point cloud since the result is influenced by how well the model is trained. How well the model is trained depends on the quality and quantity of the training data and how many epochs or iterations the model has run through the training data. The training of the model is performed using the default parameters for training in the algorithm.

Category	Class	Color
Man-made terrain	1	Blue
Natural terrain	2	Maroon
High vegetation	3	Pink
Low vegetation	4	Green
Buildings	5	Red
Hard scape	6	Purple
Scanning artefacts	7	Navy
Cars	8	Olive (Yellow)

Table 2.4: Description of categories. To visualize the different categories the points are colored individually.

If the point cloud already has a label file, then it is possible to assess how well the segmentation went by comparing the label file to the segmented point cloud. The pre-labeled file is considered as the "true value" which the segmented point cloud will be compared to. This comparison will end up in a confusion matrix that sums up how well the segmentation went.

In figures 2.2 and 2.3 the point cloud is shown respectively with RGB-colors (RGB colors from the camera of the scanner) and with colors colored by the labels from the segmentation. From a distance, it looks overall satisfying but there are also some errors. These errors are for example in the segmentation of cars (one of them is mostly blue which indicates man-made terrain class). Furthermore, some terrain on the left side of the pictures is segmented wrong (it is classified as buildings but it is low vegetation). A more quantitative method to assess the segmentation is to compare the segmented point cloud to the pre-labeled point cloud, this is done in the normalized confusion matrix in table 2.5.

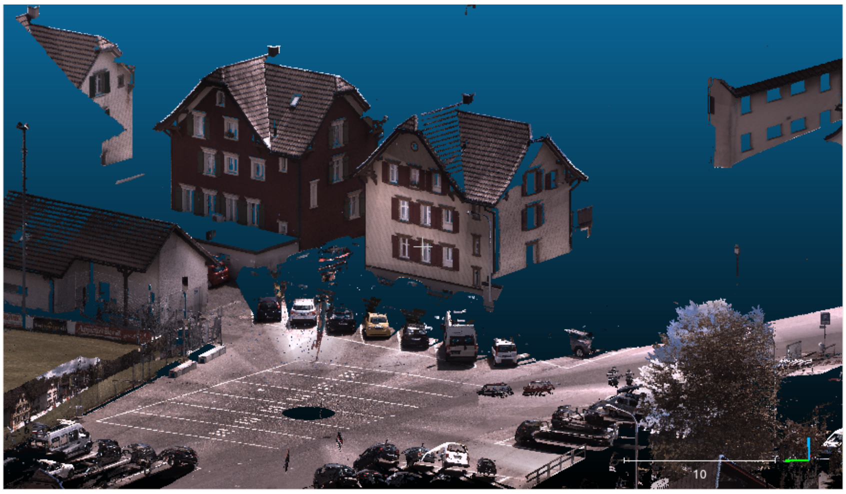

Figure 2.2: Data set: *Sg27_station9_intensity_rgb* with RGB colors

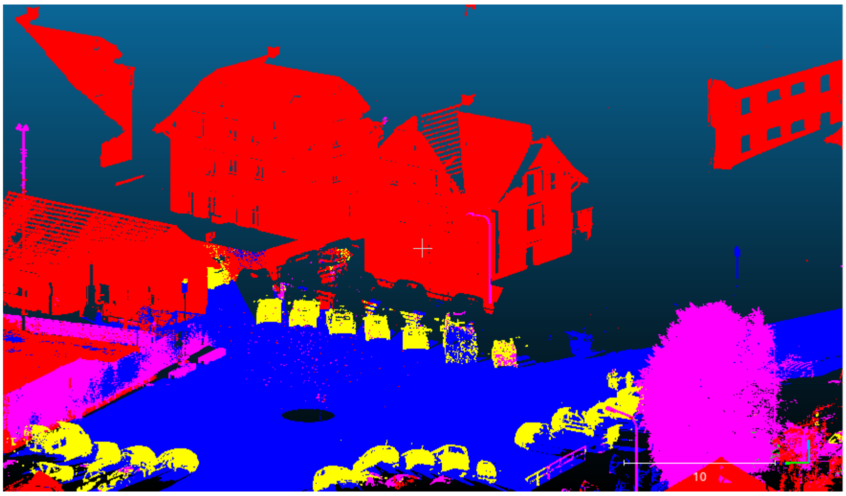

Figure 2.3: Result of segmentation. Data set: *Sg27_station9_intensity_rgb* data set with class labels colors. The point cloud is colored based on the classes in table 2.4

In table table 2.5 the proportion between pre-labeled points and predicted points is documented. The diagonal (the grey boxes) describes the share of the total points that have been predicted correctly as the given class.

		Predicted labels								TS[3] %	Accuracy
		1	2	3	4	5	6	7	8		
Tl[1]	1	17.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	17.3	99.4
	2	0.3	7.9	0.0	0.0	0.0	0.0	0.0	0.0	8.3	94.6
	3	0.0	0.0	17.5	0.4	0.2	0.0	0.0	0.0	18.2	96.3
	4	0.0	0.0	0.6	0.4	0.3	0.3	0.0	0.1	1.7	21.4
	5	0.1	0.0	0.1	0.1	46.8	0.2	0.0	0.0	47.4	98.8
	6	0.1	0.1	0.0	0.1	0.9	2.8	0.4	0.1	4.4	63.1
	7	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.3	75.7
	8	0.1	0.0	0.0	0.0	0.0	0.1	0.1	2.1	2.4	89.1
PS[2] %		17.8	8.0	18.3	1.0	48.4	3.5	0.7	2.3	Overall accuracy	
Precision		96.9	98.8	95.7	36.3	96.7	79.1	28.6	91.9	94.9	

Table 2.5: Normalized confusion matrix for *Sg27_station9_intensity_rgb* (Hackel et al. 2017). Explanation for abbreviations: [1]: True labels, [2]: Predicted shares, and [3]: True shares.

The diagonal elements in relation to the true shares generally look successful, especially because the overall accuracy is almost 95 %. Only label 4 (low vegetation) has a very poor accuracy. This could indicate that it is hard to classify low vegetation with the present model. The model is a result of the training data and if this low segmentation category is not very well represented in the training data it affects the model's ability to segment low vegetation. Furthermore, low vegetation can be hard to generalize, which is why it might be hard to classify. Interestingly class 7 (scanning artefacts) has decent accuracy but a very low precision, meaning that the model confuses other classes with this class. Even though the result of the diagonal elements is satisfying overall, the IoU must be studied further. The IoU for each class can be seen in table 2.6.

Class	1	2	3	4	5	6	7	8	mIoU
IoU	0.96	0.94	0.92	0.16	0.96	0.26	0.54	0.83	0.70

Table 2.6: IoU for *Sg27_station9_intensity_rgb*

Generally, the IoU's show the same tendencies as the respective accuracies for each class show in table 2.5 - some classes have been classified very well, while class 4, 6, and 7 have been classified poorly.

To gain a better understanding of where the errors are happening in the point cloud, it is also possible to color the points in the point cloud based on whether each point has been predicted correctly or not. This functionality is not included in the GitHub repository or as a standard way of evaluating the performance of segmentation models. It is however relatively easy to subtract the true labels from the predicted labels - if this subtraction results in 0 then the points class has been predicted correctly, and if it is anything else the point's class has been predicted wrongly. Using this information the points can be colored based on whether or not the prediction is correct. The point cloud shown in figure 2.4 is a "true/false point cloud" for the segmentation of the point cloud described above. The error point cloud is calculated using the script *deviation_colors.py*, which can be seen in appendix A with ref.no. 2.1.

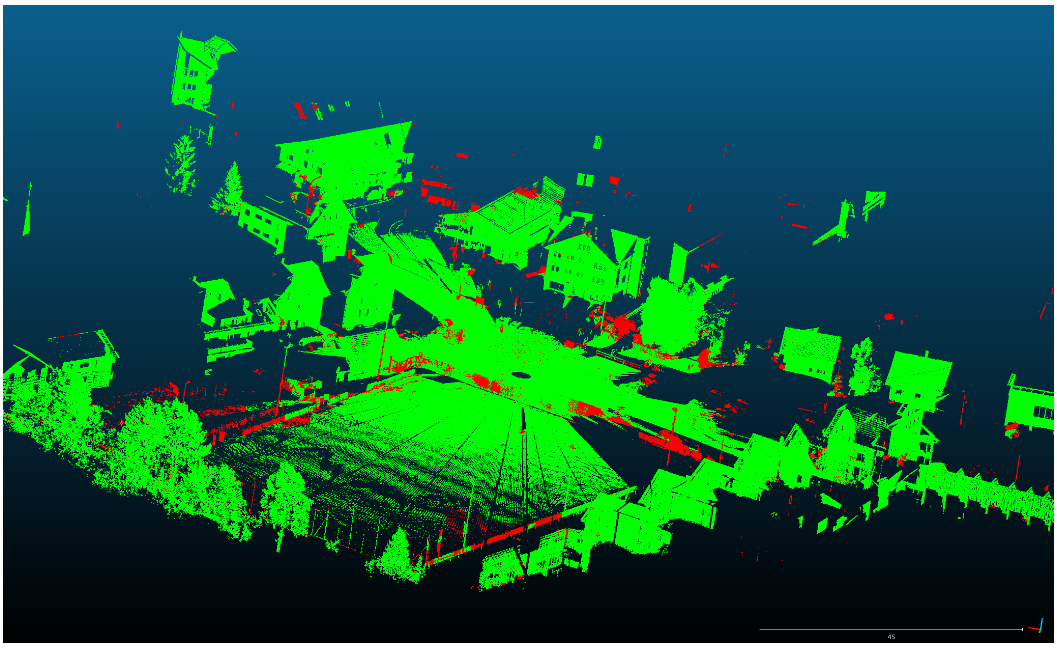

Figure 2.4: True/false plot of the segmented point cloud. Data set: *Sg27_station9_intensity_rgb*. The point cloud is colored green for points that have been predicted correctly and red for points that have been predicted wrongly.

Many of the red points in figure 2.4 seem to be points that are not in direct relation with points all the way around them, possibly because many of these points indicate "incomplete" objects, or because the missing surrounding geometry makes it harder for the model to guess which class these points belong to. Based on this figure, the overall accuracy, and the IoU, this segmentation can be concluded to have been mostly successful, with only few errors mostly centered around segmentation of low vegetation and hard scape.

Custom example - Aalborg

As described above, the first example is from the same source as the training data set, which is why it would be interesting to segment a completely different point cloud but still with an urban scene. LE34 Aalborg has lent out a point cloud for testing usage. This point cloud is also produced from a laser scanner. The scan covers a part of Vesterbro and Borgergade in Aalborg City. An image of the raw scan is documented in figure 2.5 and 2.7. In this example, it will be tested if the segmentation works on this data set as well. The segmented point cloud is shown in figure 2.6 and 2.8.

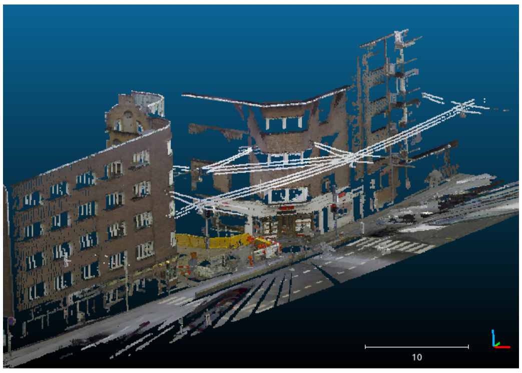

Figure 2.5: Scan Vesterbro with RGB-values seen from south east

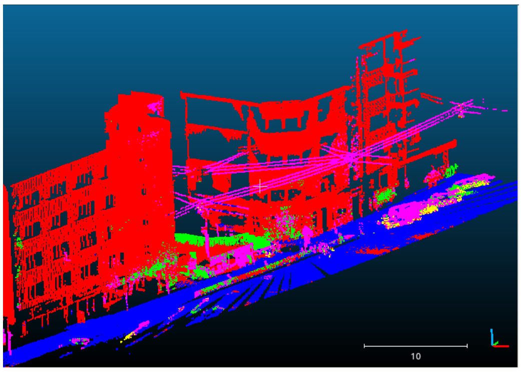

Figure 2.6: Scan Vesterbro with class-values seen from south east

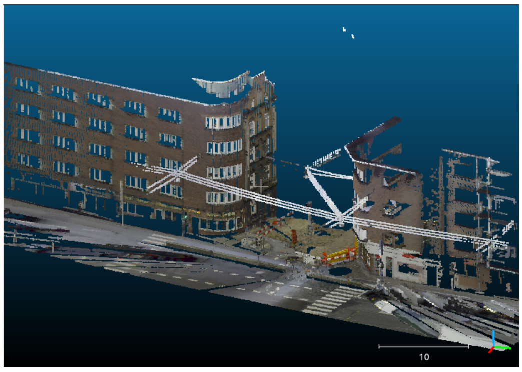

Figure 2.7: Scan Vesterbro with RGB-values seen from north east

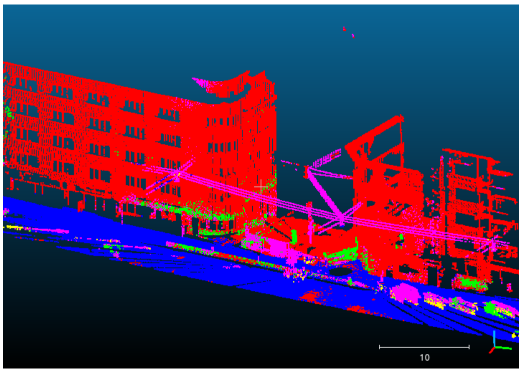

Figure 2.8: Scan Vesterbro with class-values seen from north east

In figures 2.5, 2.6, 2.7 and 2.8 the same point cloud is visualized from two different angles and colored in RGB-colors (camera colors) and colored by point classification values. Again, the result of the segmentation is overall satisfying but with more errors than the other example which is expected. Especially on the buildings on the east side of Vesterbro, the segmentation has some errors. It has predicted facade walls as low vegetation which is wrong. Furthermore, the road is not very well segmented since it is segmented as a combination of buildings and man-made terrain. This is especially visible in the perspective shown in figures 2.9 and 2.10.

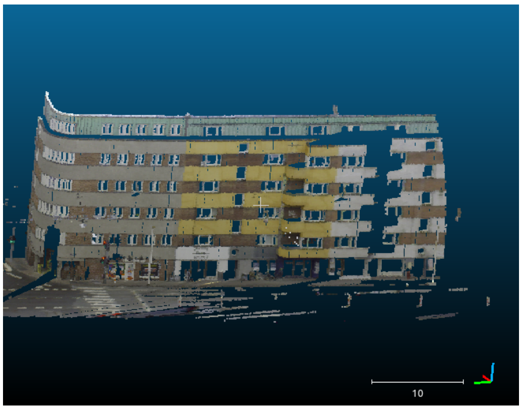

Figure 2.9: Scan Vesterbro with RGB-values seen from west

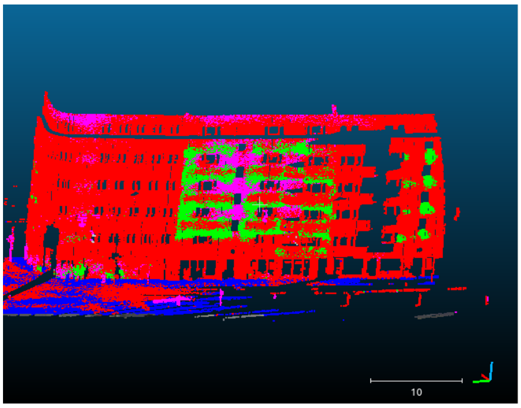

Figure 2.10: Scan Vesterbro with class-values seen from west

Even though there are errors in the segmented point cloud from Vesterbro in Aalborg, in general, it works well. The result could be even better if the model was trained on data that resembles the data from Vesterbro. The type of scanner might for example be relevant, as RGB values and intensity values can vary from laser scanner to laser scanner. In an urban setting like the one tested in this section, the environment and architecture may also be a factor, as the appearance and geometry of buildings and other elements can vary from country to country or place to place. In general, the segmentation classifies buildings and man-made terrain well. This example confirms that semantic segmentation has potential in the evaluation of point clouds as it is possible to augment the point cloud with information about what semantic class each point belongs to.

2.3 Conclusion on the initial problem

It can be concluded from the research in this chapter that it is possible to segment a point cloud by using the GitHub repository Open3D-PointNet2-Semantic3D (Lao 2019) with Deep Learning methods. The segmentation can be run with the associated test data but can also be run with custom point clouds which makes it interesting to research further. The results from the two above examples were overall satisfying, however, more segmentation errors were detected in the point cloud data from Aalborg. An explanation for this could be that the model is trained on scenes that are significantly different from the point cloud from Vesterbro, be it in the type of scanner used or the appearance of the environment in itself. Accordingly, it is important to train the model on representative data since it influences the model's ability to segment point clouds. As such, it may be relevant to produce custom training data that is directed towards a specific kind of segmentation; both in terms of the environment and the semantic classes. Custom training data is not necessarily constrained to the 8 semantic classes defined in the Semantic3D dataset, and as such it is theoretically possible to train models that are specialized in handling specific types of segmentation assignments.

The purpose of this chapter was to research if semantic segmentation works and is implementable in practice, which can be confirmed. Therefore, it would be interesting to study further how these methods can improve the evaluation of point clouds in land surveying companies.

Chapter 3
Problem Statement

In the initial problem, it is briefly researched how to segment point cloud with deep learning technologies. It was concluded that it is possible to segment point clouds with an approach called PointNet++, implemented through the GitHub repository Open3D-PointNet2-Semantic3D. This method can also be used on custom point clouds if the scenes segmented are similar to the scenes used to train the model. However, this is not necessarily ideal - both because the environments in Denmark differ from the environments that the Semantic3D data has been collected from, but also because the 8 predefined classes in this dataset are not necessarily relevant in land surveying perspectives. As such, it seems reasonable that to use this segmentation approach in danish land surveying companies it is relevant to train a model on custom data that has been classified into custom classes directed towards use on specific types of assignments. But how is this performed, how will it influence the assignments, and what quality of segmentation can be expected from this? To direct the project towards researching these points a general problem statement is set up:

> **How can semantic segmentation via deep learning be used in danish land surveying companies?**

The problem statement is meant to find out how semantic segmentation can be implemented and used in a traditional land surveying perspective. This means that it will be relevant to examine which types of land surveying assignments that involve point clouds, how to use semantic segmentation in the workflow of these assignments, and assess the quality of a segmentation used in this context. Ultimately, it will be relevant to assess the potential that semantic segmentation has from a land surveying perspective.

In the above, "quality" refers to the performance of the segmentation, cf. the evaluation parameters described in 2.

Three research questions have been formulated to substantiate the answering of the problem statement. These three research questions are formulated below and will individually contribute to answering the problem statement in a joined conclusion of this report.

Research questions:

1. What types of assignments involving point clouds are provided by surveying companies, and how can semantic segmentation hypothetically be used to improve the workflow of these assignments?
2. How can a semantic segmentation model directed towards a specific land surveying assignment be trained and how does such a model perform?
3. Which potentials and challenges are tied to the use of semantic segmentation models in a land surveying perspective?

Research question 1 will be used to clarify the types of assignments involving point clouds that surveying companies provide and then examine how semantic segmentation hypothetically can be used to help solve these assignments. This will be used to set up criteria for how models aimed at segmenting each type of point cloud assignment can be trained. Thus, this research question is used to direct the training of models that can segment each specific point cloud assignment. A specific type of assignment will be chosen accordingly as the focus for the practical implementation of semantic segmentation. One type of assignment is chosen as the focus, as the methodical process for training and evaluating a semantic segmentation is the same for all models regardless of the type of assignment.

Research question 2 will concern the practical challenges of training a segmentation model that is targeted towards improving the point cloud processing of a specific type of point cloud assignment relevant in land surveying businesses. The performance of the model will be assessed using the evaluation parameters introduced in chapter 2. The answering of this research question will include setting up criteria for the segmentation model, substantiation of the decisions made regarding the labelling and training process and the model will ultimately be evaluated based on results from the training and the segmentation.

Based on the experiences obtained in the two first research questions the third and last research question is about clarifying the general potentials and challenges related to the use of semantic segmentation in a land surveying perspective. Thus, the research question will concern evaluating and assessing the potentials and challenges of semantic segmentation in a land surveying perspective based on results and experiences from research question 2.

When the research questions are answered it is documented which land surveying assignments semantic segmentation can be applied for (research question 1). The implementation of semantic segmentation for a specific task and the performance hereof is documented through research question 2. Lastly, the potentials and challenges that semantic segmentation has in land surveying are assessed through research question 3. Based on these findings the problem statement is answered.

CHAPTER 4

Method

This chapter will describe and discuss the method and structure of this report and how the problem statement and the three research questions stated in chapter 3 will be answered. The problem statement is formulated based on the concluded points of the initial problem and general knowledge about the challenges when processing point clouds. Figure 4.1 illustrates the report structure and how the project is put together. The circles in figure 4.1 refer to which chapter the content is in. The rectangular boxes illustrate the content of each chapter.

The project is built up in chronological order where each following chapter depends and builds on the previous chapters, with no exceptions. The report begins with an introduction in chapter 1 and will be rounded off by putting the project into perspective in chapter 10. From this chapter and onward, the report will be focused on answering the general problem statement by answering the more specialized research questions. The general methods used to answer these research questions will be described in the section below. Research questions 1 and 3 are answered in their respective chapters, whereas research question 2 is divided into two chapters (6 and 7). The more concrete method for answering each question will be described in their respective chapters for research questions 1 and 3, for research question 2 the concrete method is described in chapter 6. After answering each research question, the conclusions from each question will contribute to answering the problem statement in the conclusion of the project in

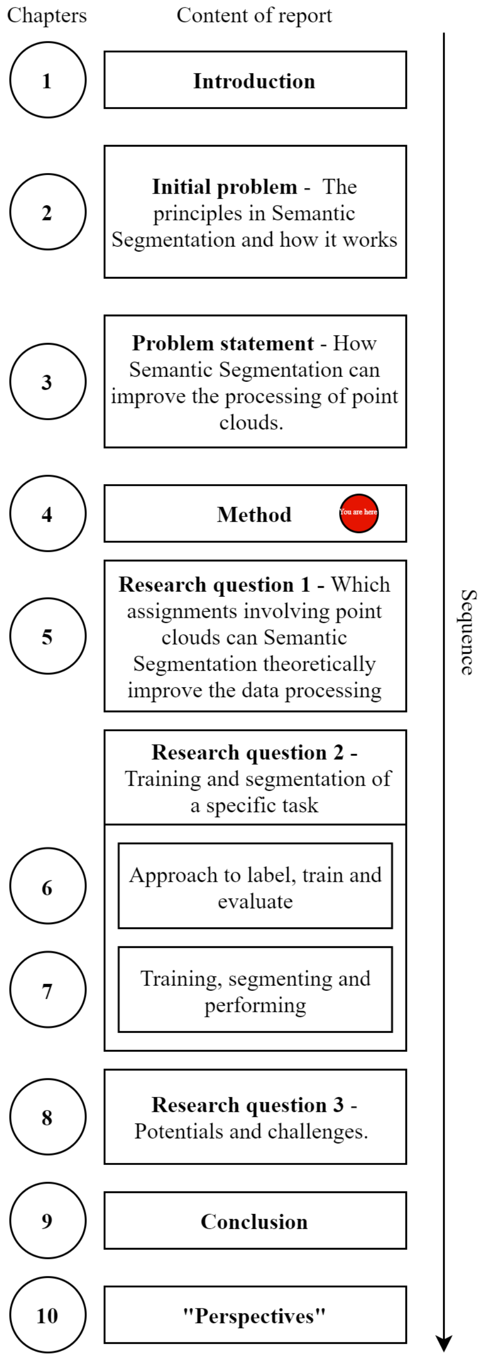

Figure 4.1: Report structure

chapter 9. Finally, the project will be rounded off in the "Perspectives" chapter where the findings of the project will be discussed and put into a perspective.

4.1 Method/approach to answer the research questions.

This section will describe the overall method and approach to answer the 3 research questions formed in chapter 3.

Research question 1
The first research question is about clarifying the types of assignments involving point clouds that theoretically can be improved by using semantic segmentation in the data processing. To achieve knowledge about which point clouds solutions surveying companies offer, interviews with two major surveying companies, LE34 and GeoPartner, have been set up. These companies offer a wide range of 3D solutions services including point cloud data editing and collecting, which is why it is relevant to interview specialists from these companies. The purpose of the interviews is to clarify which tasks involving point cloud data that they provide. Furthermore, the results of the initial problem will be presented for them to find out if they see a potential in using semantic segmentation using deep learning, if they have similar solutions that they use now, or if they have ideas for usage in specific tasks. The interviewing method is preferred to achieve knowledge about point cloud assignments performed in practice since this project is intended to research how semantic segmentation can be implemented in "real" surveying tasks.

When it is clarified which tasks usually contain point cloud data and how the data is evaluated/processed, the next step is to research if semantic segmentation hypothetically can improve the processing of these tasks. This is assessed through information about how each task is processed from the interviews and by combining this with the information about the segmentation of point clouds achieved in the initial problem, chapter 2. Each task likely has different demands or criteria, and therefore the segmentation as shown in chapter 2.2 will probably work with varying success depending on the task. To get better results the model must be designed for the specific task, so it has its own trained model (experience-file) that can be used in the processing of that specific task. Some tasks may be similar in properties that are relevant to extract and in the type of environment the point cloud represents, and may as such be able to use the same model. Therefore, the point cloud tasks will be categorized based on which model the point cloud can be segmented with. Finally, one of the categorizations is chosen as the focus for further work in the following research question where semantic segmentation will be practically implemented in the workflow of this type of point cloud assignment.

Research question 2
The second research question is about finding out how semantic segmentation can be implemented practically for a chosen point cloud task. Which task the segmentation

will be implemented for, is chosen at the end of chapter 5. The analysis in training and segmentation is based on collected point cloud data which will simulate real-world land surveying tasks. The answering of this research question is divided into two parts, a methodical part in chapter 6 and the specific implementation in chapter 7.

Based on the findings in research question 1 some criteria for the specific task will be formulated. These criteria are meant to define the function of the segmentation model, i.e how detailed the model is, the level of abstraction (numbers of classes), etc. These parameters are important to clarify before the segmentation model is designed so it will work appropriately compared to the criteria.

In the design of the model, the major factor is the training data (labeled point clouds). Before the training point cloud can be chosen and labeled, the type of scene that the point clouds describe must be defined, and also the level of abstraction that the data is labelled with. When the scene is chosen, it is important to find representative point cloud training data. Accordingly, some scenes in the area in and around Aalborg are chosen and scanned to produce some training material. Furthermore, a scene for evaluation data will also be found and scanned. This point cloud functions as independent test data and will therefore be segmented by the model trained using all the other data. The chosen scenes will be documented and the characteristic of them will be described.

When the training data set has been collected the labeling of them can begin. Some point cloud editing and visualization software will be used to label the point clouds. The number of classes must be defined before the labeling begins. The criteria from earlier will be used to define how many classes are required for this segmentation model. How this is done is described in its respective section.

The training and evaluation process that is found through the answering of this research question will be general for training and evaluation of semantic segmentation models. This means that this process in principle can be used for the training of an entirely different segmentation model that handles another type of point cloud assignment than the one chosen as the focus in this research question.

Research question 3
The third and last research question is related to examining the potentials and challenges of using semantic segmentation in a land surveying perspective.

The achieved knowledge about which tasks semantic segmentation can be applied for and the quality of a segmentation is the starting point of answering this research question. This knowledge obtained in research questions 1 and 2 is used to describe the actual capabilities and also to describe the limitations of semantic segmentation. These capabilities and limitations of the semantic segmentation are concrete and factual and are an important parameter when evaluating the overall functionality of the segmentation tool. Though, one of the main purposes of this research question is to find out if the semantic segmentation can be used from a land surveying perspective. Therefore, the author's own experiences regarding the use of semantic segmentation will be used to point out some potentials and challenges regarding the use hereof. Furthermore, land surveying companies will be presented with the results from research question 2. In the correspondences with the companies, the key points are what potentials and challenges they see in the use of semantic segmentation in a land surveying perspective now that more specific results are obtained via research question 2. The intention is that the professionals that are interviewed will provide feedback on the results, both in terms of potentials that segmentation models have and which challenges or issues must be taken into account when using segmentation models in land surveying. Through the potentials and challenges identified by the authors and through the correspondences it will be possible to categorize the potentials and challenges and thereby answer the research question in chapter 8. More specific methodical choices will be described in chapter 8.

CHAPTER 5

Processing of point clouds

This chapter will be used to document the answering of research question 1, *What types of assignments involving point clouds are provided in surveying companies and how can semantic segmentation theoretically be used to improve the workflow of these assignments?* As the research question suggests, the question is divided into two parts; a part describing the typical assignments that involve point clouds in land surveying companies and a part meant to discuss how semantic segmentation hypothetically can be used to streamline the processing of these assignments. This research question will also be used to choose one of the types of surveying assignments involving point clouds as a focus for the rest of the report.

In figure 5.1 the sequence and approach for answering this question are illustrated.

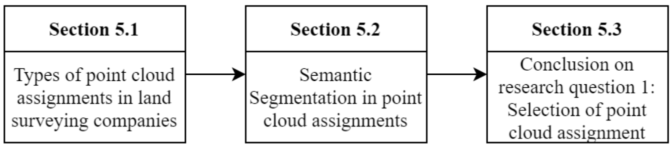

Figure 5.1: The overall approach of answering research question 1

5.1 Point cloud assignments in surveying companies

As written in chapter 4 this research question will be answered based on interviews with the land surveying companies LE34 and GeoPartner since these are the two largest land surveying companies in Denmark, and since it is known that they use point clouds in their respective businesses. The representative from LE34 was Malte Holm-Christiansen from LE34's offshore department, while the representative from GeoPartner was Morten Hellemann who is a 3D specialist in GeoPartner. The interviews with the two companies were informal and unstructured and set up as a conversation. The only planned elements in the interviews were that the talk should include questions about the type of point cloud assignments that each of the representatives work with in their respective companies, how they perform the processing of these point clouds and if they see a potential for using semantic segmentation in these assignments. Summaries for respectively the interview with LE34 and GeoParner document the contents from the interviews and can be found in the zip-file with the ref.no. 5.1 and 5.2, see also appendix A. The below text is based on the representatives' answers.

Types of assignments

The below is an attempt at aggregating the different types of point cloud assignments that are provided in LE34 and GeoPartner.

- Point clouds of buildings both indoors and outdoors
- Point clouds of building facades, roads, and other elements in an urban environment
- Point clouds of longer stretches of roads and the surrounding elements
- Point clouds of industrial elements like oil platforms, train tracks, factory floors, or pipes
- Point clouds gathered from UAVs for visualizations or terrain modelling

This list is non-exhaustive but captures some of the more prevalent point cloud assignments.

Workflow

Each of these tasks requires different processing to reach an end product that is fit for use by the customer. Generally, the process includes preliminary processing (registration of the point clouds), assignment-specific processing or calculations, for example, spatial subsampling, preparing the point cloud for vectorization in software like Revit or AutoCAD, or performing various analyses based on the point clouds. How much of this processing is needed is very dependent on the individual customer, as some customers, like architects, may prefer the registered point cloud as a product, while others may only want vectorized data in a CAD format. None of the two representatives use semantic segmentation of any kind in their current workflow with point clouds.

The part of this workflow that semantic segmentation may help improve is the process after the point cloud has been registered. According to both LE34 and GeoPartner, the main potential of semantic segmentation in this process lies in filtering out unnecessary data from the point clouds. If the model can semantically label each point in the point cloud then each class, or group of classes, can be exported as individual point clouds. As such, the original point cloud can be split into multiple different point clouds. A scan of an urban environment can for example be split into a point cloud containing building facades, a point cloud containing the road, a point cloud containing signs and lampposts, and a point cloud containing everything else, i.e the unnecessary elements (dependent on the specific assignment but for example vegetation, cars, bikes, people, etc.) and noise. These point clouds containing only a few semantic elements are easier to work with as there are fewer elements to consider when performing the manual labor. They will also take up less memory on the computer because of the fewer points, which enables more the computer to work with larger areas of data while also lowering the import/processing/export times of the data.

From the above, it can be inferred that the main advantage of semantic segmentation in point cloud processing in land surveying companies is the ability to automatically filter out unnecessary data from the point cloud so that the point clouds can be more focused on the elements that are important for the task at hand.

This is somewhat in contrast with the authors' preconception of the advantage of semantic segmentation; that the semantic segmentation would allow for isolating very specific elements that need to be vectorized in point clouds tasks. However, according to LE34 and GeoPartner, this is a somewhat unreliable approach, as potential errors in the segmentation are more likely to end up causing the user to make errors when vectorizing the data based on more specialized semantic groups. This is less of a problem when segmenting the point cloud into larger more general groups, as more of a sense of the whole of the point cloud is retained. As such, it is considered more reliable to segment the point clouds into fewer, larger semantic groups of points rather than segment the point cloud into many, smaller semantic groups of points. As the segmentation cannot be expected to be 100 % correct, it is important to not delete the points classified as unnecessary data or noise, as these points can be used to perform quality checks of the segmentation, to make sure that no important features in the point cloud are segmented wrongfully.

5.2 Semantic segmentation in point cloud assignments

Using the above information about point cloud assignments and processing in land surveying companies and the potential that semantic segmentation has in this context according to land surveyors it is possible to consider if any of these assignments can be grouped so that only one segmentation model has to be trained for each group of assignments. In doing this it is relevant to involve the information about the process of semantic segmentation using deep learning gathered in chapter 2. Although an accurate understanding of the process behind semantic segmentation is hard to obtain without a degree in computer science, the general method behind segmentation in PointNet++ is based on finding geometric structures that define each class, and to some degree, colors are used as well. Thus, if point clouds tasks are to be grouped into categories that can be segmented with only one model, they have to consist of the same few semantic classes with consistent geometrical properties, preferably with similar color palettes. Segmenting an urban scene with a model trained on industrial data would make no sense, as the semantic classes and the geometric properties are different. As such, point cloud assignments can only be grouped under the same segmentation model if the environments in the point clouds can be expected to be similar. In figure 5.2 the point cloud assignments identified through the interviews are shown to the left and are grouped based on likeness in environments on the right.

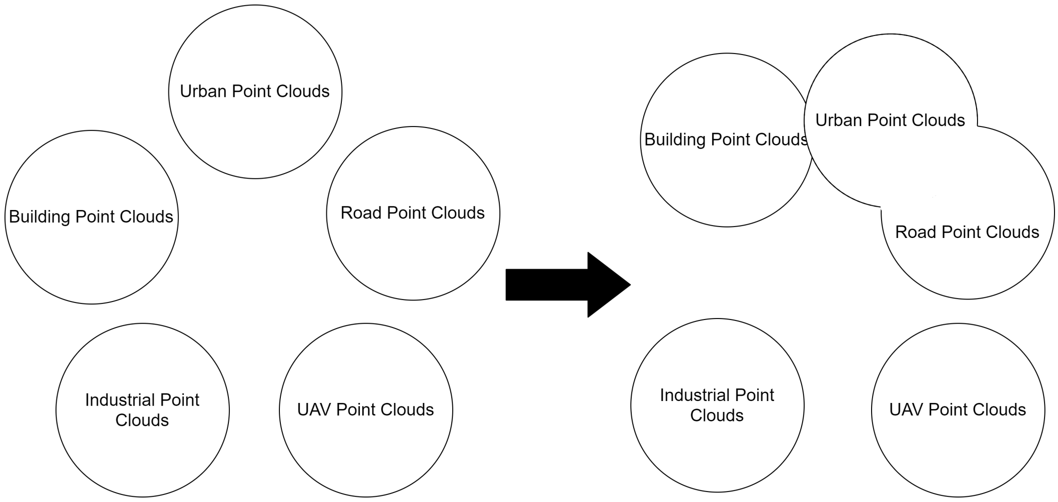

Figure 5.2: The point cloud assignments visualized in no specific order (left) and the point cloud assignment grouped by environment and semantic classes present (right).

As it is indicated in the right side of figure 5.2 urban point cloud and road point clouds overlap. This is because the environment scanned in these point cloud assignments are similar and expected to contain the same semantic classes. A scan of roads is likely performed in an urban environment, while urban environments are likely to contain roads. The point cloud assignment focused on buildings is similar to urban scans, but point clouds of buildings contain indoor data as well, which is not considered to be similar to the outdoor environments as the semantic classes and general geometric properties differ significantly between these point cloud assignments. However, point clouds depicting the facade of buildings contain similar environments as in urban scans and road scans. As such, it may be relevant to split point clouds of buildings into indoor point clouds and outdoor point clouds. This way, the outdoor part of the building point clouds can be set in cohesion with the urban point clouds and the point clouds of roads, while the indoor part of the building point clouds can be segmented using a dedicated segmentation model.

Even though UAV point clouds almost exclusively depict outdoor scenes it is not assessed to be compatible with the segmentation model for outdoor scans. The properties of a UAV point cloud are different from point clouds measured with a scanner because of the difference in measuring method and perspective to the scene. Also, the purpose of UAV point clouds differs from the purpose of regular laser scan point clouds, and as such, there is a need for an individual segmentation model for UAV point clouds.

Industrial point clouds cannot be put in cohesion with any of the other models, because the purpose varies greatly from customer to customer and from assignment to assignment.

Based on the above, a list of segmentation models that applies to a large portion of the point cloud assignments that land surveying companies offer can be devised:

- Indoor point clouds
- Outdoor point clouds
- UAV point clouds
- Industrial point clouds

The goals and criteria for each of these segmentation models are set up below:

Indoor point clouds
Indoor point clouds mostly consist of apartment and house scans. These point clouds are mainly used for project planning and restoration of the interior of existing buildings. In these cases, the important elements in the point clouds are the ceilings, floors, walls, windows, and doorways, as the rest of the interior is removable as a general rule. Therefore, all the interior furniture is irrelevant in these point clouds, and as such a segmentation can focus on segmenting the relevant from the irrelevant data. Training a model to perform this segmentation requires access to indoor data that insofar as possible portrays representative indoor areas. Acquiring this data may be a challenge, as this requires permission to measure in multiple different houses/apartments.

An example of an indoor point cloud can be seen in figure 5.3.

Figure 5.3: Indoor point cloud. Image source: (Yusuf 2019)

Outdoor point clouds
Outdoor point clouds have many different purposes depending on the assignment. In city environments, outdoor scans are used for building facades and road scans among others. In this case, the segmentation could help the processing by dividing the point cloud into different point clouds containing only building facades, terrain, technical elements, and noise. Acquiring data to train a model to do this is relatively easy, as scans of public areas do not require permissions. Thus, the main challenge with training a model to do this lies in collecting representative data, so that a model that can be used for general outdoor scan data can be trained.

An example of an outdoor point cloud can be seen in figure 5.4.

Figure 5.4: Outdoor point cloud. Image source: Own measurements

UAV point clouds

Point clouds from UAVs are often used for generating terrain models used in volume calculations or for visualizations of the area. If the point clouds are used for terrain modelling, then the only points that are relevant in the point cloud are the ones describing the terrain, and as such a semantic segmentation may focus on segmenting the terrain from everything else, be it vegetation, houses, cars, etc.

An example of a UAV point cloud can be seen in figure 5.5.

Figure 5.5: Point cloud generated from UAV data. Image source: (DJM-Aerial n.d.)

Industrial point clouds
Industrial point clouds are often aimed towards specific analyses of different industrial elements, for example, floor analyses, pipe analyses, or analyses of train tracks and the surroundings. Semantic segmentation could help remove all the irrelevant features from these point clouds so that only the features that are meant to be analyzed are present. Data from industrial environments is hard to access, as the data is often confidential, and it is hard to generalize industrial data because the nature of this data varies depending on the specific customer and specific assignment.

An example of an industrial point cloud can be seen in figure 5.6.

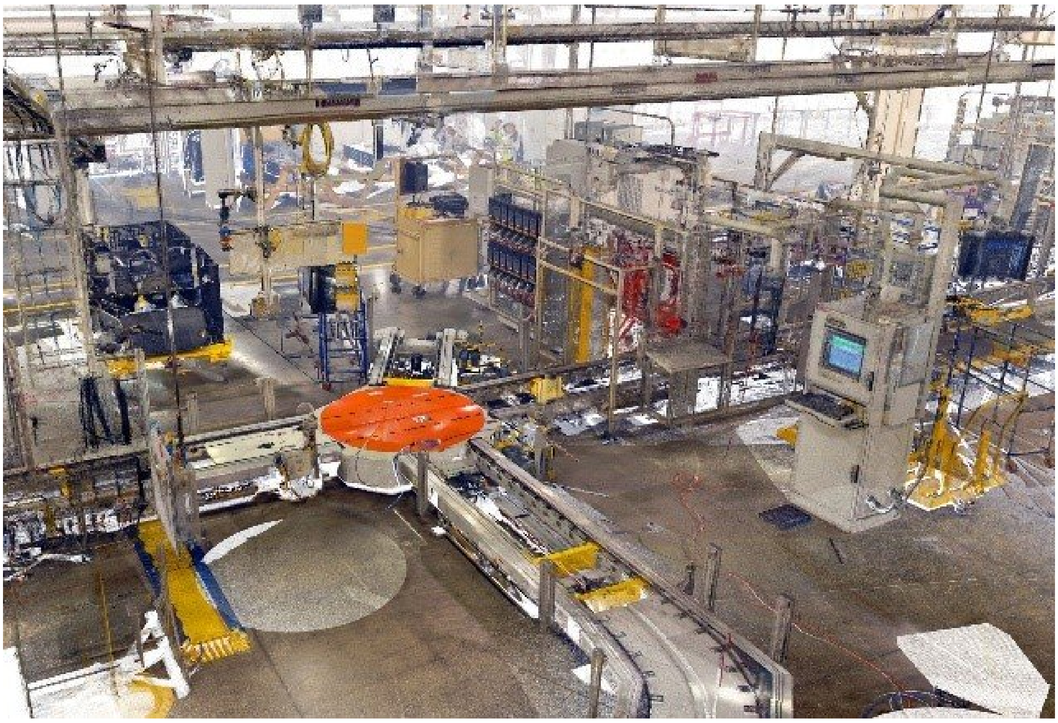

Figure 5.6: Industrial point cloud. Image source: (Bures, Martirosov, and Polcar 2019)

5.3 Conclusion on research question 1

The present sub-conclusion will select the type of point cloud assignment used in the further research in the following chapters 6 and 7. Although it would be relevant and interesting to train semantic segmentation models for all of the above aggregations of point cloud assignment, it is necessary to choose a focus for this project. Based on the above, a specific type of point cloud assignment will be chosen. As training data for industrial point cloud tasks is hard to access and generalize, this category is assessed to be sub-optimal in relation to the others. UAV data is easier to obtain, especially for areas outside cities, and although training a model to segment terrain from vegetation and other irrelevant categories is relevant, it is considered more interesting to investigate the potential of segmentation of either indoor or outdoor point clouds. Experimenting with training a model to segment indoor point clouds would be very interesting, as this could make indoor point clouds more storage-efficient by removing everything that is not walls, floors, or roofs. On the other hand, training a model to segment outdoor point clouds would make it possible to segment point clouds in assignments aimed at, for example, roads or building facades. As such, segmentation of outdoor point clouds offers a broader range of use-aspects than indoor point clouds. Furthermore, it is easy to collect data of outdoor scenes, as this does not require permission from anyone, whereas collecting indoor point clouds require access and permission from the respective owner(s) of the indoor space.

For that reason, *outdoor point clouds* are chosen as the focus for the practical implementation of a semantic segmentation model. Specifically, the focus will be on urban point clouds.

CHAPTER 6 Approach for research question 2

This and the forthcoming chapter regards the answering of the 2nd research question "*How can a semantic segmentation model directed towards a specific land surveying assignment be trained and how does such a model perform?*". The question concerns the practical implementation of a semantic segmentation model. Based on research question 1 answered in chapter 5 it was chosen to aim this segmentation model at segmenting outdoor point clouds. As the implementation of such a model requires making a lot of choices regarding success criteria, gathering, and labeling of point cloud data, choices about the training itself, and the evaluation thereof, the answering of research question 2 is divided into two parts: The current chapter describing and substantiating the approach/methods for doing so and the forthcoming chapter 7 documenting the practical process.

The structure of this chapter is illustrated in figure 6.1.

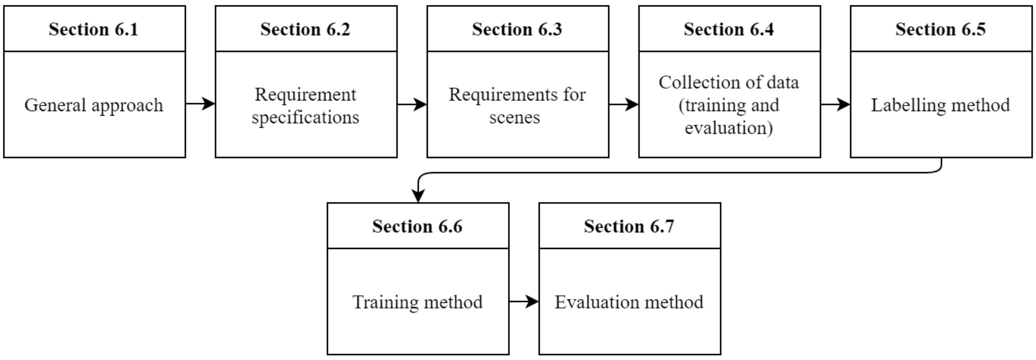

Figure 6.1: Flow diagram showing the structure of the current chapter.

6.1 General approach

Initially, the overall approach will be described for answering this research question. Figure 6.2 illustrates the general structure of the decision-making that lays the foundation for answering research question 2. Firstly, it is relevant to set up some requirements or criteria for the segmentation model, so that the rest of the process can be guided towards living up to these requirements. The next important part lies in choosing the data that will be used in the training of the segmentation model. In this part, the collected data and the reasoning behind choosing this data for training and evaluation will be described. The next step is to label these datasets so that each point in the dataset belongs to a semantic class. This also contains some methodical decisions, described in the respective section. When the point clouds are labeled the training can start. In this part, the considerations related to training a model and evaluating said model will be described. Based on the evaluation it will be possible to assess whether the decisions made earlier could be tweaked to train an even better model. As such, this process is iterative, and multiple models will likely have to be trained.

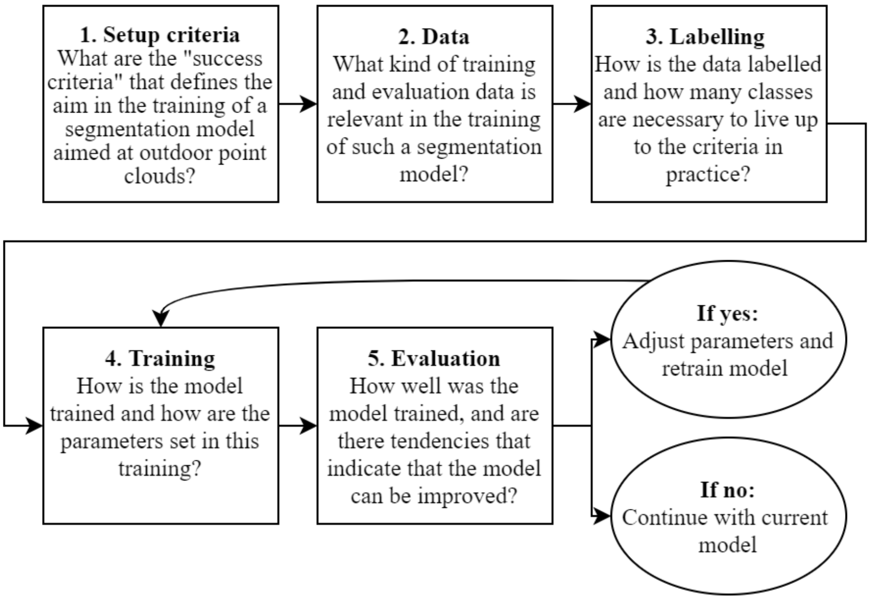

Figure 6.2: Flow diagram showing the process of training and evaluating a segmentation model.

6.2 Requirement specification

This section will be used to set up a requirement specification for the segmentation model so that the rest of the process is guided and targeted towards some specific goals, which makes it possible to make decisions about data, and methodical choices regarding labelling, training, and evaluation.

Through the answering of research question 1 urban point clouds (in urban areas) were chosen as the subject for the training of a semantic segmentation model. This type of point cloud assignment is in most cases focused on either building facades or roads. As such, these are the important classes to be able to identify in a point cloud and these classes should therefore be segmented with high accuracy. Other relevant, although less critical, classes include vegetation and technical facilities like road signs, bus stops, street lights, etc. Everything else can in principle be classified as irrelevant points, as they in most point cloud assignments play no significant role. As described in chapter 5, it is more relevant to segment point clouds into fewer, larger semantic classes rather than the opposite. Consequently, the 4 classes mentioned above are enough classes in themselves, if the rest of the point cloud can then be classified as irrelevant data. Therefore, the ideal model for this task will be able to load in point clouds of a scene and then segment this scene into point clouds containing the above 4 classes and a "noise" point cloud. This principle is illustrated in figure 6.3.

The end goal tied to the training of a segmentation model targeted towards urban point clouds is that it should be viable for processing of point clouds in land surveying companies. Thus, it is also relevant to consider the requirements that land surveying companies would have for the quality of the segmentation model; when is the accuracy acceptable? In practice, it is hard to assess when the accuracy is high enough, or the other way around, when it is too poor to be of any help in the point cloud processing. The model is not meant to be a fully automatic solution that will be the basis of the entire workflow after registration of the point clouds - it is meant as a help tool that aids in speeding up the rest of the workflow by making it possible to load in the relevant subsections of the original point cloud and work on that, without having to spend memory or processing power on irrelevant data. Because of the model's role as a "helper function", high accuracy is not a criterion per se. That being said, high accuracy is still preferable. In the initial problem, a model was trained to have an overall accuracy of around 95 % by training on the benchmark data from Semantic3D. Based on this, it should be possible to reach a comparable overall accuracy of a segmentation model if the training data is of equally high quality in terms of labelling and if the environments in the different scenes have the same general properties. Thus, an accuracy of around 95 % will be the goal. However, it may not be necessary to have a high accuracy across all of the classes. As mentioned above, the important classes are the buildings and/or the roads in many outdoor point cloud assignments. If the model can identify roads and buildings with a high accuracy, then the accuracy of the model's predictions of other classes is less important, as these points in many cases will be discarded anyway. Generally, the accuracy of the segmentation of the different classes should be assessed in relation to the precision of their respective segmentations.

To sum up, the criteria that are used to assess whether the model is usable in a land surveying aspect is based on the model's ability to segment the point cloud into a few larger classes with a focus on roads and building facades. The overall accuracy of the predictions of this model should be around 95 %, although less is fine as long as the accuracy of the segmentation of buildings and roads remains in the vicinity of 95 % - errors in the predictions internally between the other classes is less important in this specific context. The list below sums up the primary criteria:

- $\sim$ 95 % overall accuracy
- At least 95 % accuracy in segmentation of buildings and roads
- Segmentation $\longrightarrow$ Building, roads, vegetation, technical facilities, and noise

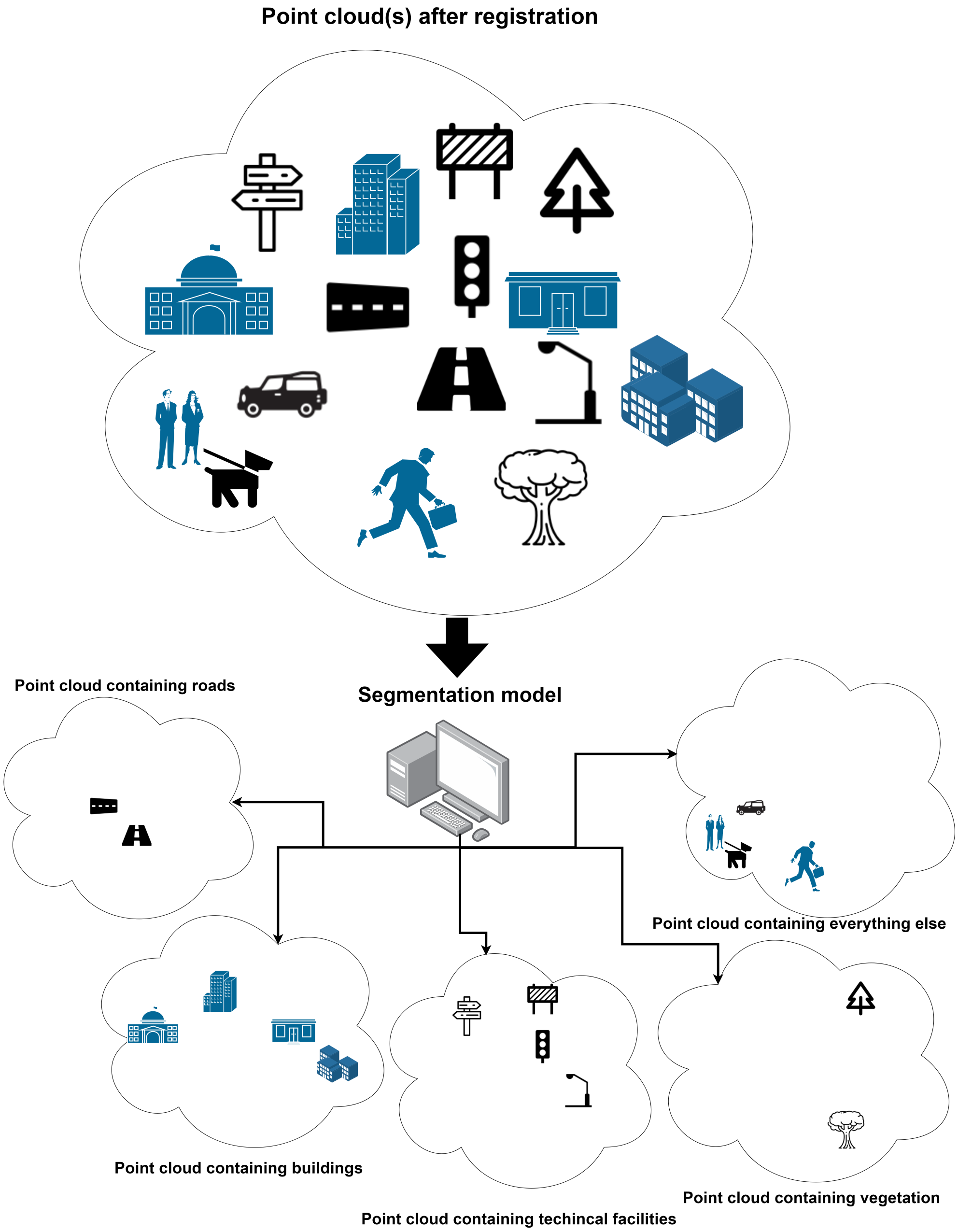

Figure 6.3: Diagram showing the principle of the segmentation model.

6.3 Requirements for scenes

With basis in the above-listed requirements, the present section will design and describe the requirements for the later chosen scenes[1]. It is particularly relevant to set up some criteria for these scenes and their contents to make sure that all the point clouds contain the same characteristics.

The first criterion for the scenes is to define the two kinds of scenes or data, which

[1]The term "scene" refers to the areas where the point clouds are scanned, meaning that for each scene there can be multiple point clouds.

are training data and evaluation data. The difference between evaluation data and training data is very simple. The training dataset is used to train the model to segment while the evaluation data is used to evaluate the segmented model without influencing the training. As such, the evaluation data is independent of the training data.

From the above requirements and some practical considerations about the physical location, the criteria for training and evaluation datasets are listed below. It is assessed that it is relevant to have several scenes for training and at least 1 for evaluation. Based on the requirements set up in the above section, the scenes must contain:

- layout of roads
- different building types
- technical facilities
- mixed vegetation
- must be placed in Aalborg City.

The scenes in the training and evaluation dataset will be chosen with a basis on these criteria. These elements must be contained in all of the point clouds, but in the interest of training a more general model, there should be some degree of variation between the characteristics of each of the elements. I.e. different types of buildings such as variations in shape, architecture, different roads, and vegetation, etc. It is hard to assess how much these characteristics can vary without influencing the model's ability to segment each class negatively. With this in mind, the attempt in this case will be to gather data that contains the above-mentioned layouts in varying shapes and forms intending to train a segmentation model that can segment urban point clouds.

6.4 Collection of data

When the scenes are selected, the collection of data can be performed. LE34 has lent out their Leica RTC360 terrestrial laser scanner for this purpose. The scans will be performed like an arbitrary scanning job. In a real situation, the job dictates the demands for the scan and is often a trade-off between economics and precision. In this situation, the precision is not the primary focus but rather the completeness of the scan, so that elements in the point cloud are well represented so that most of the elements can be identified as whole objects. Each scene will consist of 5-8 individual stations that cover a complete picture of the given scene. As the precision between the point clouds is not a key element, the stations in each scene will be registered using only a cloud-to-cloud algorithm in the software Leica Register360. There is no need to use targets for a more precise registration or for putting the point clouds in a coordinate system. Therefore, the resulting point clouds will contain X-, Y- and Z-coordinates in a local coordinate system. The point clouds will furthermore include intensity values and RGB values.

The Leica RTC360 is a professional grade terrestrial laser scanner that has a fairly high scanning resolution with 2 million points/second and each scan takes 60 seconds (medium resolution). When colored points are required, the scan takes 1 additional minute. The distance accuracy is 1.0 mm + 10 ppm and the angle accuracy is 18" (5.5556 mgon) cf. (Leica n.d.). Again, the accuracy of the scanner is not interesting because the precision of the point clouds is less relevant in this case. It is however still relevant to understand the general properties of point clouds measured with the RTC360.

To make the datasets more computationally manageable all the datasets will be spatially downsampled using PDAL's[2] voxel-centroid-nearest-neighbour downsampling. This operation divides the point cloud into voxels (cubes) with a set side length. The point closest to the centroid of points within each voxel is kept and everything else is discarded from the point cloud. This substantially reduces the file size of the point cloud, while homogenizing the point cloud so that the density of points in the point cloud is maximum the set voxel-size all across the cloud. The lower the side length of the voxels is set to be, the more of the original point cloud is kept, while larger side lengths remove more points from the point cloud. In the default settings for training a segmentation model using the GitHub repository (Lao 2019), the point clouds are also downsampled using a voxel-based method. In this case, the size of the voxels is set to 5 cm. Based on this, it is assessed that a density of one point per cube with a side length of 5 cm (one point per 125 cm^3) is high enough for the deep learning algorithm to be able to distinguish between the geometry of different semantic classes in outdoor point clouds. Therefore, all point clouds in the training and evaluation datasets are downsampled to one point per 125 cm^3 using voxel downsampling. The script used for downsampling the dataset is shown in appendix A with ref.no. 6.1.

The following step is to label these point clouds so that each point in the datasets is assigned a semantic class.

6.5 Labelling method

This section will describe the method and approach for how the point clouds for training and evaluation are labelled. Furthermore, it will describe the methodical choices in the labelling process and how the categories/classes are created.

The manual labelling is necessary so that the deep learning algorithm knows the correct semantic classes of the datasets. This means both the training datasets and the evaluation will be labelled. To do this the software Lidar360 will be used.

[2]pdal.io

Lidar360 is commercial software that provides the possibilities for labelling each point in the different point clouds.

It is important that the labelling of the datasets in this project is performed the same way in all of the point clouds. Mismatches and errors in the way that the point cloud is labelled will "confuse" the segmentation model and are likely to cause errors in the segmentation. Therefore, it is necessary to be meticulous in the labelling so that the labelling is both accurate and precise in relation to the chosen level of abstraction.

Thus, it is relevant to clarify the level of abstraction so that each point cloud will be labelled similarly.

Level of abstraction and categories.
A primary factor when labelling point clouds is the level of abstraction since it directly influences the result of the segmentation. The level of abstraction in the labelling equals the level of abstraction that the segmentation model outputs. This sub-section will describe and discuss why and how the level of abstraction is chosen as it is. The level of abstraction is a general problematic in the production of maps or models since it is a representation of reality. In this context, the abstraction consists of an interpretation of reality, where each point is put into one of a number of predefined categories that simplify the representation of reality.

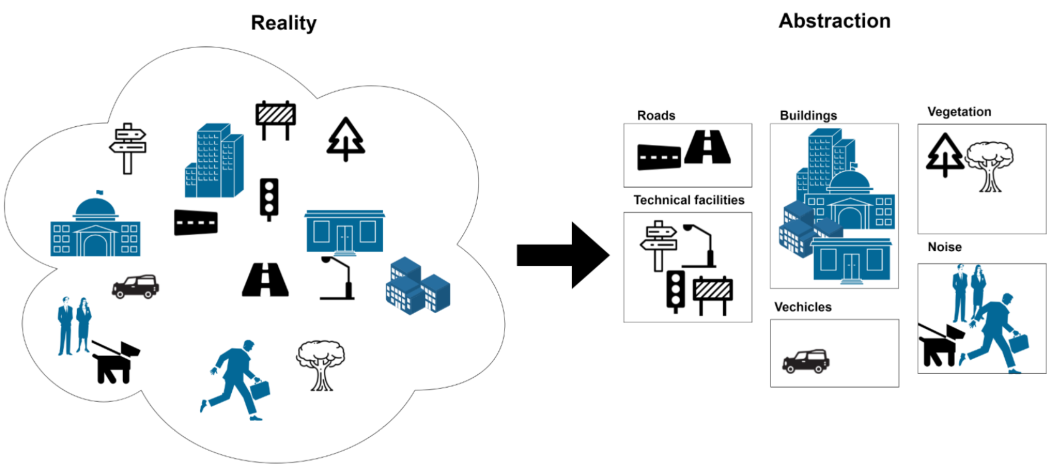

Figure 6.4: Principle of abstraction - When the reality is labelled and divided into categories

The design of the classes used in the labelling of these point clouds is made based on the 5 classes (buildings, roads, vegetation, technical facilities, and noise) defined in the requirement specification set up in section 6.3 on page 38. These classes are also illustrated in figure 6.4. However, a few more classes are added as the geometrical properties of "noise" are diffuse and noise is therefore assumed hard to generalize for the model. Therefore, elements that are unnecessary in the chosen point cloud assignment but have general recognizable geometrical properties are given their own classification. This is the case for vehicles, which are irrelevant in most point cloud assignments but are somewhat geometrically consistent across all the point clouds. Therefore, cars, bikes, trailers, and trucks are labelled as vehicles with the sole purpose of combining this class with the noise class later.

In table 6.1 the classes are documented and their semantic class code and color are described. Below table 6.1 each class's characteristics are described and why that class is included.

Category	Class	Color
Not classified	0	White
Grey terrain	1	Grey
Green terrain	2	Green
High vegetation	3	Light green
Vehicles	4	Yellow
Buildings	5	Red
Technical facilities	6	Blue
Noise	7	Pink

Table 6.1: Classification of labelled point clouds.

Not classified
This class is entirely made for overlooked points in the point clouds. These could be hidden points. Ideally, none or only very few points will be in this class, as they are a result of human error. This class should not be exchanged or confused with the below coming class "noise".

Grey terrain
The category grey terrain covers all sealed surfaces which include roads, sealed footpaths, etc. The reason for having this category is that these points often are relevant for an urban scene.

Green terrain
The other terrain category is green terrain which is characterised by lawns, low vegetation, etc. The idea of having both grey and green terrain is to differentiate these from each other as the properties of green and grey terrain are different. Moreover, green terrain is not often a relevant element in land surveying tasks, whereas grey terrain is important in many tasks.

High Vegetation
The category high vegetation is self-explaining. This category is included to be able to identify potential trees or other high vegetation when processing point clouds.

Vehicles
The vehicles category containing bikes, cars, trailers, trucks, etc. The specific category might not be that useful for a typical scanning task of urban scenes and could therefore be categorised as noise instead. But in an attempt not to confuse the prediction model by categorizing vehicles as noise, vehicles have their own category since their geometrical shape is excepted to be recognisable.

Buildings
The building category is also self-explaining. It contains different types of buildings and it does not distinguish between them.

Technical facilities
The class technical facilities contains elements as signs, fences, airborne cables, bollards, etc. This category is included because these elements can be relevant in certain point cloud assignments.

Noise
Noise is the last category and it includes all the unrecognisable points and objects from the point clouds. An example of this could be reflections from windows, driving cars, etc. Noise can be considered to be the leftover points after labelling everything else in the point cloud. The difference between noise and unclassified points is that noise is actively labelled as noise, whereas unlabelled points are unintentionally overlooked in the labelling process.

6.6 Training method

After the training dataset has been labelled it is possible to begin the training of a segmentation model by feeding the labelled data to the deep learning algorithm. Many different parameters can be adjusted in this process[3], but many of them require a deep understanding of computer science to understand and tune based on substantiated decisions. Consequently, the training will be run with the default values for these parameters. The properties that are relevant and possible to tune in a land surveying perspective are whether the colors of the point clouds are used in the training and for how long the model trains.

As written in chapter 2, the training process is divided into epochs - the iterative process of trying to guess the semantic label of every single point in the dataset, evaluating the correctness of these guesses, and then adjusting how the guesses are made before starting over by guessing again. The idea is that for each epoch the model gets better and better at identifying the points belonging to each of the classes in the dataset until the accuracy of the model stagnates around some number that, preferably, approaches 100 %. As described in section 6.3 on page 38 the goal is to reach an accuracy of around 95 % on custom data. By default, the number of epochs used in the deep learning algorithm in this project is 500. Based on experience from chapter 2 training a model for one epoch takes around 10 minutes, meaning that it would take multiple days to train for 500 epochs with the training dataset collected for this project. This is not optimal, as it may be necessary to train multiple models because of unforeseen challenges with the datasets used. Furthermore, the time the virtual machine is running is billed by Google, which can make the long training sessions expensive.

While training, the model outputs the performance of its predictions in a log file,

[3]See *Semantic.json* in (Lao 2019) for a list of the adjustable parameters

meaning that it continuously outputs the overall accuracy and the IoU for each of the classes that it is trying to guess. Since this evaluation is based on the data that the model has been trained on there is reason to believe that this evaluation is biased. If a dedicated evaluation dataset is specified, however, the algorithm will try to predict the semantic classes of all points in the evaluation data set and compare it to the true labels every 5 epochs, yielding a more independent and unbiased evaluation of the model's performance while training. The dedicated evaluation dataset is only used to calculate the current performance of the model, and the results do not influence how the model predicts points in future epochs. Thus, it is possible to plot how the performance changes over time when training.

Based on the above, the initial training of a model is performed with a lower amount of epochs than the default 500, while still giving the algorithm time to learn from the training data. When the performance of the model is then plotted based on the number of epochs that the model has been trained for, it will be possible to assess when the curve starts to flatten and the performance stagnates. 200 epochs is chosen, as this is higher than the 100 epochs from the initial problem, while still being quite a bit lower than the default 500. From the performance of the model during the span of these 200 epochs, it should be possible to assess whether more epochs are necessary, or if it is possible to stop the training of other potential models earlier because the performance stagnates earlier than 200 epochs. The process of the training is illustrated in figure 6.5.

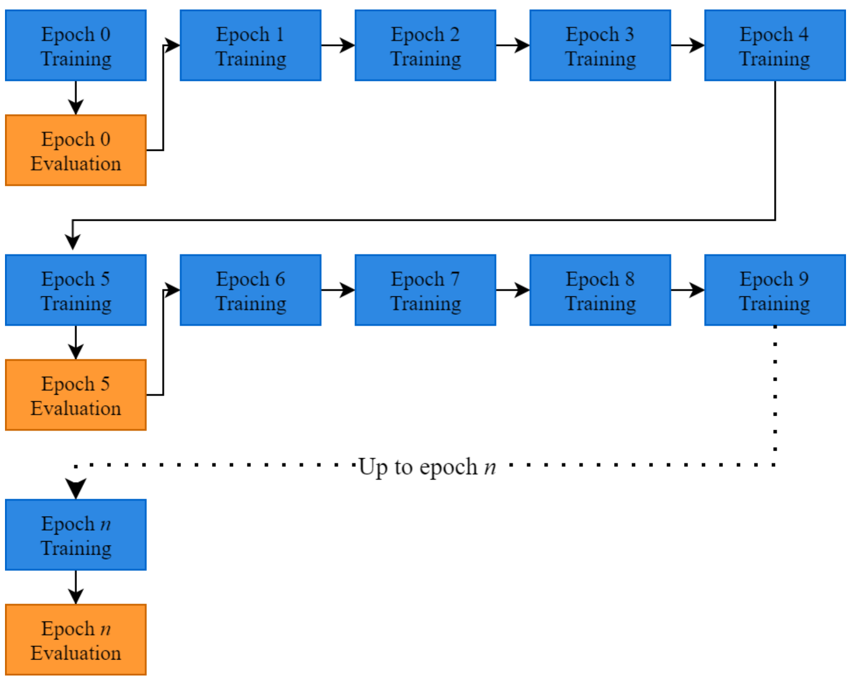

Figure 6.5: The training process illustrated with blocks that represent training epochs (blue) and blocks that represent evaluation epochs (orange). Every fifth epoch an evaluation of the model is performed by guessing the label of points in the evaluation dataset and outputting the results by comparing the guesses with the true labels.

Another choice that has to be made is whether the training of the model should include the RGB colors of the point cloud (default) or should include intensity only. Using RGB colors allow for more detail and may make it easier for the model to differentiate between elements - for example, it may notice that green terrain and vegetation are in green colors, while paved roads and sidewalks are grey-ish. The weakness of RGB colors in point clouds is that it takes longer for the 3D scanner to perform its measurements and that RGB colors are not precise, as variations in light can cause changes in the colors of the point cloud. Also, in the case of the RTC360 used to collect data from the scenes described above, the colors are added to the points by taking photos of the surrounding areas *after* the points have been measured, meaning that there is a temporal mismatch between the points measured and the colors given to them. This can cause errors in the colors of the point cloud if the scene is not relatively stationary, which is not the case in urban environments. An example hereof can be seen in figure 6.6.

Figure 6.6: Example of error in the color of points caused by a pedestrian with a red jacket walking in front of the scanner after the scan was completed, but while the scanner was taking pictures of the scene.

On the contrary, intensity values are more precise as intensity values are independent of the light in the scene and are measured at the same time as the position of the points. Furthermore, from the perspective of a land surveyor, measuring point clouds with colors take longer time and the point clouds take up much more hard disk space. Consequently, the value that colors bring to a point cloud is somewhat negligible compared to the downsides in terms of time and hard disk space that is a byproduct of measuring with colors. For this reason, it is relevant to examine if the semantic segmentation model can be trained on data without color and if the performance of such a model is comparable to the performance of a model trained on data with colors. If this is the case, then there is no reason to train a model on point clouds with colors.

Another element that will be relevant to test in the training process is whether or not to merge the point clouds for each scene into one large point cloud or keep the individual point clouds as they were scanned. Training and evaluating on individual point clouds is the most convenient, as this is how point clouds usually are exported after registration. However, individual point clouds do not give a very complete depiction of many elements, as the elements are only measured from one side, see for example figure 6.7. This may influence the segmentation model's ability to predict the semantic class of points in incomplete objects because the geometry of incomplete objects is very unsteady and is dependent on the angle between the scanner and the object. This tendency was observed to cause errors in section 2.2 on page 12.

In contrast, if the point clouds from each scene are merged, the resulting depiction will be much more complete as objects in the point cloud are measured from more than one angle. This yields a more precise geometry of objects, which may help the segmentation model in its predictions.

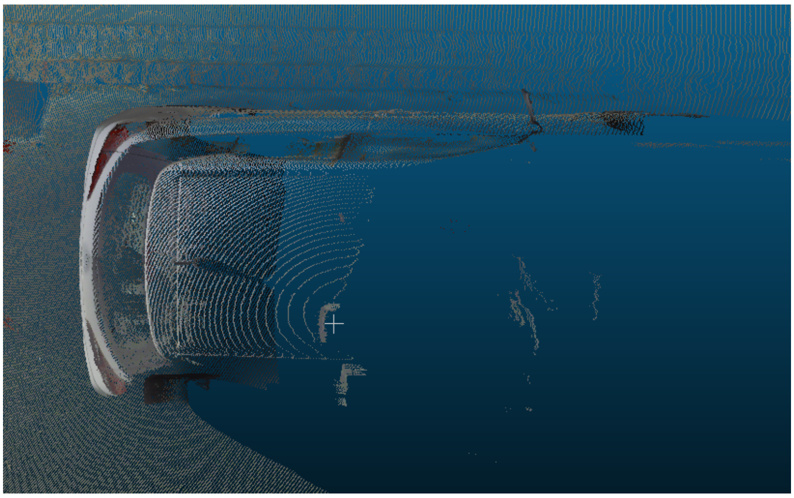

Figure 6.7: Single station point cloud depicting one side of a car.

<u>Based on this at least 3 models will be trained:</u>

- A model trained on individual point clouds with RGB-color
- A model trained on individual point clouds without RGB-color (intensity only)
- A model trained on merged point clouds with or without RGB-color - depending on the training results of the above models[4]

[4]It is considered unnecessary to train a fourth model, as the training of the first two models

6.7 Evaluation method

This section will describe how the segmented point cloud is evaluated (in contrast to the evaluation of training as described above). The result of the above-described training is a segmentation model which can segment point clouds into the 7 classes shown in table 6.1 on page 42 (Unclassified points are not included). The question is how accurate this segmentation is.

The approach for evaluating the segmented point cloud is initially to use the built-in evaluation functions/outputs in the scripts from PointNet++ such as the confusion matrix and the IoU of each class. These are the same evaluation parameters as described in chapter 2 in the pilot study of semantic segmentation. Furthermore, the segmentation will be presented with the colors corresponding to each class as well as the true/false plot to illustrate the location of potential errors. When the segmented point cloud is evaluated with these evaluation tools it should be possible to assess how well the point cloud is segmented.

will show if RGB segmentation is superior to intensity-only or if intensity-only segmentation is of comparable quality to RGB segmentation. Whether or not RGB colors are used in the training of this model will be chosen accordingly.

CHAPTER 7
Training & Segmentation

In the previous chapter, chapter 6, the methodical approach for answering research question 2 is described. The present chapter will be used to document the practical process of training a segmentation model to segment outdoor point clouds and then evaluating its performance by segmenting an outdoor point cloud.

Initially, the training and evaluation scenes will be chosen and described. Hereafter the labelling of the point clouds will be described, especially the challenges related here to. The training can begin when all the point clouds are labelled. The logs from the training will be studied further to be able to assess how the training went. Furthermore, the segmented point cloud will be evaluated by studying the confusion matrix, IoU, and true/false plots. Finally, the last section in this chapter will sum up the key findings and answer the research question "*How can a semantic segmentation model directed towards a specific land surveying assignment be trained and how does such a model perform?*". The structure of this chapter is visualised in figure 7.1.

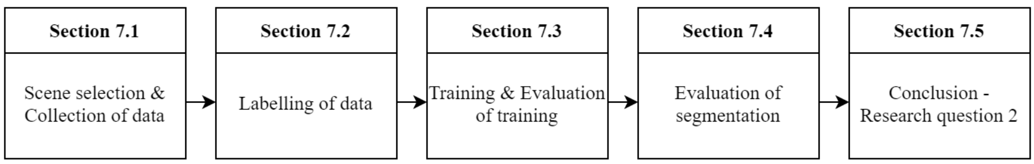

Figure 7.1: Flow diagram showing the structure of the current chapter.

7.1 Selection of scenes

With the "criteria" from section section 6.3 on page 38 in consideration training and evaluation scenes in Aalborg are found. The evaluation scene will be described first, whereafter the training scenes also will be described.

The evaluation scene is a road intersection between Ryesgade and Helgolandsgade in Aalborg. This place complies with the listed demands for evaluation data. Figure 7.2 shows an oblique aerial photo of the test area.

The chosen area is placed in the west part of Aalborg City. The scenes here include road structures, several different building styles, technical facilities, and vegetation. In this scene, it is assessed to be enough different elements to give a satisfactory segmentation to evaluate the properties of segmentation.

Figure 7.2: Test scene - Road intersection between Ryesgade and Helgolandsgade, Aalborg (SDFE 2019). The red polygon indicates the approximate area that the scans cover

7.1.1 Training scenes

Again the training scenes are found with basis in the listed criteria section 6.3 on page 38. Figure 7.3 illustrates the four different training scenes and also the evaluation scene placed around in Aalborg. The characteristic for each scene is described below.

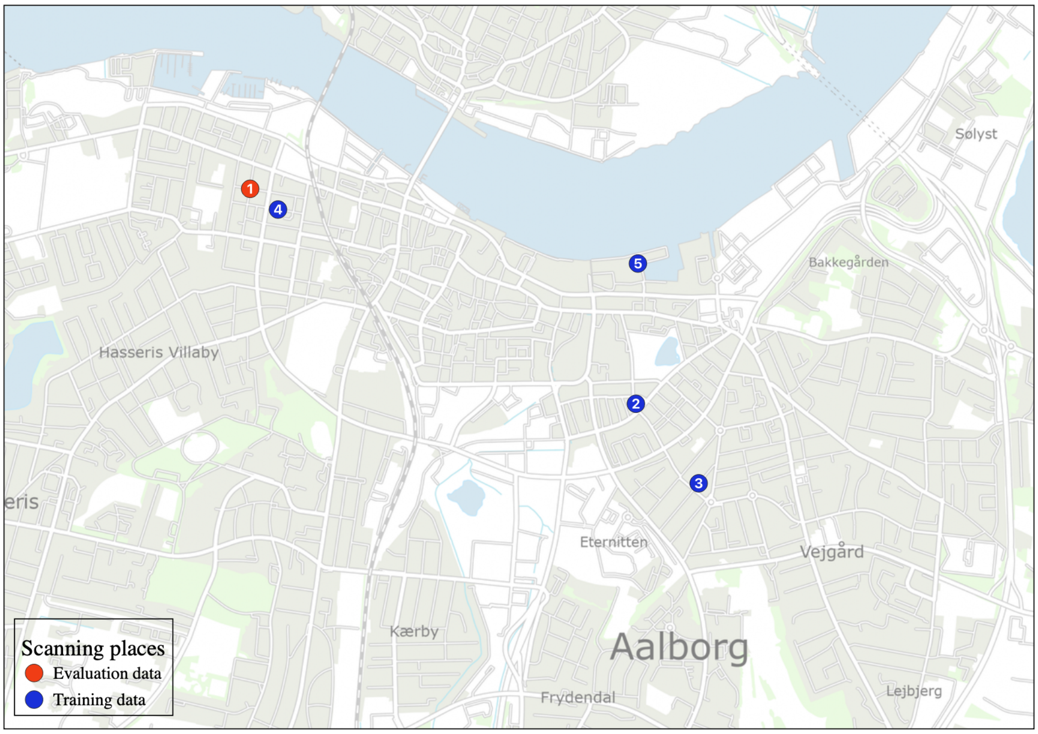

Figure 7.3: Illustrates where the training data is scanned in Aalborg City. Background map: (SDFE 2021)

The four blue places pointed out in figure 7.3 are four different scenes including road and building structures. The reason for choosing these different scenes for training is to reach a more general model that can handle multiple different urban scenes. The four scenes contain different road layouts and different building styles, some are high-density housing whereas others are low-density. Again, this variety is meant to train a more general model. The alternative is to have only one larger training scene, but this will result in a more specific model.

The following will describe each scene briefly to give an understanding of what the scenes consist of and the variety of them. The scenes are chosen to have a representative extraction of elements in typical urban scenes, but could in principle be other scenes in Aalborg.

Scene 2

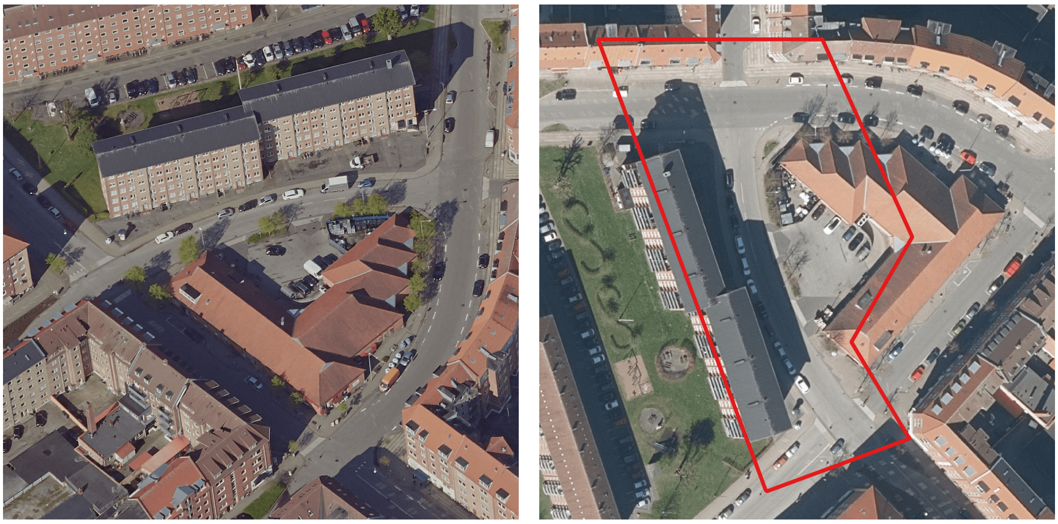

Figure 7.4: Scene 2, Samsøgade, Aalborg (SDFE 2019). The red polygon indicates the approximate area that the scans cover

The first scene illustrated in figure 7.4 is in Samsøgade in Aalborg. Herein the scene contains different building styles some 4-5 floors and some other 1-2 floors, parking places, and different road intersections. This variation of buildings types and roads is interesting to train on since it contains fairly many different objects.

Scene 3

Figure 7.5: Scene 3, Vonsyldsgade, Aalborg (SDFE 2019). The red polygon indicates the approximate area that the scans cover.

In figure 7.5 scene 3 is illustrated, this scene is in another area where the houses are lower and more spread out in the area. There is more vegetation than in the previous scene. This scene contributes with information about the structures of these smaller houses and more vegetation than the other scenes.

Scene 4

Figure 7.6: Scene 4 - Fredericiagade and Dybbølsgade, Aalborg (SDFE 2019). The red polygon indicates the approximate area that the scans cover.

Figure 7.6 illustrates scene 4 which is placed in the west of Aalborg City. This scene primarily contains 4-5 floor buildings, road structures, and a little square. The overweight in this scene is buildings and road structures. These buildings stand out compared to the previous scene because the blocks are generally connected to each other.

Scene 5

Figure 7.7: Scene 5 - Pieren, Aalborg (SDFE 2019). The red polygon indicates the approximate area that the scans cover.

Figure 7.7 illustrates scene 5 which is a newly built area at Aalborg Harbour. This scene is relevant to include in the training data since this building style differs compared to the older building styles in the other scenes.

In addition to these above-described scenes, a scene from LE34 and a scene from GeoPartner are used to train the model. The scene from LE34 is the same as in the pilot study in chapter 2, which is a scene from Vesterbro, Aalborg. The point cloud from GeoPartner is from a road structure containing a few buildings. The scene from GeoPartner will be described as scene 6 and the point cloud from LE34 will be described as scene 7.

Overview of training datasets and evaluations dataset

Table 7.1 sums up the characteristics from each scene. By the table, it is possible to check if elements in the training dataset are also represented in the evaluation dataset which is a prerequisite to get a successful segmentation and evaluation in the end.

Classes/Scenes		Evaluation	Training					
		1	2	3	4	5	6	7
Grey terrain	Road structures	✓	✓	✓	✓	✓	✓	✓
	Footh path	✓	✓	✓	✓	✓	✓	✓
Green terrain	Lawns	✓		✓			✓	
	Low vegetation		✓	✓		✓	✓	
	Green areas			✓			✓	
High vegetation	Trees	✓	✓	✓	✓	✓	✓	✓
	Bushes / Hedges	✓	✓	✓	✓	✓	✓	
Vehicles	Trucks	✓						
	Bicycles	✓	✓		✓	✓	✓	✓
	Cars	✓	✓	✓	✓	✓	✓	✓
Buildings	Single buildings	✓		✓			✓	
	Blocks	✓	✓		✓	✓		✓
	H. buildings $\geq$ (4 flr.)	✓	✓		✓	✓		✓
	L. buildings $<$(4 flr.)	✓	✓	✓		✓	✓	
Technical facilities	Signs	✓	✓	✓	✓	✓	✓	✓
	Airborne cables	✓	✓		✓	✓		✓
	Power cabinet	✓	✓	✓	✓	✓	✓	✓
	Fences	✓	✓	✓	✓	✓	✓	✓

Table 7.1: Characteristics for evaluation and training scenes (including the point clouds from GeoPartner (training 6) and LE34 (training 7). The point clouds for scene 1-5 can be downloaded from the URL: `https://tinyurl.com/SM4MadsMikkel`.

This means that there are 6 training datasets and 1 evaluation dataset - 7 datasets in total. Scenes 1-5 are scanned for the purpose of this project, whereas the two others (scenes 6 and 7) come from respectively GeoPartner and LE34, and these scenes also include urban scenes. The size of the 6 training datasets is generally very similar, at around 300-400 MB per dataset after downsampling, and therefore they are expected to contribute evenly to the training.

7.2 Labelling

When the above categories are defined and scenes are found the actual labelling of point clouds can begin. This is a time-consuming job since in theory each point in the 7 point clouds will be labelled manually. The following will describe some of the challenges concerning the labelling of these datasets. Furthermore, some examples will illustrate how the point clouds look like before and after they are labelled. The first step is to remove very incomplete data in the edges of the point cloud, to reach better overall completeness of the point cloud. Hereafter the point clouds are labelled. The labelling works by selecting the points with a selection-tool and then assigning the points with their respective classes.

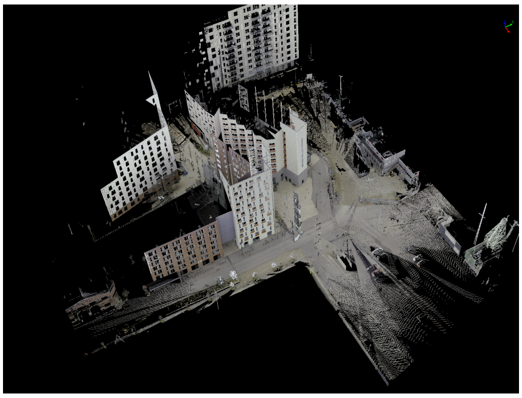

Figure 7.8: Scene 5, Pieren Aalborg in RGB-colors.

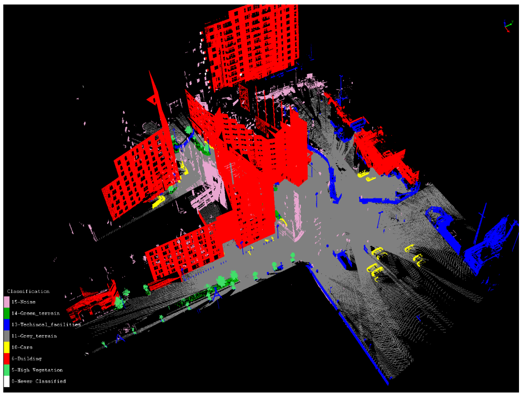

Figure 7.9: Scene 5, Pieren Aalborg after manual labelling.

In figures 7.8 and 7.9 the dataset Scene 5, Pieren, Aalborg is illustrated in RGB-colors and after it is manually labelled. From this, it can be seen that the major categories are buildings and grey terrain. This is typical for all the labelled point clouds.

One of the primary challenges that occur when labelling the point clouds is the definition of the border between two different classes. These borders are generally difficult to define in point clouds, but the downsampling to one point per 125 cm^3 may increase the difficulty of defining these borders. An example of this is illustrated in figure 7.11. It is illustrated here that it can be hard to define the borders between each category.

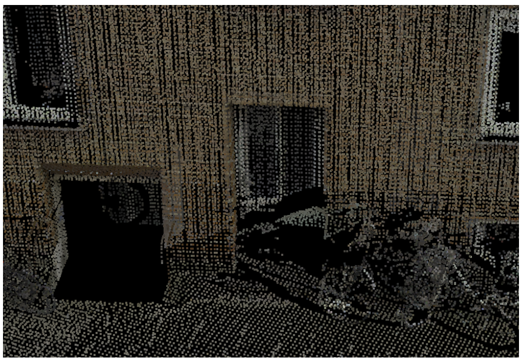

Figure 7.10: The same example as in 7.11 just in RGB-colors. This example also illustrates how hard it can be to distinguish between points from different classes using RGB-colors

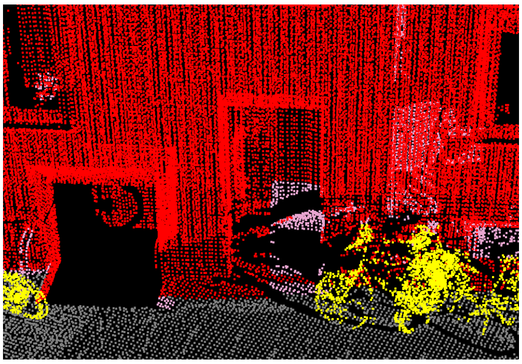

Figure 7.11: Example from scene 4 on a situation where the border between grey terrain and building can be hard to define (in class-colors)

Since these borders can be hard to define it is expected that the prediction also will have trouble segmenting these borders because the segmentation will not be better than the labelled training datasets.

Another dilemma when labelling the point cloud can be the decision if the point will be categorised after noise or after their true value. For example, there is no category "people" but there are several people in the datasets, hence they are labelled as noise. But would it be better to give them their own category? Another example is driving cars during the scanning this results in objects which only are partially scanned, an example of this is illustrated in figure 7.12.

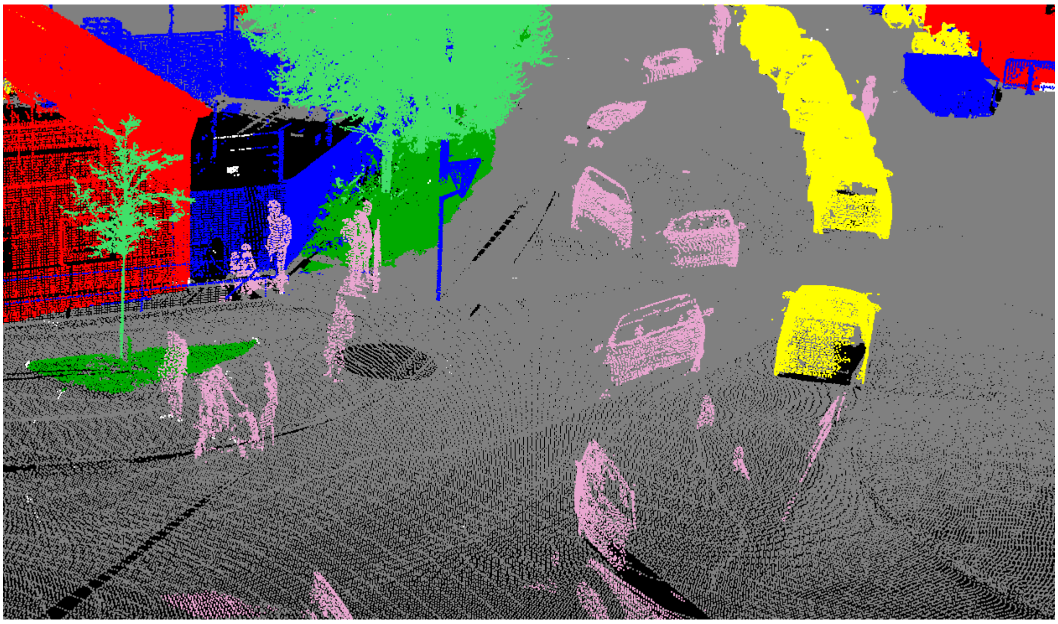

Figure 7.12: Scene 2, example of how cars and pedestrians are labelled as noise

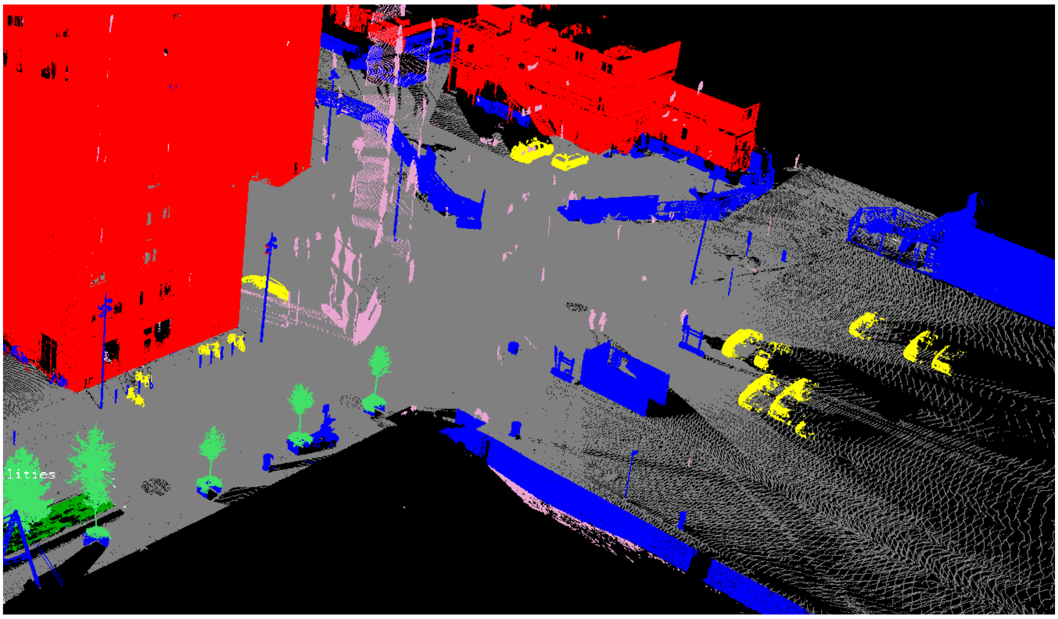

Figure 7.13: Scene 5, example of where reflections from buildings are categorised as noise

Related to the classification of noise, the reflections from windows, doors, cars, etc. are also labelled in this category. An example of this is documented in figure 7.13.

Probably, the most challenging part of the labelling process is to generalize and interpret the categories described above in a similar way throughout all scenes. It is assumed to be important that the point cloud is labelled in a similar way to get a successful segmentation. This is a challenge because it is a human decision every time a point is labelled. Inconsistency in the labelling will contribute to errors in the segmentation. Therefore a way to minimize this inconsistency is to clarify the content of each class. Though, it can still be difficult to segment some points in the right way. Figure 7.14 gives an example of inconsistency in the categories: Here, the greenish-blue boxes illustrate some disputes in the categorisation. The two boxes close to the ground, a lift and an excavator, are respectively categorised as technical facilities and noise. But if the lift is a technical facility, why is the excavator not a technical facility? In this case, the reason for labelling the excavator as noise is that the points defining it are sparse, meaning that it is relatively incomplete and therefore seen as noise. This is a general problematic in the labelling phase.

Even though it seems simple to categorise the points in a point cloud there are several problematics connected to it.

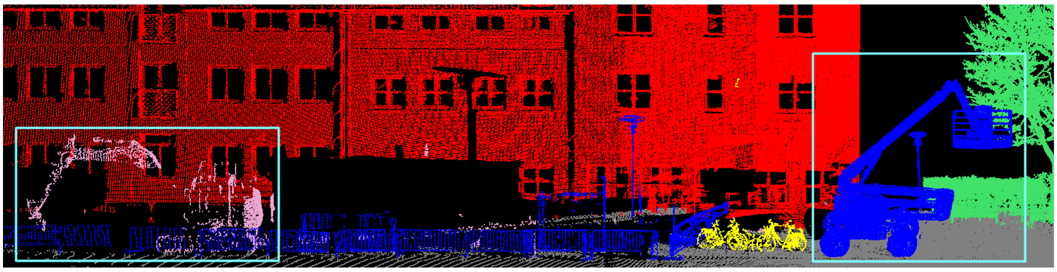

Figure 7.14: Scene 4, example of error in categorisation of technical facilities and noise.

When the labelling is completed, the point cloud file is exported from Lidar360 in a text format including x-coordinate, y-coordinate, z-coordinate, intensity, red, green, blue and label for each point. To match the format used in PointNet++ the labels-column is exported to a .labels-file which is an ASCII file containing the number corresponding to the semantic class of each point, one row for each point in the point cloud. This operation is performed using the python script *split.py*, see appendix A ref.no. 7.1.1.

Distribution of classes - quality check

To research if the evaluation dataset and the training dataset are comparable in the distribution of classes the following examines the distribution of points in the datasets.

The distribution between the different classes in the training dataset and evaluation dataset are plotted in figures 7.15 and 7.16. This distribution shows that points classified as buildings are the vast majority of points in both the training and evaluation dataset, followed by grey terrain and vegetation. The last four categories are less abundant in the scenes. That which strikes the eye about these two figures is that green terrain is not represented in the evaluation dataset - or at least not represented by enough points to make it visible on this bar plot. This may cause the evaluation of the segmentation model to be misleading when it comes to the model's performance regarding this class. However, choosing another dataset for evaluation would compromise some of the aspects that the evaluation dataset was chosen based on - especially the variety in building types.

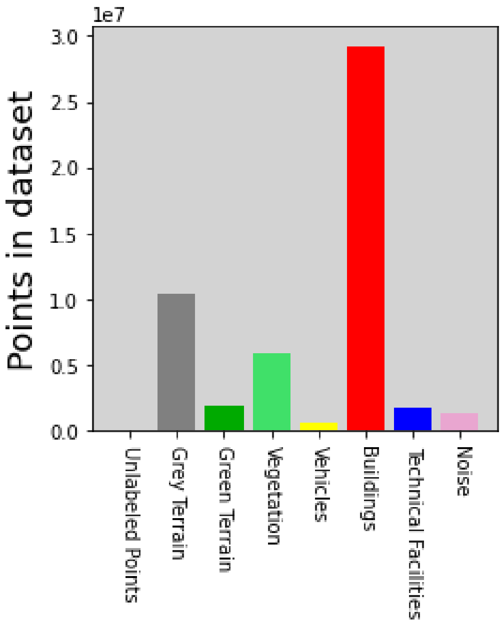

Figure 7.15: Distribution of points in the training dataset

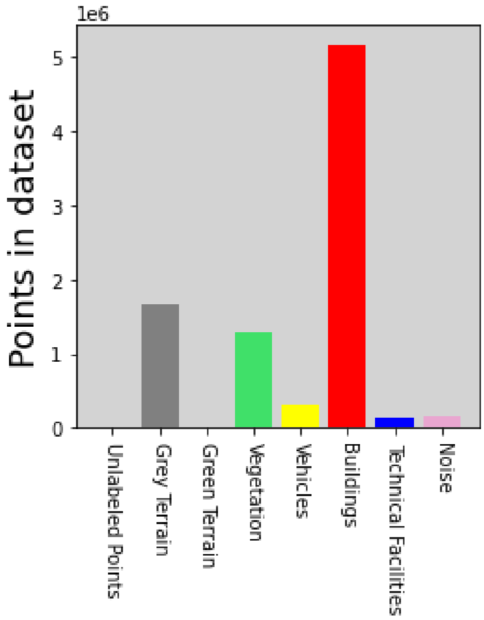

Figure 7.16: Distribution of points in the evaluation dataset

If the absence of green terrain in the evaluation dataset is overlooked, the semantic distribution of points seems to be similar in the two datasets. In figure 7.17 the percentage that each semantic class takes up in each of the datasets is plotted beside each other. Based on this, it is clear that the semantic classes in general are similarly distributed. Noise and technical facilities are not quite as abundant in the evaluation dataset, but if the model has been trained to identify these elements properly, then that should not be a problem.

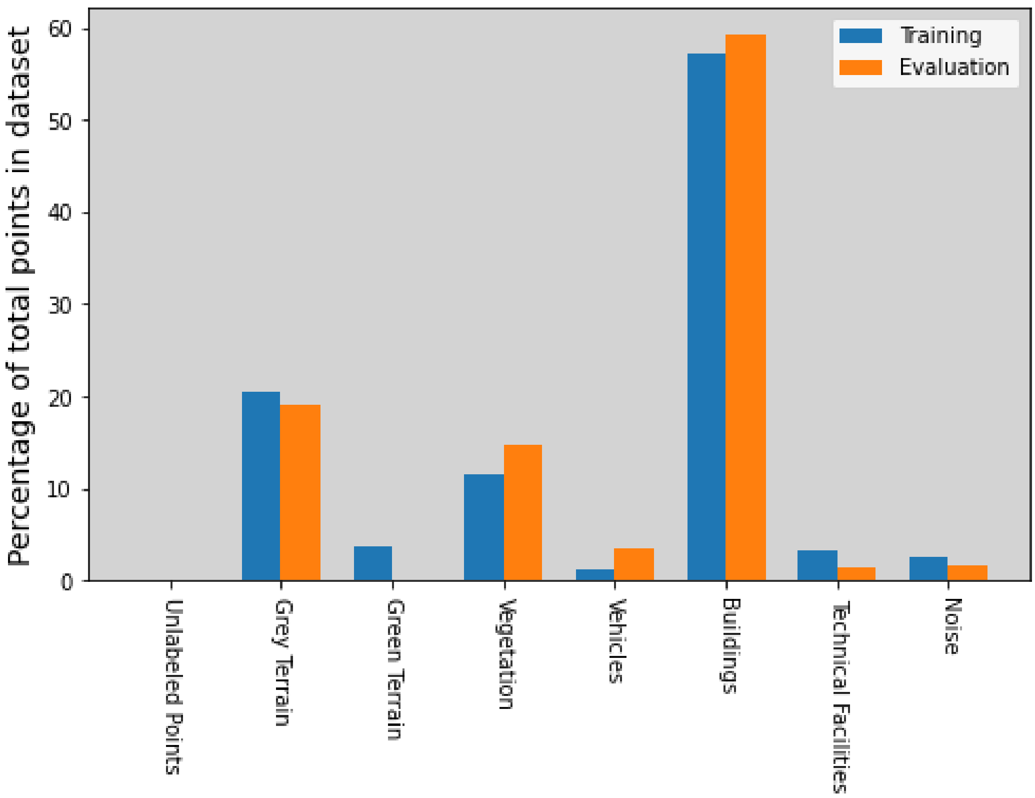

Figure 7.17: Comparison of the percentages that each semantic class takes up in the training dataset and evaluation dataset.

Thus, the evaluation dataset is considered acceptable for evaluating the performance of the segmentation model.

7.3 Training results

This section will go through the training results for the 3 trained models documented in table 7.2. The difference between the 3 models cf. table 7.2 is the input parameters for training the models. Model 1 and 3 are trained on both geometry, intensity, and RGB values, whereas Model 2 is trained without RGB values. Model 3 is trained using merged point clouds, whereas Model 1 and Model 2 are trained with individual point clouds.

Name	G^1	RGB	Intensity	I^2 or M^3	Training	Evaluation	Epochs
Model 1	✓	✓	✓	Individual	Scene 2 - 7	✓	200
Model 2	✓		✓	Individual	Scene 2 - 7	✓	350
Model 3	✓	✓	✓	Merged	Scene 2 - 7	✓	150

Table 7.2: Overview of trained models and which inputs are used to train them. Abbreviations: ^{1}: Geometry ^{2}: Individual point clouds and ^{3}: Merged point clouds

The training is performed as described in section 6.6. The output of each training is a segmentation model and a log file. The segmentation model is the "experience file" that can be used to segment point clouds and the log file documents the training and its performance for each training. When evaluating the training it is relevant to examine the log file to see how the performance of the model changes over the epochs. An extract of a log-file is documented in figure 7.18[1]

```
**** EPOCH 000 ****
2021-04-20 09:38:39.586038
mean loss: 1.913893
Overall accuracy : 0.828875
Average IoU : 0.348595
IoU of grey terrain : 0.767449
IoU of green terrain : 0.212654
IoU of high vegetation : 0.658954
IoU of vehicles : 0.106230
IoU of buildings : 0.851870
IoU of technical facilities : 0.117697
IoU of noise : 0.073903
2021-04-20 09:49:44.746282

---- EPOCH 000 EVALUATION ----
mean loss: 0.000000
Overall accuracy : 0.892818
Average IoU : 0.373813
IoU of grey terrain : 0.888762
IoU of green terrain : 0.015497
IoU of high vegetation : 0.693682
IoU of vehicles : 0.249131
IoU of buildings : 0.930813
IoU of technical facilities : 0.106418
IoU of noise : 0.106201
Model saved in file: log/semantic/best_model_epoch_000.ckpt
Model saved in file: log/semantic/model.ckpt
```

Figure 7.18: Screenshot of the first epoch in the log-file from the training of Model 1.

The log file first of all documents the model's ability to segment the training data after each epoch, which results in IoUs for each class, an average IoU, and an overall accuracy. Mean loss is also logged, although this parameter is not evaluated in this project, as mean loss is harder to understand in the context of this project than the other parameters evaluated.

Most of the log-file is the results of the model's predictions on the training dataset, which means the results are biased. When setting up the training process it is optional to include an evaluation dataset, which, if included, allows the training script to evaluate the model's ability to segment an independent point cloud every fifth epoch - as shown in figure 6.5 on page 44. The evaluation dataset is not used in the training and therefore the result is not biased.

The following three subsections 7.3.1, 7.3.2 and 7.3.3 will document and describe how the training of each of the above-mentioned models went. Ultimately, the training results will be assessed and compared to find out which training method is best.

[1]Note that figure 7.18 is edited to fit the classes used in this segmentation. The log-files in appendix A, ref.no. 7.2, are raw and in the same format as in the benchmark, meaning that text for the classes is wrong.

7.3.1 Model 1 - Results from training with colors and intensity

This subsection will examine how the training of Model 1 went. In Model 1 the input parameters are RGB colors and intensity, it trains on individual point clouds (which means they are not merged for each scene) and it trains for 200 epochs. Ahead of the training, it is not known if 200 epochs are less or more than needed, this will be researched when analyzing the log file. The log file for the training of Model 1 can be found in appendix A ref.no. 7.2.1.

The analysis is performed by using a python script (plot_log.py in appendix A ref.no. 7.2.4) which plots the results for each training in 3 different graphs.

The first plot is illustrated in figure 7.19 where the IoU (evaluated from the same training dataset) of each class is plotted over epochs. The y-axis is the IoU (1 is equal to 100 %) and the x-axis epochs from 0 - 200. The average IoU is also computed and illustrated as the dotted line. From the graph, it can be concluded that the model rapidly (after about 25 epochs) is able to segment grey terrain, high vegetation, and buildings with at least 90 % IoU. From the four other classes, it shows that they are harder to segment and also takes more epochs before their curves stagnate - it takes about 150 epochs (it may still increase even after the 150 epochs). These categories are not well represented in the point clouds and might be hard to generalize. The average IoU (the dotted line) is obviously in the middle of the other graphs and ends up in an average IoU at about 90 %.

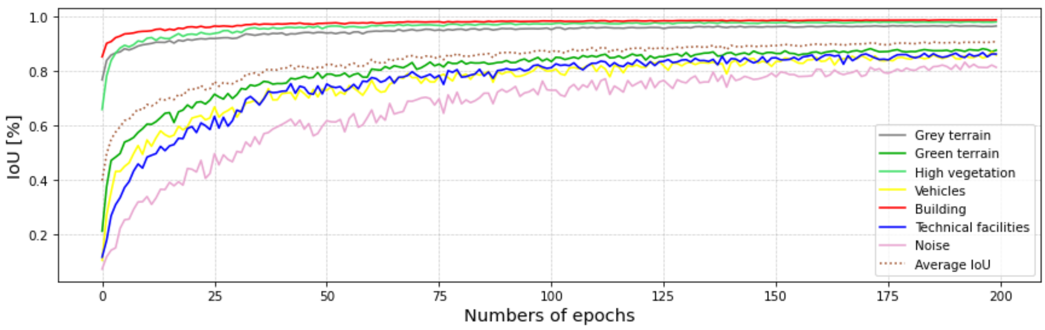

Figure 7.19: Training results for Model 1 - The models ability to segment the training data (biased) over epochs

As mentioned, the training also has the opportunity to evaluate the model compared to the evaluation dataset. The model's ability to segment the evaluation dataset is documented in figure 7.20. Here the tendencies are different: There is not the same development in how well the model can predict the evaluation data over the epochs. It immediately looks like the model does not segment the evaluation data significantly better after epoch 200 than it does after epoch 0. The IoU scores are still fairly high for buildings, grey terrain, and high vegetation they are still a bit lower than in figure 7.19, which is expected because of the bias.

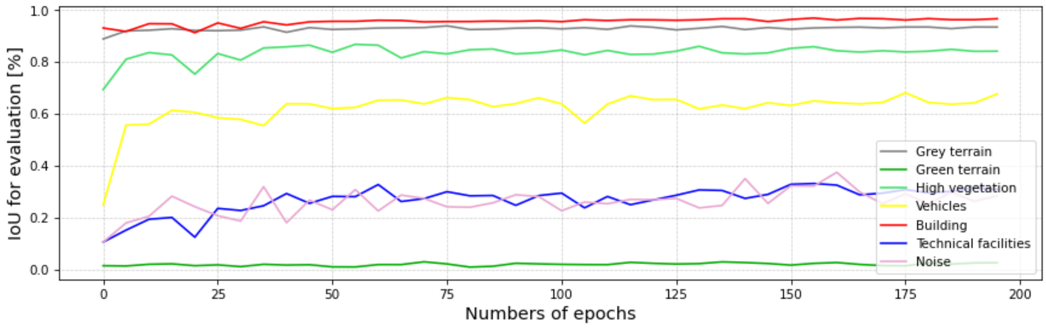

Figure 7.20: Training results for Model 1 - The model ability to segment the evaluation dataset (unbiased) over epochs

Figure 7.21 documents the overall accuracy for the comparisons between the training data (the blue graph) and also overall accuracy for the evaluation data (the orange graph).

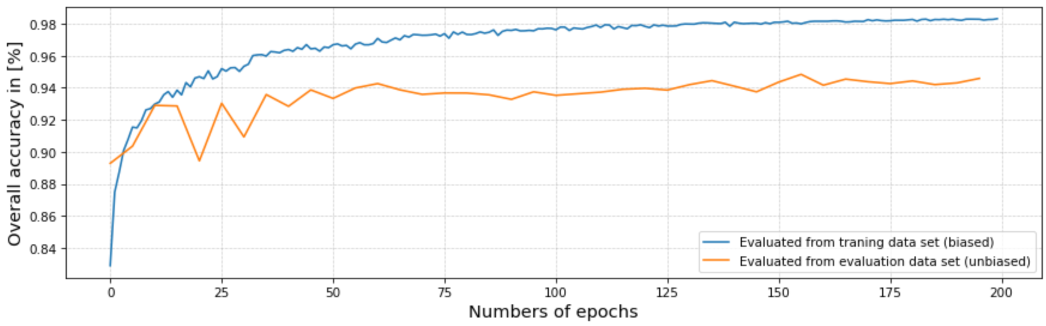

Figure 7.21: Training results for Model 1 - The overall accuracy from the training dataset and the overall accuracy for the evaluation dataset

From this research of these three plots, it can be concluded that after 150 epochs the model's ability to segment stagnates for the training dataset. For the evaluation dataset, the training only increases the model's ability to segment the point cloud slightly. Though, the overall accuracy is still about 94 %. In conclusion, it can be assessed that it generally works fine to train the model with RGB colors. The following will examine how the training goes when relying only on intensity and no RGB colors.

7.3.2 Model 2 - Results of training without RGB-values

In this subsection, the results from the training of Model 2 will be researched. In the training of Model 2, the RBG-values are omitted, which means the training is based on the geometry and the intensity values. This training also trains on individual point clouds. The model is trained over 350 epochs as a consequence of a technical mistake, as it was only supposed to be trained over 200 epochs like for Model 1. The log file for the training of Model 2 can be found in appendix A ref.no. 7.2.2.

Figure 7.22 documents Model 2's ability to segment the training data (trained by intensity). From the plot, it can be concluded that it is generally harder to segment the training data when the model is trained without RGB values. Though, it is the same classes that have respectively the highest and the lowest scores as in figure figure 7.19 on page 61.

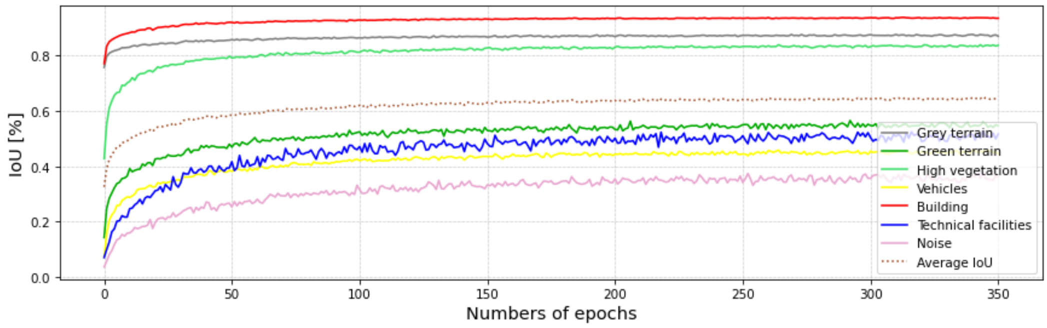

Figure 7.22: Training results for Model 2 - The models ability to segment the training data (biased) over epochs

In figure 7.23 the IoU for segmenting the evaluation dataset is plotted.

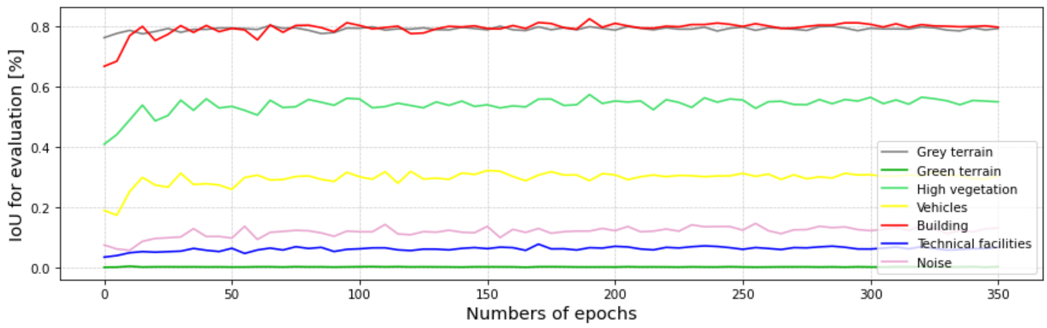

Figure 7.23: Training results for Model 2 - The model ability to segment the evaluation dataset (unbiased) over epochs

Finally, the overall accuracy for the training dataset and for the evaluation is plotted. Again there is a significant difference in how well Model 2 can segment the point cloud for respectively the training dataset and evaluation dataset. The training data is more recognisable than the independent evaluation data.

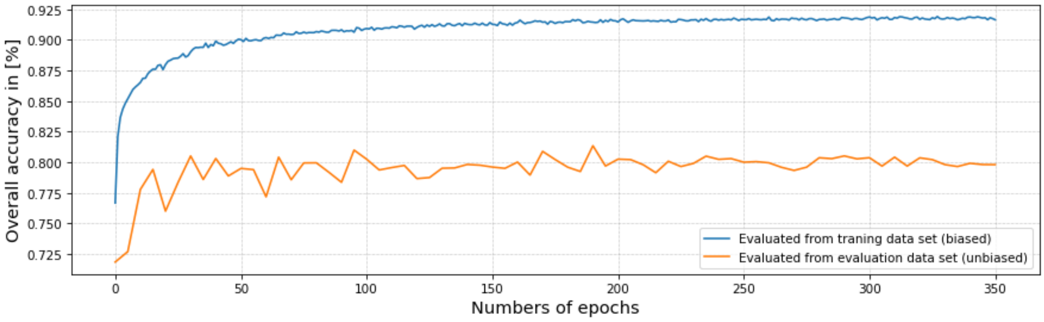

Figure 7.24: Training results for Model 2 - The overall accuracy from the training dataset and the overall accuracy for the evaluation dataset

The performance of Model 2 generally stagnates after 150 epochs. For example, the scores from the overall accuracy compared to evaluation data is about 80 % (after 350 epochs).

7.3.3 Model 3 - Results of training where point clouds are merged

The third and last model is trained over 150 epochs and it uses RGB values and intensity values. The primary difference for this model is that the training and evaluation data is merged into one point cloud for each training scene. The log file for the training of Model 3 can be found in appendix A ref.no. 7.2.3.

In the same way as the two above scenarios, the log-file is researched and plotted into three plots. The first plot in figure 7.25 documents that the IoU generally is high for vegetation, building, and grey terrain and their final IoU is around 0.95. The other categories generally score lower.

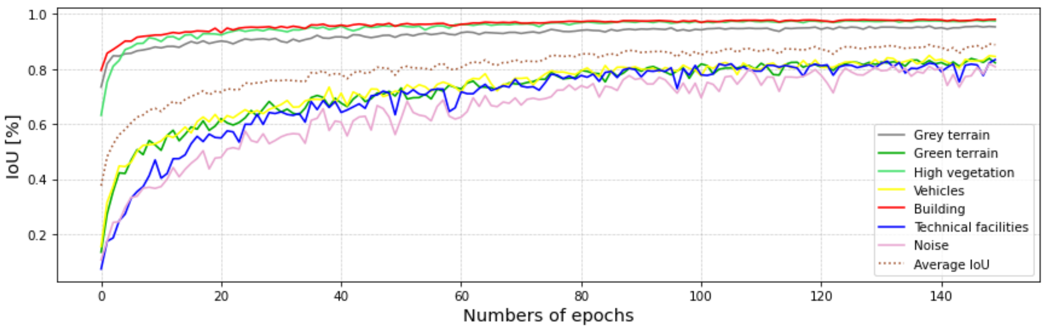

Figure 7.25: Training results for Model 3 - The models ability to segment the training data (biased) over epochs

The plots of Model 3's ability to segment the evaluation point cloud are documented in figure 7.26. The general tendency is that the IoU of the model does not increase significantly during the training epochs. Though, the classes high vegetation and vehicles' IoU increases a bit from epoch 0 to epoch 20.

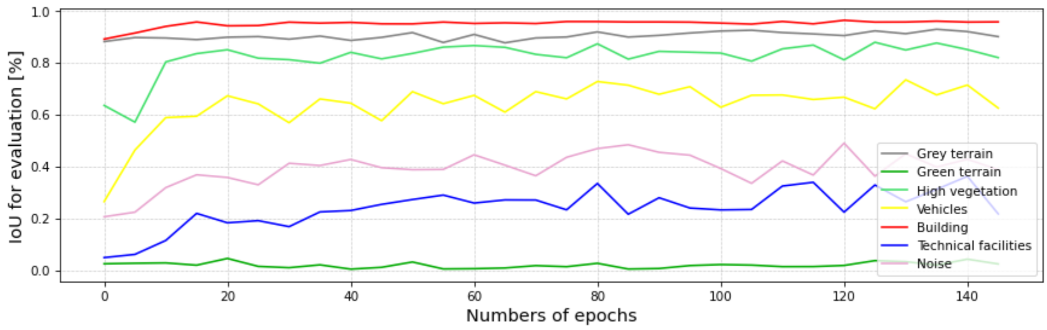

Figure 7.26: Training results for Model 3 - The model ability to segment the evaluation dataset (unbiased) over epochs

In figure 7.27 the overall accuracy for respectively the training dataset and evaluation dataset is plotted. The overall accuracy for the training data is significantly higher than for the evaluation dataset. The overall accuracy for the training dataset stagnates after 60 epochs. The overall accuracy for the evaluation dataset increases at the start (from epoch 0-20), whereafter it varies during the 150 epochs but ends in around 0.93 for the overall accuracy.

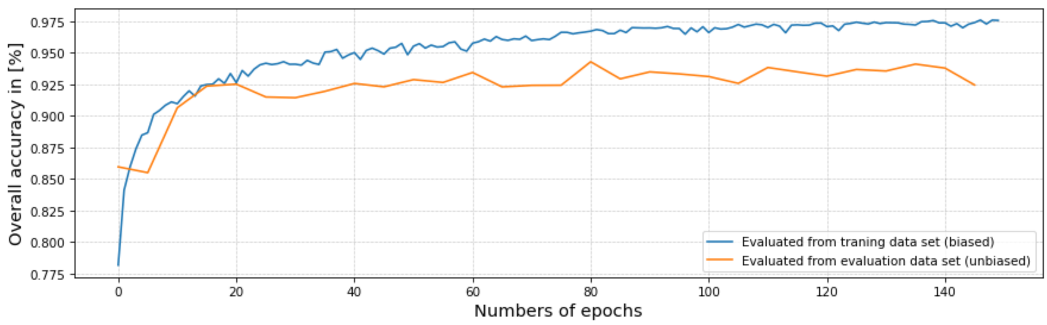

Figure 7.27: Training results for Model 3 - The overall accuracy from the training dataset and the overall accuracy for the evaluation dataset

7.3.4 Evaluating the training methods

The three different training methods have been analyzed by researching their log-file. This subsection will with background in the research of the above plots compare the training methods and assess which of the training method is the most efficient for segmenting urban scenes. Table 7.3 documents the three models' abilities to segment the training dataset and the evaluation dataset after 140 epochs. The comparison is for the 140$^{\text{th}}$ epoch since it is the latest saved version of the Model 3.

	Model 1	Model 2	Model 3
Overall accuracy (training)	0.98	0.91	0.97
Average IoU (training)	0.89	0.63	0.88
Overall accuracy (evaluation)	0.94	0.80	0.94
Average IoU (evaluation)	0.57	0.38	0.61

Table 7.3: Comparison of the "Overall accuracy" and the "Average IoU" for the three trained models after 140 epochs.

In table 7.3 the overall accuracy and the average IoU for the 140$^{\text{th}}$ epoch of each model is documented. Generally, Model 1 has the best scores in the training dataset where Model 3 has the best scores in the evaluation dataset. On the contrary, Model 2 has the poorest score in all classes, thus it indicates that RGB-values matter when segmenting urban point clouds. Since Model 2 has significantly lower scores it is not chosen to be studied further.

Immediately, the log file does not describe where the model fails, therefore the following will research for Model 1 and Model 3 where the errors in the segmentation occur. This is performed by plotting a true/false (green = true / red = false) point cloud. The point cloud illustrates where each model fails in the segmentation of the evaluation data.

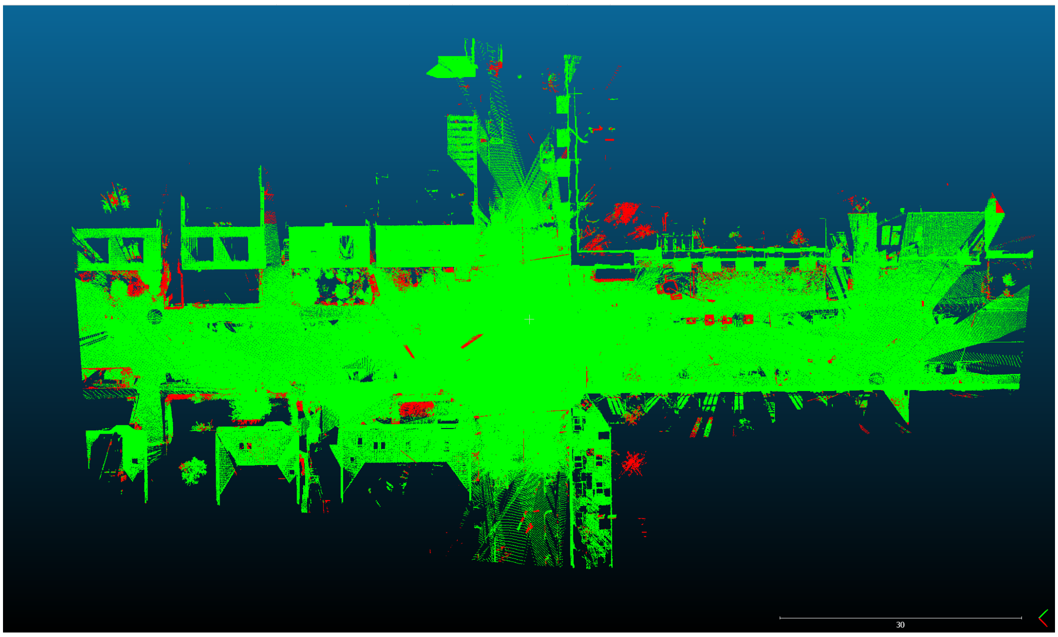

Figure 7.28: True/false plot of Model 1. Illustrates where Model 1 has problems by segmenting the evaluation point cloud. The point cloud is shown in a top-view.

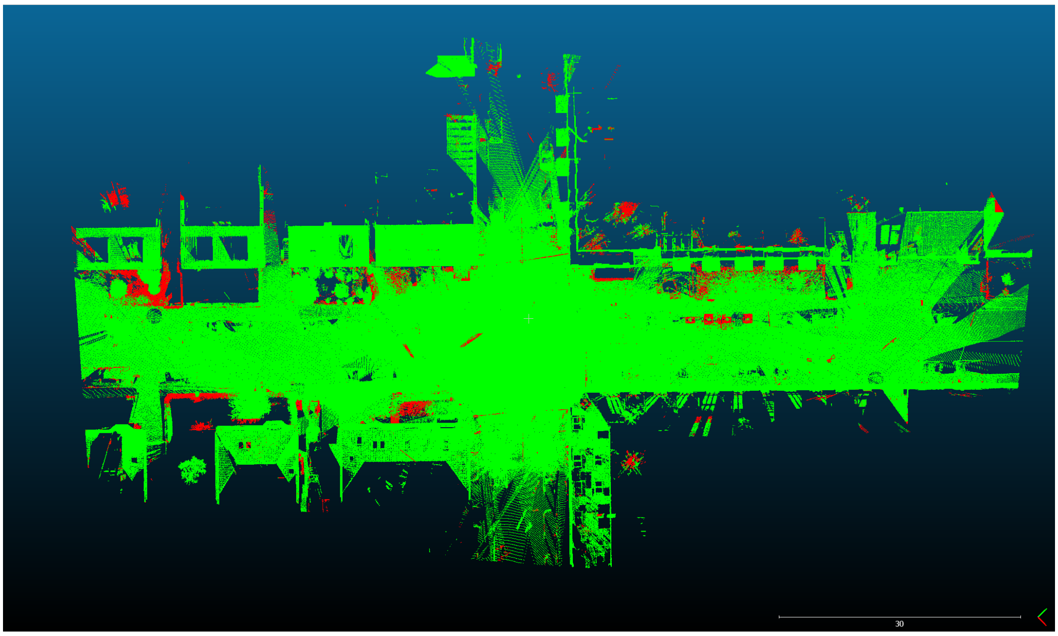

Figure 7.29: True/false plot of Model 3. Illustrates where Model 3 has problems by segmenting the evaluation point cloud. The point cloud is shown in a top-view.

In figure 7.28 and 7.29 there is no significant difference in true and false values. It figures that the two models have nearly the same problems in segmentation of the evaluation dataset, though there are small differences without real impact on the result.

Concluding, from the previous research of Model 1-3 it is found out that Model 2's scores are significantly lower than Model 1 and 3. Additionally, the difference between Model 1 and 3 is minimal. Model 1 scores highest for the training dataset whereas Model 3 scores highest for the evaluation dataset. Since Model 1 and Model 3 is so close to each other but Model 3 is slightly better in segmentation of the evaluation dataset, why only this one will be evaluated further.

7.4 Evaluation of segmentation results - Model 3

This section will evaluate the segmentation of the evaluation data *Scene 1 - Road intersection between Ryesgade and Helgolandsgade*. This dataset is segmented with the latest trained model (Model 3), where the point clouds were merged for each to a single point cloud for each scene and the model is trained for 140 epochs. Initially, the confusion matrix and IoU will be presented and studied. Hereafter, the segmented point cloud and a true/false plot are presented and researched.

Confusion matrix

The confusion matrix in table 7.4 documents the ratio between predicted labels and true labels for the evaluation dataset. The result in the confusion matrix should generally be similar to the plotted log files from the last training. The results are however not expected to be completely the same, because the training is performed on sparse versions of the point clouds, whereas the segmentation interpolates the point cloud back into its full size[2].

		Predicted labels								TS[4]	A[5]
		0	1	2	3	4	5	6	7		
TL[1]	0	0.00	0.00	0.00	0.01	0.00	0.02	0.00	0.00	nan	nan
	1	0.00	18.42	0.19	0.18	0.07	0.18	0.01	0.02	19.07	96.56
	2	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.03	44.54
	3	0.00	0.14	0.09	13.92	0.12	0.12	0.22	0.17	14.77	94.25
	4	0.00	0.09	0.00	0.06	2.84	0.19	0.19	0.30	3.53	80.52
	5	0.00	0.08	0.00	0.07	0.08	58.75	0.06	0.27	59.32	99.05
	6	0.00	0.07	0.02	0.25	0.07	0.29	0.74	0.04	1.47	49.94
	7	0.00	0.03	0.00	0.26	0.03	0.28	0.03	1.14	1.78	64.17
PS[2]		nan	18.84	0.32	14.75	3.20	59.84	1.11	1.94	Overall accuracy	
P[3]		nan	97.78	3.99	94.35	88.63	98.19	66.10	58.91	95.82	

Table 7.4: Confusion matrix for prediction of the evaluation data using Model 3 in %. Abbreviations used in the table: [1]: True labels, [2]: Predicted shares, [3]: Precision, [4]: True shares and [5]: Accuracy and [6]: Overall Accuracy. These evaluations parameters are the same as defined in section 2.1.2 on page 9.

Category	Class	Color
Not classified	0	White
Grey terrain	1	Grey
Green terrain	2	Green
High vegetation	3	Light green
Vehicles	4	Yellow
Buildings	5	Red
Technical facilities	6	Blue
Noise	7	Pink

Table 7.5: Legend table - Same table as 6.1

In table 7.4 it is documented how well the classes is categorised. The class-code is given in table 7.5. Class 1 or grey terrain is successfully segmented with an accuracy of 96.56 % and a precision of 97.78 %, this means the model almost perfectly finds the grey terrain points. As for green terrain, or class 2, the tendency is significantly different as the accuracy is 44.54 % and the precision is at 3.99 %, which means it has predicted too many points as green terrain. The green terrain is generally hard to segment since it is under-represented in the evaluation dataset. Classes 3,4, and 5 perform well with an accuracy between 80.52 to 99.05 % and a precision between 88.63 to 98.19 %. These classes are respectively high vegetation (3), vehicles (4), and buildings (5). The last two classes are technical facilities (6) and noise (7) - these are performing with medium success but they are also expected to be hard to segment since they contain varying forms, shapes, and elements.

[2]See appendix B for a short explanation

	1	2	3	4	5	6	7	mIoU
IoU	0.95	0.38	0.89	0.72	0.97	0.41	0.44	0.63

Table 7.6: IoU's for each class and the mean IoU (mIoU).

The other evaluation parameter is the IoU for each class. Table 7.6 sums up the IoU for each class. The IoU generally varies through the different classes, but overall it follows the same tendencies as seen in table 7.4. The classes grey terrain, high vegetation, vehicles, and buildings also score high in the IoU, whereas technical facilities, noise, and especially green terrain have a lower IoU.

Segmented point cloud - evaluation data

Figure 7.30 shows the evaluation dataset as it is predicted by the segmentation model.

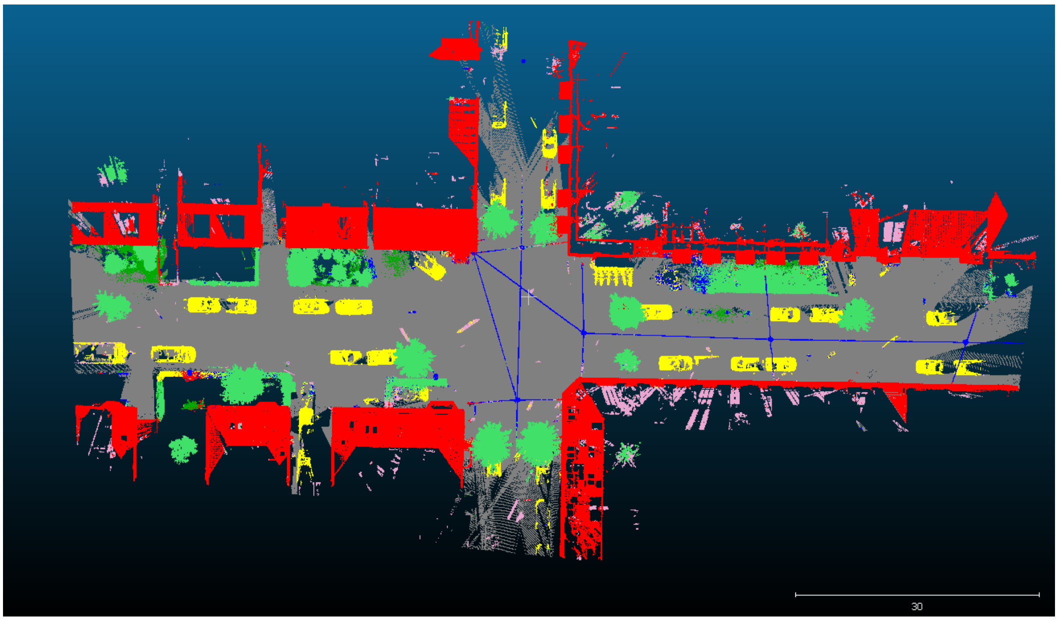

Figure 7.30: Segmentation of the evaluation dataset (top-view). The color scheme is the same as in the labelling process, see table 7.5 on the preceding page.

Generally, the segmentation in figure 7.30 looks very promising, as buildings, roads, cars, and vegetation generally seem to be labelled correctly and with high accuracy. From this view, it is hard to spot any large/meaningful errors. To better visualize what this segmentation can do for the point cloud figure 7.31 shows two screenshots of the point cloud - one with every class enabled, and one with only road and buildings included.

Figure 7.31: *Top image:* Full evaluation point cloud - all classes enabled.
Bottom image: Segmented evaluation point cloud - only roads and buildings enabled.

From figure 7.31 it is clear that removing every class that is not roads or buildings drastically reduces the number of elements in the point cloud. Airborne wires, vegetation, and cars are simply removed from the point cloud resulting in a point cloud that is less messy and "cleaner" to look at. If only buildings and roads are kept in the point cloud and the rest of the classes are discarded, the file size of the point cloud is reduced by around 19 % in this case. Table 7.7 documents the file size of each class. The effect that the segmentation has when toggling the different semantic classes on and off is illustrated in the gif that can be seen in appendix A ref.no. 7.3.1. Furthermore, the segmented point cloud can be seen in .e57 format in appendix A with ref.no. 7.3.2.

Classes	Share in %	File-size for txt format
Grey terrain	19 %	70.4 MB
Green terrian	0 %	1.2 MB
High Vegetation	15 %	54.6 MB
Vehicles	3 %	12.0 MB
Buildings	60 %	221.0 MB
Technical facilities	1 %	3.8 MB
Noise	2 %	7.2 MB
Full size	100 %	370.2 MB

Table 7.7: This table documents the file-size of each class in txt-format with a 5 cm downsampling.

To compare the segmentation with the ground truth as it was labelled, the script *deviation_colors.py* is used, which can be found in appendix A with ref.no. 2.1. The resulting point cloud can be seen in figure figure 7.29 on page 67.

From figure 7.29 it is clear that there are small groups of points around the point cloud that have been predicted incorrectly. Some of these groups are illustrated in figure 7.32.

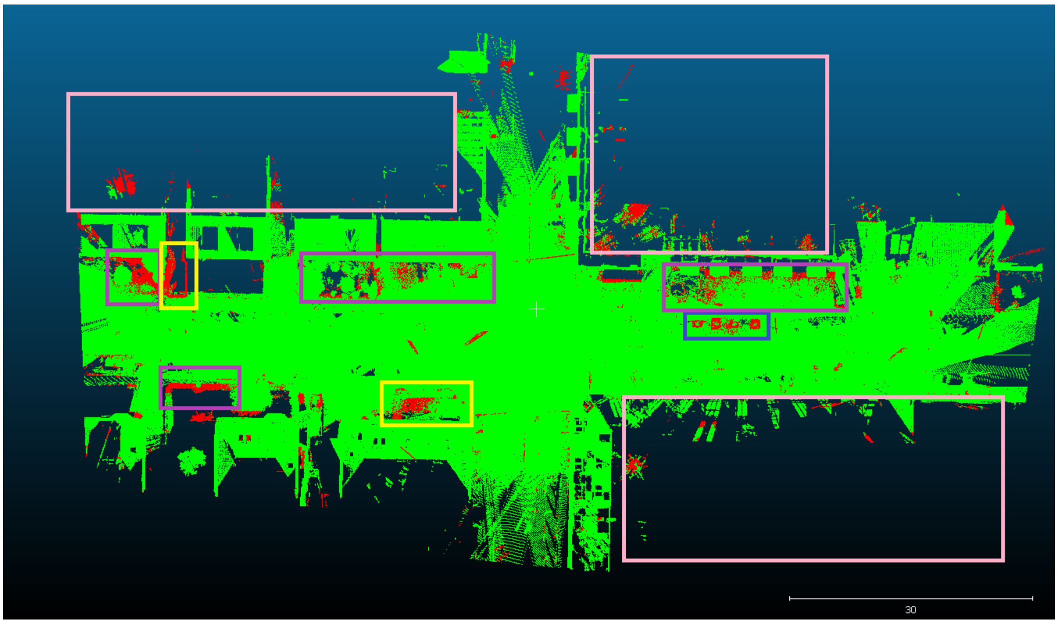

Figure 7.32: Figure 7.29 with boxes grouping some of the errors in the segmentation. Purple boxes indicate errors in green terrain/vegetation, blue boxes indicate errors in technical facilities, yellow boxes indicate errors in vehicles and pink boxes indicate errors in noise.

The errors shown within the boxes shown in figure 7.32 include a large portion of the total errors in the point cloud. The errors in the purple boxes are mostly tied to errors in green terrain. The areas with errors in the vehicle class are caused by a van of a kind that is not present in any of the training datasets and by a car that is hidden behind vegetation making points representing half the car very sparse and incomplete. The area with errors in technical facilities is caused by half-finished buried waste containers of a kind that is not represented in the training dataset. As for areas with errors in noise, these are mostly caused by scanning through windows. These areas can be hard to label correctly without also labelling the buildings as noise, and therefore some of the noise from measuring through windows is wrongfully labelled as building. An example hereof is shown in figure 7.33. These errors in the labelling of point clouds cause errors in the segmentation. Some of the errors seen in the pink upper-right rectangle in figure 7.32 can be explained by these errors in the labelling.

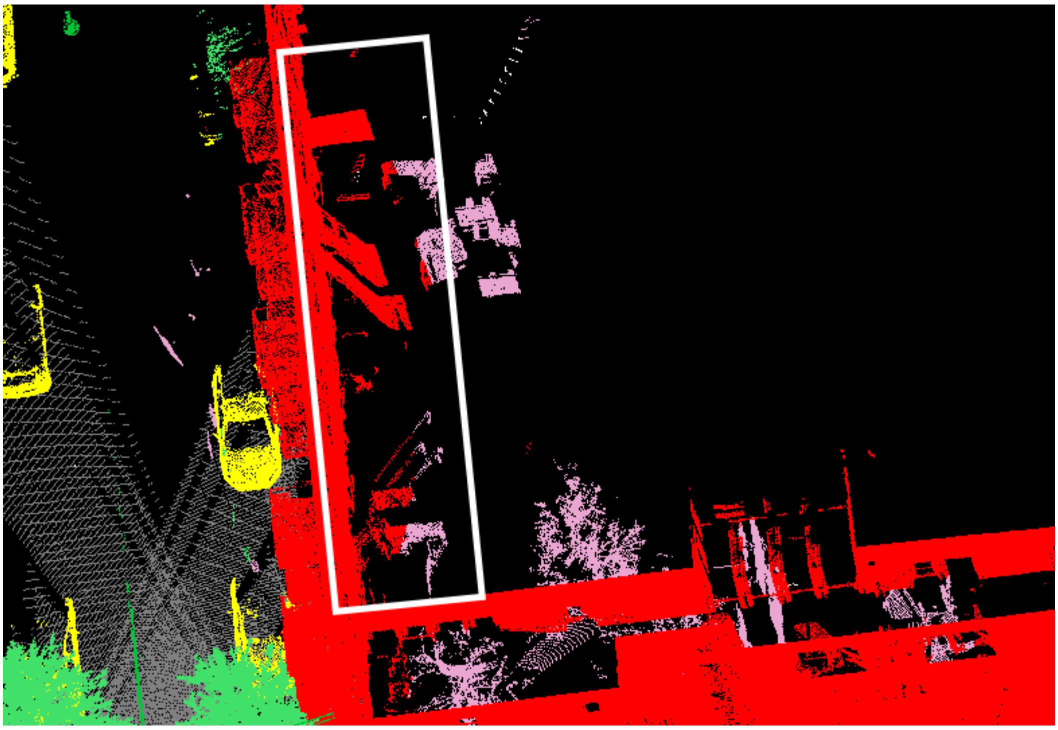

Figure 7.33: Example of errors in the labelling of noise. The red points within the white rectangle are wrongfully labelled as building although they should be labelled as noise.

In general, it would appear that the segmentation model can predict the semantic class of a large portion of the points in the evaluation dataset. The areas that are not predicted correctly are in many cases either a result of:

- Incomplete data
- Data with geometry not represented in training data
- Errors in the true labels for the evaluation dataset

Incomplete data is often present towards the edges of the point cloud where the data is getting sparse or is caused by blind angles in the scan as in the case with one of the cars in the error point cloud above. As implied, data with geometry not represented in training data is simply errors that are caused because the segmentation model has not been trained to recognize these specific geometries. This is hard to avoid, as it is practically impossible to create a generalized training dataset that includes all kinds of elements that may occur in an urban setting. The errors in the labelling are impossible to avoid as human errors will occur in this kind of work.

7.5 Conclusion - research question 2

Now that a segmentation model aimed towards segmenting outdoor point clouds has been trained and evaluated in terms of accuracy and IoU's, it is relevant to see if this model lives up to the requirement specification that was set up in section 6.2 on page 36. In section 6.2 the requirements were that the segmentation model should be able to segment point clouds with an accuracy of around 95 % with a higher weight on the accuracy of roads and buildings rather than the rest of the semantic classes.

The overall accuracy of the model trained on the merged point clouds depicting whole scenes lives up to the requirements in terms of overall accuracy as this reached almost 96 % with accuracies for roads and buildings of 96.5 % and 99.0 % respectively. The accuracy of classes such as technical facilities and noise were however quite a bit lower at around 50 % and 64 % respectively. These semantic classes are not considered as important as buildings and roads, and the inaccuracy of these classes does not impact the classification of buildings and roads. As such, the trained model lives up to the requirement specification, and the model is considered fit for use in a land surveying perspective.

Chapter 8 Potential and challenges

This chapter will be used to answer the third research question *Which potentials and challenges are tied to the use of semantic segmentation models in a land surveying perspective?* This research question is meant to evaluate and discuss the potential that semantic segmentation has in a land surveying perspective.

The former research questions have focused on (1): Point cloud assignments in land surveying assignments with the goal of being able to direct the training of segmentation models directed towards handling these assignments and (2): On training and evaluation of a specific segmentation model. In this research question, the experience gained from the former research questions will be used to discuss how the general concept of semantic segmentation using deep learning can be used in land surveying. The focus will be on semantic segmentation on a general level, and as such not focused directly on outdoor point clouds as was the focus of research question 2. The considerations will be on a more general level than the considerations from research question 1, which was focused on which classes are relevant for segmentation and how the segmentation would help the specific task. Consequently, this research question is more focused on how semantic segmentation can help land surveying companies in general, and what challenges are tied to this.

The method for finding this out will be based on:

- The authors' experiences with and thoughts about semantic segmentation
- Further evaluation from point cloud professionals

The authors' experiences and thoughts are relevant to consider, primarily because some challenges can only be identified with experience in the use of semantic segmentation. These reflections from the authors can therefore be seen as an "expert input" regarding semantic segmentation in itself.

To gain a better understanding of the potentials and challenges that semantic segmentation has in land surveying it is relevant to include the point cloud professionals that were also involved in the answering of research question 1. Their experience will be very relevant in identifying potentials and challenges in land surveying. The point cloud professionals will be presented with the results from research question 2 so that they can gain a better understanding of the potential performance of semantic segmentation trained to be used in a specific land surveying task. The intention is that the point cloud professionals with their experiences with processing and use of point clouds in land surveying companies will be able to indicate the potential that segmentation models like the one trained through the answering of research question 2 has. They may also indicate potential challenges regarding the use of this technology.

Using this feedback together with the experience about semantic segmentation that has been attained throughout this project, it will be possible to discuss if and how semantic segmentation can be used in practice in land surveying and if any issues should be considered before implementing it.

The overall structure of this chapter is illustrated in figure 8.1. In this, the first element is the "author's reflection" and the second element is the "expert correspondence" where three land surveying companies have been contacted and asked about the use of semantic segmentation. Ultimately, the chapter will be rounded off in an assessment and conclusion about if semantic segmentation has potential in a land surveying perspective.

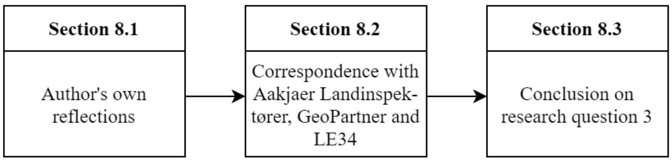

Figure 8.1: Flow diagram showing the structure of the current chapter

8.1 Authors' reflections

This section is used to present some of the authors' reflections regarding the use of semantic segmentation in land surveying. The reflections will primarily be focused on the challenges tied to semantic segmentation.

One of the main challenges that arise when trying to use semantic segmentation is the implementation itself. The practical use of software and scripts has a steep learning curve, the scripts are generally hard to understand and the algorithm has some very specific software dependencies that are not mentioned. Therefore, a somewhat large time investment is necessary to begin using this technology if the user does not have any prior experience in setting up this kind of software or in computer science in general. This process is documented in appendix B.

The scripts also require a computer with a substantial GPU and are designed to work on Linux-based operating systems only. As such there are somewhat strict requirements for both software and hardware (see for example appendix B) which also makes it harder to implement for many people. To get around this in the project, Google Cloud Platform was used to set up a virtual machine with Ubuntu and the required hardware. This is a good solution in this context but is likely not viable in business. One thing is the "computer logistic" challenge of having to first uploaded data to the virtual machine for it to be processed and having to download it again afterward. Another more critical problem with this is that the Google Computer is not always available, as Google has limited server capacities. This means that if the server is already in use by other people on the same server (in this case Europe West-1b), then the computer cannot be started. In the worst cases, this has meant that the google computer could not be started for 2 days during this project. In a business case, a problem like this can cause many problems. Consequently, if semantic segmentation were to be implemented in a land surveying company, the relevant solution to this problem is to invest in a processing computer that can run a Linux-based operating system.

Another challenge is the problem with quality control of the segmentation. In this project, it has been possible to perform quality control through the evaluation dataset. But this is only possible because that point cloud is labelled. In practice, labelling the point clouds before segmenting them would be very time-consuming and inefficient. Consequently, the segmentation of point clouds in practice would be performed without the quality control that labels can give. As such, the only available quality control is a visual control of the segmentation. Alternatively, a small portion of the point cloud could be labelled as a test sample, to see the performance of the segmentation in that particular portion of the cloud.

8.2 Expert correspondence

As described, some professionals in land surveying companies have been presented with the results from research question 2, and hereafter they are asked to comment on their opinions regarding potentials and challenges in the use of semantic segmentation in a land surveying perspective. These persons are consulted because of their expertise in and experience with processing of point clouds. The consultations were mostly performed as written correspondences over E-mail, except for the correspondence with GeoPartner, where a short interview was performed instead. The interview and the written correspondences are summarized and documented in appendix A with ref.no. 8.

LE34 and GeoPartner are contacted again since they are already acquainted with the theme of this project as they were also contacted in connection with research question 1. Furthermore, another land surveying company, Aakjaer Landinspektører has been contacted, which is a smaller land surveying company that also uses point cloud data in their work.

The correspondence is designed in a way where the overall results from research question 2 are summarized and presented for them. Concretely, the E-mail contained the segmented point cloud, a GIF which illustrates how the point cloud is segmented into classes (see appendix A ref.no. 7.3.1) and the deviation plot (the same as in figure 7.29) and finally the segmentation plot (the same as in 7.30). The land surveying companies are all presented with the same examples. The relevant remarks regarding potentials and challenges in a land surveying perspective are described in the following subsection.

It is relevant to have in mind that these professionals do not necessarily have in-depth knowledge about semantic segmentation but have an understanding of the general principles in the segmentation technology. Furthermore, it is relevant to consider that the professionals were asked to comment on the potentials and challenges via mail during their regular working hours. Therefore, it is not to be expected that they take out more than 15 minutes of their day to familiarize themselves with the results. Their remarks should be considered with this in mind.

The E-mail correspondences are attached in appendix A with ref.no. 8.1, 8.2, and 8.3, and a summary of the interview with GeoPartner is attached with ref.no. 8.4.

8.2.1 Key findings from correspondences

This subsection will account for the correspondences with LE34, GeoPartner, and Aakjaer Landinspektører. The below is a generalization of the three companies' comments regarding potentials and challenges for semantic segmentation. After explaining the potentials and challenges that the respondents have pointed out, the respective potentials and challenges will be discussed using the findings gathered throughout this project.

Potentials

The response from the land surveying companies was generally positive. All of the respondents saw potential in the segmentation's ability to remove irrelevant data from the point cloud, be it noise from scanning through windows, cars, or classes that can be irrelevant to the specific assignments like trees, technical facilities, etc. They all considered this to be a time-saver that potentially reduces file sizes and therefore also input/output time and processing time. This also makes it possible to load more data into the computer before RAM is used up. It is also assessed that it may make further algorithms, like different automatic/semi-automatic feature extraction algorithms, faster and more precise due to less noise. One mentioned that semantic segmentation may make the processing of point clouds more efficient to the point where it may be possible for the company to keep more of the work on the point clouds in-house, instead of having to outsource some of the operations. "Clean" point clouds containing only relevant data are considered to be a more attractive product for the customers, which is also an important potential from a business perspective.

In the following bullets the main potentials of semantic segmentation as found from the correspondences are listed:

- **Removes unnecessary and irrelevant data**
- This reduces file size and reduces input/output and processing time and allows for handling of more relevant data at a time

- This may also make further automatic/semiautomatic processes faster and more precise, as there is less noise that can cause errors
- Clean point clouds are considered to be a better product for the customers

From this, the primary potential that semantic segmentation has in a land surveying perspective is its ability to remove unnecessary and irrelevant data. The other potentials are all derived from this primary potential.

Challenges

The challenges that the respondents saw were mostly of practical nature:

- How does the algorithm react when presented with very large datasets, i.e. 200 GB or more?
- Can it work with point clouds from mobile laser scanners or UAV data?
- And does it work with point clouds that are a product of multiple different scanners and sources?
- Does it only work on point clouds that contain RGB colors?

8.2.2 Potentials and challenges explained

This section will comment on the above-described remarks and if they can be substantiated or explained with the knowledge about semantic segmentation achieved in this report.

Concerning the "potentials" described above, one of the primary functions/potentials is to remove noise and other irrelevant data. With the experience from research question 2 in mind, it is assessed that removing noise and irrelevant data can be done relatively easily when the point cloud is segmented. This implies that segmented point clouds are easier/faster for the computer to manage and makes it possible to manage larger point clouds using the same computational capacity since the point clouds contain fewer points. Another potential is that the point cloud could easier be targeted towards the customer's demands for the scanning job, which again is because irrelevant data easily can be removed. The potentials remarked by the point cloud experts are very much in line with the potentials for semantic segmentation that was expected based on the results from research question 2, which also illustrated this quality.

The challenges remarked by the respondents were focused on practical challenges - for example, if the segmentation model would be able to handle a file with a size of multiple 200's GB. In the research in this report, the evaluation dataset was about 350 MB which is significantly less. There has however been no indication of the scripts from PointNet++ cannot manage a 200 GB point cloud, although it will require a large computational capacity. Thus, there is no reason to believe that the segmentation model cannot manage large point clouds, although it will take a considerable amount of time to process point clouds of this size. In this case, however, it may be relevant to not merge all scan stations into one large point cloud, so that the point clouds are processed separately.

In the correspondence, it was also wondered if the segmentation would work for datasets from UAV or mobile scanners. As long as the point cloud contains cartesian XYZ coordinates, intensity, and RGB colors, then the model will be able to guess at the semantic classes of points in the point cloud - even if the point cloud is a product of multiple different scanners. For point clouds from mobile scanning, there is no reason to believe that it will not work with results comparable to the ones found in chapter 7 as long as the scenes measured are similar to the urban scenes used in chapter 7. For UAV data it is less certain that the segmentation will have comparable results, as UAV point clouds have other characteristics than regular terrestrial point clouds - especially in terms of perspective to the depicted area. Based on this, it was assessed in chapter 5 that UAV-point clouds should be segmented with a model dedicated to this task. This assessment may not be completely correct, as point clouds from UAV data may describe the geometry of an area in a way that is similar to point clouds from terrestrial laser scanning. This applies if the UAV is flown in a relatively low height and even more so if taking oblique photos, as this allows for covering data under, for example, eaves and roofs, thus creating a point cloud with geometry similar to terrestrial laser scanning. However, a UAV point cloud will never include intensity as this is tied to LiDAR scanning, and therefore the segmentation model may encounter serious problems in trying to segment a UAV point cloud. Therefore, UAV data is not considered compatible with segmentation models trained exclusively on other sources of point cloud data. Consequently, it is necessary to train a model on UAV data if the goal is to be able to segment UAV data. This is a general rule of thumb in this segmentation approach.

In regards to whether the segmentation model works with point clouds without RGB colors: Performing semantic segmentation based on geometry and intensity has been found to perform worse than segmentation with geometry, intensity and RGB-colors, cf. section 7.3.2 on page 62. The evaluation during training showed an overall accuracy of around 80 %, at which point the segmentation starts to lose some value - and it is significantly lower than the 95 % specified as a requirement in section 6.3 on page 38. Therefore, RGB colors are necessary to obtain results comparable to the results found through this project.

In the correspondences one of the respondents mentioned that they especially use laser scanning and point clouds in assignments that concern restoration/mapping of buildings, meaning that their focus often is on buildings and building details, such as walls, pillars, roofs, attics, balconies, and similar elements. Another respondent mentions the same kind of detail-oriented assignment in relation to scanning of roads, where elements like curbs, road markings, drains, man-holes, etc. Because of this, it may be relevant to research if the deep learning algorithm can be taught to identify and segment these more specific elements in either buildings or roads. A test of this is performed and documented in appendix C. From this, it can be concluded that the segmentation model can be trained to find more specific elements in buildings with an accuracy of 85.49 %, although it is assessed that higher accuracy can be achieved if more effort into labelling the training and evaluation data precisely.

8.3 Conclusion on research question 3

It can be concluded that one of the primary strengths of semantic segmentation is the capability to remove noise and irrelevant data, which can be very time-consuming compared to conventional methods of evaluating point clouds. The strength, to be able to remove noise/irrelevant data has an impact on a range of sub-processes when working with point clouds, which might aid in making a point cloud solution relevant in situations where traditional surveying methods usually would be preferred.

Another potential with semantic segmentation is the ability to segment more detailed objects than tested in research question 2. It was asked in the correspondence if the segmentation could segment building details or details in a road structure. A further test examined if building scans could be segmented into more detailed classes. A new model was trained, and segmentation of these building details worked overall. The overall accuracy was about 85.49 % where the previously trained model has an overall accuracy of 95.82 %. Though, the labelling process was reduced to a minimum of effort. It is assessed a better result could be reached by spending more resources on labelling the training and evaluation data more carefully.

Based on the findings of this chapter, there are also a couple of challenges tied to the use of semantic segmentation in a land surveying perspective. The primary challenge is the implementation of the software itself, as this is a challenge that is tied to an entirely different professional background than what most surveyors have. The challenges regarding which sources of point clouds the algorithm can handle and how much data it can handle can for the most part be negated by having a dedicated processing computer with high processing power, RAM, and GPU and by training models that are specific for respectively LiDAR clouds and photogrammetrical clouds. The algorithm does however seem to perform best when RGB data is present, and as such, it is necessary to take the extra time in the measurements to gather RGB colors as well before this can work.

The central conclusion of the research question is that semantic segmentation can be useful in a land surveying perspective, since it has a very large potential for removing unwanted data from point clouds and, except for the challenge of implementing the software, the challenges are rather negligible in comparison to the potential.

Chapter 9
Conclusion

The conclusion aims to answer the problem statement "*How can semantic segmentation via deep learning be used in danish land surveying companies?*" stated in chapter 3. Key findings from the previous sub-conclusions for each research question will be the basis of answering the problem statement.

Through the research of Semantic Segmentation via Deep Learning it can be concluded that the general approach for using semantic segmentation in a land surveying perspective can be summed up with these 3 steps:

1. Implement segmentation technologies
2. Train segmentation models
3. Segment point clouds $\longrightarrow$ Save processing time

Semantic Segmentation can in theory be implemented for any type of point cloud as long as the trained model fits the point cloud. The cornerstone in the segmentation technology is these models, and they must be aimed at a specific type of point cloud, since the result of the segmentation is directly dependent on how well the characteristics in the point cloud are represented in the given model.

From the research of the implementation of Semantic Segmentation for urban point clouds, it can be concluded that by training a targeted model an overall accuracy of 95 % can be reached. In the specific example, buildings and grey terrain were better segmented than for example noise, which is a result of the trained model. Training of a model is time-consuming and requires a high level of computing capacity but has the advantage that each model can be fitted to each surveying company's needs and the model can be modified continuously.

The primary potential of semantic segmentation is the ability to segment point clouds, which enables removal of irrelevant data. This helps reduce processing time and thereby it assists in making point clouds a more interesting and competitive solution in a land surveying company. The disadvantage or challenge related to Semantic Segmentation is the implementation of it, it takes high-end hardware and the software can be quite complicated to set up in the right way.

To sum up, the processing of point clouds can be improved by implementing and using Semantic Segmentation via Deep Learning for point clouds. The segmentation can be implemented by training a model to a specific type of task, where after the segmentation can be performed. The segmentation enhances the point clouds' potential and makes the general workflow with point clouds more manageable since the number of human operations is reduced. In general, it is assessed that Semantic Segmentation has great potential for use in danish land surveying companies.

Chapter 10
Perspectives

Based on the findings of this thesis the present chapter will round the report off by putting the project into a perspective. The intention is to describe and discuss further relevant challenges/potentials with Semantic Segmentation which is not fully researched in the report. The content of this chapter can then be used as inspiration for further studies or researches.

In this project, the focus has been on semantic segmentation of terrestrial point clouds through the use of the deep learning algorithm PointNet++. PointNet++ is an algorithm from 2019, which in this field is somewhat old. Newer and better technologies are developed continuously and more than likely there already exists more accurate options for semantic segmentation of terrestrial point clouds like the ones used in land surveying. As such, it may be relevant to research other algorithms further in the hopes of finding a more accurate deep learning algorithm.

Another element that is relevant to examine is the relationship between the evaluation data and the segmentation model. It is hard to assess if the accuracies and IoUs that were achieved with the model trained through this project is a result of training data and evaluation data being very alike or if the datasets actually are not alike, meaning that higher accuracies and IoUs are achievable with more similar datasets. Therefore, it would be relevant to evaluate the performance of the segmentation model on multiple different evaluation data sets to gain a better idea of the performance of the model.

The segmentation algorithm tested through this project works well as a tool that can be used after registration of the point clouds but before further processes and calculations like feature extraction/vectorization or calculation of digital elevation models. However, the process of segmenting point clouds is somewhat messy with many different sub-processes that need to be performed in order for the segmentation to work. If semantic segmentation is to be implemented in modern land surveying companies, then it would be relevant to streamline the process of segmenting point clouds, so that it is more user-friendly and does not require the user to do anything after the process has started. This is assessed achievable if a dedicated processing computer with a Linux-based operating system is used.

In an ideal scenario, the company has managed to train multiple different models that are directed towards handling the specific point cloud assignments that they perform. Thus, a wide array of models are available in a database or "model-bank", and these can be used to segment the point cloud to meet the specific needs of the customer in that assignment. The idea of collecting different models already exists on *Modelzoo.co* where pre-trained models for many different algorithms can be downloaded. However, the models here are from all kinds of deep learning frameworks and from different algorithms, and as of the time of writing, none of them seems to be involved with segmentation of point clouds. Consequently, the land surveying company has to develop their own "model-bank".

In the scenario where a land surveying company has fully implemented semantic segmentation in their workflow with both a dedicated computer, a model-bank, and a streamlining of the processes in the segmentation, it becomes easy and relatively fast to remove irrelevant data from any point cloud. The use of point clouds may begin to compete with more assignments where traditional surveying methods usually are used, because of the ease of collecting plenty and covering data and, with the implementation of semantic segmentation, ease of *only* working with the relevant data. And to top it all off: The segmentation algorithms that make all of this possible are open-source, meaning that the potentials of semantic segmentation are free for anyone.

References

Bures, Marek, Sergo Martirosov, and Jiří Polcar (July 2019). "A NOVEL PROJECTION ALGORITHM FOR PRODUCTION LAYOUT EXTRACTION FROM POINT CLOUDS". In: *Acta Polytechnica* 59, pp. 203–210. DOI: 10.14311/AP.2019.59.0203.

David, Eli (2018). *Deep Learning For Dummies*. E-Book ISBN: 978-1-119-48355-7. John Wiley & Sons, Inc.

DJM-Aerial (n.d.). *Point Cloud Survey At Washington Coal Mine*. https://djm-aerial.com/point-cloud-survey/. read: 27-04-2021.

Grigorescu, Sorin et al. (Apr. 2020). "A survey of deep learning techniques for autonomous driving". In: *Journal of Field Robotics* 37.3, pp. 362–386. ISSN: 1556-4967. DOI: 10.1002/rob.21918. URL: http://dx.doi.org/10.1002/rob.21918.

Hackel, Timo et al. (2017). "SEMANTIC3D.NET: A new large-scale point cloud classification benchmark". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. IV-1-W1, pp. 91–98.

Lao, Yixing (2019). *Semantic3D semantic segmentation with Open3D and PointNet++*. https://github.com/intel-isl/Open3D-PointNet2-Semantic3D#intro.

Leica (n.d.). *Leica RTC360 3D-Laserscanner*. https://leica-geosystems.com/-/media/files/leicageosystems/products/datasheets/leica%20rtc360%20ds%20872750%200620%20en%20lr.ashx?la=da&hash=63D80255A9256B1C1D20BD9D3EF3A5D1. read: 27-04-2021.

Liu, Weiping et al. (2019). "Deep Learning on Point Clouds and Its Application: A Survey". In:

Lohani, Bharat and Suddhasheel Ghosh (2017). "Airborne LiDAR Technology: A Review of Data Collection and Processing Systems". In: *The National Academy of Sciences, India 2017* URL https://doi.org/10.1007/s40010-017-0435-9.

Marr, Bernard (2018). *10 Amazing Examples Of How Deep Learning AI Is Used In Practice?* Read: 23-02-2021. https://www.forbes.com/sites/bernardmarr/2018/08/20/10-amazing-examples-of-how-deep-learning-ai-is-used-in-practice/?sh=173bbc5cf98a.

Oppermann, Artem (2019). *Artificial Intelligence vs. Machine Learning vs. Deep Learning*. Read: 08-03-2021. https://towardsdatascience.com/artificial-intelligence-vs-machine-learning-vs-deep-learning-2210ba8cc4ac.

SDFE (2019). *Skraafotos*. Read: 23-03-2021. https://skraafoto.kortforsyningen.dk.

— (2021). *Flyfotos og laserscanning*. Read: 23-04-2021. https://sdfe.dk/saadan-arbejder-vi-med-data/flyfotos-og-laserscanning.

Singh, Satya P. et al. (2020). *3D Deep Learning on Medical Images: A Review*. arXiv: 2004.00218 [q-bio.QM].

Uniqtech (2020). *Understand Jaccard Index, Jaccard Similarity in Minutes.* https://medium.com/data-science-bootcamp/understand-jaccard-index-jaccard-similarity-in-minutes-25a703fbf9d7. read: 31-05-2021.

Yusuf, Bulent (2019). *How to work more efficiently with large point cloud datasets.* https://www.navvis.com/blog/how-to-work-more-efficiently-with-large-point-cloud-datasets. read: 27-04-2021.

Figures

Tables

APPENDIX A

List of contents - Zip-file

This appendix documents the structure of the zip-file handed in along with the thesis. Table A.1 can be used in conjunction with the ref.no. given within the thesis to find the respective files in the zipped directory.

Filename / folder	**Explanation / function**	**Ref.no.**
chap_2		
deviation_plot.py	Plots errors in the segmentation	2.1
chap_5		
1_Resume_MHC_LE34.pdf	Documents the meeting	5.1
2_Resume_MH_GP.pdf	Documents the meeting	5.2
chap_6		
voxel.json	help file to downsampling	6.1
voxeldownsampling_e57.py	downsampling script	6.2
chap_7		
1_labelling		
1_split.py	Extracts labels from lidar360	7.1.1
2_logs		
1_Model1_log.txt	Logs from training	7.2.1
2_Model2_log.txt	Logs from training	7.2.2
3_Model2_log.txt	Logs from training	7.2.3
4_plot_log.py	Plots the log file	7.2.4
3_segmentation		
1_Result_gif.gif	GIF illustration seg. pc	7.3.1
2_SegEvaPC.e57	The seg. point cloud in e57 format	7.3.2
chap_8		
1_MailSV_MHC_LE34.pdf	Documents the correspondence	8.1
2_MailSV_PHJ_Aakjaer.pdf	Documents the correspondence	8.2
3_MailSV_HVJ_Aakjaer.pdf	Documents the correspondence	8.3
4_Resume_MH_GP_V2.pdf	Documents the meeting	8.4

Table A.1: Table showing the structure of the zip-file.

APPENDIX B

Implementation of semantic segmentation

This appendix is meant to describe in a technical way how to implement the Open3D-PoinNet2-Semantic3D (PointNet++) locally on a computer. The description refers to the names and folders in the repository (Lao 2019). First, it will be described how to set up the hardware and software on the computer required to run the PointNet++ segmentation approach.

Description of hardware and software

Since it takes high-end hardware to segment point cloud a virtual machine has been set up. A virtual machine is set up at Google Cloud Platform which offers cloud computing solutions such as virtual machines. The virtual machine is chosen since normal computers cannot manage the segmentation task. The benefit of using Google Cloud Platform is, that it is fast to set up your wanted configuration, and the configurations can be changed easily.

Configurations for the virtual machine:
- Machinetype: n1-standard-8 (8 vCPUs, 30 GB memory)
- Graphic card (GPUs): 1 x NVIDIA Tesla P100 (16 GB memory)
- Harddisk: 200 GB SSD

In the interest of reproducibility the software configurations are listed below:

Software configurations[1]:
- Operating system: Ubuntu 16.04
- CUDA 9.0, cuDNN 7.4.1
- Tensorflow 1.12 and Tensorflow-GPU 1.12
- Python 3.5
- Open3D (built from source)

The above settings are the software settings on the computer used in this project. The operating system is Ubuntu 16.04, CUDA 9.0 and cuDNN are programs that ensure the computer can use the deep learning functions on the GPU. Where Tensorflow is the deep learning version. The computation is performed in a Python program with the library Open3D installed.

Implementation of repository

[1]These software versions has shown to be an important factor since not all newer versions are backward compatible

The following will describe how to run these scripts in the repository PointNet++, the names of folders and scripts in the repository will be referred (Lao 2019). Before the overall method or approach is described a flowchart is illustrated in figure B.1. The flowchart and the following description give an understanding of how to implement the repository.

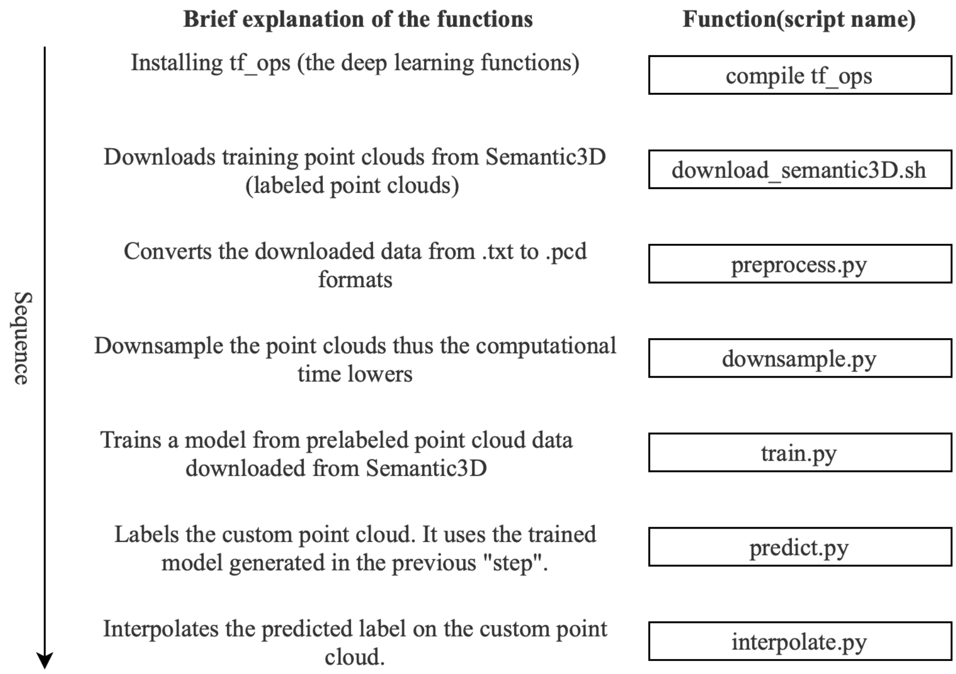

Figure B.1: The process of semantic segmentation. (Lao 2019)

First of all the above-mentioned software or similar has to be installed on the computer and the repository has to be downloaded locally on the computer. Each script in the repository has to be run individually in the described sequence cf. figure B.1. Before the elements in the repository can run it is crucial to build the Tensorflow kernels. These TensorFlow kernels are used to train a model and to segment point clouds. Building these kernels can be challenging as it will fail if the right versions of the software are not installed properly or if the environment paths are not set up properly.

In the repository, the first step is to download training data (already labeled point clouds) which the script *download_semantic3D.sh* does when it runs. The further step is to change the format of the downloaded files from .txt to .pcd since pcd-files is more effective to process, this is done in the script *preprocess.py*. To bring down the computation time the point clouds are downsampled in the script *downsample.py*. When the Tensorflow kernels are installed properly the training can begin which the script *train.py* does. The inputs to the training script are the pre-labeled point clouds and the output is a file containing experiences achieved by running the training script - this is called a checkpoint file, or .ckpt file. As standard, it uses files from

Semantic3D to train on. How much the script will be trained is an assessment of how much training data is available, iterations, and how good the model is needed. When the training has ended up in a model file the prediction can then begin by running the script *predict.py*. The output from this is a file that includes a label to each point in the point cloud. To run this prediction it must be given which trained model is used, which data set to be predicted and how many samples are used. If the segmentation is not satisfying the trained model can be improved by training it further or using more qualified training data. Finally, the *interpolate.py* script returns the points removed by *downsample.py* and interpolates the label of the returned points. Furthermore, it joins the information from the "label"-file over to the point cloud file and it is colored by labels.

To apply this method to custom data a few settings has to be changed in the script *dataset/semantic_dataset.py*. Herein the input data is specified by giving the names and path for the training, testing, and validating data. The data (the training, validation, and the test data) is placed in *dataset/semantic_raw*. When these elements are changed the repository can be used to train new models or segment custom point clouds.

The results of the segmented point clouds is documented back in the report section 2.2 on page 13.

APPENDIX C

Detail segmentation

This appendix will research if semantic segmentation can be used to classify more detailed objects than tested in research question 2. The research originates from the "*expert correspondence*", chapter 8, where it was questioned if semantic segmentation would work on more detailed objects in point cloud classifications.

C.1 Segmentation of building details

The present research will examine if point cloud can be segmented in a more detailed level than tested in research question 2. The specific research is centered around if it is possible to segment building elements, which is a specific example from the correspondence, described in chapter 8. In this research, it will be examined if a newly trained model can classify building objects, concretely which objects are described further below. The research is focused on buildings objects, but it could be any other category, since the focus is the principle, to see if segmentation works on a more detailed level.

The approach for testing if the segmentation works here is similar to the research approach in chapter 7. The test is performed on the same dataset scene 1-5 (without the dataset from LE34 and GeoPartner), all points not involving buildings are deleted. One of these datasets is chosen to be the evaluation dataset, which means there are 4 training datasets and 1 evaluation dataset. Again, the evaluation dataset is scene 1 at Helgolandsgade/Ryesgade. It is worth having in mind, that these datasets include fewer stations than scans performed with the purpose of mapping buildings. For this reason, the completeness of the buildings in the datasets in this training is poor in some areas.

Earlier, the point clouds were downsampled to 1 point pr 125 cm^3 (voxel: 5x5x5 cm), where they now are downsampled to 1 point pr 1 cm^3 (voxel: 1x1x1 cm), so that smaller and more specific objects can be detected.

To sum up the method for training a new model: First, the categories will be defined and then training and evaluation data will be labelled, whereafter the training of the model will begin. Lastly, the trained model will be evaluated by comparing the segmentation results to the evaluation dataset.

Labelling/Classes:

Step 1, is to define categories and afterward label the training and evaluation datasets. How the categories are defined is not so important in the present research. Contrary, it is relevant that the objects are more detailed than in the previous researches. The specific categories are designed with a basis in the expert correspondence from chapter 8 since there were some specific enquiries to interesting objects for segmenting, these categories are documented in table C.1.

Category	Class	Color
Walls	1	Red
Windows	2	Yellow
Roof	3	Grey
Doors	4	Green
Drain pipes	5	Blue
Balconies (including French balconies)	6	Pink

Table C.1: Legend table for the building segmentation

The labelling of the training and evaluation data is again performed in Lidar360. In each scene, 2-4 buildings are chosen and labelled and the rest of the points in the point cloud is deleted since it will take too many resources to label all buildings in all scenes. 2-4 buildings from these 4 training scenes are assessed to be a minimum for the training performance. Unlabelled buildings are discarded from the datasets in the training and evaluation.

Training:

When the categories are defined and the training data is labelled, the training can be started. Initially, the training is set to 150 iterations (epoch), but the training log will be studied to ensure the model has gained full potential from the training data. When the training was performed, the evaluation data was not labelled, which is why performance on evaluation is not included in the plots of the log file.

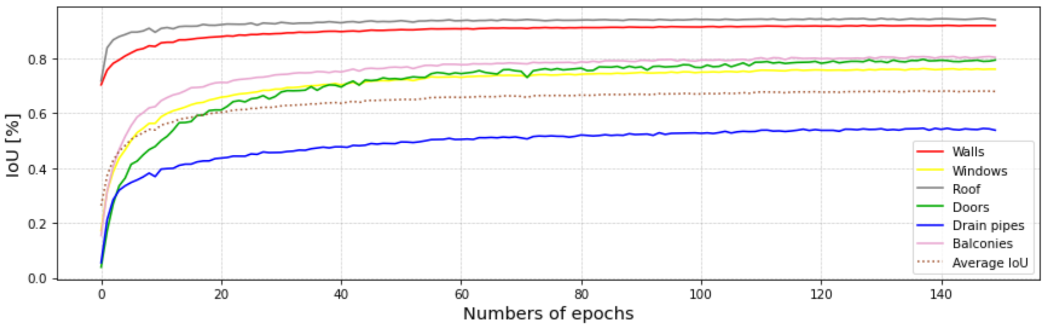

Figure C.1: Training results for segmentation of buildings - The models ability to segment the training data (biased) over epochs

The plot in figure C.1 illustrates the model's ability to segment its own training data. After about 30 epochs most of the curves stagnate, though the class "doors" stagnates later, after about 60 epochs. Generally, the IoU is lower compared to earlier training logs (compared to the log files from section 7.3.3) which is expected since the training dataset is smaller and the classes may be harder to generalize and label precisely and accurately.

Segmenting result:

The result of the above training is the building model, which should be able to segment point clouds containing buildings. This subsection will present and evaluate the results of the segmentation in the same way as in section 7.4, where the confusion matrix and the IoU's are presented. Furthermore, a plot of the segmented point cloud and the plot of errors (deviation plot) will be illustrated and commented on.

The confusion matrix is documented in table C.2. The segmentation performs differently across all classes. Walls and roof scores fairly high accuracies, whereas windows and drain pipes score medium and doors has a poor score.

		Predicted labels							
		1	2	3	4	5	6	TS[4]	A[5]
TL[1]	1	63.63	1.61	0.33	0.20	0.70	1.25	67.73	93.96
	2	3.16	7.18	0.69	0.31	0.01	0.43	11.78	60.97
	3	2.95	0.69	13.40	0.00	0.54	0.00	17.58	76.20
	4	0.63	0.30	0.00	0.23	0.00	0.00	1.16	19.80
	5	0.55	0.00	0.15	0.00	1.05	0.00	1.75	59.74
	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	NaN
PS[3]		70.92	9.78	14.57	0.74	2.30	1.69	OA[6]	
P[3]		89.72	73.41	91.95	31.11	45.49	NaN	85.49	

Table C.2: Confusion matrix for prediction of the evaluation data using the building-model %. Abbreviations used in the table: [1]: True labels, [2]: Predicted shares, [3]: Precision, [4]: True shares and [5]: Accuracy and [6]: Overall Accuracy. These evaluations parameters is the same defined in section 2.1.2 on page 9 and in the previous confusion matrices. Row and column numbers refer to the classes shown in table C.1 on the facing page.

In table C.3 the IoU of each class and the mean IoU is documented (class 6, balconies is not included, since there were no balconies in the evaluation dataset). These also differ from class to class, though the categories which scored high in the evaluation matrix also scores a high IoU.

	1	2	3	4	5	6	mIoU (except class 6)
IoU	84.82	49.94	71.44	13.76	34.82	NaN	50.96

Table C.3: IoU from the segmentation of the evaluation dataset (only buildings). The IoU is in % and the mean IoU is determined without class 6, since class 6 was not in the evaluation dataset.

Figures C.2 and C.3 respectively show the segmented point cloud and an error plot for the segmented point cloud of buildings.

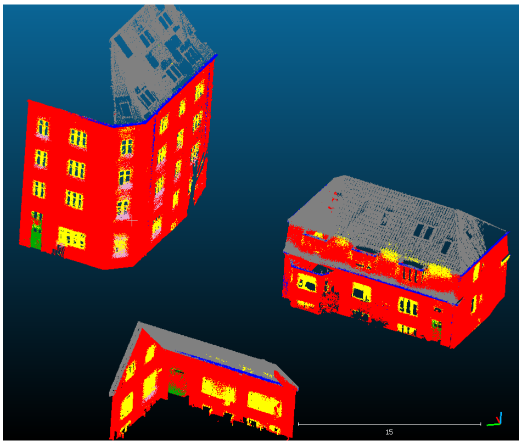

Figure C.2: Segmented point cloud of buildings. The colors are defined by the legend in table C.1.

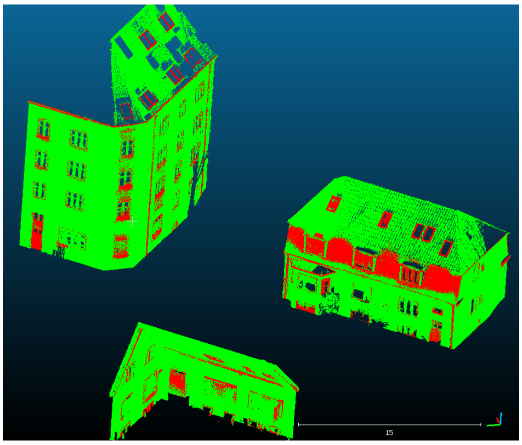

Figure C.3: Error plot of the segmentation of buildings shown in figure C.2. Green points are correctly labelled, while red points are wrongly labelled.

From figures C.2 and C.3, it is clear that this segmentation model is not quite as precise as the model trained to segment full outdoor point clouds which was also indicated in the training logs. In some cases, the errors occur where the model has predicted a completely wrong class, and in other places, it fails in the edge between two objects. A concrete example where the model's prediction is wrong is on the roof of the building on the right side in figure C.3. The steepest roof is here segmented as walls and the roof windows are classified as the roof. On the other hand, the figures show that the model actually can identify the remaining roofs, drain pipes, doors, and windows overall, though with some errors, especially in the transition between different classes.

Drain pipes are unexpectedly well segmented in the evaluation data. They are identified in almost all places they are labelled. It is unexpected because it is such a little part of the point cloud. This indicates that small identifiable objects can be classified very well.

Another relevant observation is to compare the result to the amount of training data. In this context the training data is containing 4 scenes containing 2-4 building each, this is a limited amount of training data and the model is still overall able to segment build objects in a fairly detailed level.