Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

Vilde Myren Mo
Marie Ting Falch Orre

TR3DRoofs: A Urban Roof Dataset

A New Dataset for Deep Learning-based Segmentation of Roof Structures in LiDAR Point Clouds

Master's thesis in Engineering and ICT
Supervisor: Hongchao Fan
June 2021

NTNU
Norwegian University of
Science and Technology

Vilde Myren Mo
Marie Ting Falch Orre

TR3DRoofs: A Urban Roof Dataset

A New Dataset for Deep Learning-based Segmentation of Roof Structures in LiDAR Point Clouds

Master's thesis in Engineering and ICT
Supervisor: Hongchao Fan
June 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

Measures to reduce the impact of climate change are becoming more and more critical. By increasing the use of renewable energy, up to 90% of the CO_2 emission reductions needed by 2050 can be achieved. The importance of 3D data is becoming increasingly more evident in this field, as modern cities require detailed models as a tool for in-depth planning to be both efficient and environmentally friendly.

In this thesis, we want to help contribute to make applications that use 3D data more accessible by exploring the applicability of one of today's biggest technology trends within automation, Artificial Intelligence (AI), on point cloud data. Specifically, we focus on the automation of the segmentation necessary for creation of 3D models of roof structures. We present a new dataset to be used for the task of 3D point cloud part segmentation of roof structures using deep learning. The goal is to propose a high-quality dataset based on real-life structures, yielding predictions of roof segmentations appropriate for applications in Norway.

The dataset is established from Light Detection and Ranging (LiDAR) data, collected across Trondheim municipality. Two versions of the dataset are proposed. The original dataset consists of 906 roofs present in the Trondheim area, and both datasets contain points manually annotated with one out of seven defined roof types, and further labelled into individual roof planes. Data augmentation methods is proposed and implemented to produce an alternative version of the dataset that is large enough for training purposes.

To evaluate the suitability of our dataset for the use in the training of a deep neural network, we adopt a recognized network for point cloud processing, PointNet++, and train it using the augmented dataset. The trained network is tested on a portion of the dataset, which results in a predicted plane segmentation of roof structures. The results indicate that our 3D dataset is suitable for training of a deep neural network. In addition, this indicates that deep learning proves to be promising in automation of the segmentation step in 3D modeling.

Sammendrag

Stadig blir behovet for tiltak for å redusere effekten av klimaendringene mer kritisk. Ved å fremme bruken av fornybar energi, kan man sørge for opptil 90% av CO_2 reduksjonene som behøves innen 2050. Betydningen av 3D data blir stadig tydeligere innenfor dette fagfeltet, da detaljerte modeller kreves for å gjøre dagens moderne byer mer effektive og miljøvennlige.

I denne oppgaven ønsker vi å bidra til å tilgjengeliggjøre 3D data ved å utforske anvendbarheten til punktsky-data i en av dagens største trender innenfor automatiserings-teknologi, kunstig intelligens. Vi vil spesifikt sette et søkelys på automatisering av segmenterings-steget i etableringen av 3D modeller av tak-strukturer. Vi presenterer her et nytt datasett for bruk i dyp læring ment for å utføre semantisk segmentering av 3D-punktskyer bestående av tak-strukturer. Vårt mål er å tilby et datasett av høy kvalitet, basert på ekte tak-strukturer, som skal resultere i gode prediksjoner av tak-segmenter, og være anvendbart for bruk i Norge.

Datasettet er basert på "Light Detection and Ranging" (LiDAR) data, samlet inn over Trondheim kommune. To ulike versjoner av datasettet er etablert. Det originale datasettet består av 906 tak i Trondheims-området. Begge datasett inneholder punkt manuelt annotert med én av syv definerte taktyper, samt en videre inndeling i individuelle takplan. Metoder for å utføre data augmentering er foreslått og anvendt for å etablere en alternativ versjon av datasettet med flere treningseksempler.

Videre er datasettets egnethet for bruk i trening av dype neurale nettverk evaluert ved hjelp av et velkjent nettverk for prosessering av punktskyer, PointNet++. En stor del av det augmenterte datasettet er brukt for treningen av nettverket, før testing er gjennomført på den gjenværende delen. Resultatet fra denne prosessen er predikerte segmenter av tak-strukturer inndelt i ulike plan. Resultatene indikerer at vårt 3D-datasett er velegnet for å trene dype neurale nettverk. I tillegg finner vi indikasjoner på at dyp læring kan være gunstig i automatiseringen av segmenteringssteget i etableringen av 3D modeller.

Preface

This paper is a master thesis written for the Department of Civil and Transport Engineering at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. The thesis is part of the study program Engineering and ICT with a specialisation in Geomatics, and was written in the spring of 2021.

We would like to thank our supervisor, Hongchao Fan, for his invaluable encouragement, help and motivation. For this, we are forever grateful.

We are also grateful to Trondheim municipality for providing us the LiDAR point cloud of the Trondheim area and to Chaoquan Zhang for providing technical support. Lastly, we want to thank Thorleif Orre and Jonas Myren Mo for proofreading this master thesis.

Trondheim, June 2021
Marie Ting Falch Orre
Vilde Myren Mo

Contents

Abstract . . . . **iii**
Sammendrag . . . . **iv**
Preface . . . . **v**
Contents . . . . **vi**
Figures . . . . **ix**
Tables . . . . **xi**
Acronyms . . . . **xii**
1 Introduction **1**
1.1 Goal and Research Questions 4
1.2 Research Method 5
1.3 Defining the Scope 5
1.4 Outline of the Thesis 6
2 Background and Related Work **7**
2.1 Fundamental Principles 7
2.1.1 Point Cloud Data 7
2.1.2 Airborne LiDAR Scanning 8
2.1.3 Traditional Segmentation 10
2.2 Deep Learning 15
2.2.1 Principles of Deep Learning 15
2.2.2 Influential Deep Learning Methods for Point Cloud Part Segmentation 19
2.3 Existing Benchmark Datasets 27
2.3.1 Image Derived Datasets 27
2.3.2 LiDAR-based Datasets 28
3 Roof Segmentation Dataset **32**
3.1 Proposed Guidelines 32
3.2 Data Acquisition 33
3.3 Data Labelling Convention 34
3.4 The Manual Segmentation Work 39
3.5 Data Augmentation 41
3.6 Preparing the Dataset for Segmentation 43
3.7 Experiences with Manual Labelling 44

3.8 The Overview of the Resulted Datasets 45
3.8.1 The Original TRD3DRoofs Dataset 48
3.8.2 The Augmented TRD3DRoofs Dataset 51
4 Deep Learning-Based Roof Segmentation using TRD3DRoofs 56
4.1 Experimental Setup 56
4.1.1 Software 56
4.1.2 Hardware 57
4.1.3 Part Segmentation using PointNet++ 57
4.2 Experimental Results 58
4.2.1 Model Configurations 59
4.2.2 Optimal Number of Points: Final Results on PointNet++ 61
5 Evaluation and Discussion 64
5.1 Evaluation 64
5.2 Manual Evaluation 68
5.3 Discussion 70
5.3.1 Labelling Strategy for Deep Learning Purposes . . . 70
5.3.2 Discussion of the Achieved Results 71
5.3.3 Calculation of Normal Vector using *k*-nn: Effects of Varying the *k* 73
5.3.4 Rule-based Post-Processing 74
5.3.5 Hyperparameter Optimization and Training Split . . 74
5.3.6 Comparison of our Results with the Results from PointNet++ 76
5.3.7 Choice of Neural Network 78
5.3.8 Deep Learning vs Traditional Segmentation 79
6 Conclusion and Further work 80
6.1 Limitations 82
6.1.1 TRD3DRoofs 82
6.1.2 PointNet++ 83
6.2 Further Work 83
Bibliography 86
A Train, test and validation splits for all models 98
B Visual results MSG_80 100
B.1 Type 1: *Flat* 100
B.2 Type 2: *Hipped* 103
B.3 Type 3: *Gabled* 106
B.4 Type 4: *Corner Element* 109
B.5 Type 5: *T-Element* 112
B.6 Type 6: *Cross Element* 115
B.7 Type 7: *Combination* 118

C Visual results MSG_100 121

C.1 Type 1: *Flat* 121

C.2 Type 2: *Hipped* 124

C.3 Type 3: *Gabled* 127

C.4 Type 4: *Corner Element* 130

C.5 Type 5: *T-Element* 133

C.6 Type 6: *Cross-Element* 136

C.7 Type 7: *Combination* 139

D Manual Evaluation on MSG_80 142

E Manual Evaluation on MSG_100 145

Figures

2.1 Time-of-flight principle used in LiDAR. 9
2.2 Example of a roof structure segmented into its contained planes. 11
2.3 Primitive fitting: Workflow of the building model reconstruction by applying a building primitive library. Image origin: [29]. 11
2.4 Region growing as proposed by [41]: Result of roof segmentation. Image origin: [41]. 14
2.5 A simple artificial neural network with three hidden layers. 16
2.6 Illustration of the PointNet++ architecture. 20
2.7 Illustration of the IoU metric. 26
2.8 H3D: Class labels present in the H3D dataset. No segmentation of roofs into planes are present. Image origin: [15]. . 30

3.1 Visualizations of LiDAR Point Cloud. 34
3.2 Overview of the area of 3D capture surrounding Trondheim. 35
3.3 Visualization of six of the defined roof types. 36
3.4 Visualization of the defined plane shapes labelled with the corresponding digit. 37
3.5 Visualization of desired result from manual segmentation and labelling. 40
3.6 A step-by-step illustration of the manual segmentation process of a single roof structure. 41
3.7 Data augmentation by splitting of roof structure of roof type 7 *Combination*. 42
3.8 Visualizations of all different label combinations of a roof structure of type 2 *Hipped*. 42
3.9 Example of polygon including balcony. FKB-data from ©Kartverket. 45
3.10 Map showing the location of building footprint of all roofs in TRD3DRoofs. 46

3.11 Examples of the labelled point cloud data in a selected residential area in Trondheim. 47
3.12 Examples of manually labelled ground truth data. 49
3.13 Pie chart of the roof type distribution in the original TRD3DRoofs dataset. 50
3.14 Histogram showing the distribution of roofs based on number of points for the original TRD3DRoofs dataset. 50
3.15 Pie chart of the roof type distribution in the augmented TRD3DRoofs dataset. 52
3.16 Stacked histogram showing the distribution of roofs for both the original and augmented TRD3DRoofs dataset. 52
3.17 Distribution of plane labels before rotation. 53
3.18 Distribution of plane labels after rotation. 54
3.19 Violin diagram showing the point density distribution of the Augmented TRD3DRoofs dataset. 55
3.20 Distribution of plane labels after rotation and filtering. . . . 55

4.1 Plot of the achieved mIoU vs. the number of roofs used for training of the model. 60
4.2 Plot of the achieved PA vs. the number of roofs used for training of the model. 61
4.3 Result of predictions performed by the MSG_80 model compared to the corresponding ground truth data. 62
4.3 Result of predictions performed by the MSG_80 model compared to the corresponding ground truth data. 63

5.1 Train and testing PAs for model MSG_80. 65
5.2 mIoUs for model MSG_80. 65
5.3 Example of a case where the labels are switched for ground truth and predicted roof. 67
5.4 Visualization of error present along ridges in the predicted result. 75
5.5 Examples of aligned models in the ShapeNet dataset. Figure origin: [67]. 77

Tables

2.1 3D part segmentation comparisons of mIoU on points on ShapeNet Parts. 25

3.1 Overview of plane labels and geometry. 39

4.1 Train, validation and test split for MSG_100. 58
4.2 Different training configurations. 59
4.3 Train, validation and test split for MSG_80. 59

5.1 Result of type-by-type metrics for MSG_80. 66
5.2 Visual evaluation performed on model MSG_100 and MSG_80. 69

A.1 Train, validation and test splits for all models. 99

Acronyms

***k*-NN** *k*-Nearest Neighbours. 43, 73

AI Artificial Intelligence. iii, 15, 56

ALS Airborne LiDAR Scanning. 3, 8, 10, 28–32, 77

CAD Computer-Aided Design. 19, 27, 77

CNN Convolutional Neural Network. 22–24, 78, 84

CPU Central Processing Units. 57

ETRS89 the European Terrestrial Reference System 1989. 33, 34

FCL Fully Connected Layer. 17

FN False Negative. 25, 26

FP False Positive. 25, 26

FPS Farthest Point Sampling. 21

GPU Graphical Processing Units. 56, 57

H3D The Hessigheim 3D Benchmark. 8, 29

IoU Intersection over Union. 26, 64

ISPRS International Society for Photogrammetry and Remote Sensing. 29

LiDAR Light Detection and Ranging. iii, ix, 3, 4, 8, 9, 12, 13, 27–30, 32–34, 44, 45, 57, 74, 80, 82

LSA Location Spatial Aware. 22

mIoU mean Intersection over Union. x, 22, 25, 26, 58–61, 64–67, 76–79

MLP Multi-Layer Perceptron. 17, 19, 20, 22, 23

MLS Mobile LiDAR Scanning. 9, 10, 28

MRG Multi-Resolution Grouping. 21

MSG Multi-Scale Grouping. x, 21, 59–61, 64–66

MVS Multiple view Stereovision/Multiview Stereo. 27, 28

NDCs Nationally Determined Contributions. 1

NN2000 Normal Null 2000. 33, 34

NTNU Norwegian University of Science and Technology. 39, 40

OE Orientation-Encoding. 21

PA Pointwise Accuracy. x, 25, 26, 58, 60, 61, 64–68, 71

PCA Principal Component Analysis. 43

RANSAC RANdom Sample Consensus. 11, 12, 14

RGB Red Green Blue. 84

RGB-D Red Green Blue -Depth. 8

RPCA Robust Principal Component Analysis. 14

SDWs Spatial Distribution Weights. 22

SFM Structure-From-Motion. 27, 28

SPGS Super Point Graphs. 23

SRI Solar Reflectance Index. 45

SSG Single Scale point Grouping. 20, 21, 59

TLS Terrestrial LiDAR Scanning. 9, 28

TN True Negative. 25, 26

TP True Positive. 25, 26

UAV Unmanned Aerial Vehicles. 10, 28, 29, 84

UNFCCC United Nations Framework Convention on Climate Change. 1

UTM Universal Transverse Mercator. 33, 34

Chapter 1

Introduction

Geospatial data, such as three-dimensional point clouds, have the last couple of years gained increased interest among researchers. Point clouds are the basis for virtual 3D models representing real-world scenes and can be used for applications such as estimations of the biomass of a forest area [1], driving of autonomous vehicles [2] or the reconstruction of building models [3]. Such building reconstruction 3D models are applicable in a wide aspect of fields, including renewable energy applications. There is an immediate need for action to reduce the impact of climate change, and for this renewable energy is a key factor. The Paris Agreement of the United Nations Framework Convention on Climate Change (UNFCCC) is an official binding global treaty on climate change [4]. Nationally Determined Contributions (NDCs) work as the central implementation tool for countries under the Paris Agreement, and renewable energy is an essential component of this as it can provide 90% of the CO_2 emissions cuts that are needed by 2050 [5]. [1]

Solar energy is one source of renewable energy, where power is directly harnessed from the sun, using solar panels. By analysing 3D models of roof structures, simulation and estimation of potential solar energy production in urban areas can be performed [6]. Such estimations are of great relevance for solar power distributors, as well as for research on how renewable energy can be utilised in the future. Another possibility is the investigation of the most suitable placement of new solar panels in a city, a task connected to urban planning. These are all applications of virtual 3D models that could help Norway reach the goals set by the Paris Agreement.

For the final 3D model to be of value, the processing of the raw point

[1] IRENA is the Internatational Renewable Energy Agency, to read more about how renewable energy is a key component of NDCs – the central implementation tool for countries under the Paris Agreement, visit `https://www.irena.org/`

cloud is crucial. Multiple steps are necessary for the establishment of a complete 3D model. An essential step in this process is the grouping of points into segments based on common characteristics and assign semantic meaning to each segment. For each of the segments, polygons needs to be derived, before the final modeling can be performed resulting in a complete 3D model of the real-world object. Thus, automatic segmentation of roof-planes as a part of roof structure detection is of great importance in geospatial analysis of building data and is the focus of this master thesis.

The task of grouping similar datapoints and assing them meaning is a data processing task termed *semantic segmentation*. For the semantic segmentation of an object into meaningfull object parts, such as the segmentation of roof structures into separate roof planes, one can further specify this as a task of *part segmentation*. Therefore we will often talk about semantic segmentation and part segmentation of roof structures interchangeably, as part segmentation is a sub-category of sematic segmentation.

Difficulties concerning 3D data, such as its irregular structure and non-uniform densities combined with large amounts of data, has historically made the handling of 3D data a challenge. Another challenge connected to the automatic segmentation of roof structures is the fact that such structures might be complex. No general data-driven method exists for the segmentation of complex roof structures, though a lot of different approaches have been applied for different scenarios [7] [8] [9]. As new technology develops, there is a hope that this will gradually change as the ability to handle heavy computational tasks is continuously increasing.

Simultaneously, the field of machine learning and computer vision, with the invention of deep learning-based networks imitating the learning process of human brains, have entered a new era. Semantic segmentation is a key area of interest in the field of deep learning, as it allows for a deeper understanding of real-world scenes. Increased computational power, combined with advancement in acquisition technology for point clouds, have made it possible to extend the use of deep learning-based networks from segmentation of 2D images to that of 3D point clouds. We believe that such deep learning-based networks are a suitable tool in the establishment of a more general process for the segmentation of roof planes.

The task of applying deep learning methods designed for 2D on 3D data is however non-trivial, due to the differences regarding the structure of the data. Compared to 2D images arranged in pixels, 3D point clouds are often unstructured and are not consistent in density. Supervised deep learning systems depends significantly on the availability of annotated ground truth data, and for point clouds the amount of data needed is immense. This need for large amounts of labelled training data is one of the main challenges that machine learning methods, and especially data-hungry deep learning neural networks, are facing [10]. In addition, neural networks needs to be trained on high-quality data to produce good predictions.

The obtainment of high-quality point cloud data of a satisfactory density for deep learning applications are often costly and time-consuming [11]. Through a literature search it was found that dense 3D point cloud datasets designed for the task of roof segmentation do exists. However, these datasets are too dense to be suitable for large scale projects such as solar energy estimations of cities. The state-of-the-art Airborne LiDAR Scanning (ALS) equipment used as standard for survey and mapping projects today deliver a density of 10-12 points/m^2. To obtain a higher point density the expenses are very high, as it demands a need for several acquisition fly-overs of the study area.

Remote sensing data is also area dependent and cannot be easily applied in other areas. The neural network needs to be exposed to Norwegian roof types to be useful for local applications. Consequently, the need arises for a 3D dataset suitable for deep learning-based segmentation for utilization in Norway. To the best of our knowledge, no such dataset containing typical Norwegian roof structures exists.

In this master thesis, we therefore present a new 3D point cloud dataset containing manually annotated roof structures obtained in residential areas of Trondheim, named TRD3DRoofs. The original Light Detection and Ranging (LiDAR) point cloud used was obtained in 2018 and distributed to us by Trondheim Municipality. The dataset consists of 2 199 051 points belonging to approximately 900 real-world roofs. Each roof is manually segmented and annotated with semantic information about both roof structure and distinguishable planes divided into eleven plane types. As we wish to both contribute a dataset consisting of only roof structures representing real-life buildings in the Trondheim area, as well as a dataset suitable for deep learning, an additional augmented dataset is presented. This augmented dataset is derived from the TRD3DRoofs dataset but contains extended data to balance the dataset with regards to roof type. The augmented data is also included to increase the size of the dataset, due to the vast amount of training data needed for deep learning. Having a mean density of 9.07 points/m^2 and being manually annotated with ground truth labels, the augmented dataset is established with the intent of being well-suited for deep learning applications to the problem of roof plane segmentation, and to be applicable in real-world projects.

The evaluation of our datasets suitability for supervised deep learning applications is performed by implementing PointNet++, a state-of-the-art deep learning network, for direct processing of point clouds, and using our ground truth data for training and evaluation. Specifically, the usability of PointNet++ for the task of roof plane segmentation of 3D point cloud data is investigated.

1.1 Goal and Research Questions

This section formally presents the main goal of the thesis, together with two research questions defined to reach the goal.

Goal Create a high-quality 3D point cloud dataset intended for training deep learning applications for the task of segmentation of roof plane structures. The dataset is to be appropriate for applications in Norway, more specifically the Trondheim area.

Research for deep learning applied to point cloud data is increasing, indicating the possibility of a general approach to the problem of 3D roof-plane segmentation. This yields a need for area-specific high-quality training data. The main goal of this thesis is the construction of a 3D point cloud dataset with manually annotated points, intended as training data to train a deep learning model to segment roofs into separate roof planes. Two research questions are proposed that addresses challenges to be solved to reach the goal of the thesis.

Geospatial data obtained by LiDAR techniques are not formatted to be suitable as direct input in deep learning algorithms. Additionally, it lacks semantic information about each point and the surrounding neighbourhoods. To make it possible to use such data as input in deep neural networks, it must be purposefully processed and labelled. The development of guidelines for processing of geospatial data is a crucial part of the development of the dataset, and is therefore the first topic of research in this thesis.

RQ1 How can LiDAR data be processed and labelled, making it suitable as input in deep learning algorithms?

During the establishment of these guidelines, the final composition of the dataset needs to be taken into consideration. The performance of a deep learning algorithm depends on the contents of the dataset it has been exposed to during training. To get the best possible result, the training data should have a certain structure, and to achieve this there will, in most cases, be a need for augmentation of the obtained data. Additionally, the amount of manual work required to create enough data is a problem. Creating more data through augmentation is an option that is both less time-consuming and cheaper than manual labour. The second research question therefore addresses the issue of data augmentation.

RQ2 How can a dataset consisting of 3D point clouds representing roof structures be augmented to create the most suitable dataset for deep learning?

To answer these questions, guidelines based on the needed workflow will be established, together with a procedure for data augmentation. This for the purpose of reaching the research goal.

1.2 Research Method

This section describes the research method applied in this thesis to reach the goal and answer the research questions presented.

First, a literature review was conducted to gain knowledge of the methods and datasets available today. Findings from this process build a basis for the design of the dataset, the choice of network and the metrics calculated for the final evaluation. Following, a strategy for the labelling process of the point cloud data was established, intended to answer **RQ1**. To address **RQ2**, a strategy for augmentation of the real-world data was proposed and implemented. A deep neural network intended for 3D point cloud data was employed and adapted to fit the proposed dataset. The datasets usability for training a neural network was evaluated based on the network results, to measure the degree of achievement of the presented goal.

1.3 Defining the Scope

The scope of this master thesis is the construction of a 3D point cloud dataset of roof structures suitable as input in a deep learning-based approach to the problem of plane segmentation. The thesis does not address the collection and processing of the original raw point cloud. The implementation of improvements for the adapted network, PointNet++ is not addressed in this thesis. Still, the network is modified to fit the proposed dataset.

Time and hardware constraints is another limitation of this thesis. The training of neural networks on the available hardware takes several hours. This makes it impossible to test every combination of model configurations available for PointNet++, with the time available. The time limitation also excludes the possibility of adaption and training of other, more complex, networks on the proposed dataset. The reported results are also dependent on the hardware available at the time when the experiment was conducted.

1.4 Outline of the Thesis

The remaining chapters of the thesis are structured as follows: Chapter 2 Background and Related Work is included as an introduction to relevant topics further explored in the thesis. The acquisition method of the original point cloud is presented, together with a historical perspective of segmentation methods applied in earlier work. Deep learning-based methods are introduced as an alternative to classic segmentation methods, and here the need for large amounts of labelled data is explained and the state-of-the-art network chosen for the evaluation, PointNet++, is detailed. Earlier 3D benchmark datasets of roof-structure data are further presented and discussed, to substantiate the need for a new 3D dataset.

Chapter 3 Roof Segmentation Dataset presents the making of the new TRD3DRoofs dataset and the additional augmented version. Guidelines for the processing of data to make it suitable for deep learning purposes are proposed as an answer to the first research question. Detailed information about the labelling taxonomy and important pre-processing steps are then described. Finally, the two new datasets are presented, including ground truth examples of different roof structures.

In chapter 4 Deep Learning-Based Roof Segmentation using TRD3DRoofs, the deep neural network PointNet++ is trained and evaluated on the augmented version of TRD3DRoofs. Experimental aspects, such as the hardware, software together and details regarding our PointNet++ implementation are presented. The result of the predictions are shown, and later evaluated and discussed, in the following chapter.

Chapter 5 Evaluation and Discussion presents the evaluation performed on the segmentation approach outlined in chapter 4. The obtained results are discussed, seen in the light of current research and relevant theory. Further, choices made in this thesis both regarding the proposed dataset and the implementation of PointNet++ are examined.

The final chapter, Chapter 6 Conclusion and Further Work reviews the main proposals of the thesis and presents the conclusions of the work. Suggestions for further work are proposed, based on the findings of the thesis.

Chapter 2

Background and Related Work

Novel technology is continuously implemented, accepted, and discarded, leading way for what is known as today's state-of-the-art technology. The knowledge obtained by the continuous improvement of technology greatly affects today's research. In this chapter, core theory that forms the fundamental for the rest of the thesis are presented. Further, a deep learning approach to the problem of semantic segmentation of 3D data is introduced. Lastly, existing benchmark datasets for the task of deep learning-based segmentation of point clouds are introduced and discussed.

2.1 Fundamental Principles

This section presents theoretical information about 3D point cloud data and is meant as an introduction to important concepts necessary for the understanding of the work presented in this thesis. Fundamental information about 3D point cloud data is given, together with the acquisition method applied for the data used in this thesis. The concept of point cloud segmentation is detailed, including a historical perspective leading up to one of today's most promising technologies, deep learning.

2.1.1 Point Cloud Data

A point cloud represents a set of points located in 3D space, described by their respectively x-, y- and z-coordinates [12]. Together with additional optional attributes, these coordinates give valuable information to the points, which jointly form a digital representation of a real-life object. This point cloud representation is the most widespread representation of acquired 3D data [13].

The density of a point cloud describes the number of points present per unit area. Based on the density, point clouds may be divided into two categories: dense or sparse point clouds. Here, we use the the definitions from [14], giving the following categories of point clouds: (a) sparse (below 20 points/m^2), and (b) dense (from 20 to hundres of points/m^2).

Deep learning approaches from point cloud data processing are greatly related to the density of the point clouds. The different densities in a point cloud represents different qualities, as they describe the features of the objects varyingly. Datasets based on point clouds can be used for predictions when the density of points in the data used for testing is similar to that in the datasets used for training.

The density varies based on factors such as the method of obtainment, with the earliest approaches being limited by the hardware of the acquisition equipment, computation ability and matching techniques, resulting in sparse point clouds [14]. With better equipment for acquisition established the last couple of years, computer vision algorithms, and increased computational ability, the possibility for creating and processing denser point clouds emerged and has been seen in work such as datasets such as The Hessigheim 3D Benchmark (H3D) [15], DublinCity [16] and DALES [17].

2.1.2 Airborne LiDAR Scanning

Different acquisition methods may be used to obtain point clouds, such as Image-derived methods, Red Green Blue -Depth (RGB-D) cameras and LiDAR systems. For this thesis, where a point cloud representing the Trondheim area was acquired by Trondheim Municipality and later used as a basis for a training dataset, Airborne LiDAR Scanning (ALS) was the method of acquisition. By using pulses of light from a laser, the distance between the acquisition instrument and the observed object may be determined, making LiDAR a suitable remote sensing method for point cloud acquisition [18]. A point density of 12-20 points/m^2 is typically acquired through ALS when using state-of-the-art equipment for large scale projects. A higher density can be obtained through the conduction of multiple acquisition fly-overs of the same area at a higher expense.

Figure 2.1 gives a visual explanation of the time-of-flight concept applied.

The resulting 3D coordinates from the acquisition, together with other optional attributes, describe the features of the points. The x- and y-coordinates denotes the planimetric ground location, while the z-coordinate defines the elevation. As mentioned, LiDAR systems detect the echo of a pulse, and the intensity of this laser pulse at return is a potential

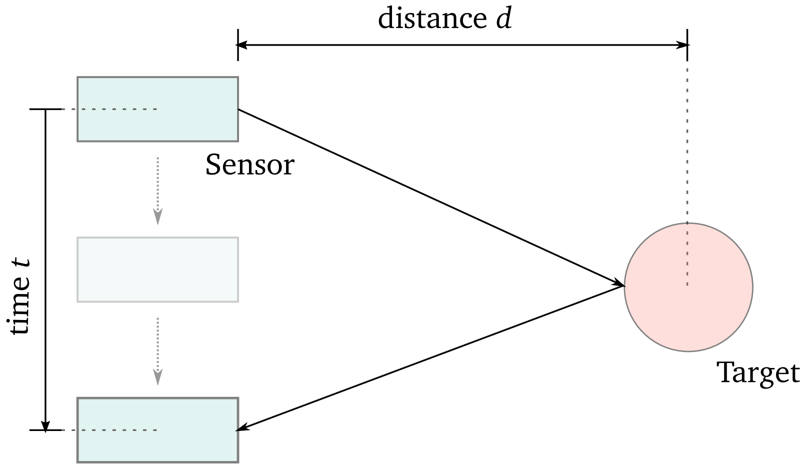


Figure 2.1: Time-of-flight principle used in LiDAR. The “echo” that is reflected after the light from the instrument hits the desired object, is detected. As the speed of propagation of the pulse is known, and the time delay between the originally released pulse and its echo may be measured, it is possible to deduce the desired distance between the device and the object. Further, the information is converted into 3D coordinates, leading to the resulting point cloud of the object and the surrounding area [19].

attribute produced by this method. Other attributes generated is a unique identifier, a timestamp for the return of a pulse, the number of returns one single pulse resulted in, and the return number for this particular pulse [19].

One laser pulse can illuminate multiple targets, as the pulse will have an energy distribution both along and across the beam direction [12]. As a result, one pulse may lead to the reflection of multiple echoes from multiple targets. When scanning buildings from above, the first echoes might be reflections from roof structures, while intermediate and last echoes can be reflections from surrounding vegetation or the ground below.

By using LiDAR systems as the acquisition method, the coordinate information is known to be reliable, as there is a direct acquisition of spatial coordinates. No complicated matching procedures are necessary, reducing the risk of information loss. On the other hand, as the information is positional, the derivation of semantic information might be a challenge. Along homogeneous surfaces, the information tends to be dense, but along break lines the data is exposed to a possible information loss, as almost no data is detected at these lines. Another potential issue is the fact that no inherent redundancy is present, leading to a possibility for corrupted data [19].

Based on the platform of the scanning device, LiDAR systems are divided into Terrestrial LiDAR Scanning (TLS)), Mobile LiDAR Scanning

(MLS) and ALS. For outdoor applications, ALS is often applied [20] [21] [22]. For the acquisition of building data in urban areas, ALS has been commonly applied such as by Morgan and Tempfli in their work on automatic building extraction [23], and Chen et al. for their approach on rooftop reconstruction [24]. Kim and Shan [25] used ALS as an acquisition method for their approach for building roof modeling, and Hu and Yangs visual perception driven building representation is another example of a method based on an ALS point cloud [11].

As ALS is conducted either from aircraft, helicopters, or Unmanned Aerial Vehicles (UAV), this is the most suitable acquisition method for the obtainment of a point cloud consisting of roof structures from buildings and is also applied in this project.

2.1.3 Traditional Segmentation

The process of classifying a point cloud into subsets based on common characteristics among the points is known as segmentation [26] [27]. Points belonging to the same area will have the same properties, and be of spatial proximity. This separation of a point cloud is a fundamental step in 3D point cloud reconstruction, making it possible to perform tasks such as object detection and classification [27]. To further exploit the point clouds, making them useful for further analyses, it is necessary to understand what kind of object each point represents. The purpose of segmentation is to correctly assign each point contained in the point cloud to a subset, giving value to the complete point cloud.

Traditional segmentation can be done by a variety of methods. Generally, the segmentation process typically consists of defining criteria, both for spatial proximity and other similarity between the points. These values are then calculated, and points are placed into segments based on the criteria they satisfy.

Roof Plane Segmentation

For tasks such as urban planning and the placement of solar systems, it is vital that the point cloud is correctly segmented into buildings and the surrounding environment. An important step in this process of classifying an entire urban environment is the segmentation of a roof structure into separate planes, as illustrated in Figure 2.2. For renewable energy applications, such as the simulation and estimation of solar energy generation, the design of each roof is of high relevance. By segmenting the roof structure into its distinctive planes, the number of planes and the angles they are placed in may be known.

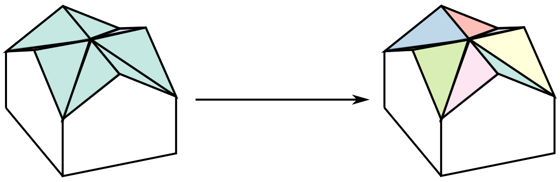


Figure 2.2: Example of a roof structure segmented into its contained planes.

Different methods for the task of roof plane segmentation have been presented in earlier work. The earliest model-driven methods has a primary focus on detecting simple shapes in the data, such as geometric structures or edges. Some early approaches utilise proximity or other attributes to find similarities. Extensive research on 2D images was already performed when the interest in 3D data increased, and consequently, several approaches were developed for the segmentation of images before they were adapted to point clouds. [28] uses primitives to perform segmentation of planar and curved surfaces on range images. The proposed algorithm was one of the first to introduce segmentation based on primitives, rather than individual pixels. Xiong et al. [29] proposed flexible building primitives for the purpose of modeling buildings in 3D. Based on basic elements in roof topology graphs, the technique facilitates the use of model-driven methods for all kind of buildings. With the use of these basic primitives, the segmentation of complex roof planes are made possible. Figure 2.3 illustrate the primitive fitting process applied in [29].

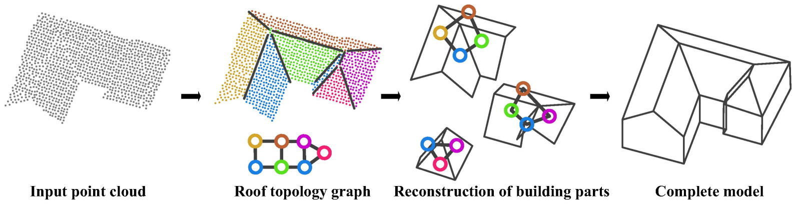


Figure 2.3: Primitive fitting: Workflow of the building model reconstruction by applying a building primitive library. Image origin: [29].

The RANdom Sample Consensus (RANSAC), was introduced by Fischler and Bolles in 1981 [30]. The algorithm is based on the concept of fitting a model to the data. Approaches cantered around RANSAC iteratively fits a model to an arbitrary subsection of the data, until a consensus as to which model describes the data most accurately is reached.

RANSAC is a robust algorithm even in cases of noise and outliers which are often present in point clouds. However, there is a chance that the algorithm produces false surfaces as it in some cases detects planes that do not belong to the same object surface. Awwad et al. proposed a modified version of the RANSAC algorithm, the seq-NV-RANSAC algorithm, to prevent the extraction of such spurious surfaces [31]. Their approach sequentially checks that the normal vector between the point cloud and the calculated RANSAC plane is below a given threshold. This approach gives, in addition, an improvement of the quality of the generated planes. Another approach for reducing the tendency of generating false planes was introduced by Xu et al. [32]. Their approach addresses the problem by introducing the weighted RANSAC, where the hard threshold for the normal vector consistency is changed into a soft threshold founded on two weight functions.

As RANSAC and other model fitting-based segmentation methods contain a solely mathematical principle, they are robust against outliers and noise. Another benefit is the ability to process large amounts of point cloud data in a relatively short time [33]. The main challenge with approaches utilising RANSAC, is the fact that it is a non-deterministic algorithm, meaning that the same input data possibly will yield different results, and the produced result will not necessarily be the optimal solution.

Clustering methods are considered an unsupervised learning problem, as the method does not require any knowledge about the different classes prior to the segmentation. The purpose of the clustering process is to distinguish between different groups of points, based on their characteristics [34]. From the result of the clustering, it is possible to distinguish between hard and fuzzy clustering. In fuzzy clustering, each point might have a varying degree of membership to each output cluster, while they would either belong to a cluster or not, in a hard clustering [35]. Sampath and Shan [7] iteratively uses the K-means clustering algorithm, first proposed by MacQueen [36] to create a polyhedral model of building roofs based on LiDAR point clouds. Normal vectors for small groups of points are calculated and clustered together, giving the principal direction of the roof planes. By identifying intersecting planes and break lines, the polyhedral roof models are constructed. Sampath and Shan [37] improved their work in 2010, using a fuzzy K-means approach and optimizing the clustering process by using a potential-based approach to estimate the number of clusters.

Occasionally the normal vectors of neighbouring planes are hard to distinguish or untrustworthy, making the fuzzy K-means algorithm less reliable for point cloud segmentation. Kong et al. [8] introduced a combination of the K-means and K-plane algorithms giving a more satisfying result for the segmentation of roof structures. Their approach estimates the clustering centres for the K-means algorithm directly from the elevation of the point cloud, improving the initialization.

[38] proposed a method for automatic roof plane segmentation where the raw LiDAR points are classified into two groups: ground and non-ground points. To extract the planar roof segments, clustering is applied based on coplanarity and neighbourhood relations of a point. Lastly, rule-based post-processing is applied to refine the segmentation.

Albano investigated a fuzzy c-means clustering method for the automatic segmentation of 3D point clouds containing roof structures [39]. A fuzzy c-means clustering method is implemented to determine the clusters, where the data points are iteratively relocated among various clusters until the largest difference possible is attained. A data point might belong to any of the clusters, and this degree of belonging is determined by the similarity. Further, the method is refined through a density clustering and connectivity analysis where planar and coplanar planes are separated. Such planes might have roof segments that are parallel or mathematically identical but that are spatially separated. In terms of geometric accuracy, the method yields good results.

While clustering-based segmentation is easy to understand and implement, it still has some limitations in the case of 3D point cloud data. Features of points are typically generated using local neighbourhoods, making techniques utilising clustering sensitive to noise and outliers. The choice of neighbourhood will also affect the result of the segmentation.

Earlier work also includes those based on the simple and effective region-growing method. Such approaches iteratively perform a set of steps until they reach a termination criterion; (1) one or more seed points is to be selected and used to initialize a new segment. These seed points cannot be present in an existing segment; (2) a homogeneity criterion is decided; (3) all the neighbouring points of a segment is tested against the criterion and included into the segment if they meet the criterion; (4) the segment grows from the included point(s) until no more additional points are available [26]. Such region growing methods are primarily sensitive to three factors: the choice of initial seed point(s)and the homogeneity criterion together with the growth unit.

Vo et al. [40] proposed a novel region-growing algorithm for point cloud segmentation in urban areas, such as the segmentation of building roofs. Two stages compose the algorithm that is based on a coarse-to-fine concept. Their approach uses octree-based voxelization, meaning that the input point cloud is represented as voxels. A region growing step is performed on this representation on the original point cloud, resulting in the extraction of the major coarse segments. Later, the output from this step goes through a refinement process giving the result of the segmentation.

Further, Xu et al. [41] uses a voxel-based region growing method for the segmentation of building roofs. By exploiting the fact that roofs consist of planar surfaces and are easily geometrically separated from other objects, they present a method using region growing with Robust Principal Component Analysis (RPCA) on a voxelized point cloud. Figure 2.4 shows the results from the roof segmentation proposed in this work.

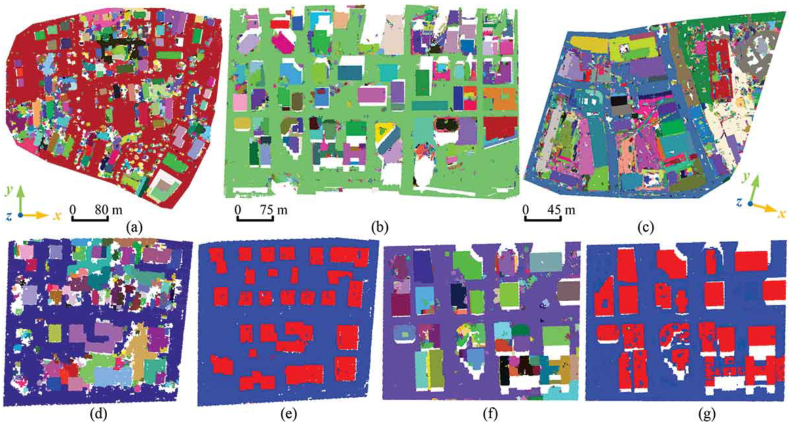


Figure 2.4: Region growing as proposed by [41]: Result of roof segmentation. Image origin: [41].

In his investigation on roof segmentation, Albano also proposed a region growing approach combined with RANSAC [39]. A region growing method where each rooftop is described with the finest spatial detail possible is used, inspired by the work of Sun and Salvaggio [42]. The initial seed point is found by an examination of the points surface smoothness, where the point with the smallest curvature is chosen. Using the normal vectors and curvatures of the neighbouring points, the region growing process segments points together. RANSAC is applied to each segmented area, with the purpose of fitting a virtual plane from the candidate points, and then force the points to move on to this plane in order to assign an impeccable flatness property to each surface. Compared to the fuzzy c-means clustering method, this approach achieved slightly better performance, but with greater computational time.

Shao et al. [43] proposed a novel method for the extraction of roofs in 3D point clouds, with a top-down strategy implemented rather than the traditional bottom-up approach usually applied. Based on cloth simulation, seed point sets containing semantic segmentation is detected at the top of the scenario. Instead of a single seed point, the method extracts multiple initial points for the region growing. This region growing technique is further exploited to extract building roof points. The authors claim that their method simplifies the roof extraction workflow and gives way for rapid extraction, at the same time as the risk of over-segmentation is reduced.

Compared to clustering-based methods, region-growing methods utilise global information, making them more robust to outliers and noise present in 3D point cloud data. They do, however, typically tend to over- or under-segment, and the accurate determination of region boundaries is a challenge [33].

Generally, the segmentation and further processing of a point cloud is a rather challenging task. The unordered nature of the points combined with the varying density distributions and large amounts of data makes the segmentation a complex and time-consuming assignment. The advent of machine learning and especially deep learning-based neural networks have introduced a possible solution to this problem, leading to a revolution in the case of 3D point cloud processing.

2.2 Deep Learning

As an opening note to the coming decade, Forbes wrote that "the increasing ability of machines to learn and act intelligently will absolutely transform our world" [44] and accordingly placed AI at the very top of the list of technology trends that will define the next 10 years. The interest in AI has been rapidly growing for some years now, from simple single-layer, feed-forward neural networks to what is largely considered today's state-of-the-art in most AI disciplines: deep learning. Making AI understand real, sensed data through for example object detection, classification, and segmentation has been particularly in focus as it facilitates automatic, in-depth understanding of the world around us. In this section, a brief introduction to the principles of deep learning is given, before a number of influential deep learning methods designed for the task of part segmentation of point clouds is presented.

2.2.1 Principles of Deep Learning

Deep learning is a sub-division of AI that can be described as a more sophisticated and mathematically complex branch of machine learning. When talking about machine learning today, people generally refer to deep learning. The core concept of this technology is to allow machines to learn to recognize patterns the same way we humans do – through experience. Mimicking the network of neurons in a biological brain, the algorithm is made up of layers of artificial neurons that learn a so-called activation function mapping from input to output [45]. The structuring of the neural network is termed the network architecture and can be looked upon as a deep, weighted and directed graph made up of layers of neurons. An example of a simple neural network is shown in the figure Figure 2.5 below.

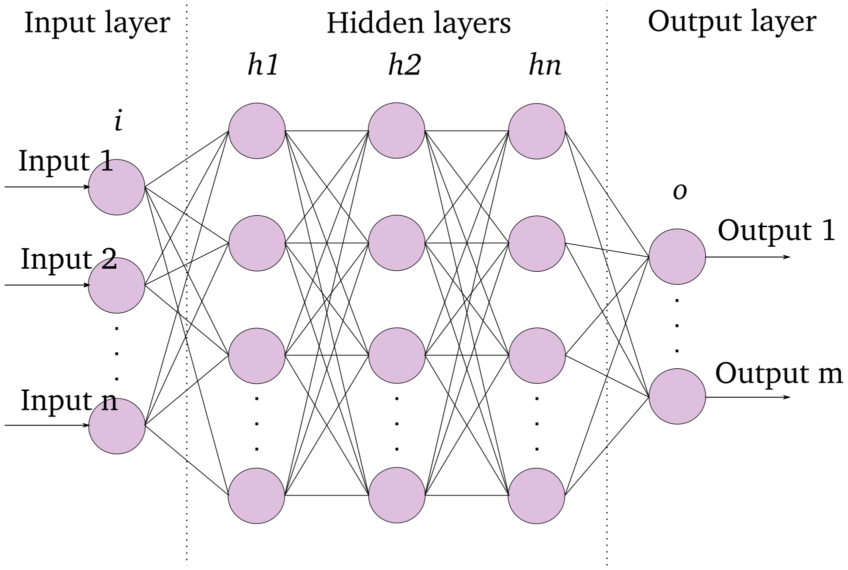


Figure 2.5: A simple artificial neural network with three hidden layers.

Similar to a person, the network will learn a feature if it is exposed to enough examples. In training, the example data are inputted together with its corresponding ground-truth with correctly labelled data that tells the algorithm what it is looking at. After processing the data, the network will conclude as to the meaning of the input data, presented in the form of a predicted output label for each datapoint. The algorithm learns by implementing a loss function that calculates the difference between the predicted label and the given ground truth, given some error criteria, and updating the network weights so that it minimizes the loss and consequently maximizes the probability of the network predicting the correct label next time it sees a similar example.

To ensure that the network has enough parameters to learn a precise mapping, especially for more intricate features, it is important that its architecture is sufficiently complex [46]. If not, the model will not be able to accurately capture relationships present in the data. The complexity is constituted by the number of layers, referred to as the depth of the network, as well as the arrangement of different layer types and their dimensions. All neural networks have one input layer and one output layer with dimensions that corresponds to the dimensions of the input- and output data, respectively. All other intermediate layers and are referred to as hidden layers and accounts for all computations performed in the neural network.

Deep neural networks are often defined as networks that utilise numerous hidden layers, where each layer learns specific features at different abstraction level, e.g., object parts, contours and colours, corners, edges and smaller patterns [47].

Some of the most common hidden layer types are fully connected, convolutional, pooling, upsampling and recurrent layers. Fully Connected Layers (FCLs) are perhaps the most frequently used and are found in most architectures. They connect every neuron in one layer with every neuron in the next. Multi-Layer Perceptrons (MLPs) are the simplest form of artificial neural networks and consist of the input layer, one or more FCL(s) and the output layer. The next type is convolutional layers. In these, the presence of smaller features is searched for by convolving one or more kernels, with associated kernel weights, over the data. Such layers are often followed by a pooling layer. Pooling layers reduces the dimensions of the data by combining the outputs from multiple neurons in the previous layer into a single input to a neuron in the next layer. Typically, by preserving the maximum or average value. Finally, we have the recurrent layers. These types of layers can be used to give a network a memory resembling property by adding a feedback loop that includes the output from a previous calculation done by the same layer as input together with the output from the preceding layer.

As the name suggests, what is learned by the network in the hidden layers are somewhat of a mystery. For this reason, deep learning is commonly referred to as a "black box" [48]. We simply do not understand exactly which information is emphasized and which is ignored when a deep neural network arrives at a prediction, and to an even lesser extent can we control it. The only thing we can control is the examples we expose the network to and the correctness of their labels. Because of this, the quality of the data used as input becomes all the more important.

Given the fact that a neural network only learns what it is shown during its training process, there are several factors that are important to consider when generating a dataset for the purpose of deep learning. First of all, it is important that the labels are correct and accurately determined as poorly labelled data can confuse the model and deter it from reaching an optimal mapping. Furthermore, it is crucial that the dataset is large enough so that the network will have been exposed to enough examples to be able to properly learn the object in question. Another essential consideration is that the data must contain a good variety of, for example, possible shapes, positions, rotations, colours, surroundings, and combinations of such traits, to become able to generalize well within a class of objects. A common practice to increase the size and diversity of a dataset is to perform data augmentation. This entails generating synthetic example data by shifting, scaling, rotating, skewing or in similar ways alter the initial data. This is a particularly critical step when dealing with unordered point cloud data as such models need to be invariant to all permutations of input order for a point set.

Preliminary to training a deep neural network, a dataset is typically split into a training, and testing set, with the training set containing the bulk of the data. The network is first trained using solely the training set for a fixed number of iterations before the final network is evaluated using the never-before-seen test data. This provides an unbiased assessment of the network performance. The aspiration of the training phase is that the model should learn the general characteristics of the data in such a way that it also performs well on the unseen data in the testing set. A challenge is to train the network long enough for it to learn necessary complex features, but not too long because it might start to memorize the training data in general. This is known as overfitting. An overfitted model is not desirable, as it will perform inadequately when exposed to new, unseen data [49].

Often the dataset is split into an additional portion called the validation set. This is used to evaluate the model during training as a tool for tuning the model hyperparameters. The hyperparameters comprise a number of model-specific parameters that affect the training process, for example, the learning rate, momentum, batch size, number of iterations, step size, random dropout, activation function, loss function, and decay rates, to mention a few [50]. These differ from other model parameters, like the model weights and the activation function coefficients, by the fact that they are set in advance and not learned during training. They are used to gain more control over the training process, and fine-tuning of these parameters is crucial for the performance. A vital part of designing a good model is to identify good values for these parameters, and these should therefore be optimized to prevent both over- and underfitting of the network. This process is called hyperparameter optimization [51].

In this section, we have only superficially remarked on the most vital aspects surrounding how a supervised deep neural network learns to understand and recognize features. We have talked about the network architecture, the importance of the input data, the training process and the numerous parameters that must be decided. With this many variables, it is almost an impossible task to point out a single optimal solution.

2.2.2 Influential Deep Learning Methods for Point Cloud Part Segmentation

For a long time, advancements in this field were mainly reserved for 2D image processing and similar problems within the field of computer vision. Countless, large-scale datasets with annotated images, like ImageNet [52], KITTY [53], Microsoft COCO [54] PASCAL VOC [55] and Cityscapes [56], has been made publicly available over the years and has allowed for deep learning algorithms to achieve incredible results in various image recognition tasks [57], [58] [59] [60] [61]. Even though these methods have come extremely far, they all have one undeniable limitation: they can never be better than approximations, as the real world has three dimensions and not two.

Unfortunately, the task of adapting algorithms designed for 2D applications to 3D point cloud data represents a considerable engineering challenge. This is not only due to the high dimensionality but also to the fact that point clouds, unlike 2D data, is unordered by nature and that the data density is extremely varying, making it unfeasible to directly apply these methods to 3D cases [62]. To elude these problems, many researchers focus on volumetric methods where the point cloud is typically transformed to a regular voxel grid or a collection of multi-view images before processing [63] [64] [65]. Whereas these methods benefit from the fact that they can adopt 2D techniques, they also adopt the accompanying inaccuracies of the necessary quantization.

However, with computers becoming progressively more powerful and 3D data acquisition tools becoming cheaper and more precise, spurring the release of several new benchmark datasets, more and more work is being conducted on directly applying deep learning on 3D point clouds [66]. Methods that do not use an intermediate transformation, are generally termed point-based methods and can be further be categorized into MLPs, convolution, and graph based methods. Here we present some of the point-based deep learning architectures in each category that have been highly influential or are considered state-of-the-art at the task of part segmentation of 3D point cloud data.

For comparison purposes, the methods performance on the the synthetic dataset ShapeNet Parts [67] are proposed. This is a well-known benchmark dataset containing shapes represented by 3D Computer-Aided Design (CAD) models. Novel work often evaluate their performance on part segmentation by training and testing on this dataset.

MLP-based methods

One of the first deep learning networks that successfully processed a raw point cloud, that is, without first transforming the data, is called PointNet and was presented by Qi et al. [62]. This novel network architecture laid the foundation for the new branch of point-based 3D deep learning techniques. Similar methods, whose network mainly consists of MLPs, are termed MLP-based methods.

PointNet takes an entire point cloud as input, where each point is represented by its coordinate values (x, y, z), and outputs a per point label for each point in the input data. Optionally, other attributes, such as colours, normal, e.g., can be included as input. The architecture is made up of three main elements: (1) A stack of MLPs that learns per-point features, followed by a pooling layer that extracts global features using the symmetric function max pooling. This combination lets the network be invariant to the input order of points. (2) A feedback mechanism that combines global features with local point features, enabling per-point prediction of semantic point labels. (3) Two joint alignment networks that preserve invariance to rigid transformations of the point cloud by aligning the input point and point features.

Although groundbreaking, PointNet has one significant shortcoming – it is, by design, not able to capture local relations between neighbouring points in metric space. Because it does not consider the physical closeness of points, it struggles to capture finer patterns and understand more complex scenes. Realizing a solution to this problem, the creators of PointNet shortly after release their improved architecture under the name PointNet++ [68]. This method rapidly attracted attention as it outperformed the current state-of-the-art at point cloud recognition tasks by a large margin on several benchmark datasets.

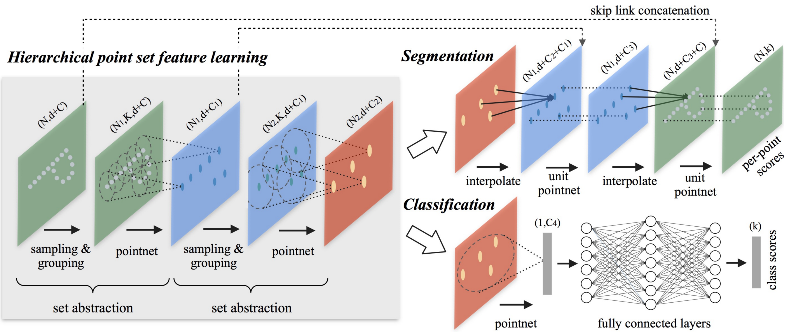


Figure 2.6: Illustration of the PointNet++ architecture and methods using 2D points as an example. Segmentation and classification is exemplified using SSG. Image origin: [68].

PointNet++ is an hierarchical neural network that applies the original PointNet recursively on subsets of points grouped into progressively larger local regions. This way it can learn both local structure information as well as the global context. The abstraction of the local regions is performed using a number of set abstraction levels. Each level consists of three key components: (1) A sampling layer that uses iterative Farthest Point Sampling (FPS) to select a subset of points that acts as centroids for their respective local region. This is a sampling algorithm that always selects the data point that is furthest from form any previously selected points until k points are selected. (2) A grouping layer that defines the local regions by locating neighbouring points for each centroid using a ball query. (3) The PointNet layer utilizing a miniature version of PointNet to learn local patterns and then construct summarizing feature vectors for each region. Using a local coordinate system, with a basis in the centroid coordinates, it can preserve relative point-to-point relations within the local regions. Figure 2.6 illustrates the architecture of PointNet++.

Additionally, PointNet++ introduces two novel density adaptive layers that intelligently combines features from different scales based on local densities. This improves the network's ability to handle data with non-uniform sample densities, something which is very common in remotely sensed point clouds. The first layer is the Multi-Scale Grouping (MSG) layer. It makes use of random point drop out for input points during training to expose the network to training data with varying density. The second layer is the Multi-Resolution Grouping (MRG) layer. This layer is less computationally expensive than the MSG layer but performs slightly worse. It combines the feature vectors from different abstraction levels using density-dependent weights. They show, through testing, that the model performance greatly improves when MSG or MRG is used, compared to when the network is trained using only Single Scale point Grouping (SSG). To this date, PointNet++ is still considered state-of-the-art due to its low complexity paired with high performance. Following their release, numerous researchers have been inspired by PointNet and PointNet++ and several improvements have been suggested.

One such improvement is a module designed by Jiang et al. [69], to be integrated with various PointNet-based architectures to optimize their performance for semantic segmentation. It uses an Orientation-Encoding (OE) unit to convolve the features of neighbouring points in eight different directions, improving the networks ability to learn shapes invariant to their orientation. Increased ability to handle multi-scale features was also realized by stacking multiple OE units and implementing shortcuts between them.

[70] uses PointNet++ as a feature extractor in their proposed Similarity Group Proposal Network (SGPN). They then introduce a similarity matrix to represent the similarity between any two point features. Exploiting the fact that points belonging to the same object instance should have a similar feature, they use the rows in the similarity matrix to combine similar point into group proposals, followed by a PointNet layer that predicts a confidence map for the similarity matrix. Finally, they use a semantic segmentation scheme to classify each group before filtering out proposals with a confidence score below a certain threshold. This, as well as using Non-Maximum suppression to create non-overlapping object instances, makes SGPN the very first point cloud instance segmentation framework.

Chen et al. [71] argue that PointNet++ and its early derivatives, are unable to learn geometric patterns accurately and robustly, as they do not consider the spatial distribution of the point cloud when creating the subregions for the feature extraction. They utilise the FPS and the ball query algorithms proposed by PointNet++, as well as their upsampling architecture, but present a new Location Spatial Aware (LSA) layer together with deeper MLP, for the feature learning a set of Spatial Distribution Weights (SDWs) in a hierarchical fashion based on the spatial relationships in local regions. They further propose LSAnet that implements the LSA layer and show that it is highly effective regarding extracting fine-grained patterns.

Convolution-based Methods

Convolutional Neural Networks (CNNs) have, for a long time, been state-of-the-art at 2D image recognition tasks due to their high accuracy and efficiency [72]. These are a category of neural networks that uses convolution layers as core components in their architecture. However, traditional convolution cannot be directly to point clouds because of their irregular and unordered nature, making the designing of new convolutional operators a popular, but challenging, research topic.

For instance, Li et al. propose PointCNN [73], a network that learns an X-transform from the grouped input points using a regular grid. This way, they achieve a weighting of the associated input features as well as the permutation of points into a local convolution order. Conventional convolutional operators, such as element-wise product and sum, can thus be applied to the transformed features. PointCNN attained state-of-the-art mean Intersection over Union (mIoU) for part segmentation on the ShapeNet Parts dataset but takes long to converge at training time [74].

Alternatively, the lightweight architecture Shellnet [74], implements a novel convolutional operator that makes it able to achieve even better results in a highly effective manner. It uses concentric sphere shells to define a point neighbourhood for each point, calculate representative features, and resolving the ambiguity of the point order, permitting the appliance of traditional convolution on the aggregated features.

Thomas et al. proposed Kernel Point Convolutions (KPconv) [75]. They preserve the point order by using kernel points to store the convolution weights in Euclidean space and correlates these to close input points using a linear function. As KPconv has the capacity to handle any number of kernel points and their location in continuous space can be learned by the network to adapt local geometry, it becomes more flexible than CNNs such as PointCNN that uses fixed-grid convolutions, making it better at handling arbitrary sized point clouds.

Opposed to the explicit correlation function implemented by KPconv, [76] learn the kernel-to-input relation using a MLP. Furthermore, the method separates the spatial and feature components of the kernel. The location of the spatial kernel elements are randomly sampled from the unit sphere.

The current benchmark on ShapeNet Parts [77] was published in December 2020 and is held by FG-Net [78]. It suggests three novel contributions for effectively handle large-scale point cloud processing: (1) A geometry-sensitive modeling module using per-point correlated feature extraction. (2) A residual learning architecture based on feature pyramids, facilitating memory-efficient, multi-scale feature learning. (3) Enhanced performance and efficiency by presenting a swift outlier and noise removal, together with a down-sampling scheme of extensive point clouds.

Graph-based Methods

Another popular design choice for neural networks is the graph-based approaches. Graphs are especially useful when it comes to capturing the structural relations between points. This allows for the local and global context to be considered to a larger degree when predicting per-point labels for segmentation tasks.

One of the firsts to propose a graph-based approach for point cloud deep learning was Landrieu and Simonovsky [79]. Using Super Point Graphs (SPGS), they were able to preserve the relations between points organized into similar geometric elements represented as superpoints. The contextual relationship amongst these elements is encoded in the edge features linking the superpoints in the SPG. Assuming that points in a superpoint are homogenous, a descriptor is calculated for each superpoint using PointNet. Finally, a graph convolutional network is used to segment the superpoints into meaningful partitions.

To overcome the problem of early methods only considering points one by one in an independent fashion when calculating features, Wang et al. proposed specGCN [80]. This is a neural network that utilises spectral graph convolution on local, nearest neighbour graphs in addition to a novel graph pooling scheme, forcing joint feature learning and the deduction of local structural information.

Another well known graph-based architecture is DGCNN [33]. Introducing a novel operator, EdgeConv, they build upon the original PointNet architecture to improve the capture of local geometric features. It constructs a local neighbourhood graph in feature space and applies EdgeConv to the edges connecting neighbouring points. Opposed to other graph CNNs, the graph is dynamically updated after each layer in the architecture. This way, they do not only exploit closeness in Euclidean space but also the similarity of features, to achieve excellent results in various point cloud recognition tasks. Zhang et al. [81] further improves this method by adding shortcuts between layers, enabling better hierarchical feature learning, and removing PointNet's transformation network. This boosts performance while reducing model complexity.

In Table 2.1 a summary of the performance of the mentioned methods is given in the form of their reported results on ShapeNet Parts, number of model parameters, as well as inference times, for comparison. The inference time refers to the time a network needs to make a prediction for a single input, i.e. one forward pass.

Table 2.1: 3D part segmentation comparisons of mIoU on points on ShapeNet Parts. The table also include number of model parameters and inference for methods where these were obtained. Shellnet is not included as it reports a different metric.

Methods	mIoU	Parameters [M]	Inference time [s]
PointNet [62]	83.7	3.48	0.015
PointNet ++ [68]	85.1	1.48	0.027
SGPN [70]	85.8	-	-
PointCNN [73]	86.1	0.6	0.012
SpecGCN [80]	85.4	2.05	11.252
DGCNN [33]	85.1	1.84	0.064
LDGCNN [81]	85.1	1.08	-
LSAnet [71]	85.6	2.3	0.06
KPConv [75]	86.4	15	12.2
ConvPoint [76]	85.8	-	-
FG-Net [78]	86.6	-	0.055

Evaluation Metrics

The mentioned mean Intersection over Union (mIoU) is one of the most common evaluation metrics used to assess a segmentation result outputted by a neural network. Another frequently used metric to evaluate network performance is Pointwise Accuracy (PA) [66]. For the evaluation of PointNet++ trained on the proposed dataset, both metrics are reported. These values are computed between the ground truth and the predicted output. The following quantities are defined to make it possible to describe these evaluation measures.

True Positive (TP): Number of points correctly predicted as belonging to a part type.
True Negative (TN): Number of points correctly predicted as not belonging to a part type.
False Positive (FP): Number of points incorrectly predicted as belonging to a part type.
False Negative (FN): Number of points incorrectly predicted as not belonging to a part type.
Number of parameters (N): The total number of part types possible.

True values indicate that the predictions are correct, while false values correspond to wrong predictions.

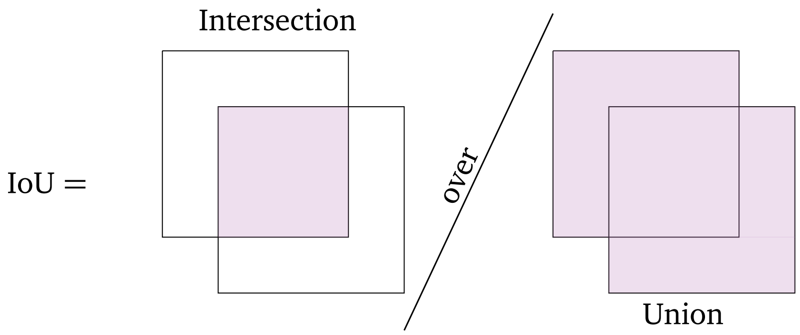


Figure 2.7: The illustration shows how the IoU metric is found by taking the intersection of the areas over the union.

PA is calculated by dividing the number of correctly predicted points by the total number of points present. This will result in the metric defining the percentage of correctly classified points for the complete dataset. Formally, it can be defined as:

$$PA = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

While being a metric easy to understand, it might give a skewed impression of the performance. If the method is especially excellent at segmenting physically larger planes, as might be the case in roof plane segmentation, the PA metric will favour this performance and increase above what is expected. Another weakness is that it does not take class imbalance present into consideration.

The Intersection over Union (IoU) is found by dividing the overlap between the predicted area and the ground truth area, by the total of both their areas. Figure 2.7 shows an explanation of how a single-class IoU is found. The mIoU is obtained in a class-wise fashion, where predictions for a given class is evaluated before the mean score is found over all classes. In a more formal matter, the metric is computed using the following formula:

$$mIoU = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i + FN_i} \tag{2.2}$$

This score provides a balanced indication of the performance, making it a preferred metric in the evaluation of the segmentation of point clouds [10].

2.3 Existing Benchmark Datasets

A challenge of supervised deep learning approaches is the emerging need for more training data in cases where the input is three-dimensional rather than two-dimensional [49]. Consequently, large 3D benchmark datasets with correctly annotated data is essential for the training and testing of neural networks. Outdoor and indoor environments require separate benchmark datasets, as there is a significant difference between objects appearing indoors and outdoors. Remote sensing data is region dependent, meaning that data obtained in a specific area not necessary are suitable for other regions, as both nature and constructions present varies with the location. Task-specific data is therefore required for tasks where deep learning approaches are to be utilised for geospatial data.

Additionally, there is a major difference between real-life and synthetic data. Synthetic data constructed from 3D CAD models are both cheap to obtain and easily accessible. Such data is of high quality containing few errors. Real-world data, on the other hand, demands the use of costly equipment and manual labour, increasing the cost and reducing the availability. A lot less datasets containing real-world data is accessible compared to synthetic datasets. Nevertheless, there is a large demand for real-world data, as synthetic data cannot fully represent real-life scenes. Tests performed by Uy et al. [82] shows that networks trained using synthetic datasets perform poorly when tested on real-life data.

With regards to the task of reconstruction of building data, several real-world benchmark datasets have been established in the last couple of years. These include datasets obtained both through image derived methods and with the use of LiDAR technology, and some of these important benchmark datasets are detailed in this section.

2.3.1 Image Derived Datasets

One commonly applied image-based method is Structure-From-Motion (SFM). The method automatically extracts features from a set of numerous overlapping images, before an iterative bundle adjustment procedure is performed based on this dataset. By using this procedure, SFM is capable of handling multi-view images instantaneously [83]. Multiple view Stereovision/Multiview Stereo (MVS) [84] is used for matching and reconstruction of 3D point data from pictures, and is important for the automatic obtainment of data [85]. Large amounts of point cloud data can be generated by the use of MVS algorithms, making it a suitable method for obtaining dense 3D point clouds [14].

Li et al. [86] proposed Campus3D as a point cloud benchmark for hierarchical understanding of outdoor scenes. Based on UAV images obtained with drones over the National University of Singapore, SFM with MVS was used to construct the 3D point cloud. The point cloud contains 937.1 million points and is annotated with point-wise labels, using a strategy where the projected 2D images are labelled, before these labels are assigned to the corresponding 3D points. Campus3D have hierarchical labels for better scene understanding, and instance labelling is used to distinguish between different instances, including different planes of a roof structure. As the site of obtainment is a campus, the buildings are quite complex and very different from typical Norwegian roof structures.

Like Campus3D [87], the SensatUrban dataset is based on UAV images constructed into a 3D point cloud with the use of SFM. The dataset was proposed by Hu et al. as an urban-scale point cloud and separates from earlier work because of its large number of points. 2847 million points are annotated into 13 classes describing typical outdoor areas. However, no instance segmentation is performed, and roofs are not separated from the rest of the building. This makes the dataset unsuitable for the task of roof plane segmentation.

The use of SFM for the obtainment of dense 3D point clouds have increased in the last couple of years, but the quality of such points clouds is not as good as those obtained with LiDAR systems [88].

2.3.2 LiDAR-based Datasets

For the obtainment of building data, both terrestrial, mobile, and aerial LiDAR scanning is commonly applied. TLS differs from ALS in that it operates from a ground-based stationary sensor rather than an airborne platform. TLS is known for collecting data of high accuracy at a fast speed and is commonly applied in medium to close-range environments. The result is a point cloud of high density [19]. Opposed to TLS, MLS is performed from vehicle-based mobile platforms, making it possible to gather data along a path [89].

Terrestrial and Mobile Datasets

Datasets obtained through TLS includes semantic3D.net, a large-scale 3D point cloud presented by Hackel et al. containing four billion manually labelled points [49]. Paris-Lille-3D [90] and SemanticKITTI [91] are two well-known large-scale datasets obtained through MLS, both of them containing a large amount of annotated points. Even though these benchmark datasets are of high quality and contains a vast number of points, they contain little to no points describing the roof structures of buildings, as they are obtained from ground-based stations.

Airborne Datasets

ALS are the most common obtainment method for datasets meant for roof segmentation tasks, as the data is acquired from an aerial point of view. One of the first high-quality point cloud datasets meant for this task was The Vaihingen 3D Benchmark (V3D) [92]. This is one of the most well-known benchmark datasets in the field of remote sensing, and it contains around 1.2 million points obtained by ALS. The points are annotated and categorized into nine categories, one of them being roofs. The dataset are however out-of-date with a median point density of 6.7 points/m^2 and a quite low number of points, making it unsuitable for deep learning techniques [17].

The Hessigheim 3D Benchmark (H3D) [15] recently replaced V3D as the International Society for Photogrammetry and Remote Sensing (ISPRS) Benchmark for semantic segmentation of 3D point clouds [93]. By using UAV for data obtainment, Kölle et al. established a dataset manually labelled into eleven classes. Figure 2.8 shows an example of the class labels present in the dataset. The dataset has a point density of 800 points/m^2, meaning that it is a very dense point cloud. Too high densities leads to limitations of the use of the dataset in deep learning applications. A datasets with such a high point density can merely be used for small scale projects due to the high cost of data acquisition and processing. Such high densities is however valuable for small scale projects such as cultural heritage or construction planning. The area of obtainment is the village of Hessigheim in Germany, and the buildings found in this area are to some extent similar to buildings found in Norwegian cities such as Trondheim. H3D includes a roof-class, but no further labelling into different planes is performed, making the dataset unfit for segmentation of roof planes.

Another aerial LiDAR dataset annotated for use in deep learning is the DublinCity dataset established by Zolanvari et al. in 2019 [16]. Obtained in the city of Dublin, the dataset contains 260 million manually labelled points that have been categorized into 13 classes at three hierarchical levels, including roofs. No further segmentation is performed for these categories, meaning that the dataset contains no plane information for roofs. Compared to the H3D dataset, the dataset has a lower density with around 348 points/m^2, but is still a dense point cloud. As for H3D, is the dataset too dense to be suitable for large scale projects. As the area of obtainment is the city of Dublin, the roof structures present will greatly dif-

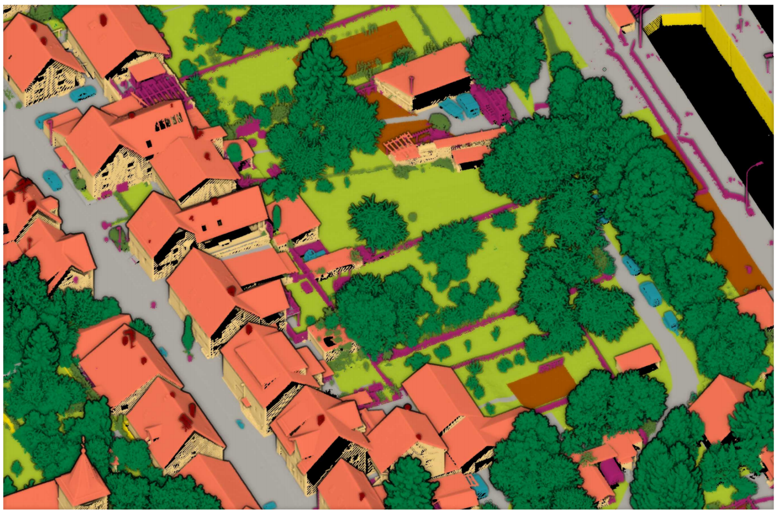


Figure 2.8: H3D: Class labels present in the H3D dataset. No segmentation of roofs into planes are present. Image origin: [15].

fer from those found in Norwegian cities, due to the difference in building construction.

LASDU, proposed by Ye et al., is another large-scale dataset for semantic labelling acquired through ALS [94]. Around 3 million points are manually labelled, and the density of the point cloud is approximately 4 points/m^2. Compared to other similar datasets, LASDU only contains five categories, and the roofs are not separated from the building structures, and consequently, no segmentation of roofs is possible with this dataset.

The ALS obtained dataset DALES is one of the latest benchmark datasets with regards to large-scale point cloud data acquired through LiDAR [17]. Compared to the LASDU dataset, the dataset is denser, containing approximately 500 million manually labelled points giving a density of 50 points/m^2. Eight categories are defined, including roofs and facades of buildings. The large size of the dataset makes it suitable for deep learning purposes such as the segmentation of roofs. However, the roof structures are not further divided into planes.

Established for the purpose of building reconstruction and with a focus on including semantic roof type information, the RoofN3D dataset [95] was obtained in the New York area for use in deep learning approaches. Unlike other benchmark datasets, information about roof type is included in the point cloud, and the roofs are further segmented into planes. Consequently, the dataset can be considered state-of-the-art when it comes to datasets for roof plane segmentation of 3D point clouds. However, the point cloud has a density of only approximately 4.7 points/m^2, and the data was obtained in the New York area in 2013 and 2014, making it out-of-date. As the area of obtainment differs a lot from the city of Trondheim and typical Norwegian buildings, the roof types present in the dataset is less relevant for use in Norway.

As mentioned, remote sensing data is region dependent. Building structures vary greatly in different areas, and as a result, different datasets need to be generated based on the location of the area of interest. As most of the existing datasets are international, this creates demand for a separate dataset containing typically Norwegian building structures. Another drawback of current benchmark datasets is the lack of an instance labelling of roof planes. In addition, no dataset with a density corresponding to that obtained in standard ALS operations exists. This leads to a need for data suitable as input for data-hungry deep learning networks meant for roof plane segmentation. With this work, we aim to close this gap, by creating a real-world 3D point cloud dataset suitable for deep learning methods applicable in Norwegian survey and mapping projects.

Chapter 3

Roof Segmentation Dataset

Segmentation of 3D roof structures into planes is a challenge possible to solve with deep neural networks. This demands a need for datasets containing large amounts of roof-plane structures. In this chapter, the aim is to present how we establish a dataset suitable for deep learning-based techniques. Guidelines for the processing of LiDAR data is introduced, including the labelling strategy for the annotation of ground truth data. Based on these guidelines, the TRD3DRoofs dataset is established. This dataset is obtained in the Trondheim area and manually annotated into typical Norwegian roof structures and planes. The original TRD3DRoofs dataset consists of roofs representing real-world buildings. An additional augmented version based on this original dataset is also presented, where augmented data is added to create a more well-balanced dataset of a larger size well-suited for deep learning. All implemented code for data augmentation and preparation was written using Python as the programming language.

3.1 Proposed Guidelines

In this section, guidelines for the establishment of a training dataset from an ALS point cloud is proposed. The result will be a dataset suitable as input for machine learning algorithms. These guidelines are established as an answer to the first research question.

The input data is a raw LiDAR point cloud acquired through ALS. As the data lacks semantic information, a taxonomy for the annotation of points is established. This taxonomy is based on what we consider Norwegian standards for roof structures and are designed to be suitable for data meant as input in deep learning algorithms. A selection of objects from the original dataset is performed, as the raw point cloud covers a vast area. These selected roof objects will form the foundation for the dataset, indicating a need for a selection strategy. For the concrete labelling task, manual segmentation is performed using the software CloudCompare [96]. This process takes the raw LiDAR point cloud data as input and results in manually labelled roof structures annotated with ground truth semantic information. Using the taxonomy for semantic labelling, each point will contain information about the plane segment and concrete roof it belongs to, together with the type of roof structure.

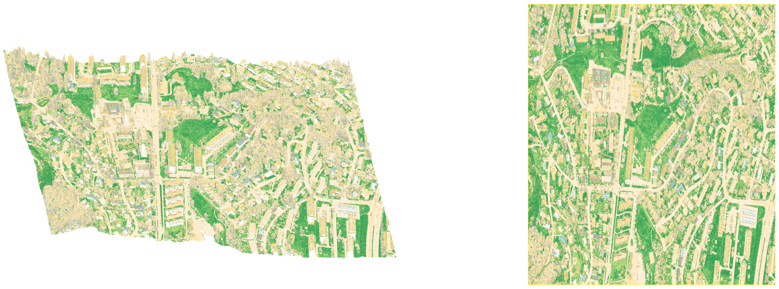
The next steps are crucial for the final design of a dataset suitable for deep learning. As the number of manually segmented roof structures are too low for data-hungry networks, augmentation of the data is necessary to increase the size of the dataset. Additional roof structures are created by shifting the labels of the planes within one category. This is performed for all roof structures, except for those containing only rectangular plane types, resulting in a more balanced dataset. The rectangular planes are omitted from the augmentation, as most of the planes in the original dataset are rectangular. The original dataset is kept as a separate dataset, resulting in the establishment of two separate datasets, respectively the Original TRD3DRoofs dataset and the Augmented TRD3DRoofs dataset.

Preparing the data by performing normalization of the coordinates and a shifting of origin is done for both datasets. The result is a 3D dataset representing real-world buildings in Trondheim Municipality, and a well-balanced 3D point cloud dataset annotated with semantic information, on a format exploitable as input for a deep neural network.

3.2 Data Acquisition

The original LiDAR data is provided by Trondheim Municipality and was obtained on 04. April 2018. For the georegistration, the coordinates are represented by the the European Terrestrial Reference System 1989 (ETRS89), while the projection used is the Universal Transverse Mercator (UTM) zone 32N. For the vertical datum, Normal Null 2000 (NN2000) is used.

Obtained over a vast amount of Trondheim Municipality, the LiDAR data consists of both rural and urban areas. The topography of this area varies between both relatively flat regions and more hilly terrain. For the annotation, the study area contained several areas mainly in the city of Trondheim. These areas were primary residential areas surrounding the city centre. Figure 3.1 shows a visualization of the LiDAR point cloud in a typical residential area in Trondheim.The original point cloud was divided into separate files based on the area of obtainment. To locate the desired roof structures and the location of these, the bounding box of each file



(a) Tilted view of the area.

(b) Area seen from above.

Figure 3.1: Visualizations of LiDAR Point Cloud.

were visualized using ArcGIS® Pro [97].

As the roof structures are obtained from existing buildings, it was of interest to make sure that each building present in the point cloud were identified and connected to the real-world building. This was desirable as the raw labelled dataset is to be applied in another thesis where the task is to connect each point in a building to an existing building footprint. As a result, it is necessary to identify all buildings. In Norway, each building is given a building ID established by the Norwegian Mapping Authority, Kartverket. This building data may be found in the common map database, "FKB", maintained by the Geovekst-parties in each municipality. The data is made available at the online service `https://geonorge.no`, and for this work the dataset "FKB Bygning" for Trondheim has been downloaded from this service. The projections used is the same as for the LiDAR data, the vertical datum is ETRS89 UTM zone 32 and NN2000 is the height datum. Licenced under the "Norge-digitalt" license [1], this data is restricted to non-commercial use. Figure 3.2 gives an overview of the area of obtainment. The bounding boxes corresponding to the LiDAR data is visualized as slightly grey, transparent squares.

3.3 Data Labelling Convention

A taxonomy for labelling each roof plane was established and is presented in this section. This was done to make sure that all needed information connected to the plane is preserved.

[1] For more information, see `https://www.geonorge.no/Geodataarbeid/Norge-digitalt/Avtaler-og-maler/Norge-digitalt-lisens/`

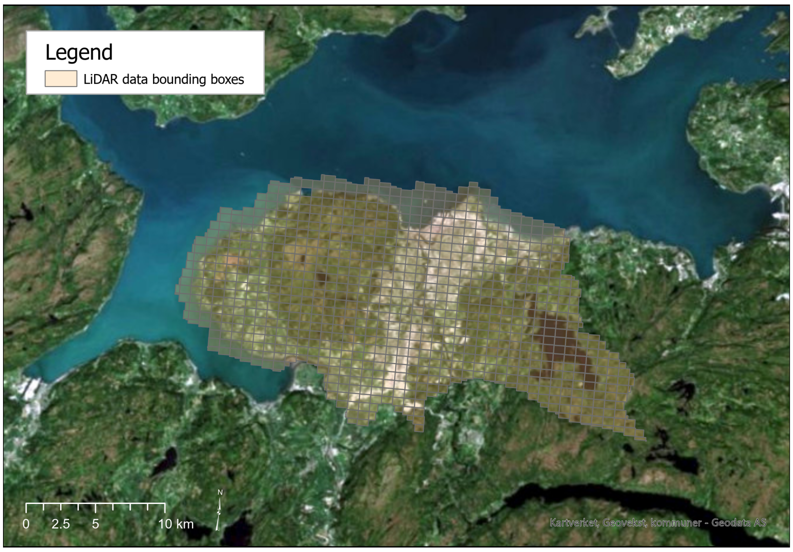

Legend
LiDAR data bounding boxes
0
2.5
5
10 km
N
Kartverket, Geovekst, Kommuner - Geodata AS

Figure 3.2: Overview of the area of 3D capture surrounding Trondheim.

As the dataset is to be used in a separate thesis where the roof structures manually labelled in the 3D point cloud are to be connected to the 2D polygon footprint representing the same building, some considerations regarding the manual labelling were necessary. Because of this interest in preserving the building information, every single segmented roof is labelled with the corresponding building ID. This gives the possibility of connecting the correct building in a point cloud to the corresponding footprint.

For the same task, buildings that are physically connected but have different IDs needs to be labelled as two individual buildings. In the point cloud, these separate buildings will in some cases be impossible to split apart. To solve this problem, each building is given a number from 1 to n where n is the largest number of connected buildings present in the dataset. For example, if the roof only belongs to one building, the plane is labelled 1. If three connected buildings share the same roof, the plane is labelled 3.

For the semantic annotation of the data, it was decided that each roof is to be assigned to a roof category based on the structure of the roof. The selection criteria for roofs segmented were based on the eight roof types proposed by Kada [98]. These roof types are based on primitives, and their simple nature makes them suitable as the fundament for other more complicated roof types. Additionally, they are established from German residential buildings, which are closer to Norwegian building structures than those found in residential areas in other parts of the world. The roof types proposed by Kada were *flat*, *shed*, *hipped*, *asymmetric hipped*, *gabled*, *corner element*, *T-element*, and *cross element*.

After an examination of the data set and the area of study, it was discovered that some changes to the definition of roof types would be suitable for Norwegian areas. One discovery was the fact that a lot of the roof structures were a combination of several types, and therefore an additional category named combination was proposed. As no buildings with an asymmetric hipped roof structure were discovered, this category was discarded. Only a few shed-structured roofs were observed, and it was decided that the flat and shed roof structures were both to be a part of the first category. As a result, seven categories for typical types of roofs in Norway were proposed; *flat*, *hipped*, *gabled*, *corner element*, *T-element*, *cross element* and *combination*. For analysis purposes, it was an interesting aspect to see how well the neural network performs on the different roof types. To be able to gather this information, each plane is labelled with a number from 1 to 7 corresponding to a type, describing the roof structure:

1. Flat
2. Hipped
3. Gabled
4. Corner Element
5. T-Element
6. Cross Element
7. Combination

Figure 3.3 contains a visualization of examples of the different roof types established, but variations to these figures occur in the actual dataset. Roof type 7 *Combination* consists of elements from all other categories and is not visualized in the figure, due to large variations in this category.

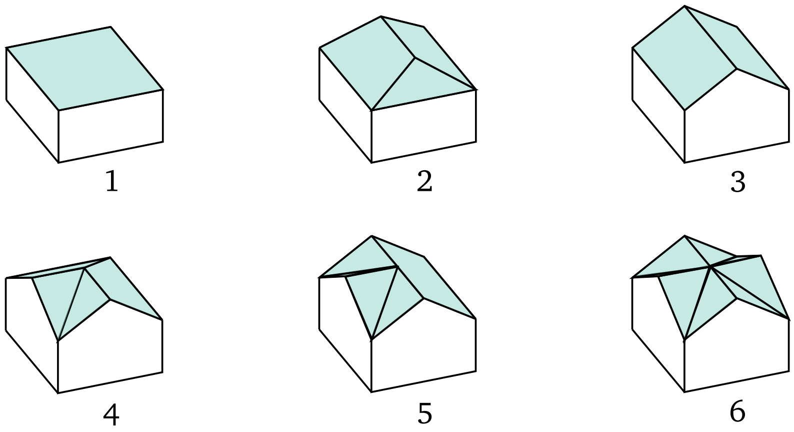


Figure 3.3: Visualization of six of the defined roof types. The numbers represents a roof type, and this corresponds to the list from section 3.3.

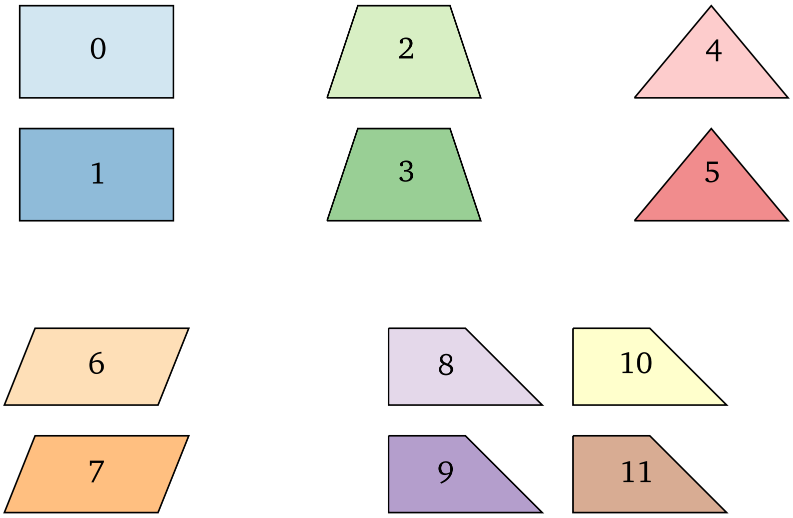


Figure 3.4: Visualization of the defined plane shapes labelled with the corresponding digit. The description of each geometric type and the labelling scheme may be found in Table 3.1.

A part labelling for each roof is performed, where the roofs are separated into different planes and each point annotated into the correct label. The taxonomy for this labelling is of importance, as it will affect the semantic information available for all points. For the original dataset, each roof-plane was labelled with a number from 1 … n where n is the total number of planes belonging to the same roof. No rules were defined for labelling within a plane, and each plane was arbitrarily numbered. However, testing of this data showed that plane annotation was of great significance for the results of the deep neural network. For the augmented dataset, a need for another labelling strategy consequently emerged. After further analysis, a more suitable taxonomy was established and is presented here. This forms the basis for the labelling of each plane in the augmented dataset.

The goal of the segmentation process is to make the deep neural network learn to classify each point into the correct plane. The planes that belong to the same category need to have some features other than proximity that makes them similar. Considering this in the manual labelling of the training data could enhance the learning process of the neural network, demanding less training data.

Five different categories for plane types were defined, and each of these categories given between two and four labels, being digits from 0 to 11. The categories are proposed based on the geometry of planes present in Norwegian roof structures and are as follows: *rectangular*, *isosceles trapezoid*, *triangular*, *parallelogram* and *ladder-shaped*. Figure 3.4 shows a visualization of the different geometric shapes, labelled with the correct digit, and coloured correspondingly based on the information in Table 3.1. Using geometric shape as a feature for segmentation purposes is a common approach and can be related to the earliest methods for segmentation, such as primitive fitting, where simple shapes in the data were detected [99] [28] [13]. Deep neural networks are able to learn features such as the local geometry of the neighbourhood surrounding a point, and this includes the shape of the plane the points compose.

Another important aspect is the fact that the neural network can learn the relationship between planes by analysing a set of planes such as a complete roof structure. This can be seen as a case of shape co-segmentation, a field of segmentation where a set of shapes is simultaneously segmented [100]. Instead of processing the shapes separately, a set of shapes are inputted and segmentations carrying consistent semantics across the shapes are generated. This can be exploited in the case of roof plane segmentation, as planes in a roof structure belonging to a certain roof type tend to have a similar fundamental relationship to other planes contained in the same roof structure. By labelling the planes based on geometric shape and inputting point clouds containing complete roof structures, the network can learn both the typical shapes of the planes and the connection between different plane types, possibly yielding better result in the task of roof plane segmentation.

Roofs of a certain type are most likely to contain the same kind of planes and the same number of each kind. By using this, fewer categories of planes and labels are necessary. For visualization purposes are each label is given a specific colour. The colour map used for this task is created using ColorBrewer[2]. The qualitative 12-class Paired is used, as this palette is well-suited for visualizing categorical data due to its variation in hue. Table 3.1 contains information about the colour used for each plane label, together with which plane geometry it describes, and which roof types contain such plane types.

[2] ColorBrewer is a tool for choosing choropleth map color schemes, based on the research of Dr. Cynthia Brewer. Visit `https://colorbrewer2.org/` for more information.

Table 3.1: Overview of plane labels and geometry.

Plane label	Plane geometry	Roof types
🟦 0	Rectangular	Flat, Gabled, T-Element, Cross-Element, Combination
🟦 1		
🟩 2	Isosceles trapezoid	Hipped, Corner-Element, Combination
🟩 3		
🟥 4	Triangular	Hipped, Corner-Element, Combination
🟥 5		
🟧 6	Parallelogram	Corner-Element, Combination
🟧 7		
🟪 8	Ladder shaped	T-Element, Cross-Element, Combination
🟪 9		
🟨 10		
🟫 11		

3.4 The Manual Segmentation Work

In this section, the methodology for the manual generation of ground truth data is presented. The main purpose of the resulting dataset is to provide labelled data for the training and testing of deep learning techniques applied in semantic segmentation of roof planes from point clouds. This requires the data to be annotated with the correct class label to generate the ground truth, meaning the ideal expected result from the learning process. Figure 3.5 visualizes an example of the desirable result from the segmentation process for each roof type, where each plane is coloured with the corresponding class colour. The manual segmentation was conducted over a period from December 2020 to March 2021.

Approximately 900 buildings spread across residential parts of the Trondheim were chosen and form the data foundation. Both smaller and larger buildings were selected, as the size of a building does not affect the result of the final segmentation. In the selection process, a focus on choosing different variations of the roof structures was prominent. If the network is only exposed to similar-looking roof-planes, it will not be able to learn planes outside of these structures. Roofs representing the same types, but with variations with regards to rotation, size of planes and elevation were desirable. As a balanced dataset is preferable in machine learning related tasks, a focus on extracting roofs evenly distributed among the types were incorporated.

For the generation of the ground truth, a manual process was established and carried out by research assistants at Norwegian University of Sci-

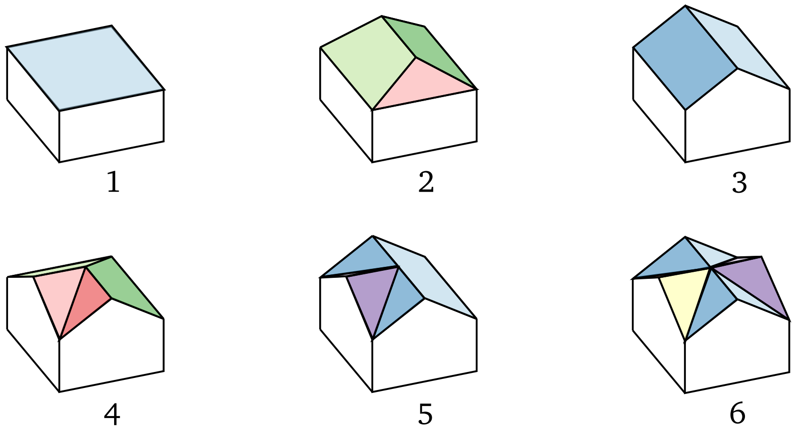


Figure 3.5: Visualization of desired result from manual segmentation and labelling. Each plane are colored corresponding to a correct plane label.

ence and Technology (NTNU), including the authors of this thesis. CloudCompare, a 3D point cloud and mesh processing software [96] were used for the manual labelling. First, each individual roof was manually segmented to distinguish it from the surrounding environment. Following, the roof of interest was segmented into subsets representing its respective planes. The manual segmentation into subsets was done by identifying natural boundaries in the structures, such as the roof ridge, or other edges. Based on the labelling taxonomy, each point contained in a plane was manually labelled with semantic information and finally each plane was stored individually as a point cloud, with the possibility of merging planes belonging to the same building into one singular point cloud. Figure 3.6 illustrates this workflow for manual segmentation a single roof.

Quality control of the manual work was achieved in a two-step fashion. First, all research assistants checked the labels of all roofs in unison. Second, the authors separately verified the annotations as the last instance. As manual annotations are prone to human error, it is not possible to avoid all label noise, and the authors are aware that this is also the case for our dataset, despite the cross-check performed.

Some challenges were encountered when identifying roof types with a corner or cross-element, and this is further discussed in section 3.7. Resultingly, a lot fewer roof structures of these types were found and manually labelled, leading to a skew in the dataset with regards to the type of roof. As the plane types are connected to the type of roof, a skew in plane-type will also appear in the finished dataset. For deep learning applications, a skewed dataset is unfortunate as the networks tend to have a bias towards the data there is a majority of [101]. Minor classes of data may in extreme

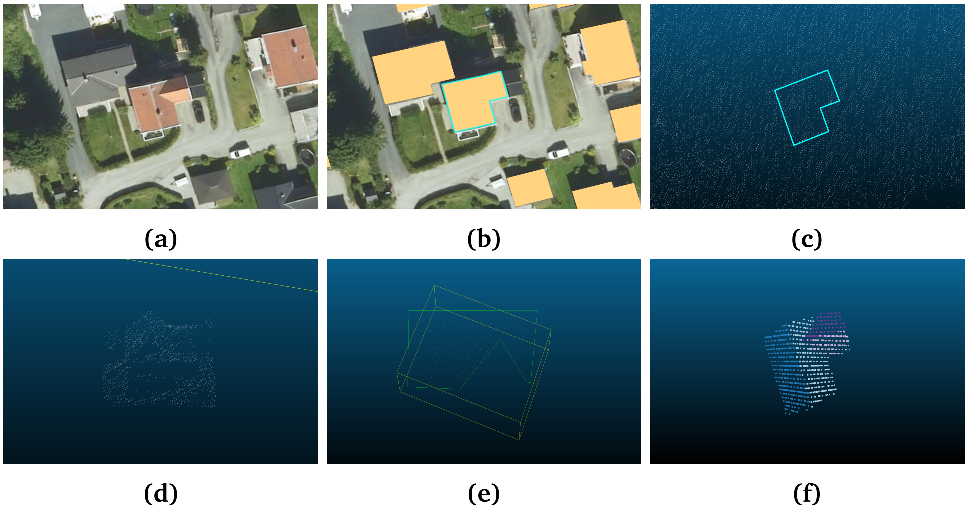


Figure 3.6: A step-by-step illustration of the manual segmentation process of a single roof structure. (a) Satellite image for a clearer view of the roof type. (b) Building footprint from the FKB-data, containing the building ID. (c) Building outlined in the original point cloud. (d) Initial rough segmentation of the building. (e) A single segmented roof plane. (f) Final segmentation results, where the colour of the points indicates the plane label.

cases be completely ignored. To avoid this bias, ensuring that the network learns in the most unbiased way possible, some altering of the data is necessary. At the same time, there is a desire to preserve a dataset where no altering is done, and only data of real-world buildings are present.

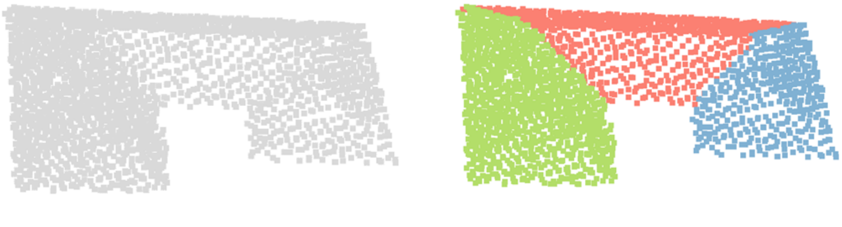
Consequently, two separate datasets are proposed in this thesis. The first is the original TRD3DRoofs dataset containing the segmented ground truth data of roof structures found in the Trondheim area. The second dataset is based on the original dataset, but contains additional augmented data, resulting in a larger, more well-balanced dataset.

3.5 Data Augmentation

In this section, the augmentation implemented and applied to the roof structures, answering the second research question, are detailed.

To balance the appearance of the different roof types, and consequently of plane types, in the dataset, two manual data augmentation steps were performed. Firstly, the combination roofs were divided into separate “sub-roof structures” based on similarity to the other roof types using CloudCompare. This way it was possible to create more training examples for specific plane types and rare roof structures. For example, looking at the roof shown in Figure 3.7, the original roof structure (left) is split into three new roof structures (right).

Secondly, the fact that each plane type is associated with two or more plane labels were exploited to generate more roofs by supplementing the dataset with all possible permutations of part label combinations. Figure 3.8 shows these combinations for a roof with a hipped structure. This was done for roof type 2, 4, 5, 6 and 7, as there already were preponderance of type 1 and 3.



(a) Before split

(b) After split

Figure 3.7: Data augmentation by splitting of roof structure of roof type 7 *Combination*.

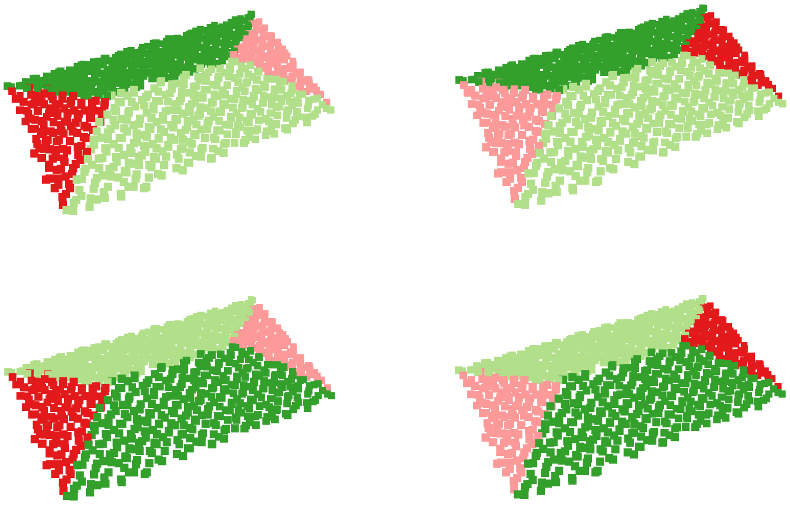


Figure 3.8: Visualizations of all different label combinations of a roof structure of type 2 *Hipped*.

3.6 Preparing the Dataset for Segmentation

The points of the original dataset provided by Trondheim Municipality are in a format unsuitable for direct use by most deep learning neural networks. Preparing the data is necessary to transform it into a dataset fit for such purposes. Consequently, a data preparation framework was established and applied to the datasets, and this is presented in this section.

First, recentering of the data was performed to decrease the size of the coordinates, and thereby reducing the computational efforts needed. A new origin was established and all remaining points subsequently moved relative to this new origin. This was done by calculating the mean for all three coordinates and subtracting these values from the coordinates of all points. The second step in the preparation of the framework is the normalization of the point cloud. When features in the data have various ranges, normalization is done to achieve a universal scale for all features. For each of the x-, y- and z-coordinates, the maximum distance from the origin was found. The coordinates of all points were divided by this maximum distance, achieving normalized values for all points in a similar range between 0 and 1.

For the last step in the process necessary for the preparation of the data, generation of normal vectors was performed for each point. The annotation of the points gives information about the two-dimensional geometry of the neighbourhood. However, one singular roof structure might contain separate planes with the same geometric shape. As there is an interest in segmenting these geometric shapes into their distinctive planes, normal vectors are included for all points as an additional feature. These normal vectors indicate the three-dimensionality of the neighbourhood of a point, as they are calculated based on several of the point's closest neighbours.

Using a k-Nearest Neighbours (k-NN) algorithm, a Principal Component Analysis (PCA) was performed to obtain the normal of the tangent plane best fitting the group of neighbouring points. The number of neighbours was based on the point density of the roofs contained in the augmented dataset, and the final value was found through trial and error. To balance the problem of including enough similar neighbours but excluding neighbours along ridges and edges belonging to other planes, the number of neighbours were decided to be $k = 6$. The density of the resulting dataset is further detailed in chapter 4. For the normal vectors to be of use in the learning process of separating distinct planes, they must be consistent when it comes to orientation. To make sure that all normal vectors point outward of the shape, a check was performed where the orientation of all normal vectors that do not exhibit this behaviour was turned. The calculated normal vector was added to each point as a new feature, giving three new values.

3.7 Experiences with Manual Labelling

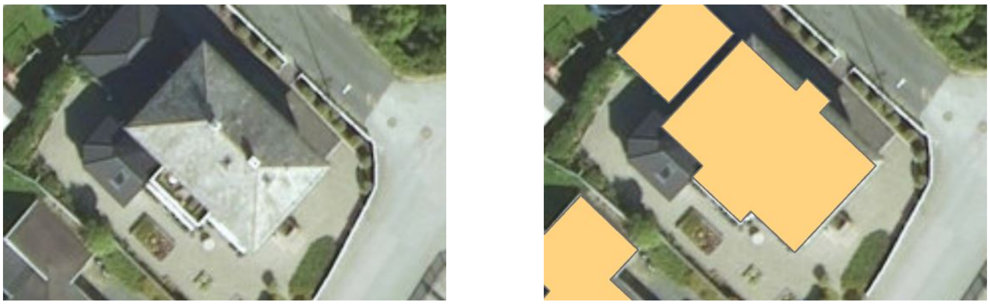
Some challenges were encountered regarding the original data, and during the manual labelling process, and these are presented and discussed in this section.

One consequence of handling a large amount of data is issues regarding storage. For the original LiDAR point cloud, the large number of points were divided into smaller separate files. These files were organized in an unstructured matter, making it challenging to locate the desired data. The point cloud was separated into distinctive files based on the area of obtainment, but these files were arbitrarily named. Consequently, the task of retrieving the desired roof structures and their location were problematic. To overcome this problem, the bounding box of each file were derived and later visualized together with a map of the Trondheim area using ArcGIS® Pro.

During the manual labelling process, several obstacles were met, some of them affecting the resulting dataset. Segmentation and annotations of training data performed manually by assistants tend to be both time-consuming and expensive [102], and this sets a limit for the number of roofs achievable in the proposed dataset. It was decided that 1000 manually segmented roofs would be enough. This would make the dataset contain a sufficient number of unique roof structures, enabling the use of data augmentation for further generation of training data.

A discovery during the segmentation process further increased the time needed for manual segmentation. It was noticed that a lot of the 2D polygons representing the footprints included parts of the buildings such as porches or balconies which are not part of the actual roof structure. An example of this is found in Figure 3.9. It was decided that such buildings are to be ignored, and not labelled, as they unnecessarily complicate the task of connecting the 2D polygons to the 3D point clouds. These types of polygons are found to be common in the Trondheim area, making the task of obtaining desirable roof structures harder.

Another finding was the fact that most buildings in Trondheim have a gabled or T-element roof structure, giving a large predominance of roof structures belonging to these two types. To even out this imbalance, buildings of the remaining types were targeted, leading to a new discovery. The original point cloud data were found to have a lot of missing points, especially of buildings with a cross or corner element. Points belonging to flat



(a) Satellite image of a building with a balcony.

(b) Example of building footprint including the balcony.

Figure 3.9: Example of polygon including balcony. FKB-data from ©Kartverket.

roof structures were also missing to some extent.

Different causes for this have been discussed to discover the origin of the fault in the dataset. As LiDAR works by observing the reflected light from an object, the surface must be reflective for the object to be detected. Specular surfaces, such as windows, will absorb most of the light instead of reflecting it, leading to the target being practically invisible to LiDAR. How reflective a roof is, is dependent on the material used in the construction. The Solar Reflectance Index (SRI) is a measure describing the solar reflectance and emissivity of materials [103]. Materials with a low SRI will reflect less light than those with a high SRI. Roofs covered with materials such as a black bituminous membrane will therefore not reflect enough light to be detected by the LiDAR sensor.

It is therefore reasonable to assume that the roofs missing from the LiDAR point cloud obtained in the Trondheim area are covered by materials that absorb light instead of reflecting it. Another sensible assumption is that flat roofs and roofs with a cross- or corner element are more likely to be covered by such materials, and therefore are missing to a larger extent than other roof types. To overcome this challenge, augmentation of the data was performed and used to establish the extended version of the dataset.

3.8 The Overview of the Resulted Datasets

In this section, the resulting datasets are presented together with statistics detailing them, and results of the labelled ground truth data. First, the original dataset, named TRD3DRoofs is presented, before the result of the augmented version is detailed further.

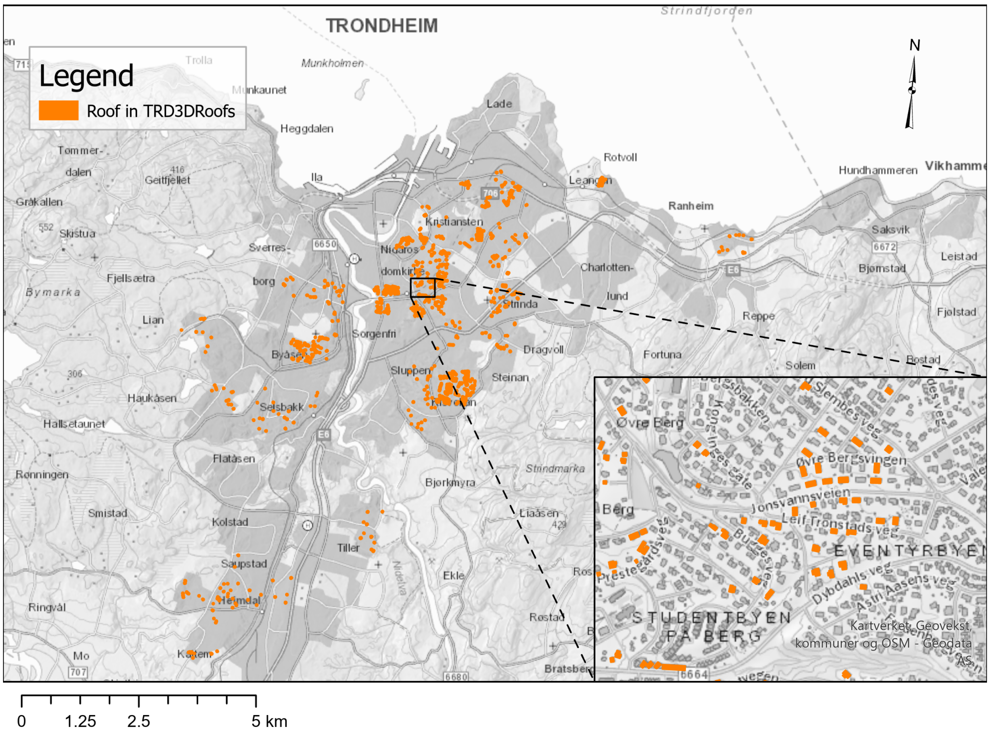
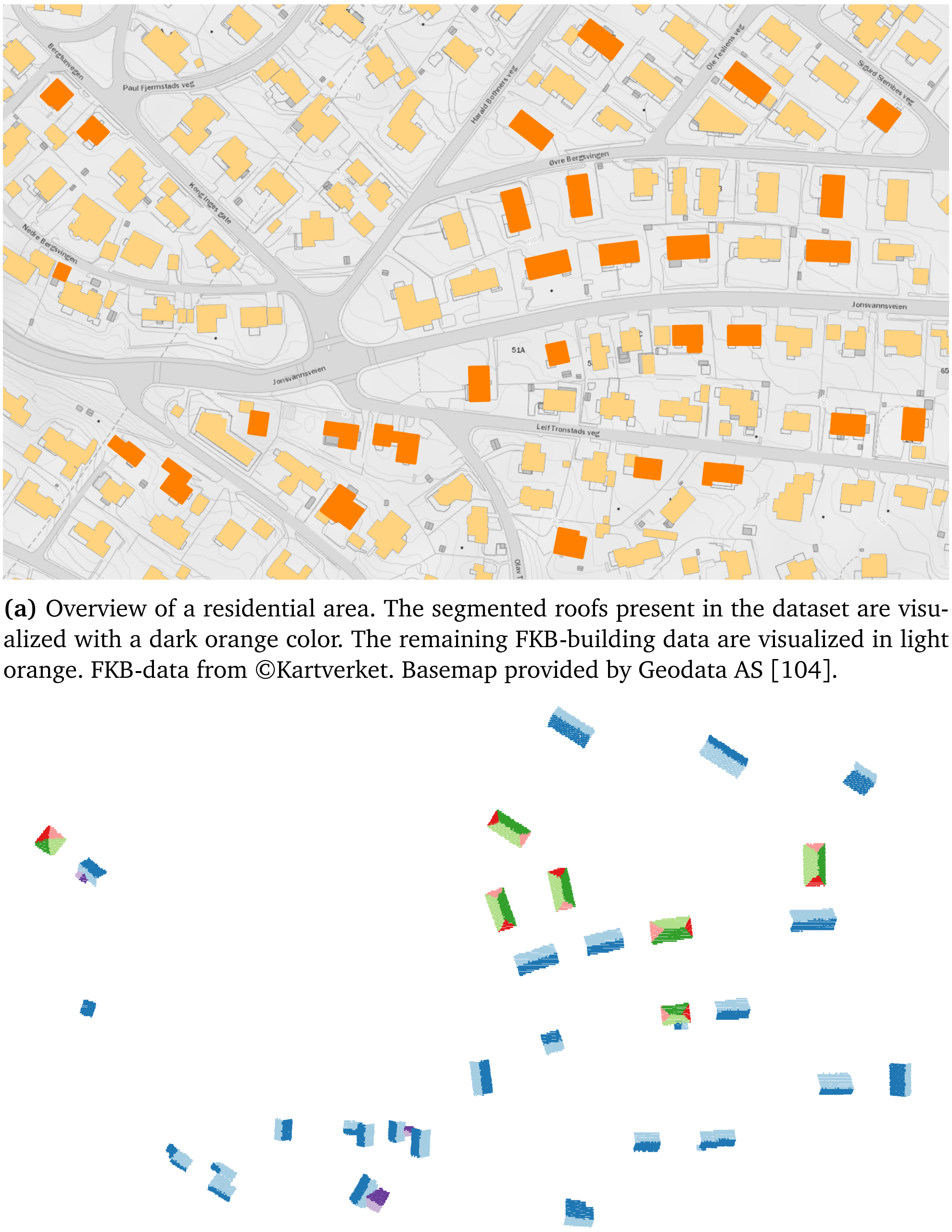


Figure 3.10: Map showing the location of building footprint of all roofs in TRD3DRoofs. The main map has a padding of 2 pixels around each polygon to improve visibility, whilst the zoomed in portion of the map displays the real area for the footprints, The scale is 1:110 000 and 1:11 000 respectively. FKB-data from ©Kartverket. Basemap provided by Geodata AS [104].



(a) Overview of a residential area. The segmented roofs present in the dataset are visualized with a dark orange color. The remaining FKB-building data are visualized in light orange. FKB-data from ©Kartverket. Basemap provided by Geodata AS [104].

(b) Corresponding labelled point cloud data of the roofs. The different colors of the planes corresponds to the definitions in Table 3.1.

Figure 3.11: Examples of the labelled point cloud data in a selected residential area in Trondheim.

3.8.1 The Original TRD3DRoofs Dataset

The original TRD3DRoofs dataset consists of a total of 2 199 051 points belonging to 906 different roofs segmented into 3 344 planes. All roof structures present are corresponding to a real-world roof structure found in the Trondheim area. The building location of the sampled roofs are illustrated in Figure 3.10. All maps presented in this thesis was created using the software ArcGIS® Pro by Esri, FKB-data from ©Kartverket is filtered and overlaid a basemap provided by Geodata AS [104].

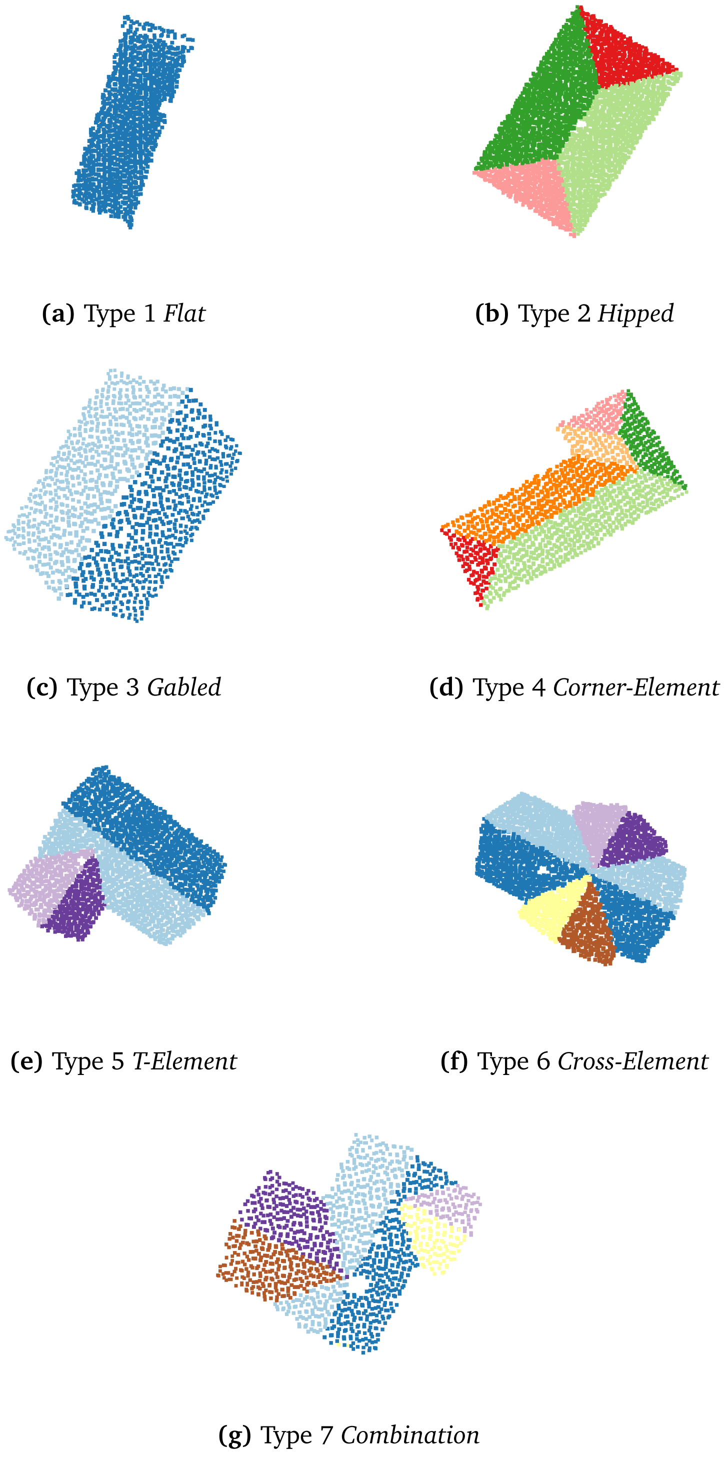
The dataset is well suited for applications such as urban planning or accurate renewable energy simulations in Trondheim. Additionally, it forms a good base for training data needed for machine learning technology.

An example of a residential area selected is shown in the zoomed in portion of Figure 3.10. The same area are presented in Figure 3.11, where both the labelled (dark orange) and unlabelled (light orange) buildings of this area are included. Figure 3.11b shows the labelled point cloud data of the corresponding roof structures in the resulting dataset.

Figure 3.12 shows a selection of resulting ground truth roof structures segmented into distinctive roof types. An example is shown for each roof type, but other variations of these are present in the dataset. Each plane is coloured based on plane number, corresponding to the definitions in Table 3.1.

The distribution of the different roof types in the original TRD3DRoofs dataset is visualized in the pie chart in Figure 3.13. Each roof type is visualized with a distinct colour, and these colours are consistent in all statistics presented in the result section. From the pie chart, a skew in the data with regards to roof type is apparent. The complete dataset contains only a small percentage of type 4 *Corner Element*, containing 38 roofs, and type 6 *Cross Element*, containing 14 roofs. An overweight of *Gabled* (type 3) and *T-Element* (type 5) roof structures are present, and respectively 262 and 191 roofs are present in these two categories. This is coincident with the distribution of real-world roof structures, as the use of gabled and T-element roof structures are quite common, while cross and corner elements are rather rare in Norwegian residential areas. A more even distribution is present with regards to the remaining roof types 1 *Flat*, 2 *Hipped* and 7 *Combination*, each containing around 130 roofs.

The histogram in Figure 3.14 visualizes the distribution of points among the different roof structures. From this, it can be deduced that most roofs contain between 1 000 and 4 000 points. A fair number of roofs also contain less than 1 000 points, and some outliers are also present in the dataset, including a few roofs containing up to 40 000 points. Roofs with such a significant number of points are often large, flat roof structures be-



(a) Type 1 *Flat*

(b) Type 2 *Hipped*

(c) Type 3 *Gabled*

(d) Type 4 *Corner-Element*

(e) Type 5 *T-Element*

(f) Type 6 *Cross-Element*

(g) Type 7 *Combination*

Figure 3.12: Examples of manually labelled ground truth data present in the TRD3DRoofs dataset. Each plane is colored corresponding to a correct plane label.

longing to either industrial buildings or residential blocks.

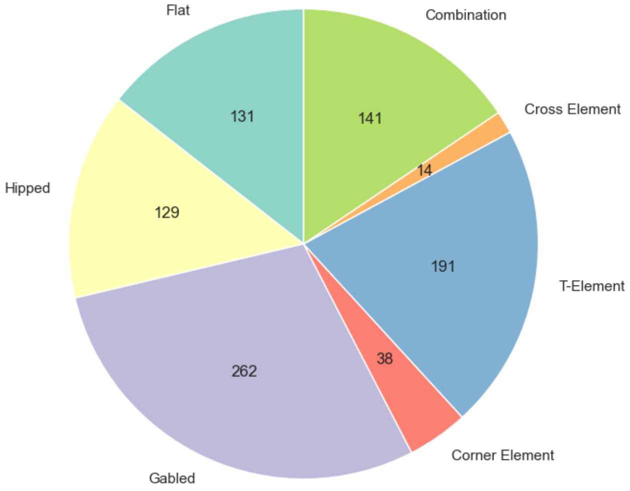


Figure 3.13: Pie chart of the roof type distribution in the original TRD3DRoofs dataset.

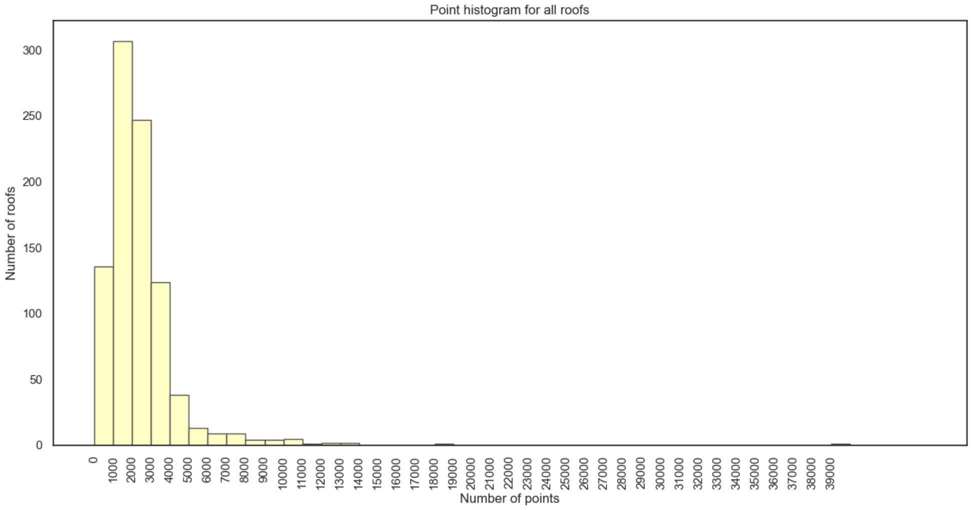


Figure 3.14: Histogram showing the distribution of roofs based on number of points for the original TRD3DRoofs dataset.

3.8.2 The Augmented TRD3DRoofs Dataset

The augmented TRD3DRoofs dataset is a more balanced dataset containing a larger number of roofs based on the original dataset. A total of 6 723 450 points belonging to 2 641 different roofs segmented into 12 007 planes are present. This dataset is constructed to be suitable as input for deep learning purposes. Such deep learning networks can then be used to predict the segmentation of similar roof plane data in other areas of Norway with similar buildings as those found in the Trondheim area. This will automate the process of establishing segmented roof data suitable for urban planning or renewable energy simulations. For this dataset, the augmentation described earlier is applied, adding new roof structures where a rotation of the plane labels is performed, as well as new roofs from the separation of those present in the combination category.

As can be seen in the pie chart found in Figure 3.15 the skew in the data have been reduced, and the dataset has a more even distribution of structures belonging to different roof types. The number of roofs in the *combination* category (7) has increased to 674 roofs, as several structures present in this category has been split into multiple roofs. A significant increase in the number of roofs is also present in roof type 2 *Hipped*, now containing 514 roofs. No augmentation is performed on roof structures only containing planes of a rectangular shape, thus, the number of roofs in type 1 *Flat* and 3 *Gabled* is the same as for the original dataset.

The histogram in Figure 3.16 visualizes the distribution of points among the different roof structures for both the augmented and original TRD3DRoofs dataset. The difference in the number of roofs present in the original and augmented dataset is clearly visible. In total, additional 1 740 roofs have been added to the dataset to increase the number of training data.

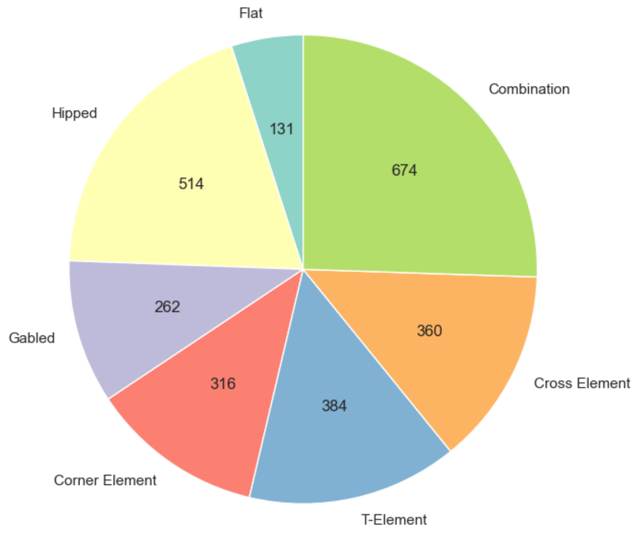


Figure 3.15: Pie chart of the roof type distribution in the augmented TRD3DRoofs dataset.

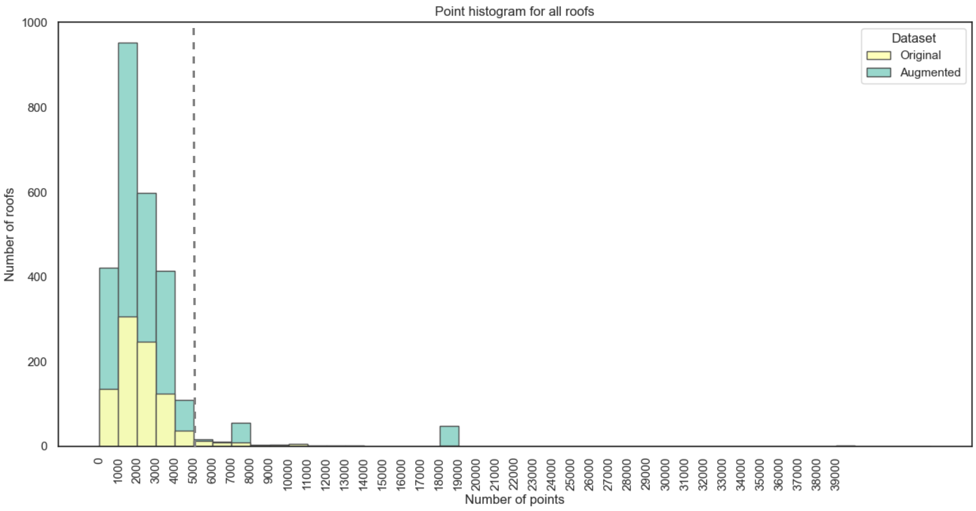


Figure 3.16: Stacked histogram showing the distribution of roofs for both the original and augmented TRD3DRoofs dataset.

Figure 3.17 presents the distribution of roof planes in the dataset after the split of roof type 7, before further augmentation is performed. Each bar is coloured corresponding to the label it belongs to, and these colours are the same used for the visualization of the ground truth data. As expected, a large overweight of the rectangular planes (0 and 1) is present, as these appear in multiple roof types. For the ladder-shaped planes, the dataset contains a lot fewer planes with labels 10 and 11 than 8 and 9. This is also as expected, as only a few roof structures contain a total of four different ladder-shaped planes. It can also be observed that few planes of a parallelogram (6 and 7) shape are present in the dataset. These are planes present in some, but not all corner-element roof types, as well as some roofs of the combination type.

The effect of the further balancing of the dataset by augmentation can be seen in the plane label distribution in Figure 3.18. The large overweight of rectangular planes (0 and 1) has been reduced, by an increased number of planes in all other categories. Still, the number of planes belonging to the rarer plane types are lower than the other planes. However, we find that the dataset contains a large enough amount of training data to make this skew less evident, indicating its suitability in deep learning technologies.

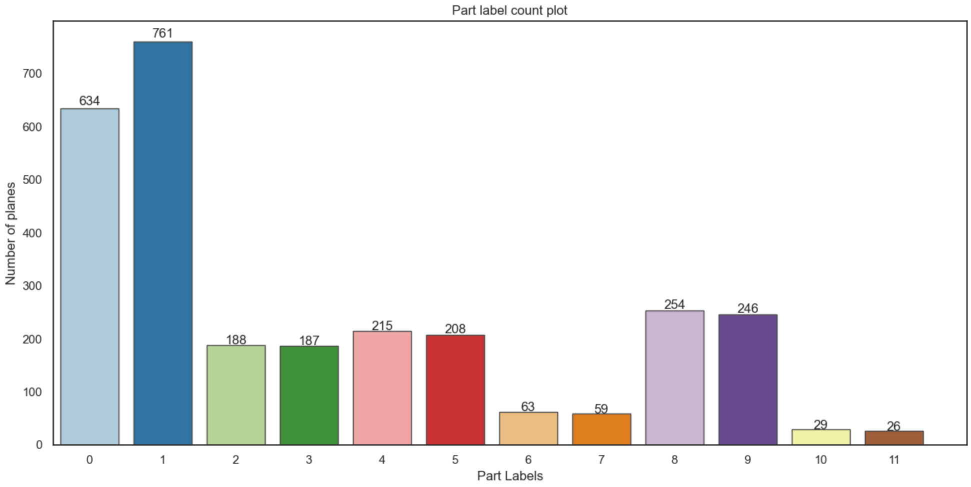


Figure 3.17: Distribution of plane labels before rotation.

The point density distribution of roofs in each roof type is found in the violin diagram in Figure 3.19. For visualization purposes, the same colours as for the pie chart is used to visualize the roof types. The shape of each plot describes the distribution of densities within a roof type. The number of roofs having a certain point density is indicated by the wideness of the plot at that density. Keep in mind that the area of each plot is constant, and no scaling is done based on the number of roofs present in each category.

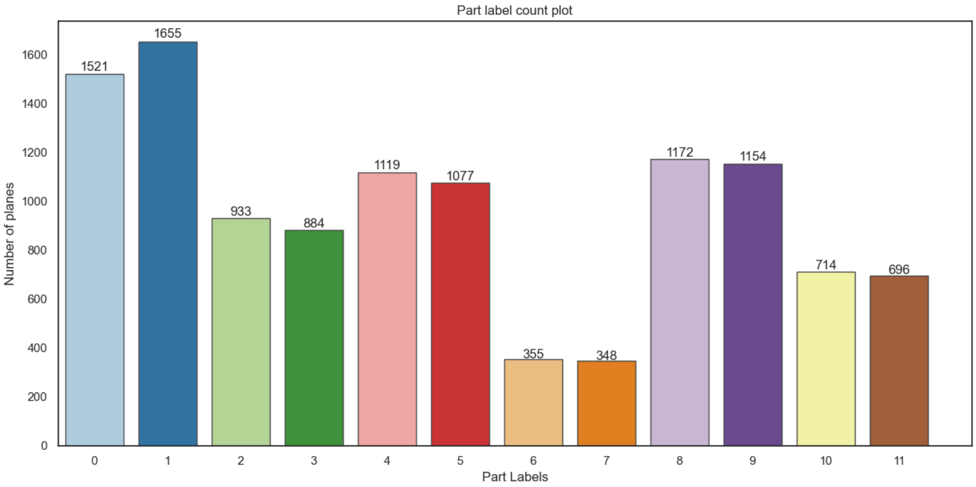


Figure 3.18: Distribution of plane labels after rotation.

Except for some outliers in roof type 5 *T-Element*, all densities are found to be between approximately 2.5 and 17.5 points/m^2, with a mean density of 9.07 points/m^2. Roof type 4 *Corner Element*, 5 *T-Element* and 6 *Cross Element* have a similar distribution of densities, with two major clusters of densities centred around 5 and 10 points/m^2. Additionally, the distribution of densities in type 1 *Flat* is found to be similar to these, but contains a larger degree of roofs with a lower density. Roof type 7 *Combination* also have a similar density distribution, but this is slightly shifted as these roofs have an overall lower density. Type 3 *Gabled* differs from the beforementioned roof types in that it only has one major cluster of densities, centred around 10 points/m^2. Roofs of type 2 *Hipped* tend to be denser than other roof types, and most of the roofs are grouped into two major clusters centred around 10 or 15 points/m^2.

The augmented TRD3DRoofs dataset is established to work as deep learning training data for the purpose of 3D roof plane segmentation. The original dataset used as the base consists of roofs with a varying number of points, and this includes outliers with a point number far exceeding that found in most roofs. This variation in the number of points is undesirable for deep learning applications, and a subset of the augmented data is established to overcome this obstacle. In this subset filtering of outliers is performed. All roofs with over 5 000 points are considered outliers, only roofs containing a lower number of points than 5 000 are included in the subset, indicated by the dotted line in Figure 3.16. This filtered dataset consists of a total of 5 083 462 points belonging to 2 497 distinctive roofs further separated into a total of 11 139 planes. The resulting distribution of plane types in this subset can be observed in Figure 3.20.

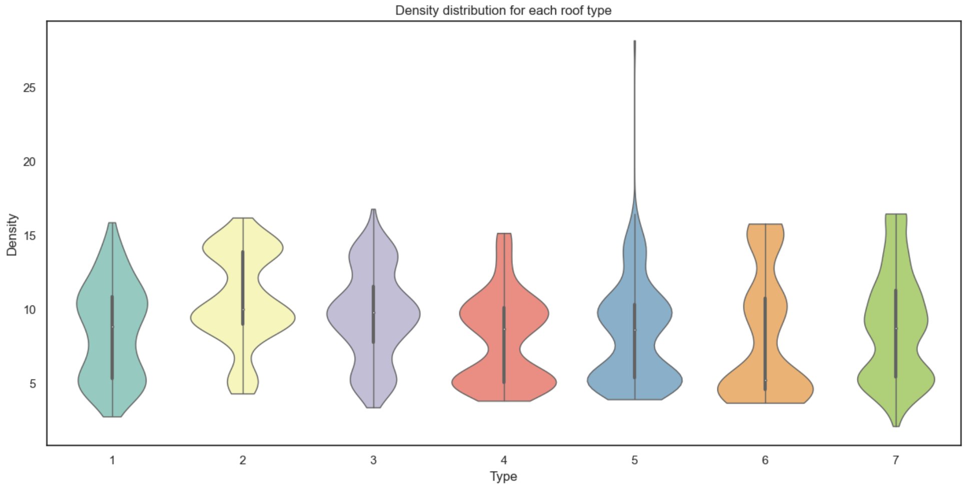


Figure 3.19: Violin diagram showing the point density distribution of the Augmented TRD3DRoofs dataset.

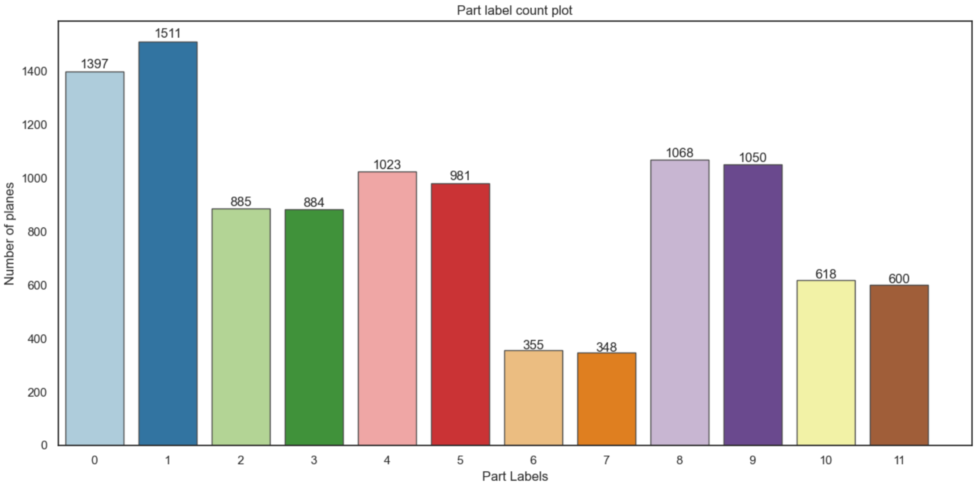


Figure 3.20: Distribution of plane labels after rotation and filtering.

Chapter 4

Deep Learning-Based Roof Segmentation using TRD3DRoofs

In this chapter, the usability of the Augmented TRD3DRoofs dataset in deep learning applications is investigated. The slightly adjusted PointNet++ network described in chapter 2 is used for the evaluation of the dataset. First, a description of the experimental setup used is given. Details about the adjustments applied to the state-of-the-art network PointNet++ is included before the results of the predicted roof structures are presented.

4.1 Experimental Setup

Details about the experimental setup are explained in this section, including the software and hardware specifications, as well as the training procedure used for the applied deep neural network.

4.1.1 Software

The chosen software for the experiment was PyTorch [105] 1.2 for Python 3.5 and CUDA 10. PyTorch is developed mainly by the Facebook AI research lab and is an open-source machine learning framework commonly applied in the implementation of deep learning neural networks. It enables powerful tensor computations on Graphical Processing Units (GPU) useful for neural network implementation. PyTorch was chosen as we have more experience with this framework, it integrates better with Python and because it is generally considered to be more transparent and developer-friendly, making it easier to get familiar with existing code [106]. An existing implementation of PointNet++ with PyTorch was used as the basis for the neural network [107]. The software LAStools [108] and more specifically the tools *text2las* and *lasinfo -compute_density* was used for the processing of LiDAR data. For visualization of the results, the open-source software MeshLab [109] was used. This software facilitates easy rendering of large 3D meshes and therefore is suitable for 3D point cloud visualizations.

4.1.2 Hardware

All experiments have been completed by a remote workstation with the following technical specifications regarding the Central Processing Units (CPU) and GPU:

Processor: Intel Xeon(R) Gold 6146 CPU @ 3.20GHz x 45
Graphics card: Nvidia Quadro GV100/PCIe/SSE2
Memory: 250.6 GB

4.1.3 Part Segmentation using PointNet++

PointNet++ were chosen as the deep neural network to assess how suitable our suggested dataset is for the task of segmenting roofs into different roof planes. This task can be defined as a part segmentation task, where each roof plane is a separate part of a roof. This network was selected based on several considerations. Firstly, an architecture that took a point cloud directly as input was sought, as this is the nature of the proposed data. Secondly, the network needed to handle highly fluctuating densities because the density in our dataset varies between 2.5 and 17.5 points/m^2. Additionally, it was important to find a low-complexity high-performance network as time and access to computation power would be limited. In addition to fulfilling these requirements, PointNet++ was chosen as it is regarded as state-of-the-art at point cloud segmentation tasks and is deemed a pioneering network within its field. Because it being such a renowned network, it is well explored and documented by other researchers, meaning that it also has many questions and solutions readily available on the internet.

The network was trained using our presented dataset, the Augmented TRD3DRoofs datasets consist of 4 641 roofs after pre-processing, adhering to the following procedure: The roofs were split into training/validation/testing sets, using a ratio of 80:10:10 split for each roof type, as shown in Table 4.2. We chose a to reserve a large portion of the data for training, due to the size limitation of our dataset. The training and validation data were then randomly arranged into 32 batches and normalized. The network was trained batch wise for 251 iterations using the Adam op-

Table 4.1: Train, validation and test split for MSG_100.

	Training	Validation	Testing
Type 1	89	11	11
Type 2	412	51	51
Type 3	208	26	26
Type 4	253	32	31
Type 5	302	38	38
Type 6	288	36	36
Type 7	446	56	56
Sum	1998	250	249

timizer. For each batch, it takes one roof at the time as the input, selects 3 000 points using Nearest Point Sampling, each knowing its xyz-values, estimated normal vector and part label, and outputs a prediction and lastly updates the network weights. An initial learning rate of 0.001 and a momentum of 0.1 were used, both with a decay rate of 0.5 and step size 20. All parameters are the same as those chosen for the original network, except for the number of sample points. This is sat higher to better learn more complex roofs.

Furthermore, two data augmentation steps were added at training time to improve variation in the training data and thereby reducing the possibility of overfitting the network. The first was a random shifting of the roofs within one batch within the range of 0.1 normalized unit. This way an increased number of possible roof positions was simulated. The other augmentation technique was a random scaling of each roof within one batch, making the roofs somewhere between 20% smaller and 25% larger. Again, the purpose of this is to expose the network to an increased variation in roof sizes.

4.2 Experimental Results

Here, the final results achieved for the part segmentation task on the TRD3DRoofs dataset using PointNet++ is presented. The PA, mIoU and the training time are reported for different training configurations of the deep neural network. Furthermore, the PA and mIoU are reported for each roof type the model attaining the highest mIoU score. These findings will later be used as a basis to do a final evaluation of our datasets suitability for deep learning tasks, as well as reporting our experience when training

Table 4.2: Different training configurations.

Model	Number of roofs	PA	mIoU	Training time (hours)
SSG_100	2248	0.541	0.746	6.5
MSG_100	2248	0.558	0.752	6.5
MSG_80	1799	**0.657**	**0.794**	3.3
MSG_60	1351	0.627	0.787	2.4
MSG_40	897	0.603	0.776	**1.0**
MSG_20	449	0.603	0.778	**1.0**

on PointNet++.

4.2.1 Model Configurations

Various models were trained using different configurations to observe which base model and input data yielded the best results on our dataset. Initially, the models were trained on the complete dataset using both the MSG and SSG model variation presented by PointNet++, see Table 4.2. For these models we achieved a mIoU of 0.7456 for SSG_100 and 0.75172 for MSG_100. Both models took an equal number of hours to train. These results are consistent with the results reported in the PointNet++ paper, where the MSG approach also attains the higher results. On this basis, the MSG variation is chosen for the training of the following models.

Table 4.3: Train, validation and test split for MSG_80.

	Training	Validation	Testing
Type 1	71	9	11
Type 2	330	41	51
Type 3	166	21	26
Type 4	202	26	31
Type 5	242	30	38
Type 6	230	29	36
Type 7	357	45	56
Sum	1598	201	249

To decide the optimal roof number for training, four models were trained on a diminishing number of roofs. Using the model trained on the original 2 497 roofs as the starting point, the number of roofs in the training- and validation sets were reduced from 100% down to 20%, with

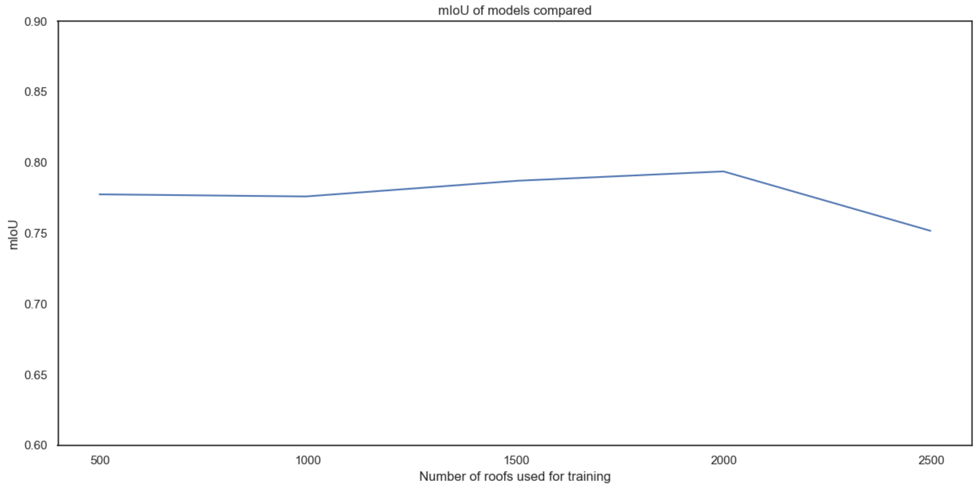


Figure 4.1: The plot shows the achieved mIoU vs. the number of roofs used for training of the model. The mIoU increases steadily, before it drops when trained on the complete dataset. mIoU of models trained.

a step size of 20%, between each of the four models. All models were, however, tested on 100% of the 249 roofs in the test dataset to procure comparable results. An example of this train/val/test split is shown in in Table 4.3 for the model utelizing 80% of the test and validation data. All other splits are listed in in Appendix A. The reported test PA, mIoU and training time for each model can be found in table Table 4.2. Here we can see that the model performs increasingly better with more training data, reaching a peak utilizing 80% of the data, before showing a drop in performance when trained on the full dataset. The best model (MSG_80) was trained using 80% of the training- and validation data, corresponding to 1799 roofs. It obtained a test PA of 0.658 and mIoU of 0.794 in the test data, after training for 3.3 hours. Figure 4.2 and Figure 4.1 shows plots of how, respectively, the PA and mIoU varies based on the number of roofs used for training.

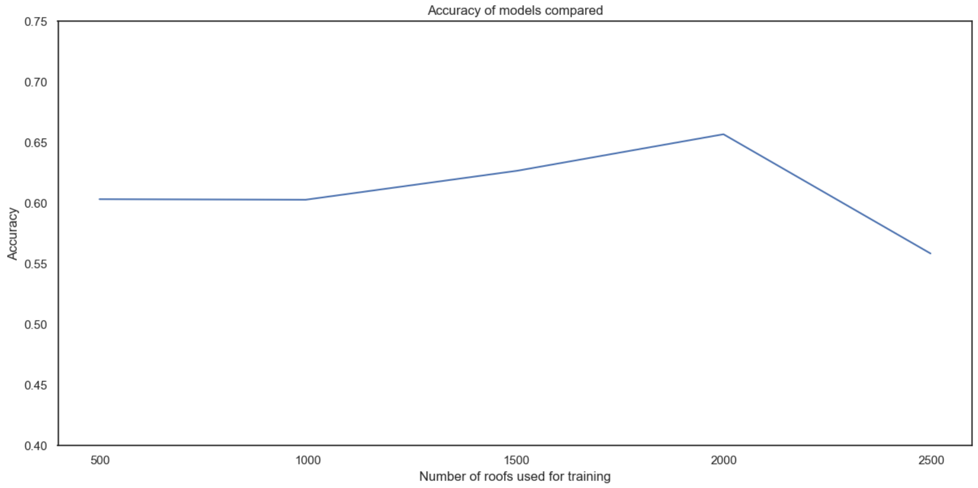


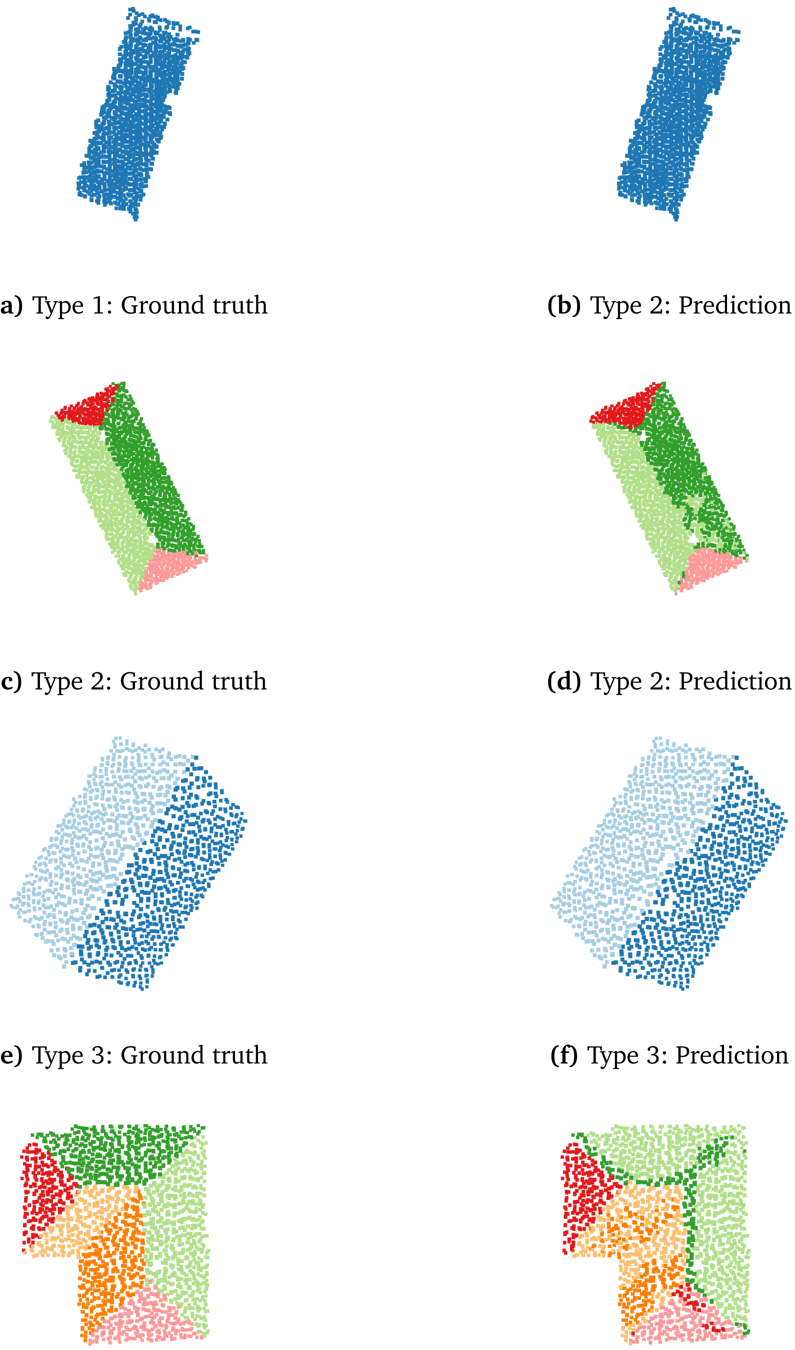
Figure 4.2: The plot shows the achieved PA vs. the number of roofs used for training of the model. The same tendency as for the mIoU is also present here.

4.2.2 Optimal Number of Points: Final Results on PointNet++

Based on the model configurations, the MSG_80 model was found to achieve the best results for PA and mIoU. In Figure 4.3, a selection of predictions performed on the test dataset are shown, additional results may be found in Appendix B.

From the results, it can be observed that model is good at segmenting and separating rectangular planes. It has more difficulties with parallelogram and ladder shaped planes. The worst result are found for roofs of type 2 *Hipped*, where points part of the isosceles trapezoidal planes are mistaken for rectangular planes.

Additionally, result of predictions performed on the complete dataset may be found in Appendix C.



(a) Type 1: Ground truth

(b) Type 2: Prediction

(c) Type 2: Ground truth

(d) Type 2: Prediction

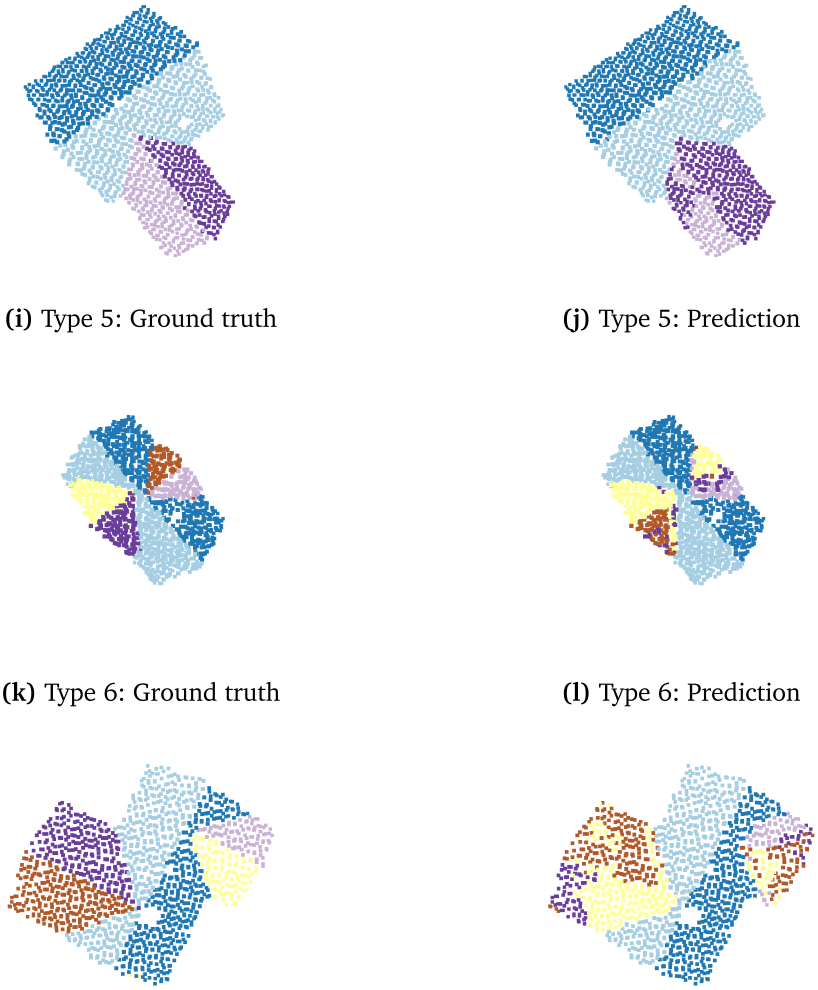
(e) Type 3: Ground truth

(f) Type 3: Prediction

(g) Type 4: Ground truth

(h) Type 4: Prediction

Figure 4.3: Result of predictions performed by the MSG_80 model compared to the corresponding ground truth data.



(i) Type 5: Ground truth

(j) Type 5: Prediction

(k) Type 6: Ground truth

(l) Type 6: Prediction

(m) Type 7: Ground truth

(n) Type 7: Prediction

Figure 4.3: Result of predictions performed by the MSG_80 model compared to the corresponding ground truth data.

Chapter 5

Evaluation and Discussion

This chapter presents the result of the evaluation performed for the experimental segmentation process outlined in chapter 4. Furthermore, the result of the segmentation, aspects regarding the proposed TRD3DRoofs dataset, and the applied network, PointNet++, are discussed in light of relevant theory and work.

5.1 Evaluation

The evaluation of the network yielding the best results chosen from the model configuration, MSG_80, is elaborated in this section. This will indicate the performance of the model for the task of roof plane segmentation. For the evaluation, the metrics introduced in chapter 2, PA and mIoU are used. The mIoU is calculated over all distinct planes present in a roof structure, meaning that this value represents the average of the IoU metric for each plane, given for each roof.

The plot in Figure 5.1 shows the training and test PA for the model, reported for each epoch. Both accuracies increase fast, yielding a PA of approximately 55% after only a couple of epochs. The training PA continues to increase, but at a slower rate. This is contrary to the test PA which comes to a halt, before slightly dropping. This large gap between test and training PA indicates that the model might be overfitted. This could also explain the reason for the drop in the performance for the MSG_100 model trained on the complete dataset.

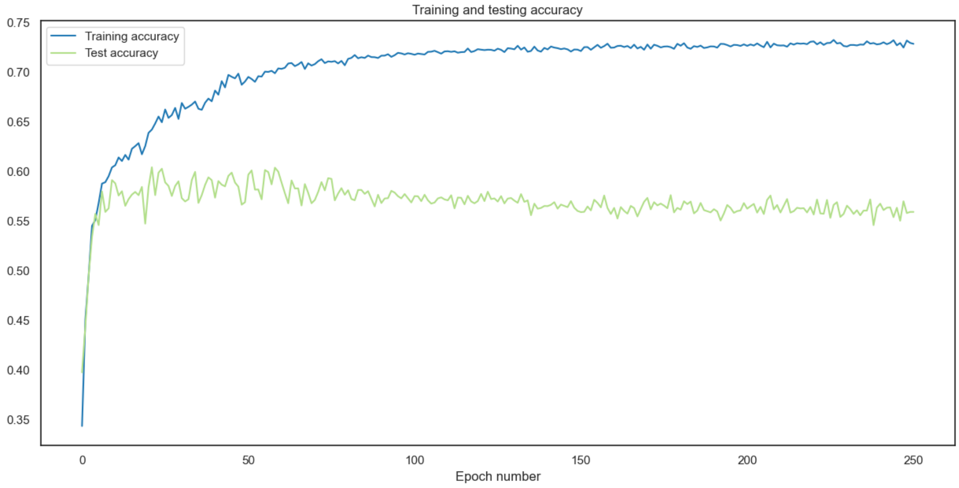


Figure 5.1: Train and testing PAs for model MSG_80.

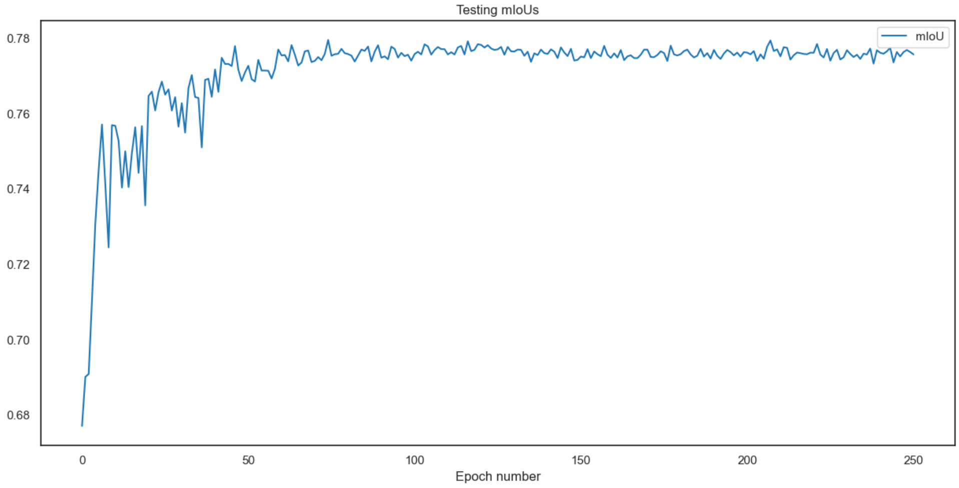


Figure 5.2: mIoUs for model MSG_80.

Table 5.1: Result of type-by-type metrics for MSG_80.

Roof type	PA	mIoU
1	0.997	0.992
2	0.475	0.751
3	0.804	0.970
4	0.432	0.642
5	0.851	0.885
6	0.722	0.691
7	0.704	0.791

Another observation from this plot is that no further improvement is present regarding the PA after approximately 50-70 epochs. This is an indication that the model could benefit from early stopping. Early stopping is a regularization approach that can be implemented to avoid overfitting[110]. This will provide guidance regarding the number of iterations that can be performed before the model becomes overfitted.

The plot in Figure 5.2 shows the mIoU for the validation set of the model. During the first epochs, the mIoU is varying, but have a gradually increasing tendency. After approximately 50-70 epochs, the value stabilizes and does not get any better. This is a similar tendency to that of the PA metric, further indicating that early stopping would be beneficial. A mIoU of 1 indicates that the predictions are perfect, and the mIoU reached here is 79.4%.

The MSG_80 model's performance on different roof types was additionally evaluated. To make this possible the model was tested on a variation of the test set where the roofs are separated based on the roof type. For these results, the average of the mIoU over all roofs is reported. The result of the performance is displayed in Table 5.1.

The best performance is found for roof type 1 *Flat*, with both a PA and mIoU close to 100%. Roofs of type 3 *Gabled* and 5 *T-Element* also yields good results, with PAs of 80% and 85% and mIoUs of 97% and 88%. For roof type 4 *Corner Element*, the worst PA is obtained with a score of 43%, and mIoU of 64%. The performance on roof structures of type 7 *Combination* is surprisingly good as this category is assumed to contain the highest number of complex roofs, with a PA of 70% and a mIoU of 79%.

The PA metric does as mentioned favour large areas, yielding a skewed result in cases where the separation of large planes is either remarkably good or bad. Based on the calculated metrics, there is an indication that this might be the case for this dataset. The model tends to predict a worse segmentation of the larger planes when exposed to roofs of type 2 *Hipped* and 4 *Corner Element*, compared to type 3 *Gabled* and 5 *T-Element*. Resultingly, the mIoU will give a better impression of the performance of the model.

Based on the labelling strategy and taxonomy established, it is desired that the points are recognized as correctly predicted if they are annotated with a label from the correct geometric class and segmented into separate planes. This means that roof planes can be arbitrary annotated with labels in the same geometric class and still be recognized as correctly predicted. An example of this is shown in Figure 5.3, where the labels of the ground truth data and the predicted results have been switched. A prediction like this should give a PA of approximately 100%, as both planes are almost completely correctly predicted into their geometric shape and separated by the roof ridge. PointNet++ does not take this fact into consideration in their calculation of the PA metric. Resultingly, the PA of the predicted results for this case will be calculated to approximately 0%, indicating that none of the planes are correctly segmented.

(a) Ground truth

(b) Prediction

Figure 5.3: Example of a case where the labels are switched for ground truth and predicted roof.

Therefore, evaluation performed by only investigating the PA metrics proposed and used in PointNet++ will not correctly demonstrate the performance of the network trained on the Augmented TRD3DRoofs dataset. While the metrics are not optimal for detailing the actual performance of the models, they are useful for the comparison between the different models proposed. To solve the lack of suitable metric indicating the performance of the models, a visual assessment was manually performed as part of the evaluation process.

5.2 Manual Evaluation

To capture a more complete overview of the performance of the trained network, two different accuracy measurement were established for the manual evaluation. The first metric, part accuracy, describes how well the network is at segmenting the different planes of a roof from each other. This metric is the same as the original PA, but is altered to solve the problem of switched labels. The second metric, geometric accuracy, describes the performance of the segmentation of the planes into the correct geometric shape. This means that if the label given corresponds to the correct geometric shape, the prediction of the point will be recognized as correct. Both metrics are found by a visual evaluation, performed manually, and are done for the entire roof structure as a whole. Consequently the metrics are prone to a human degree of error. Both metrics found by manual evaluations are only estimations of the accuracy as they are found through visual inspection, and a scale increasing by 5% for each step is used.

The manual evaluation is performed for two of the trained models, the model trained on the complete dataset, MSG_100, and the model yielding the best results from the calculated metrics, MSG_80. 10 different roof structures of each roof type were randomly chosen from the test dataset, giving a total of 70 roofs used for evaluation. These were visualised and manually evaluated, resulting in estimated scores for the complete roof. Tables for both models containing estimated values for each roof are to be found in Appendix D and Appendix E. The average of both measurements for each roof type is calculated, together with the mean for all roof type classes. Table 5.2 contains a full overview of the result of these calculations for both models.

The MSG_80 model performs slightly better than the MSG_100 with regards to the mean geometric accuracy with scores of respectively 97% and 96%. Scores for both models are high for this metric, indicating that PointNet++ recognizes the different geometric shapes defined in the TRD3DRoofs dataset to a large extent. It can be observed that this behaviour seems to be independent of the roof structure, as there is a consistency in the performance yielded for each type. For roofs of type 1 *Flat*, both models excel, yielding scores of 100% for both part and geometric accuracy. The lowest geometric accuracy obtained for the MSG_80 model is a score of 96%, obtained for roofs of type 2 *Hipped*, 4 *Corner Element*, 5 *T-Element* and 6 *Cross Element*. For the MSG_100 model, the lowest geometric accuracy is obtained for type 2 *Hipped* with score of 90%.

For the part accuracy, the performance of both models is somewhat worse, and a larger difference is present between the two. MSG_80 have a

Table 5.2: Visual evaluation performed on model MSG_100 and MSG_80.

Roof type	Trained model	Part accuracy	Geometric accuracy
1	MSG_100	**1.00**	**1.00**
	MSG_80	**1.00**	**1.00**
2	MSG_100	0.71	0.90
	MSG_80	**0.72**	**0.96**
3	MSG_100	0.92	**1.00**
	MSG_80	**0.96**	**1.00**
4	MSG_100	0.65	**0.96**
	MSG_80	**0.73**	**0.96**
5	MSG_100	0.86	**0.98**
	MSG_80	**0.87**	0.96
6	MSG_100	0.76	0.95
	MSG_80	**0.82**	**0.96**
7	MSG_100	0.76	**0.97**
	MSG_80	**0.82**	0.97
Mean	MSG_100	0.81	0.96
	MSG_80	**0.84**	**0.97**

mean part accuracy of 84%, while the same metric for MSG_100 is 81%. Contrary to the geometric measurement, there is a variance in the performance dependent on the roof type. This variance is present in both models. The part accuracy metric drops noticeably for roofs of type 2 *Hipped* and type 4 *Corner Element*. For the MSG_80 model, type 2 *Hipped* structures yield the worst performance, with a score of 72% , while the MSG_100 performs worst on type 4 *Corner Element*, with a score of 65%. A similar tendency as the calculated metrics regarding the accuracy and the favouring of large planes is also present in these visual evaluation results.

By observing the predicted results, it can be noted that most of the wrongly labelled points are present along ridges in the roof. For roofs with a small degree of error, the wrongly labelled points are often those found on or close to the ridge to another segment. The presence of such error is most likely due to the variance in the neighbourhood of such points, as they will include points belonging to multiple planes.

It is important to notice that the metrics calculated are based on a subset of the test dataset, not all predictions performed by the models. This indicates a limitation in the validity of the results and evaluations presented.

5.3 Discussion

This section discusses certain aspects of the new TRD3DRoofs point cloud dataset, together with the results obtained from the segmentation performed using the neural network PointNet++. These aspects are discussed in light of related work and theory, as a tool in understanding why certain results are yielded, and how the methods have been conducted.

5.3.1 Labelling Strategy for Deep Learning Purposes

As indicated by the visual evaluation, the labelling strategy of the ground truth data might affect the result of the predictions performed by a neural network.

PointNet++ is presented as a neural network capable of both capturing local and global features of the input point cloud [68]. This means that not only the local neighbourhood of a point is considered during the segmentation, but the point's complete roof structure is evaluated as a whole as well. Resultingly, the network has the ability to learn the relationships between adjacent planes in the roof structures. One approach to the problem of labelling is consistency in plane labelling with regards to global factors. By labelling the planes with an approach that keeps the relationship between adjacent planes consistent, the network could learn this connection, improving the prediction of the roofs.

Most of the roofs of the same roof type have a similar structure, meaning that they will contain the same number of planes with consistent relations between them. By implementing consistency in the labelling of planes within roof types, there is reason to believe that the plane accuracy of several roof types would increase. For roof types where this has unconsciously been adapted in the labelling process, such as 3 *Gabled* and 5 *T-Element* the achieved part accuracy is significantly higher. The same strategy could with benefit have been applied to other roof types such as 2 *Hipped*, 4 *Corner-Element* and 6 *Cross-Element*. However, this is not the case for roofs of type 7 *Combination*. These roofs have an arbitrary number of planes, and no reliability is present in their positioning to each other. Thus, this approach is not guaranteed to solve the problem of part segmentation for these roof structures.

Based on the presented results, there is an indication that the connection between plane label and normal vector affects the performed predictions. There is a possibility that PointNet++ is dependent on consistent normal vectors to correctly segment the planes into distinct planimetric shapes, thereby yielding weaker results for the part segmentation. However, as roofs present in real-world environment are placed with arbitrary rotations, there is no possibility of keeping a consistent relationship between a roof plane label and the normal vector of a plane in such data.

[111] adapted their solution to a labelling where the planes were annotated based on cardinal directions rather than geometry. This is especially suitable in cases where a fixed number of planes is present in the roof structures. This labelling strategy would give a better consistency both for the relationship between adjacent planes and for the normal vector of each type. Both factors would be consistent across different roof structures. Norwegian roof structures are, as discussed, prone to arbitrary rotations. This would complicate the definition of the cardinal directions in the taxonomy for this strategy. To achieve the desired benefits from this method, there is also a need for a fixed number of planes. This is not suitable for the roof types defined in this thesis, both due to the variations across the classes and because of the large variation of roofs present in the combination category. Resultingly, we find that the drawbacks of this approach make it unsuitable for annotation of Norwegian roof structures.

There is a desire for the network to learn to separate the planes independent of the rotation and the connection between planes. Solving these issues is a non-trivial task. As mentioned, deep neural networks are often referred to as “black boxes”. One can only assume which features are of the highest importance, and thoroughly testing of different approaches is needed to prove these theories. For point cloud applications, the amount of research performed is still sheer, and proposed novel networks such as PointNet++ are prone to weaknesses.

5.3.2 Discussion of the Achieved Results

Based on the given evaluation, the following discussion is proposed to better understand the obtained results and possible explanations.

The drop of PA for the model tested on the complete dataset, together with the large gap between test and training PA indicates that the model might be overfitted. For relatively small datasets, supervised learning algorithms tend to overfit [49]. One explanation as to why this happening, in this case, is that the model hasn’t been exposed to enough examples of different variations of data. In datasets where the data is very similar, overfitting is more common. This problem could be solved by increasing the size of the dataset, including variations of different structures. Obtainment of such data is possible by further manual annotation of roof structures, or through data augmentation. Manual labelling is both time-consuming and expensive, leaving further data augmentation as the best option. Another possible explanation for these issues is that the hyperparameters in the network are not optimized. The setting of these hyperparameters is discussed further in subsection 5.3.5.

One question that arises from the proposed results is why the network is better at segmenting the roofs based on geometric features rather than planimetric feature. The labelling strategy proposed might be a factor that affects the methods ability to segment the planes. Each plane is segmented into different geometric classes in the training data, and this labelling of a geometric shape is kept consistent across all roof types. A reasonable possibility is that the network learns the shape and connects it to a label, and if this is done consistently in the data it is exposed to during training, it will recognize such shapes when exposed to the test data.

For the segmentation of the planes based on their planimetric features, it is reasonable to assume that other attributes of the points are studied during the learning process. PointNet++ considers both local and global features during the segmentation [68], and global features might include factors such as the connection between adjacent planes. The results indicate that PointNet++ is dependent on consistent relationships between the planes, to exploit this as a global feature. For the manually labelled data in the TRD3DRoofs dataset, these relationships are not consistent, as the structures of the roofs vary both within the same roof type and across the different classes.

Further study of roof types yielding bad results for planimetric segmentation substantiates this theory. By inspecting the ground truth data labelled for type 2 *Hipped* and 4 *Corner-Element*, it can be observed that no consistency with regards to plane labelling is present. It appears like the manual labelling strategy has been unconsciously less consistent during the annotation of these roof type structures. This might explain the network's lack of ability to segment planes in these roof structures properly. Especially roofs of type 3 *Gabled* and 5 *T-Element* have consistent labelling in the ground truth data, and as the plane accuracy for these two is among the highest achieved, the assumption that the relationship between labels is of great importance is further supported.

The model performs surprisingly well for roof structures of type 7 *Combination*, even though this category contains some of the most complex structures. A reason for this might be the fact that a lot of the roofs of more complex structures has a basis consisting of rectangular planes. These planes are well-represented in roofs of type 3 *Gabled* and 5 *T-Element*, and the model has learned to predict these planes adequately. Resultingly, these will to a large extent be correctly predicted for the combined roof structures, increasing the performance in this class.

For the determination of planimetric shape, the normal vector of each point is another important feature. Each roof obtained and segmented in the dataset are based on a real-world object present in the Trondheim area. In real-world data, buildings are placed with arbitrary orientation in relation to the other roofs. This orientation is kept the same in the dataset, meaning that a roof of a certain type might have several different arbitrary orientations around the z-axis. Inconsistency in the direction of the normal vector might be a factor affecting the predicted results of part accuracy. The number of neighbours in the k-NN algorithm is an important factor. For the proposed TRD3DRoofs dataset, the six nearest neighbours were considered during the calculation of normal vectors for each point. This number might not be optimal, yielding inferior results for the plane segmentation. This value, as well as the effect of this on both the part and geometric accuracy, is further discussed in subsection 5.3.3.

5.3.3 Calculation of Normal Vector using k-nn: Effects of Varying the k

Based on the evaluation, there is an indication that the calculation of the normal vector affects the prediction performed by the network. The calculation of the normal vector is based on the k-NN algorithm. In this section, the effects of varying the value of k, the number of nearest neighbours to retrieve, is discussed.

If the number of retrieved neighbours is too small, the algorithm will be sensitive to outliers. This would result in too much variance in the values, and the normal vector calculated for each point of a plane would be inconsistent. If a too large number of neighbours are retrieved, the neighbourhood may include points from adjacent planes. In cases where most points present in a point neighbourhood are belonging to bordering planes, this is a problem. The points normal vector would in this situation represent the neighbouring plane, leading to the point possibly being wrongly segmented. This is particularly a problem along the ridges of a roof structure.

The visual inspection performed, indicates that most of the wrong predictions are done along ridges, indicating that a $k = 6$ is not optimal for this dataset. Research for finding this optimal value is an aspect possible to investigate in future work.

5.3.4 Rule-based Post-Processing

As mentioned in the previous section, a lot of the errors of the predictions occur along ridges. Figure 5.4 shows one example of how the points along ridges are prone to errors, being wrongly segmented into the adjacent planes.

If the error present in the segmentation occurs along the ridge of the roof structures, a higher degree of error could be considered sufficient for the intended applications presented in this thesis. When seen in light of the 3D modeling process, the segmentation step of the roof structures does not need to be flawless. Before the final 3D model is to be constructed, polygons must be derived for each of the segments. Post-processing of the segmented dataset is therefore a necessity, and this process could exclude errors present at the ridges.

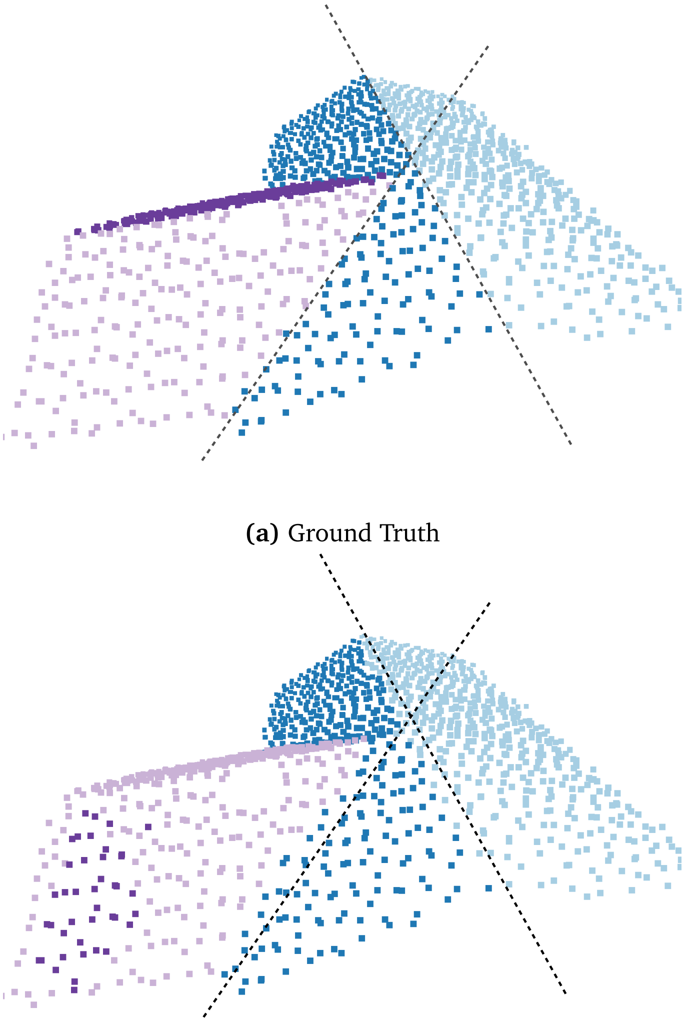
If not automatically done, rule-based methods can be implemented to correct small errors in the segmentation. Rule-based segmentation have earlier been applied for LiDAR point clouds [112]. By defining a set of rules for the proper segmentation of wrongly segmented points along the ridges, the post-processing will reduce the amount of error present in the predictions.

Based on this reasoning, it is sensible to find the result of the roof plane segmentation of TRD3DRoofs using PointNet++ sufficient for modeling purposes.

5.3.5 Hyperparameter Optimization and Training Split

The performance of a neural network is highly dependent on the configurations of the parameters. Optimization of hyperparameters are important aspects that can affect the training process, and thereby also the predicted results [51]. Based on this, there is reason to believe that adjustment of the used hyperparameters could yield better results for the segmentation of roof planes.

To find the most suitable settings, tests and analysis are needed for each of these parameters. This is done through trial and error, and is often a hugely time-consuming task as the network needs to be trained for several variations of each of the parameters. The network currently uses 3-6 hours for training. With the limitations of the thesis regarding available hardware and time, this a task not feasible to perform. Consequently, hyperparameter



(a) Ground Truth

(b) Predicted Result

Figure 5.4: Visualization of error present along ridges in the predicted result. The dotted lines indicates the ridges of the roof. It can bee seen that points from the ladder shaped plane (light purple) along the ridge has been wrongly segmented into the neighbouring rectangular plane (blue). This is also the case along the ridge between the rectangular planes (blue-shades), where several points are segmented into the wrong plane.

optimization goes beyond the scope of this thesis. This is a topic that with benefit could be further investigated in future work.

Another factor that affects the training of neural networks is the train/validation/test split of the dataset. Currently, the split ratio for training and testing applied to the data is 80 : 20. Other variations of this ratio are commonly applied, such as 70 : 30 [17]. As the splits define which structures the network is to be exposed to during training, these splits might affect the resulting predictions performed. As point clouds are unordered, and the variation of roof structures present in real-life areas are very diverse, there is a need for a large number of examples of roofs. For a dataset of this size, a split where only 70% of the complete dataset is used for training could result in poor performance. Some roof type classes already contain a minimum of structures, and reducing this number even further is not desirable. With a low number of roofs used for training, each structure is of more importance. Consequently, in these cases a well-balanced dataset, both between classes and with regards to variations within a class, is desirable. This is not possible with the current data in the TRD3DRoofs dataset.

Presently, all models proposed are tested on the same number of data but trained varying percentages of the training dataset. Other variations for the amount of data in the splits could yield better results, especially for more complex structures, as the network could in these cases have been exposed to these at a higher rate in the training. E.g. for the models trained on a low percentage of the training set, the test set will contain a large share of the data. The most important consideration when deciding split ratio, is to make sure that both the train and test datasets appropriately represents the problem domain.

Another possibility is to implement and adapt a more advanced approach to the splitting of the data. By analysing and evaluating the structures present in the dataset, each structure could individually be placed in the different splits based on a set of rules, resulting in a more optimal split.

5.3.6 Comparison of our Results with the Results from PointNet++

The creators of PointNet++ shows that their network can be used for semantic part segmentation by using a subset of the ShapeNet dataset [67]. Shapes represented by point clouds are taken as input and used to predict a part label for each point. Normal vectors are added to each point, to depict the underlying shape to a larger extent. For this dataset, state-of-the-art performance at the time is achieved with a mIoU of 85.1%. On the pro-

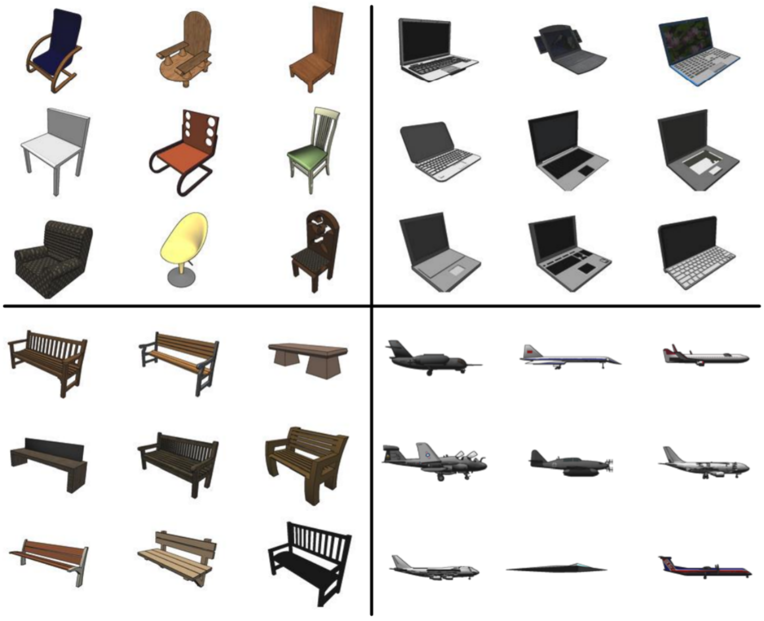


Figure 5.5: Examples of aligned models in the ShapeNet dataset. Figure origin: [67].

posed TRD3DRoofs dataset presentet in this thesis, PointNet++ achieves a mIoU of 79.4%.

Compared to our proposed TRD3DRoofs dataset, the ShapeNet Parts dataset is very different. ShapeNet [67] consists of 3D CAD models of objects labelled with a hybrid approach. Both algorithms and human effort are combined to perform the annotation of the models. 3D CAD models are contrary to our data not based on scans of real-world objects. As the data is artificially obtained, the quality of the point cloud is significantly superior. Density distributions in 3D CAD models will not contain the same degree of variety as a point cloud obtained through ALS. Another significant dissimilarity between the datasets is the orientation and annotation of the objects. For ShapeNet Parts, all objects are aligned with a rigid alignment strategy, making sure that every object has the same orientation. Examples of this can be found in Figure 5.5. This rigid alignment simplifies the training procedure, as the normal vectors of points will be consistent between different objects with regards to part label. This will however never be the case of data obtained in real-world scenarios, as these are prone to arbitrary rotations. Another benefit of the rigid alignment is that it simplifies the annotation process, making it easier to retain information about the relationship between neighbouring planes.

The predicted results and our manual evaluation indicate that the performance of PointNet++ with regards to part segmentation is dependent on the consistency of global features. When exposed to data with inconsistency in global features, such as the TRD3DRoofs dataset, the performance of the part segmentation drops. It appears that the network in this case still learns the geometric features of the local points. Nevertheless, it is not satisfyingly separating points belonging to distinct planes. To perform this separation, the global features of the complete point clouds need to be taken into consideration. The peformance results of PointNet++ for this application indicates that it in some cases are vulnerable to inconsistency in global features. This is remarkably evident when the network is trained on real-world data where arbitrary orientations of roof structures are present.

Based on the theory presented in section 2.3, the differences between our dataset consisting of real roof data and the syntetic data of ShapeNet Parts is significant. We therefore consider the obtained mIoU of 79.4 to be satisfying. This belief is further strengthened when comparing to the highest mIoU obtained on ScanNet[113], a dataset consisting of real indoor scenes. [66] reports that the best performing point based method, KPConv[75], achieves a mIoU of 68.8.

5.3.7 Choice of Neural Network

PointNet++ was selected as a basis used to test our datasets applicability for training a deep learning network, specifically for the task of part segmentation of roof structures. This method was deemed fitting because it fulfilled our requirements of directly processing a point cloud as input, the handling of varying data densities and achieves good results at the task of part segmentation. However, the main reason as to why it was preferred over other deep learning methods, such as the ones mentioned in 2.2.2, that also satisfy the necessities, was to consider the time and hardware constrains.

For example could the graph-based methods, specGCN [80] and LDGCNN [81], not be as affected by the plane labelling scheme as graph-based method are especially focused on capturing structural relations. Then again, DGCNN has a longer inference time, while achieving the same mIoU as PointNet++ on ShapeNet Parts Table 2.1. SpecGCN has a higher mIoU score, but the network is too complex for it to be practical considering our limitation.

On the other, could a CNN-based method produce good results, as several of these receive high marks at task of part segmentation. Both KPConv [75] and FG-net [78], but we deem these methods to be too new to be implemented by enough independent researchers and is therefore not as well-documented as PointNet++. This became a deciding factor, due to the time limitation. The best alternative would be PointCNN [73], as this method is less complex that PointNet++ whilst still achieving a higher mIoU score. The down side is that it is slow to converge during training.

5.3.8 Deep Learning vs Traditional Segmentation

In this thesis, we argue that a deep learning approach to the problem of roof plane segmentation of 3D point cloud data is a feasible option for a more general solution to the problem of segmentation. There is one major reason why neural networks are an improved alternative to traditional segmentation methods: the learning process.

Traditional segmentation methods have in common that they all need definitions of features and criteria to determine the similarity between points and detect objects [26]. The feature definition and selection for identifying planes is a challenging task. Humans instinctively recognize object around us that we earlier have been exposed to. However, it is not necessary possible to easily explain mathematically why we are able to recognize and categorize different objects. A varying unspecified number of factors affect this process. Thus, the task of correctly defining the features and criterions necessary to separate an arbitrary roof structure into distinctive planes is difficult. It is not possible to address all aspects that affect what is to be considered a roof plane.

This problem is solved in the case of deep learning. Artificial neural networks are inspired by the neurobiological basis of how the human brain is learning. The result of this novel field of research is advanced algorithms that, as the human brain, learn features based on the input they are exposed to. This means that no mathematical definition of features describing the roof planes is necessary. Resultingly, the neural network has the possibility of detecting roof planes outside of what is possible to define with mathematical features. This is a huge advantage in the case of segmentation of 3D point clouds. The use of deep learning-based approaches for the segmentation of roof planes in 3D point clouds can both improve the result of the segmentation, as well as reduce the time spent, by simplifying the process.

Chapter 6

Conclusion and Further work

In this chapter, we conclude our work in light of the established research questions. Limitations of the proposed work, including the dataset and the adapted network, are presented and discussed. Research that could further improve our work is highlighted and would be interesting to investigate in future work.

This thesis addresses the creation of suitable 3D point cloud data for deep learning techniques intended for roof plane segmentation needed for applications in typical Norwegian residential areas. The purpose of this thesis is to achieve this by answering the proposed research questions.

The establishment of guidelines for manual labeling of LiDAR data, which explains how data formatting and normalization are performed, can be seen as an answer to the first research question. The output is annotated ground truth data of point cloud roof structures, in a format suitable for deep learning algorithms. However, limitations are present regarding the labelling strategy established. Currently, no consistency between the labels and the normal vectors are present, and the result from the experiments indicates that this affects the predictions given. In addition, some indications show that maintaining a consistent relationship between adjacent planes across all roof types would be appropriate. This is however not consistently applied in this work. No optimal solution were found, suggesting that further research on this labelling strategy should be conducted. The second research question is answered in the guidelines proposed. In this step, multiple roof structures are separated, and rotation of the labels is performed to increase the amount of data present. This augmentation of the data, along with the formatting, results in a dataset large enough for deep learning techniques.

The result of this thesis is two new datasets, TRD3DRoofs and Augmented TRD3DRoofs. These datasets consist of manually labelled ground truth data of roof structures. Two separate datasets are established to satisfy the need for both real-life data, and a large enough amount of training data for data-hungry networks. Further, by conducting experiments, the usability of the proposed augmented version of the dataset is tested in a well-recognized neural network for point cloud data, PointNet++. Evaluation of both the usability of the Augmented TRD3DRoofs dataset in a deep neural network and the usability of PointNet++ for plane segmentation of real-world roof structures are conducted.

Based on the presented evaluation and reasoning proposed in the discussion in chapter 5, it is reasonable to assume that the results of the roof plane segmentation are sufficient for the construction of 3D models of roof structures. Nevertheless, to achieve desirable results along ridges of the roofs, a need for rule-based post-processing methods for the segmentation emerges. It should also be noted that the results are dependent on the area of obtainment and hardware used, and that manual estimations are the basis of parts of the evaluation performed.

We conclude that the proposed Augmented TRD3DRoofs dataset achieves promising results for roof plane segmentation when utilised as training data for the deep neural network PointNet++. It is reasonable to conclude that the 3D models that may be constructed based on the TRD3DRoofs dataset and the segmentation performed by PointNet++, is applicable for renewable energy applications in residential areas of Trondheim. This indicates that a deep learning approach might be a solution to the lack of a general method for segmentation of complex roof structures, but further research is necessary to confirm this indication.

This thesis distinguishes itself from earlier work by establishing ground truth 3D point cloud data labelled with semantic information about roof planes, not only roof structures. In addition, the proposed dataset has a point density similar to that obtained in most projects in surveying and mapping, opposed to other datasets. Furthermore, the roof structures present are typical for Norwegian residential areas, and to the best of our knowledge, no such dataset already exists. The use of neural networks on 3D data is a challenging task, and not much research has yet been conducted.

Our work points out the usability of the neural network PointNet++ used for the task of 3D roof plane segmentation. Factors regarding the network trained on real-world data where arbitrary real-life orientations of the roof structures are present, are discussed. We find this important to highlight for future improvements and research on point-wise deep learning algorithms, such as PointNet++.

6.1 Limitations

Based on the experiment performed on the established dataset, it was discovered certain limitations regarding the methodology of the project and the resulting TRD3DRoofs dataset. In addition, limitations of PointNet++ trained on our proposed dataset are also present. These limitations will be presented and discussed in this final section.

6.1.1 TRD3DRoofs

Datasets established from real-world LiDAR data are heavily dependent on the quality of the raw point cloud. The density of the point cloud depends on the surrounding conditions during the time of obtainment and will consequently lead to a varying point density for the roofs. Point density is a feature that might affect the prediction result, so a varying density is not optimal [68]. This limitation related to variations in density will be present both in the original and augmented dataset.

Few complicated roof structures are present in the ground truth data, leading to a lack of training data for such structures in the original dataset. These complex roof structures are however present both in the Trondheim area and other Norwegian residential areas. If the trained network is exposed to these roofs, it will provide poor prediction results not applicable for the desired applications. The validity of the original dataset is limited by both the amount of roofs, and the variation of the roofs present. The dataset is prone to a skew related to categories, as it is based on only real-world roof structures. In real-life residential areas, there is a natural imbalance based on the commonality of different roof types. Additionally, due to the cost and time needed for manual annotation, the original dataset is not large enough to be suitable as input for data-hungry networks [49].

These limitations could be overcome by additional segmentation of real-world roof structures, and the augmented TRD3DRoofs improves these weaknesses to some extent. The augmented version have a smaller skew, and a larger amount of roofs. However, issues regarding the consistency of the plane labelling are introduced. As discussed, there is a desire for consistency between adjacent planes across all roof types in the augmented dataset. Normal vectors are also added to all points as an additional feature. Their contribution is intended to support the learning process, but as these normal vectors are not consistent with regards to plane label, the labelling strategy is not ideal. Theses annotations issues are a major limitation of the current Augmented TRD3DRoofs dataset. No solution was discovered for this problem during the work of this master thesis, and further research will be necessary to improve the labelling strategy.

6.1.2 PointNet++

Based on how the metrics are calculated in PointNet++ and how the data of the proposed Augmented TRD3DRoofs is structured, the network will in theory only have a 50% chance of separating planes of a geometric class correctly (except for ladder shaped planes, where the chance is 25%). This is due to the problem highlighted in section 5.1.

This limitation indicates that no metric given by the implementation of PointNet++ is suitable for presenting the performance of the models correctly. This creates a need for a visual evaluation to determine the quality of the performance. These results will only be estimates, as they are performed manually.

6.2 Further Work

In this section, further work is proposed that addresses current limitations and which can further improve the work of this thesis. This includes the collection and processing of additional data, examination of possible post-processing methods and investigation of the usability of the dataset for training other neural network models.

The original dataset consists of approximately 900 manually segmented roof structures. For the complicated task of training neural networks on 3D point clouds, this is not enough as an excessive amount of data is needed. Additionally, as the data is only collected in the Trondheim area, the validity of the results are limited based on the roof structures present here. Further work would therefore involve the collection, segmentation, and processing of additional data obtained both in the Trondheim area and other Norwegian cities. This would also include the addition of more complex structures into the dataset, to make the model able to predict a broader type of roof structures present in Norwegian residential areas. Together, these further contributions would increase the validity of the dataset beyond that of today. Additional data augmentation, such as resizing of the points, would further increase the amount of data present, possibly resulting in an even more appropriate dataset for deep learning purposes.

The current manual labelling strategy includes limitations regarding consistency between the plane label and the normal vector, and consistency between adjacent planes. The establishment of a suitable labelling scheme is challenging, as variations in roof structures of all types complicate the task of consistency. A study where different approaches are tested and evaluated is desirable, but such a task is time-consuming and in need of large amounts of manual labour, going beyond the scope of this thesis. However, this is an interesting task for future research.

Currently, the resulting predictions include errors of varying degrees, primarily with regards to separation of distinctive planes. Future research would include the establishment of rule-based post-processing methods meant to eliminate such errors. These methods would be applied to the roof planes where certain points are incorrectly segmented, or the separation of distinct planes is at flaw. By establishing post-processing methods errors could be corrected based on a set of established rules, and the "cleaning" of the predicted data would result in an improved result of the segmentation.

Further optimization of the network hyperparameters is necessary to improve the training process of PointNet++ on real-world data, as such data is significantly different from the synthetic data it has been tested on in the past. It would be interesting to see the result of the predictions if this were to be done. It would also be of interest to test other possibilities for calculating the normal vector, for example using different values of k or letting the network learn this value, to see if that could improve the results on the edge cases.

Features that are present for each point in the dataset at its current state is coordinates, normal vector, roof type, and plane label. In the future, further investigation of additional features would be interesting. Combining the point cloud with Red Green Blue (RGB) colour data have been successful in related work on 3D point clouds [11] [86]. This indicates that combining the existing point cloud with UAV images of the same area could provide improved results and is a topic relevant for further work on the dataset.

One aspect that would be interesting to investigate further, is the usability of TRD3DRoofs in training other deep neural networks. It seems like the dataset has limitations regarding the consistency of the relationship between adjacent planes and the given labels when utilised in PointNet++. PointNet++ uses layers that imitates convolution, but this may not be optimal. It would be interesting to adopt a network using CNNs, to see if this would yield better results, indicating a more suitable technique for the proposed task of roof plane segmentation of real-world buildings. Pre-processing of the point cloud, where clustering is applied to separate the point cloud into clusters could resolve the problem of plane segmentation present when utilizing PointNet++. This is not optimal, as it complicates the segmentation process. From a mathematical perspective, the problem of clustering can be regarded as a graph-based optimization problem [14]. Several graph-based deep learning methods have been developed for point cloud processing, and these could prove to be more suitable for the proposed dataset. In further work, it would be interesting to investigate the suitability of TRD3DRoofs in graph-based networks such as PointGCR [114].

Bibliography

[1] N. Lu, J. Zhou, Z. Han, D. Li, Q. Cao, X. Yao, Y. Tian, Y. Zhu, W. Cao and T. Cheng, 'Improved estimation of aboveground biomass in wheat from RGB imagery and point cloud data acquired with a low-cost unmanned aerial vehicle system,' *Plant Methods*, 2019. DOI: 10.1186/s13007-019-0402-3.

[2] P. Narksri, E. Takeuchi, Y. Ninomiya, Y. Morales, N. Akai and N. Kawaguchi, 'A Slope-robust Cascaded Ground Segmentation in 3D Point Cloud for Autonomous Vehicles,' *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018. DOI: 10.1109/ITSC.2018.8569534.

[3] S. Malihi, M. J. Valadan Zoej, M. Hahn, M. Mokhtarzade and H. Arefi, '3D building reconstruction using dense photogrammetric point cloud,' 2016. DOI: 10.5194/isprsarchives-XLI-B3-71-2016.

[4] *Unfccc - united nations framework convention on climate change: The paris agreement*. [Online]. Available: https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement (visited on 27/05/2021).

[5] (). 'Irena - international renewable energy agency: Climate change,' [Online]. Available: https://www.irena.org/climatechange (visited on 27/05/2021).

[6] M. S. Wong, R. Zhu, Z. Liu, L. Lu, J. Peng, Z. Tang, C. H. Lo and W. K. Chan, 'Estimation of Hong Kong's solar energy potential using GIS and remote sensing technologies,' *Renewable Energy*, 2016. DOI: 10.1016/j.renene.2016.07.003.

[7] A. Sampath and J. Shan, 'Building Roof Segmentation and Reconstruction from LiDAR Point Clouds Using Clustering Techniques,' *IAPRS International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences*, 2008, ISSN: 16821750.

[8] D. Kong, L. Xu and X. Li, 'A new method for building roof segmentation from airborne LiDAR point cloud data,' *Measurement Science and Technology*, 2013. DOI: 10.1088/0957-0233/24/9/095402.

[9] S. A. N. Gilani, M. Awrangjeb and G. Lu, 'Segmentation of airborne point cloud data for automatic building roof extraction,' *GIScience and Remote Sensing*, 2018. DOI: 10.1080/15481603.2017.1361509.

[10] E. Goceri, 'Challenges and Recent Solutions for Image Segmentation in the Era of Deep Learning,' *2019 9th International Conference on Image Processing Theory, Tools and Applications, IPTA 2019*, 2019. DOI: 10.1109/IPTA.2019.8936087.

[11] Q. Hu, B. Ang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni and A. Markham, 'RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds,' Tech. Rep., 2020, pp. 11 108–11 117.

[12] H. Fan. (2018). 'Uav and laser scanning - tba4231 applied geomatics,' [Online]. Available: https://ntnu.blackboard.com/. (accessed: 19.04.2021).

[13] A. Kaiser, J. A. Ybanez Zepeda and T. Boubekeur, 'A Survey of Simple Geometric Primitives Detection Methods for Captured 3D Data,' *Computer Graphics Forum*, vol. 38, no. 1, pp. 167–196, 2019. DOI: 10.1111/cgf.13451.

[14] Y. Xie, J. TIAN and X. X. Zhu, 'Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation,' *IEEE Geoscience and Remote Sensing Magazine*, pp. 1–20, 2020. DOI: 10.1109/MGRS.2019.2937630.

[15] M. Kolle, D. Laupheimer, S. Schmohl, N. Haala, F. Rottensteiner, J. D. Wegner and H. Ledoux, 'The Hessigheim 3D (H3D) Benchmark on Semantic Segmentation of High-Resolution 3D Point Clouds and Textured Meshes from UAV LiDAR and Multi-View-Stereo,' *ISPRS Open Journal of Photogr. and Rem. Sens.*, 2021.

[16] S. M. I. Zolanvari, S. Ruano, A. Rana, C. Alan, R. E. da Siliva, M. Rahbar and A. Smolic, 'DublinCity : Annotated LiDAR Point Cloud and its Applications,' 2019.

[17] N. Varney, V. K. Asari and Q. Graehling, 'DALES : A Large-scale Aerial LiDAR Data Set for Semantic Segmentation,' *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) DALES:*, 2020. DOI: 10.1109/CVPRW50498.2020.00101.

[18] S. E. Reutebuch, H.-e. Andersen and R. J. Mcgaughey, 'Light Detection and Ranging (LIDAR): An Emerging Tool for Multiple Resource Inventory,' *Journal of Forestry*, 2005.

[19] H. Fan. (2019). 'Laser scanning i - tba4236 theoretical geomatics,' [Online]. Available: `https://ntnu.blackboard.com/`. (accessed: 10.04.2021).

[20] E. Naesset, 'Determination of mean tree height of forest stands using airborne laser scanner data,' *ISPRS Journal of Photogrammetry and Remote Sensing*, 1997.

[21] S. Xu, G. Vosselman and S. O. Elberink, 'Multiple-entity based classification of airborne laser scanning data in urban areas,' 2013. DOI: `10.1016/j.isprsjprs.2013.11.008`.

[22] P. Packalen, J. Strunk, T. Packalen, M. Maltamo and L. Mehtätalo, 'Resolution dependence in an area-based approach to forest inventory with airborne laser scanning,' *Remote Sensing of Environment*, 2019. DOI: `10.1016/j.rse.2019.01.022`.

[23] M. Morgan and K. Tempfli, 'Automatic building extraction from airborne laser scanning data,' *International Archives of Photogrammetry and Remote Sensing.*, 2000.

[24] D. Chen, R. Wang and J. Peethambaran, 'Topologically Aware Building Rooftop Reconstruction From Airborne Laser Scanning Point Clouds,' *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, 2017.

[25] K. Kim and J. Shan, 'Building roof modeling from airborne laser scanning data based on level set approach,' *ISPRS Journal of Photogrammetry and Remote Sensing*, 2011. DOI: `10.1016/j.isprsjprs.2011.02.007`.

[26] H. Fan. (2020). 'Segmentation - tba4256 3d digital modelling,' [Online]. Available: `https://ntnu.blackboard.com/`. (accessed: 10.04.2021).

[27] A. Nguyen and B. Le, '3D point cloud segmentation: A survey,' *IEEE Conference on Robotics, Automation and Mechatronics, RAM - Proceedings*, pp. 225–230, 2013. DOI: `10.1109/RAM.2013.6758588`.

[28] X. Y. Jiang, U. Meier and H. Bunke, 'Fast range image segmentation using high-level segmentation primitives,' *IEEE Workshop on Applications of Computer Vision - Proceedings*, 1996. DOI: `10.1109/acv.1996.572006`.

[29] B. Xiong, M. Jancosek, S. Oude Elberink and G. Vosselman, 'Flexible building primitives for 3D building modeling,' *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 101, pp. 275–290, 2015. DOI: 10.1016/j.isprsjprs.2015.01.002.

[30] M. A. Fischler and R. C. Bolles, 'Random Sample Paradigm for Model Consensus: A Apphcatlons to Image Fitting with Analysis and Automated Cartography,' *Graphics and Image Processing*, vol. 24, no. 6, pp. 381–395, 1981.

[31] T. M. Awwad, Q. Zhu, Z. DU and Y. Zhang, 'AN IMPROVED SEGMENTATION APPROACH FOR PlANAR SURFACES FROM UNSTRUCTURED 3D POINT CLOUDS,' *The Photogrammetric Record*, 2010.

[32] B. Xu, W. Jiang, J. Shan, J. Zhang and L. Li, 'Investigation on the weighted RANSAC approaches for building roof plane segmentation from LiDAR point clouds,' *Remote Sensing*, vol. 8, no. 1, p. 5, Dec. 2016. DOI: 10.3390/rs8010005.

[33] C. Wang, M. Ji, J. Wang, W. Wen, T. Li and Y. Sun, 'An improved DBSCAN method for LiDAR data segmentation with automatic Eps estimation,' *Sensors (Switzerland)*, vol. 19, no. 1, Jan. 2019. DOI: 10.3390/s19010172.

[34] H. Fan. (2020). 'Clustering - tba4256 3d digital modelling,' [Online]. Available: https://ntnu.blackboard.com/. (accessed: 03.04.2021).

[35] L. V. Vilson, P. S. Excell and R. J. Green, 'A Generalisation of the fuzzy c-means clustering algorithm,' 1988.

[36] J. Macqueen, 'SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS,' 1967.

[37] A. Sampath and J. Shan, 'Segmentation and Reconstruction of Polyhedral Building Roofs From Aerial Lidar Point Clouds,' *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, 2010.

[38] M. Awrangjeb and G. Lu, 'Building roof plane extraction from LIDAR data,' *2013 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2013*, 2013. DOI: 10.1109/DICTA.2013.6691490.

[39] R. Albano, 'Investigation on Roof Segmentation for 3D Building Reconstruction from Aerial LIDAR Point Clouds,' *applied sciences*, 2019.

[40] A. V. Vo, L. Truong-Hong, D. F. Laefer and M. Bertolotto, 'Octree-based region growing for point cloud segmentation,' *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 104, pp. 88–100, 2015. DOI: 10.1016/j.isprsjprs.2015.01.011.

[41] Y. Xu, W. Yao, L. Hoegner and U. Stilla, 'Segmentation of building roofs from airborne LiDAR point clouds using robust voxel-based region growing,' *Remote Sensing Letters*, 2017. DOI: 10.1080/2150704X.2017.1349961.

[42] S. Sun and C. Salvaggio, 'Aerial 3D Building Detection and Modeling From Airborne LiDAR Point Clouds,' *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING*, 2013.

[43] J. Shaou, W. Zhang, A. Shen, N. Mellando, S. Cai, L. Luo, N. Wang, G. Yan and G. Zhou, 'Seed point set-based building roof extraction from airborne LiDAR point clouds using a top-down strategy,' *Automation in Construction*, 2021. DOI: 10.1016/j.autcon.2021.103660.

[44] B. Marr, *These 25 Technology Trends Will Define The Next Decade*, 2020. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2020/04/20/these-25-technology-trends-will-define-the-next-decade/?sh=ddcdf9829e3b (visited on 15/12/2020).

[45] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu and F. E. Alsaadi, 'A survey of deep neural network architectures and their applications,' *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017. DOI: 10.1016/j.neucom.2016.12.038.

[46] J. Patterson and A. Gibson, *Deep Learning: A Practitioner's Approach*. 2017, pp. 7–7.

[47] A. L'Heureux, K. Grolinger, H. F. Elyamany and M. A. Capretz, 'Machine Learning with Big Data: Challenges and Approaches,' *IEEE Access*, vol. 5, pp. 7776–7797, 2017. DOI: 10.1109/ACCESS.2017.2696365.

[48] V. Buhrmester, D. Münch and M. Arens, 'Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey,' 2019.

[49] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler and M. Pollefeys, 'Semantic3D.Net: a New Large-Scale Point Cloud Classification Benchmark,' *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017. DOI: `10.5194/isprs-annals-IV-1-W1-91-2017`.

[50] A. Ioannidou, E. Chatzilari, S. Nikolopoulos and I. Kompatsiaris, 'Deep learning advances in computer vision with 3D data: A survey,' *ACM Computing Surveys*, vol. 50, no. 2, 2017. DOI: `10.1145/3042064`.

[51] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl and R. Bardenet, 'Algorithms for Hyper-Parameter Opti-mization,' Tech. Rep., Dec. 2011. [Online]. Available: `https://hal.inria.fr/hal-00642998`.

[52] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, 'ImageNet: A large-scale hierarchical image database,' pp. 248–255, 2009. DOI: `10.1109/cvprw.2009.5206848`.

[53] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, 'Vision meets robotics: The KITTI dataset,' *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013. DOI: `10.1177/0278364913491297`.

[54] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, 'Microsoft COCO: Common objects in context,' in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, Springer Verlag, 2014, pp. 740–755. DOI: `10.1007/978-3-319-10602-1_48`.

[55] M. Everingham, S. M. Eslami, L. Van Gool, C. K. Williams, J. Winn and A. Zisserman, 'The Pascal Visual Object Classes Challenge: A Retrospective,' *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015. DOI: `10.1007/s11263-014-0733-5`.

[56] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth and B. Schiele, 'The Cityscapes Dataset for Semantic Urban Scene Understanding,' in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 3213–3223, ISBN: 9781467388504. DOI: `10.1109/CVPR.2016.350`.

[57] A. Krizhevsky, I. Sutskever and G. Hinton, 'Imagenet classification with deep convolutional neural networks,' *Advances in neural information processing systems*, 2012.

[58] K. Simonyan and A. Zisserman, 'Very deep convolutional networks for large-scale image recognition,' in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, International Conference on Learning Representations, ICLR, Sep. 2015.

[59] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, 'Going Deeper with Convolutions,' Tech. Rep., 2015, pp. 1–9.

[60] J. Long, E. Shelhamer and T. Darrell, 'Fully Convolutional Networks for Semantic Segmentation,' Tech. Rep., 2015, pp. 3431–3440.

[61] K. He, X. Zhang, S. Ren and J. Sun, 'Deep residual learning for image recognition,' in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 770–778, ISBN: 9781467388504. DOI: `10.1109/CVPR.2016.90`.

[62] C. R. Qi, H. Su, K. Mo and L. J. Guibas, 'PointNet: Deep learning on point sets for 3D classification and segmentation,' *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 77–85, 2017. DOI: `10.1109/CVPR.2017.16`.

[63] L. Tchapmi, C. Choy, I. Armeni, J. Gwak and S. Savarese, 'SEGCloud: Semantic segmentation of 3D point clouds,' *Proceedings - 2017 International Conference on 3D Vision, 3DV 2017*, pp. 537–547, 2018. DOI: `10.1109/3DV.2017.00067`.

[64] H. Y. Meng, L. Gao, Y. K. Lai and D. Manocha, 'VV-NET: Voxel VAE net with group convolutions for point cloud segmentation,' *arXiv*, pp. 8500–8508, 2018.

[65] R. A. Rosu, P. Schütt, J. Quenzel and S. Behnke, 'LatticeNet: Fast Point Cloud Segmentation Using Permutohedral Lattices,' 2019.

[66] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu and M. Bennamoun, 'Deep learning for 3D point clouds: A survey,' *arXiv*, pp. 1–27, 2020. DOI: `10.1109/tpami.2020.3005434`.

[67] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi and F. Yu, 'ShapeNet: An Information-Rich 3D Model Repository,' Dec. 2015.

[68] C. Qi, L. Yi, H. Su and L. Guibas, 'PointNet++: Deep Hierarchical Feature Learning on,' *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, no. Dec, pp. 5105–5114, 2017.

[69] M. Jiang, Y. Wu, T. Zhao, Z. Zhao and C. Lu, 'PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation,' 2018.

[70] W. Wang and R. Yu, 'SGPN : Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation University of California , San Diego,' *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2569–2578, 2018.

[71] L. Z. Chen, X. Y. Li, D. P. Fan, K. Wang, S. P. Lu and M. M. Cheng, 'LSANet: Feature learning on point sets by local spatial aware layer,' *arXiv*, 2019.

[72] S. Khan, H. Rahmani, S. A. A. Shah and M. Bennamoun, 'A Guide to Convolutional Neural Networks for Computer Vision,' *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1–207, Feb. 2018. DOI: `10.2200/s00822ed1v01y201712cov015`.

[73] Y. Li, R. Bu and X. Di, 'PointCNN : Convolution On X -Transformed Points,' no. NeurIPS, 2018.

[74] Z. Zhang, B. S. Hua and S. K. Yeung, 'ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics,' *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, pp. 1607–1616, 2019. DOI: `10.1109/ICCV.2019.00169`.

[75] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette and L. J. Guibas, 'KPConv: Flexible and Deformable Convolution for Point Clouds,' Tech. Rep., 2019, pp. 6411–6420.

[76] A. Boulch, 'ConvPoint: Continuous convolutions for point cloud processing,' *Computers and Graphics (Pergamon)*, vol. 88, pp. 24–34, May 2020. DOI: `10.1016/j.cag.2020.02.005`.

[77] *ShapeNet-Part Benchmark (3D Part Segmentation) | Papers With Code*. [Online]. Available: `https://paperswithcode.com/sota/3d-part-segmentation-on-shapenet-part` (visited on 06/06/2021).

[78] K. Liu, Z. Gao, F. Lin and B. M. Chen, 'FG-Net: Fast Large-Scale LiDAR Point CloudsUnderstanding Network Leveraging CorrelatedFeature Mining and Geometric-Aware Modelling,' 2020.

[79] L. Landrieu and M. Simonovsky, 'Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs,' Tech. Rep., 2017, pp. 4558–4567.

[80] C. Wang, B. Samari and K. Siddiqi, 'Local Spectral Graph Convolution for Point Set Feature Learning,' *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11208 LNCS, pp. 56–71, 2018. DOI: `10.1007/978-3-030-01225-0_4`.

[81] K. Zhang, M. Hao, J. Wang, C. W. de Silva and C. Fu, 'Linked Dynamic Graph CNN: Learning on Point Cloud via Linking Hierarchical Features,' 2019.

[82] M. A. Uy, Q. H. Pham, B. S. Hua, T. Nguyen and S. K. Yeung, 'Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data,' *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, no. Iccv, pp. 1588–1597, 2019. DOI: `10.1109/ICCV.2019.00167`.

[83] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey and J. M. Reynolds, ''Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications,' *Geomorphology*, 2012. DOI: `10.1016/j.geomorph.2012.08.021`.

[84] M. Bosch, Z. Kurtz, S. Hagstrom and M. Brown, 'A multiple view stereo benchmark for satellite imagery,' *Proceedings - Applied Imagery Pattern Recognition Workshop*, 2017. DOI: `10.1109/AIPR.2016.8010543`.

[85] Y. Furukawa and J. Ponce, 'Accurate, dense, and robust multiview stereopsis,' *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2010. DOI: `10.1109/TPAMI.2009.161`.

[86] X. Li, C. Li, Z. Tong, A. Lim, J. Yuan, Y. Wu, J. Tang and R. Huang, 'Campus3D: A Photogrammetry Point Cloud Benchmark for Hierarchical Understanding of Outdoor Scene,' 2020. DOI: `10.1145/3394171.3413661`.

[87] Q. Hu, B. Yang, S. Khalid, W. Xiao, N. Trigoni and A. Markham, 'Towards Semantic Segmentation of Urban-Scale 3D Point Clouds: A Dataset, Benchmarks and Challenges,' 2021. arXiv: `2009.03137`.

[88] J. Xiao, J. Zhang, B. Adler, H. Zhang and J. Zhang, 'Three-dimensional point cloud plane segmentation in both structured and unstructured environments,' *Robotics and Autonomous Systems,* 2013. DOI: `10.1016/j.robot.2013.07.001`.

[89] S. Nikoohemat, M. Peter, S. O. Elberink and G. Vosselman, 'Semantic interpretation of mobile laser scanner point clouds in Indoor Scenes using trajectories,' *Remote Sensing*, vol. 10, no. 11, pp. 18–22, 2018. DOI: `10.3390/rs10111754`.

[90] X. Roynard, J. E. Deschaud and F. Goulette, 'Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification,' *International Journal of Robotics Research*, 2018.

[91] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss and J. Gall, 'SemanticKITTI,' *Iccv*, no. iii, 2019.

[92] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez and U. Breitkopf, 'THE ISPRS BENCHMARK on URBAN OBJECT CLASSIFICATION and 3D BUILDING RECONSTRUCTION,' *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, DOI: `10.5194/isprsannals-I-3-293-2012`.

[93] I. I. S. for Photogrammetry and R. Sensing. (). 'Isprs benchmarks,' [Online]. Available: `https://www.isprs.org/education/benchmarks.aspx`. (accessed: 07.05.2021).

[94] Z. Ye, Y. Xu, R. Huang, X. Tong, X. Li, X. Liu, K. Luan, L. Hoegner and U. Stilla, 'LASDU: A large-scale aerial LiDAR dataset for semantic labeling in dense urban areas,' *ISPRS International Journal of Geo-Information*, 2020. DOI: `10.3390/ijgi9070450`.

[95] A. Wichmann, A. Agoub and M. Kada, 'ROOFN3D: Deep learning training data for 3D building reconstruction,' *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 2018. DOI: `10.5194/isprs-archives-XLII-2-1191-2018`.

[96] D. Girardeau-Montaut. (). 'Cloudcompare,' [Online]. Available: `http://www.cloudcompare.org/`. (accessed: 18.03.2021).

[97] (). 'Arcgis pro - the world's leading gis software,' [Online]. Available: `https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview`. (accessed: 09.06.2021).

[98] M. Kada, 'Scale-dependent simplification of 3D building models based on cell decomposition and primitive instancing,' 2007. DOI: `10.1007/978-3-540-74788-8_14`.

[99] Y. Shirai and M. Suva, 'RECOGNITION OF POLYHEDRONS WITH A RANGE FINDER,' *Electrotechnica l Laboratory Tokyo, Japan*, no. 3, pp. 80–87, 1972.

[100] Z. Wu, R. Shou, Y. Wang and X. Liu, 'Interactive shape co-segmentation via label propagation,' *Computers and Graphics*, 2014. DOI: 10.1016/j.cag.2013.11.009.

[101] J. M. Johnson and T. M. Khoshgoftaar, 'Survey on deep learning with class imbalance,' *Journal of Big Data*, 2019. DOI: 10.1186/s40537-019-0192-5.

[102] X. Huang, C. Weng, Q. Lu, T. Feng and L. Zhang, 'Automatic labelling and selection of training samples for high-resolution remote sensing image classification over urban areas,' *Remote Sensing*, no. 1, 2015. DOI: 10.3390/rs71215819.

[103] N. L. Alchapar and E. N. Correa, 'Optothermal properties of façade coatings. Effects of environmental exposure over solar reflective index,' *Journal of Building Engineering*, 2020. DOI: 10.1016/j.jobe.2020.101536.

[104] *Gråtone (UTM33)*. [Online]. Available: https://geodataonline.maps.arcgis.com/apps/Embed/index.html?webmap=f7a6927a01cc46d59a279facc84b4556&extent=10.9532,59.9265,11.0982,59.9765&zoom=true&scale=false&disable_scroll=false&theme=light (visited on 08/06/2021).

[105] (). 'Pytorch - from research to production,' [Online]. Available: https://pytorch.org/. (accessed: 09.06.2021).

[106] R. Johns, *PyTorch vs Tensorflow for Your Python Deep Learning Project*, 2020. [Online]. Available: https://realpython.com/pytorch-vs-tensorflow/#pytorch-vs-tensorflow-decision-guide%20https://realpython.com/pytorch-vs-tensorflow/ (visited on 29/05/2021).

[107] X. Yan, 'Pointnet/pointnet++ pytorch,' *https://github.com/yanx27/Pointnet*$_{p}ointnet2_{p}ytorch$, 2019.

[108] (). 'Rapidlasso gmbh - fast tools to catch reality: Lastools,' [Online]. Available: https://rapidlasso.com/lastools/. (accessed: 09.06.2021).

[109] (). 'Meshlab,' [Online]. Available: https://www.meshlab.net/. (accessed: 09.06.2021).

[110] L. Prechelt, 'Early Stopping - But When?' In, Springer, Berlin, Heidelberg, 1998, pp. 55–69. DOI: 10.1007/3-540-49430-8_3.

[111] R. Pohle-Fröhlich, A. Bohm, P. Ueberholz, M. Korb and S. Goebbels, 'Roof segmentation based on deep neural networks,' *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 4, pp. 326–333, 2019. DOI: 10.5220/0007343803260333.

[112] B. Yang and Z. Dong, 'A shape-based segmentation method for mobile laser scanning point clouds,' *ISPRS Journal of Photogrammetry and Remote Sensing*, 2013. DOI: 10.1016/j.isprsjprs.2013.04.002.

[113] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser and M. Nießner, 'ScanNet: Richly-annotated 3D reconstructions of indoor scenes,' in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 2432–2443. DOI: 10.1109/CVPR.2017.261.

[114] Y. Ma, Y. Guo, H. Liu, Y. Lei and G. Wen, 'Global context reasoning for semantic segmentation of 3D point clouds,' *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision*, 2020. DOI: 10.1109/WACV45572.2020.9093411.

Appendix A

Train, test and validation splits for all models

Table A.1: Train, validation and test splits for all models. For SSG_100, the split is identical to the split used for MSG_100.

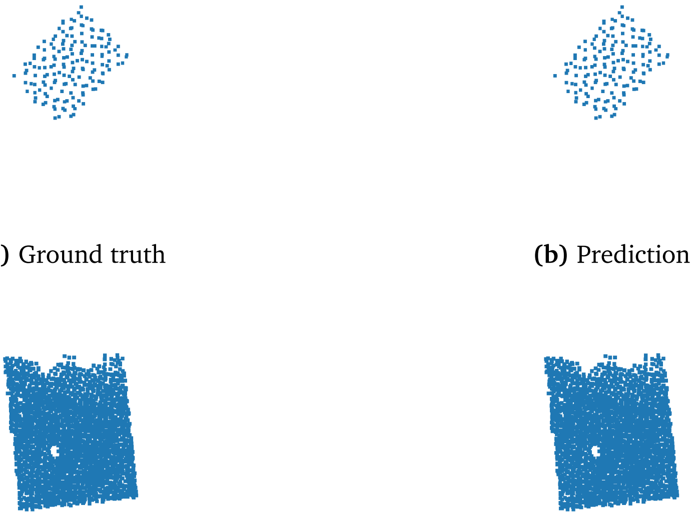
Roof type	Model	Training	Validation	Testing
1	MSG_100	89	11	11
	MSG_80	71	9	11
	MSG_60	53	7	11
	MSG_40	36	4	11
	MSG_20	18	2	11
2	MSG_100	412	51	51
	MSG_80	330	41	51
	MSG_60	247	31	51
	MSG_40	165	20	51
	MSG_20	82	10	51
3	MSG_100	208	26	26
	MSG_80	166	21	26
	MSG_60	125	16	26
	MSG_40	83	10	26
	MSG_20	42	5	26
4	MSG_100	253	32	31
	MSG_80	202	26	31
	MSG_60	152	19	31
	MSG_40	101	13	31
	MSG_20	51	6	31
5	MSG_100	302	38	38
	MSG_80	242	30	38
	MSG_60	181	23	38
	MSG_40	121	15	38
	MSG_20	60	8	38
6	MSG_100	288	36	36
	MSG_80	230	29	36
	MSG_60	173	22	36
	MSG_40	115	14	36
	MSG_20	58	7	36
7	MSG_100	446	56	56
	MSG_80	357	45	56
	MSG_60	268	34	56
	MSG_40	178	22	56
	MSG_20	89	11	56

Appendix B

Visual results MSG_80

In this appendix, additional visuals for the predictions done by the MSG_80 model are shown. The results are separated by roof type, and 9 examples are given for each type.

B.1 Type 1: *Flat*

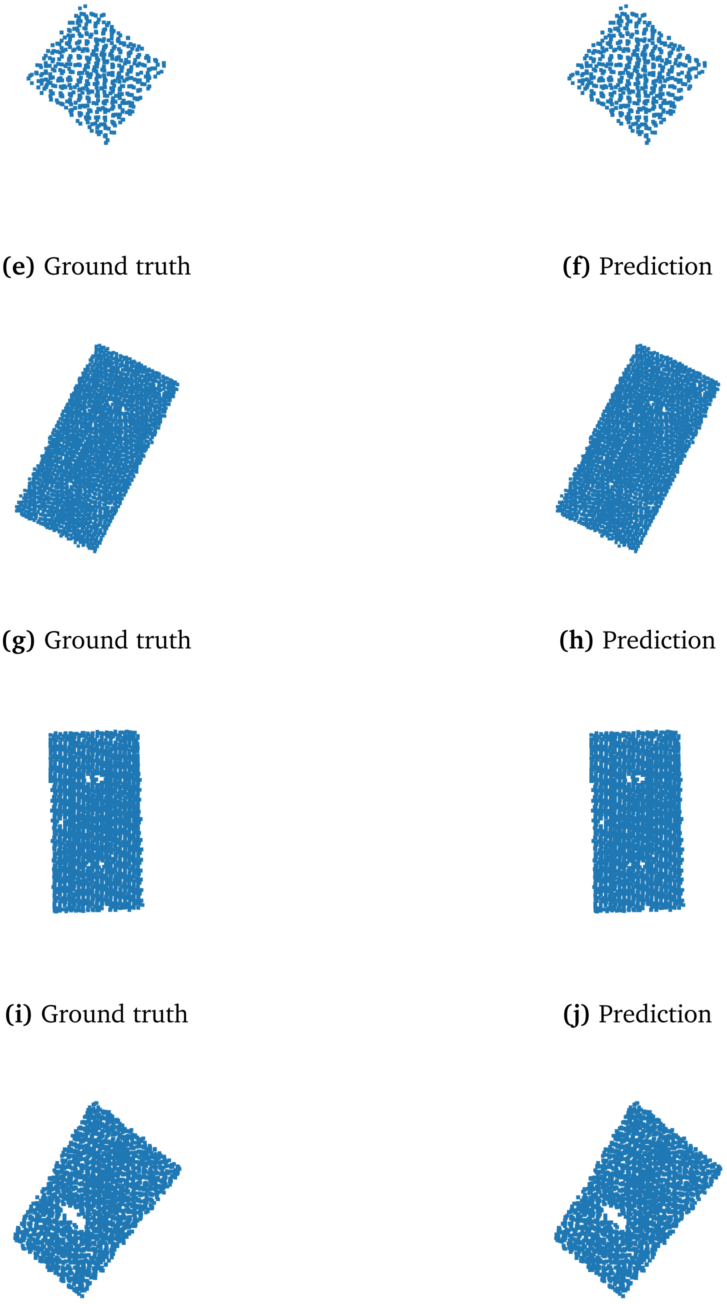


(a) Ground truth

(b) Prediction

(c) Ground truth

(d) Prediction



(e) Ground truth

(f) Prediction

(g) Ground truth

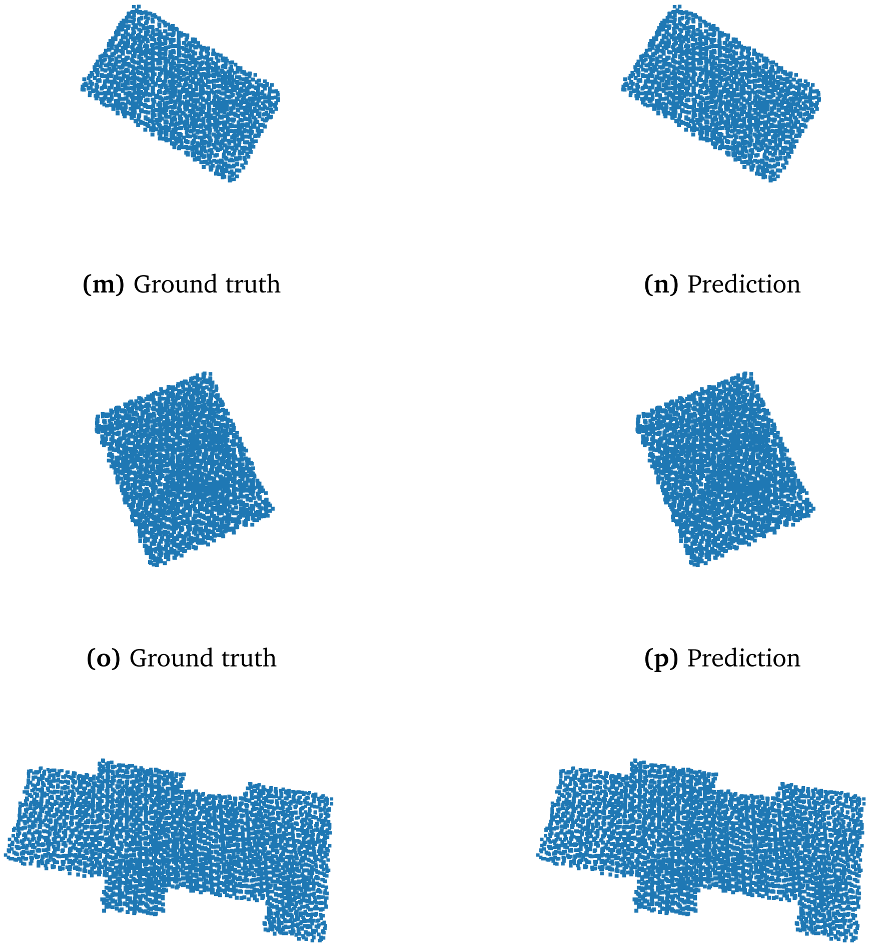
(h) Prediction

(i) Ground truth

(j) Prediction

(k) Ground truth

(l) Prediction



(m) Ground truth

(n) Prediction

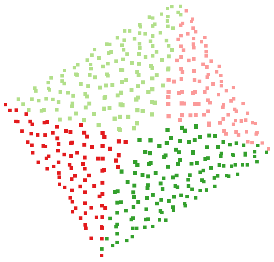
(o) Ground truth

(p) Prediction

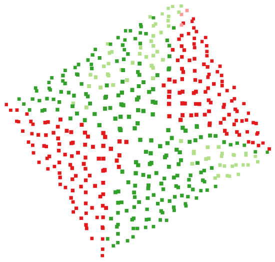
(q) Ground truth

(r) Prediction

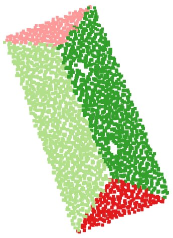
B.2 Type 2: *Hipped*



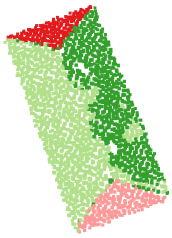
(a) Ground truth



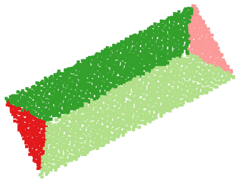
(b) Prediction



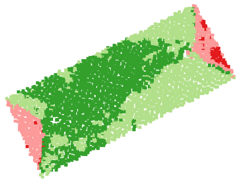
(c) Ground truth



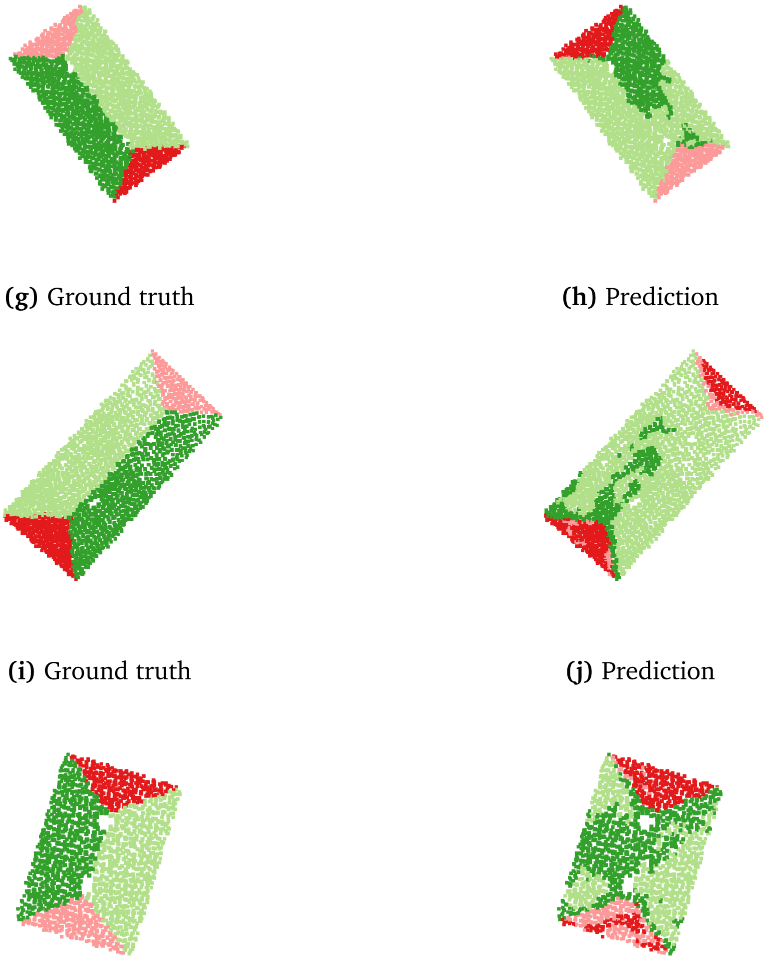
(d) Prediction



(e) Ground truth



(f) Prediction



(g) Ground truth

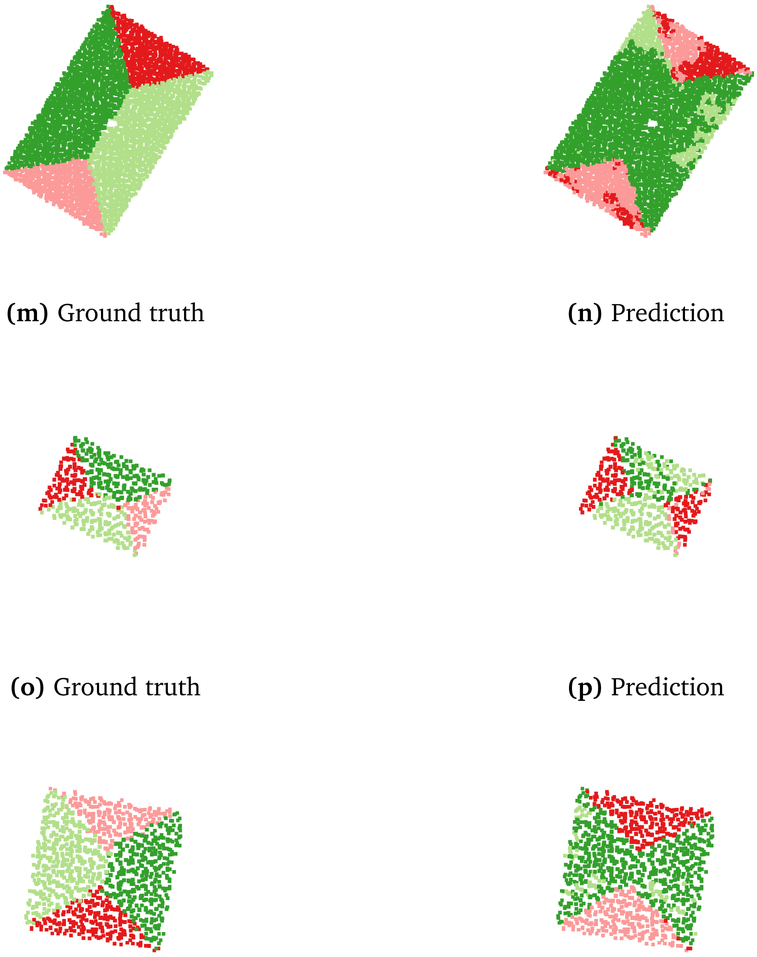
(h) Prediction

(i) Ground truth

(j) Prediction

(k) Ground truth

(l) Prediction



(m) Ground truth

(n) Prediction

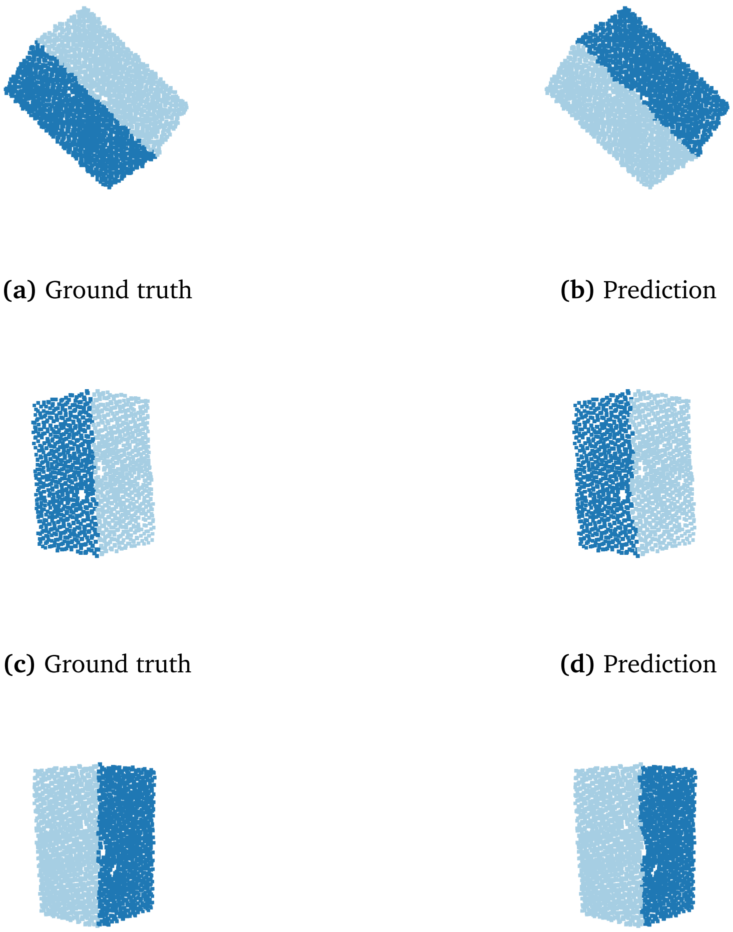
(o) Ground truth

(p) Prediction

(q) Ground truth

(r) Prediction

B.3 Type 3: *Gabled*



(a) Ground truth

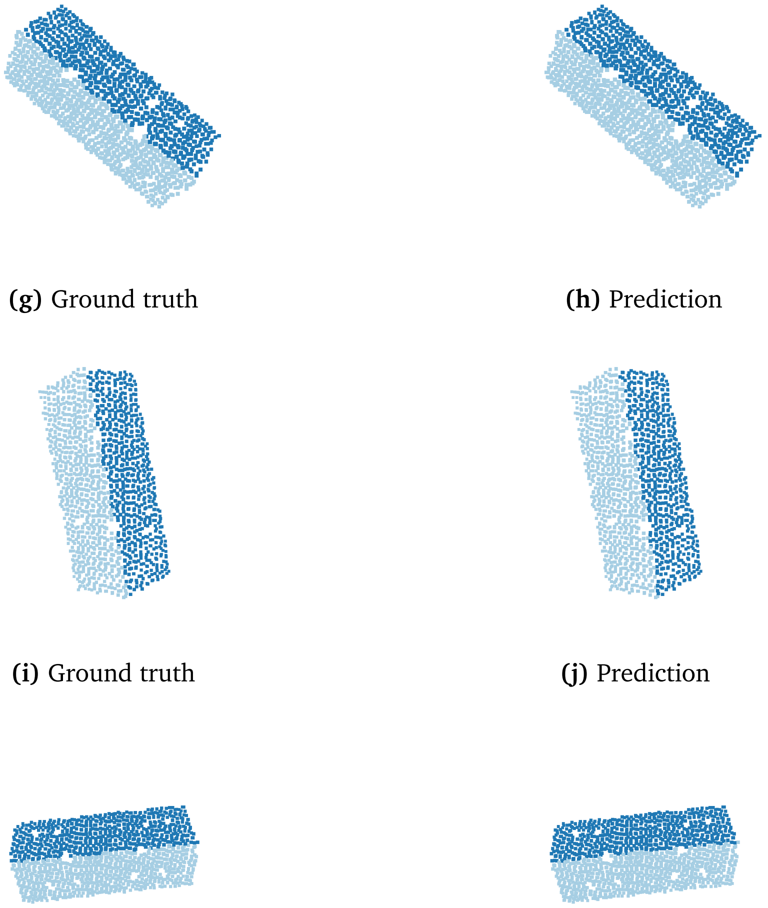
(b) Prediction

(c) Ground truth

(d) Prediction

(e) Ground truth

(f) Prediction



(g) Ground truth

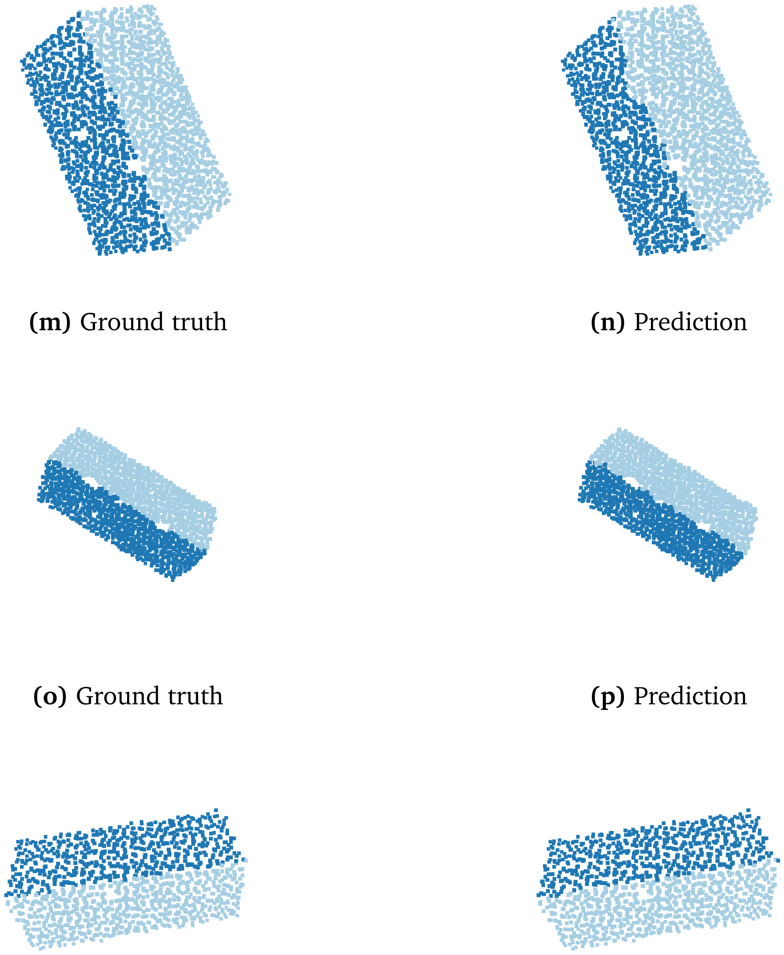
(h) Prediction

(i) Ground truth

(j) Prediction

(k) Ground truth

(l) Prediction



(m) Ground truth

(n) Prediction

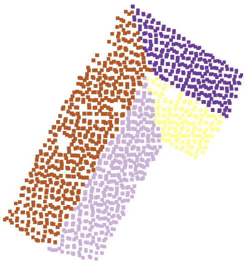
(o) Ground truth

(p) Prediction

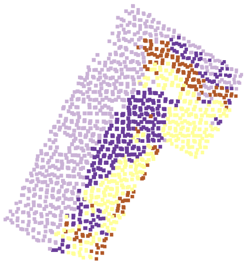
(q) Ground truth

(r) Prediction

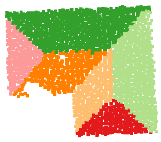
B.4 Type 4: *Corner Element*



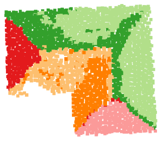
(a) Ground truth



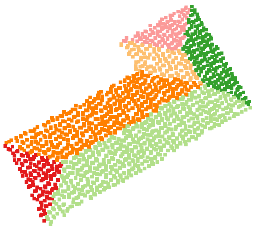
(b) Prediction



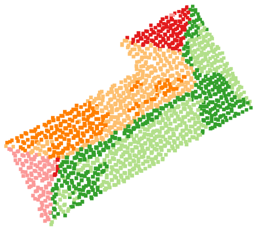
(c) Ground truth



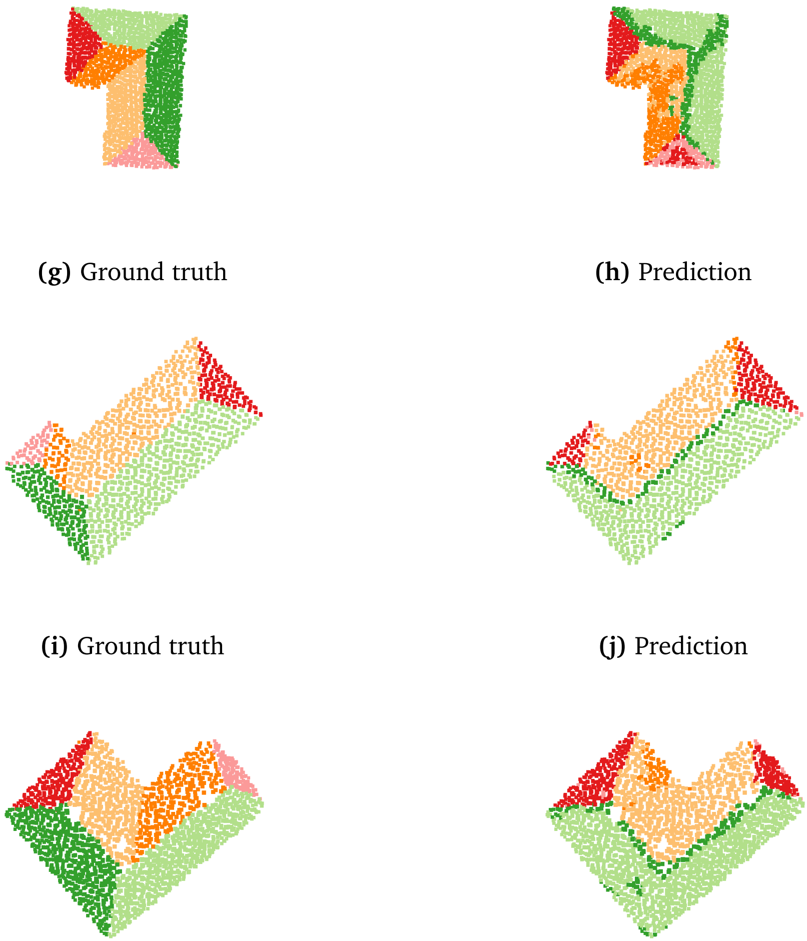
(d) Prediction



(e) Ground truth



(f) Prediction



(g) Ground truth

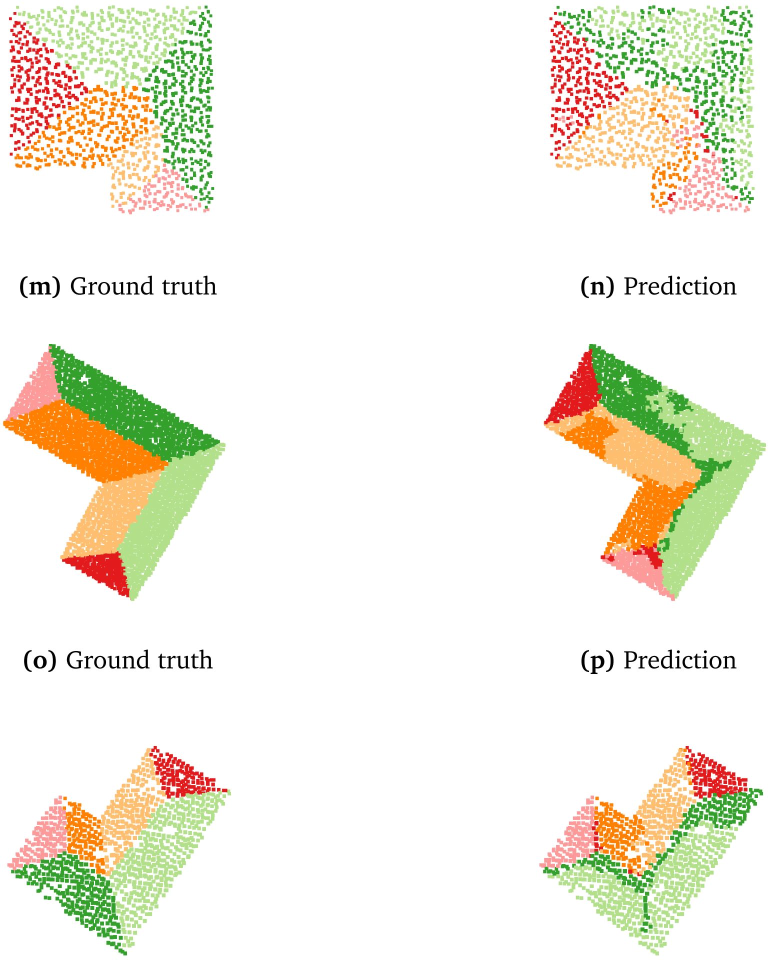
(h) Prediction

(i) Ground truth

(j) Prediction

(k) Ground truth

(l) Prediction



(m) Ground truth

(n) Prediction

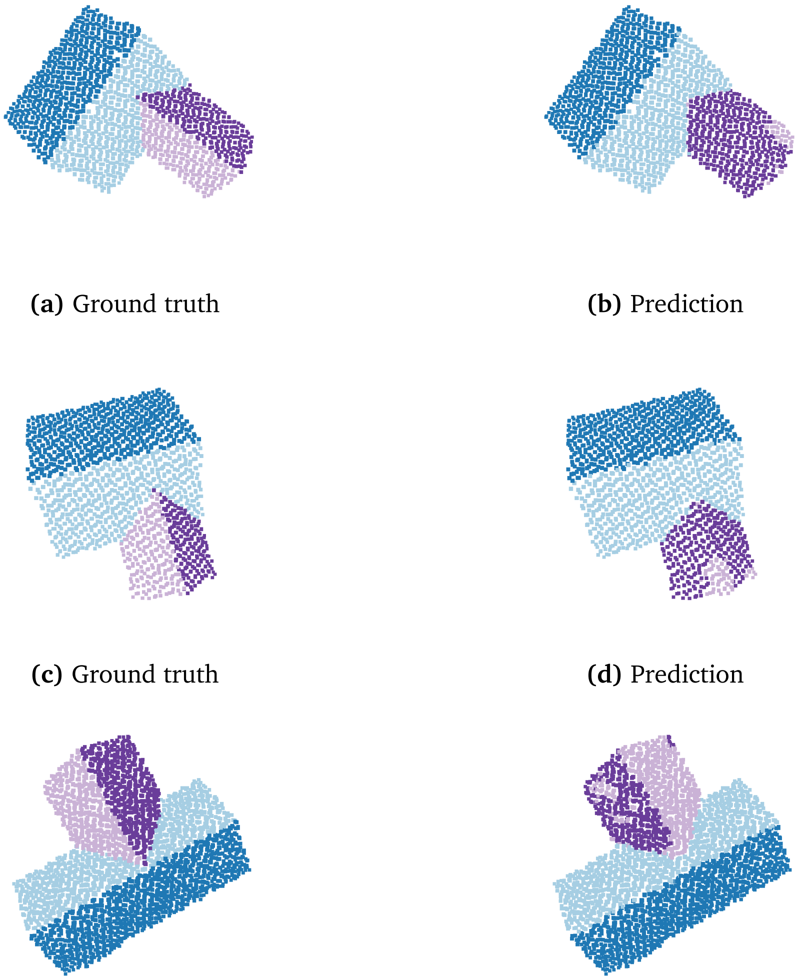
(o) Ground truth

(p) Prediction

(q) Ground truth

(r) Prediction

B.5 Type 5: *T-Element*



(a) Ground truth

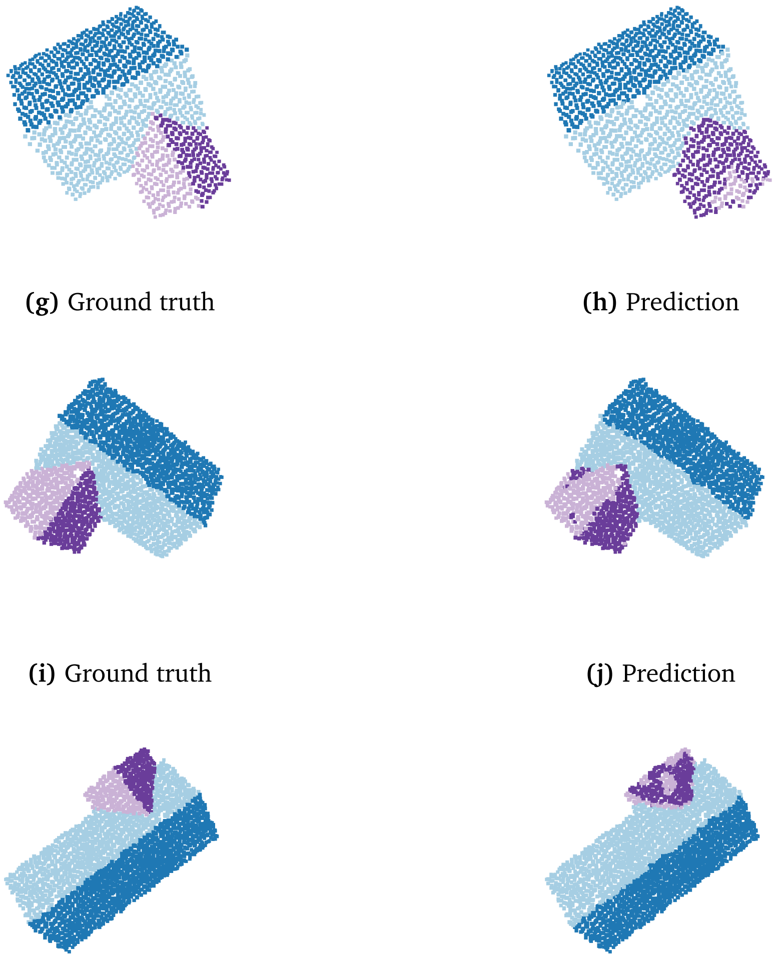
(b) Prediction

(c) Ground truth

(d) Prediction

(e) Ground truth

(f) Prediction



(g) Ground truth

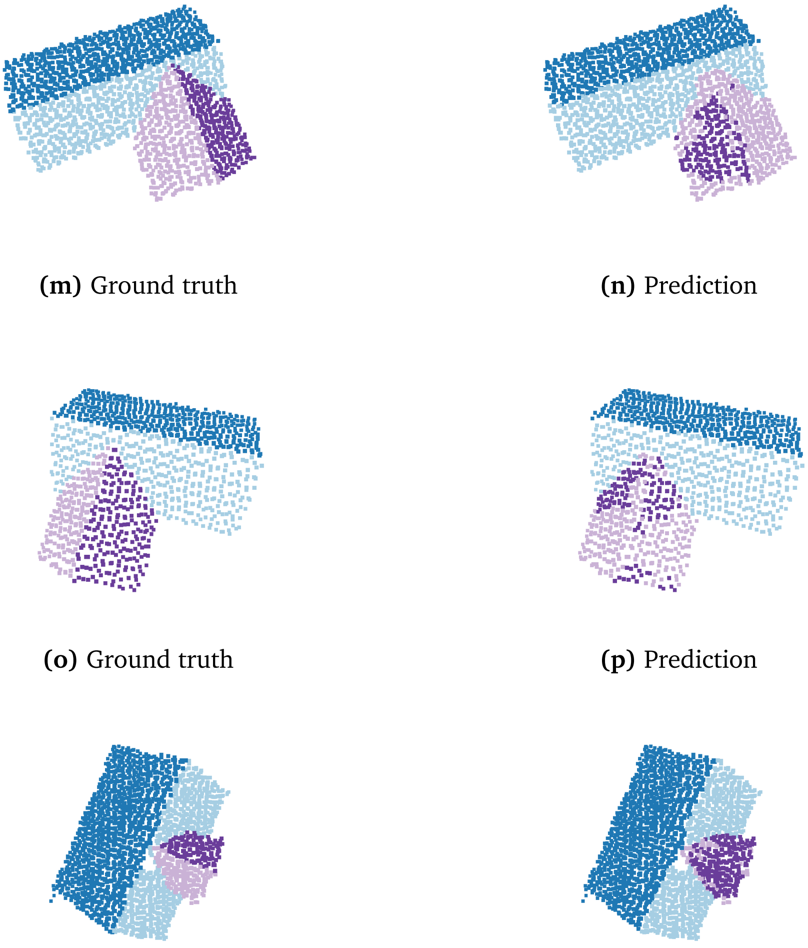
(h) Prediction

(i) Ground truth

(j) Prediction

(k) Ground truth

(l) Prediction



(m) Ground truth

(n) Prediction

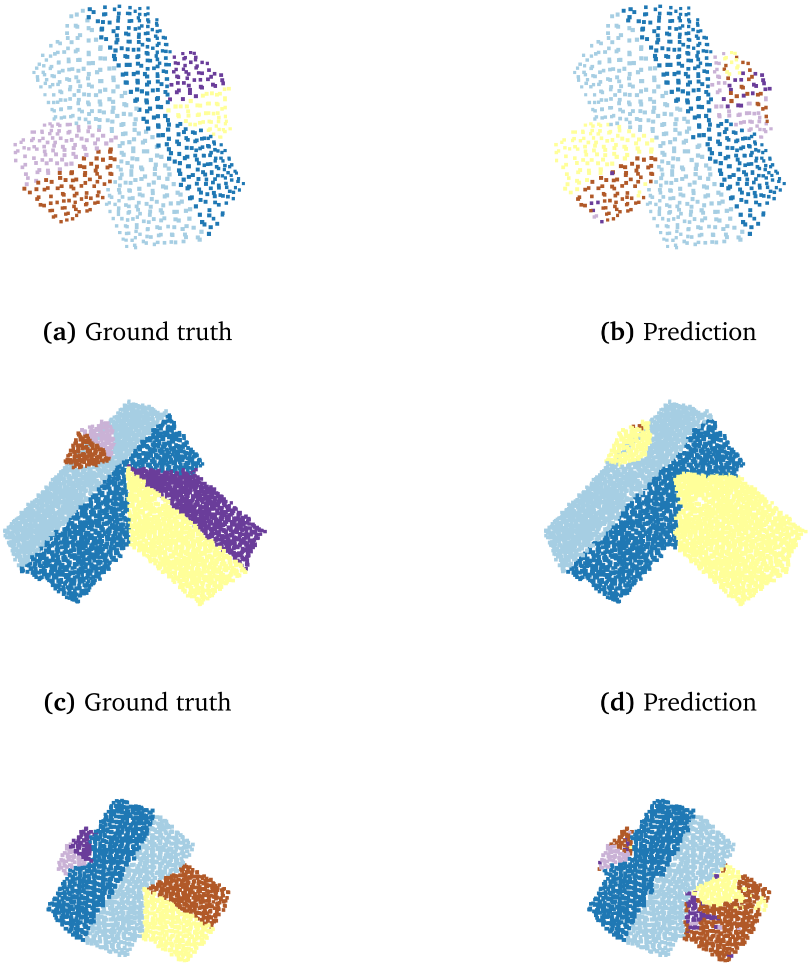
(o) Ground truth

(p) Prediction

(q) Ground truth

(r) Prediction

B.6 Type 6: *Cross Element*



(a) Ground truth

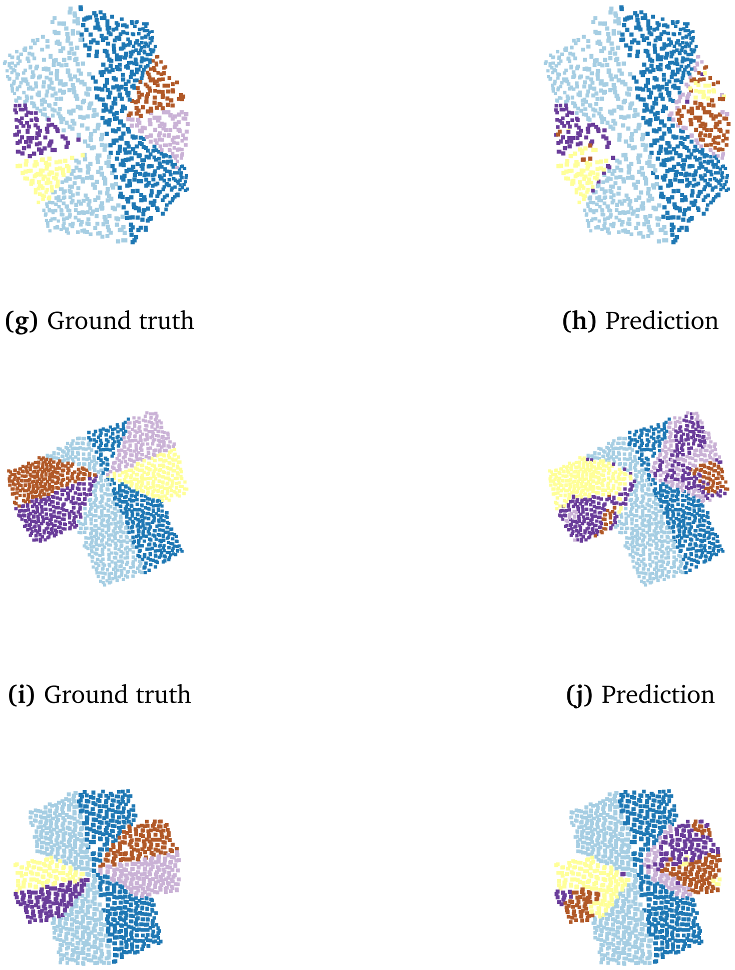
(b) Prediction

(c) Ground truth

(d) Prediction

(e) Ground truth

(f) Prediction



(g) Ground truth

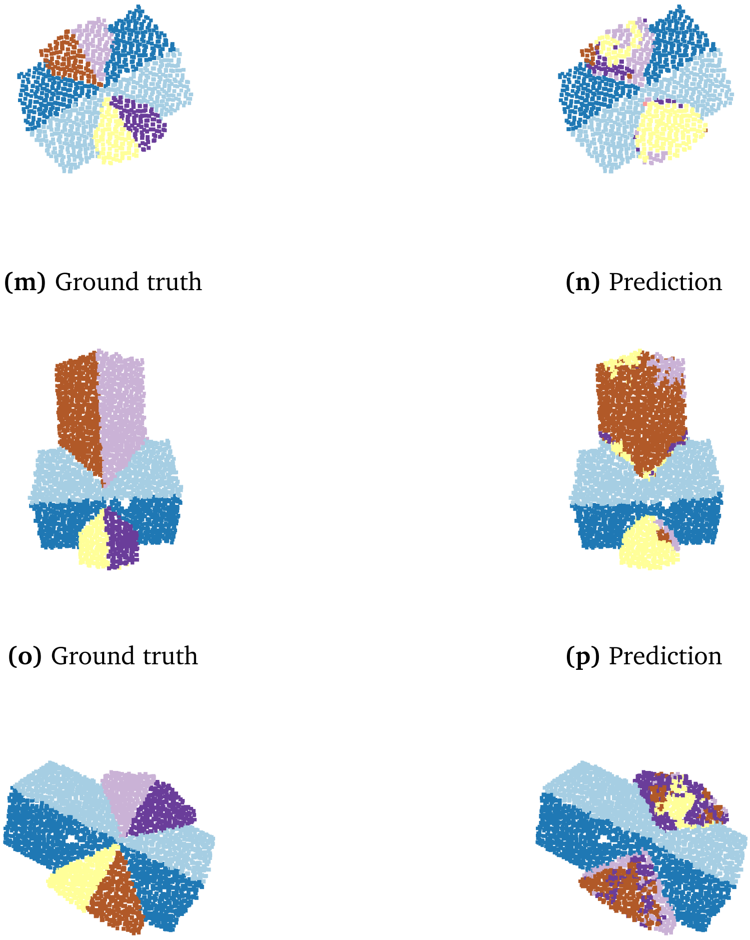
(h) Prediction

(i) Ground truth

(j) Prediction

(k) Ground truth

(l) Prediction



(m) Ground truth

(n) Prediction

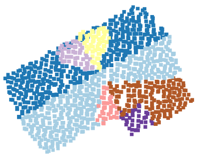
(o) Ground truth

(p) Prediction

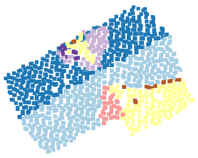
(q) Ground truth

(r) Prediction

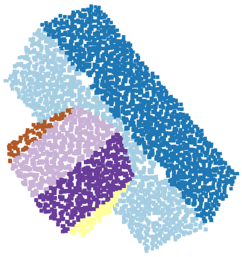
B.7 Type 7: *Combination*



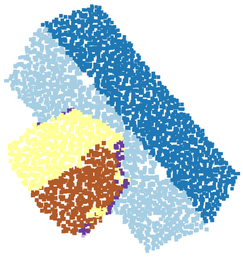
(a) Ground truth



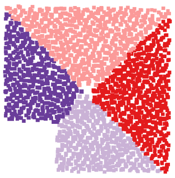
(b) Prediction



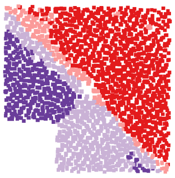
(c) Ground truth



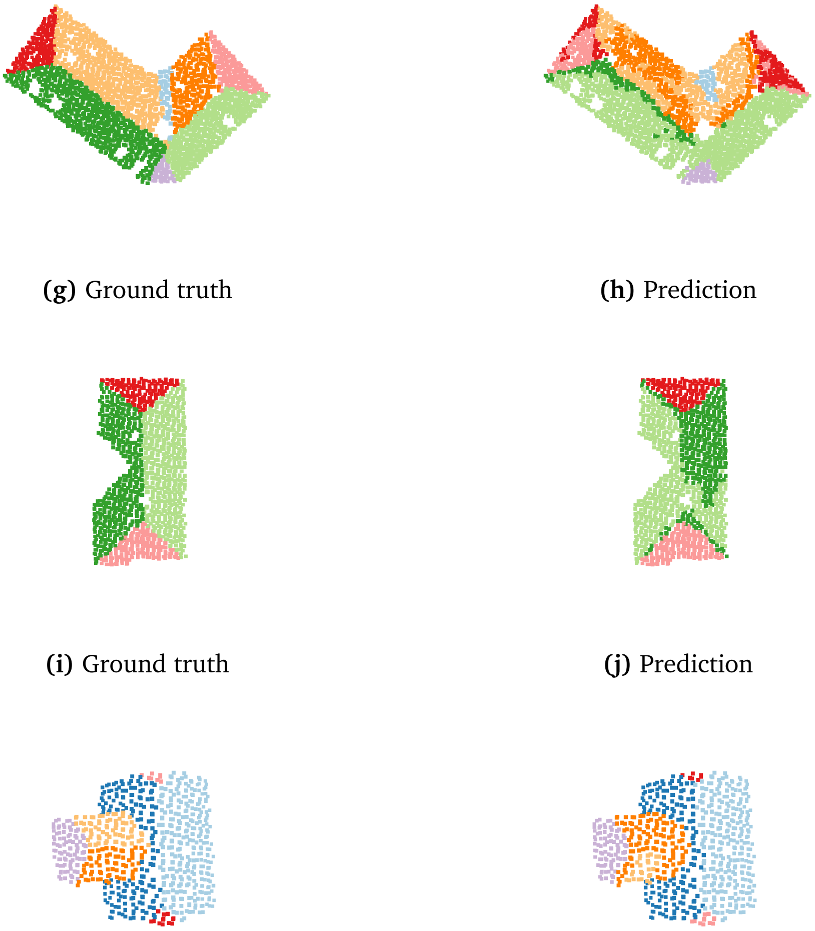
(d) Prediction



(e) Ground truth



(f) Prediction



(g) Ground truth

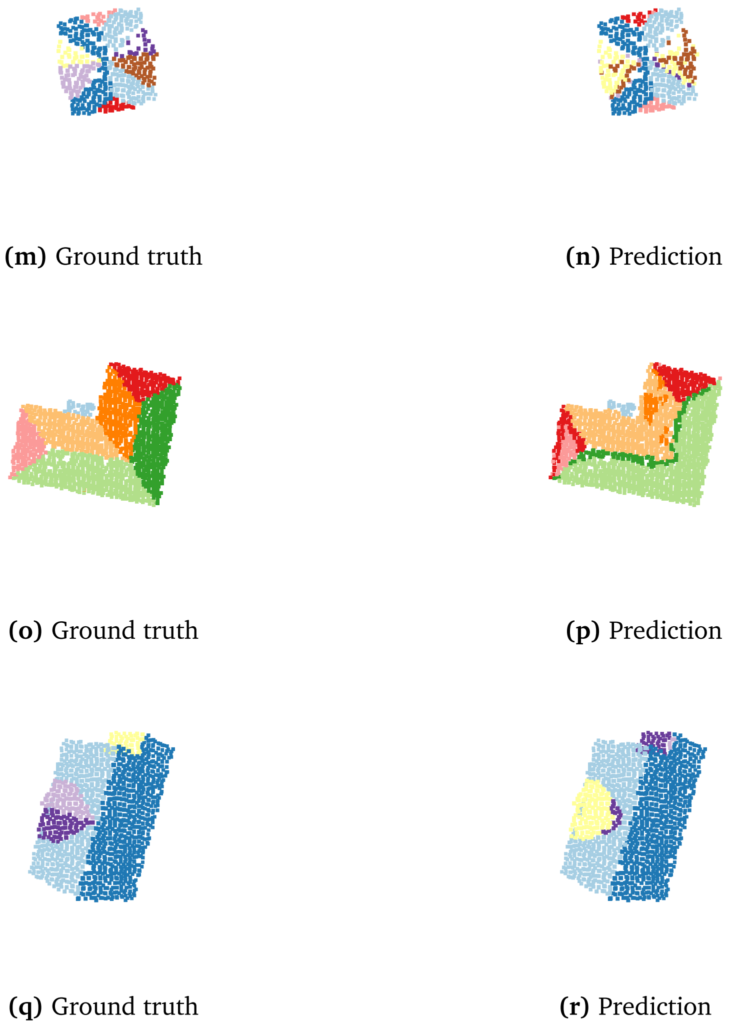
(h) Prediction

(i) Ground truth

(j) Prediction

(k) Ground truth

(l) Prediction



(m) Ground truth

(n) Prediction

(o) Ground truth

(p) Prediction

(q) Ground truth

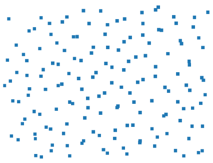
(r) Prediction

Appendix C

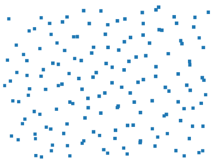
Visual results MSG_100

In this appendix, additional visuals for the predictions done by the MSG_100 model are shown. The results are separated by roof type, and 10 examples are given for each type.

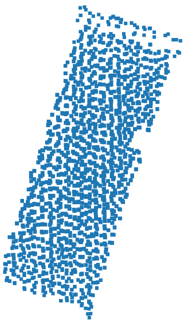
C.1 Type 1: *Flat*



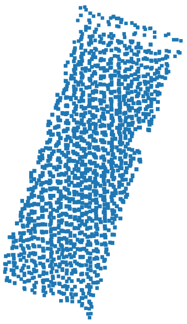
(a) Ground truth



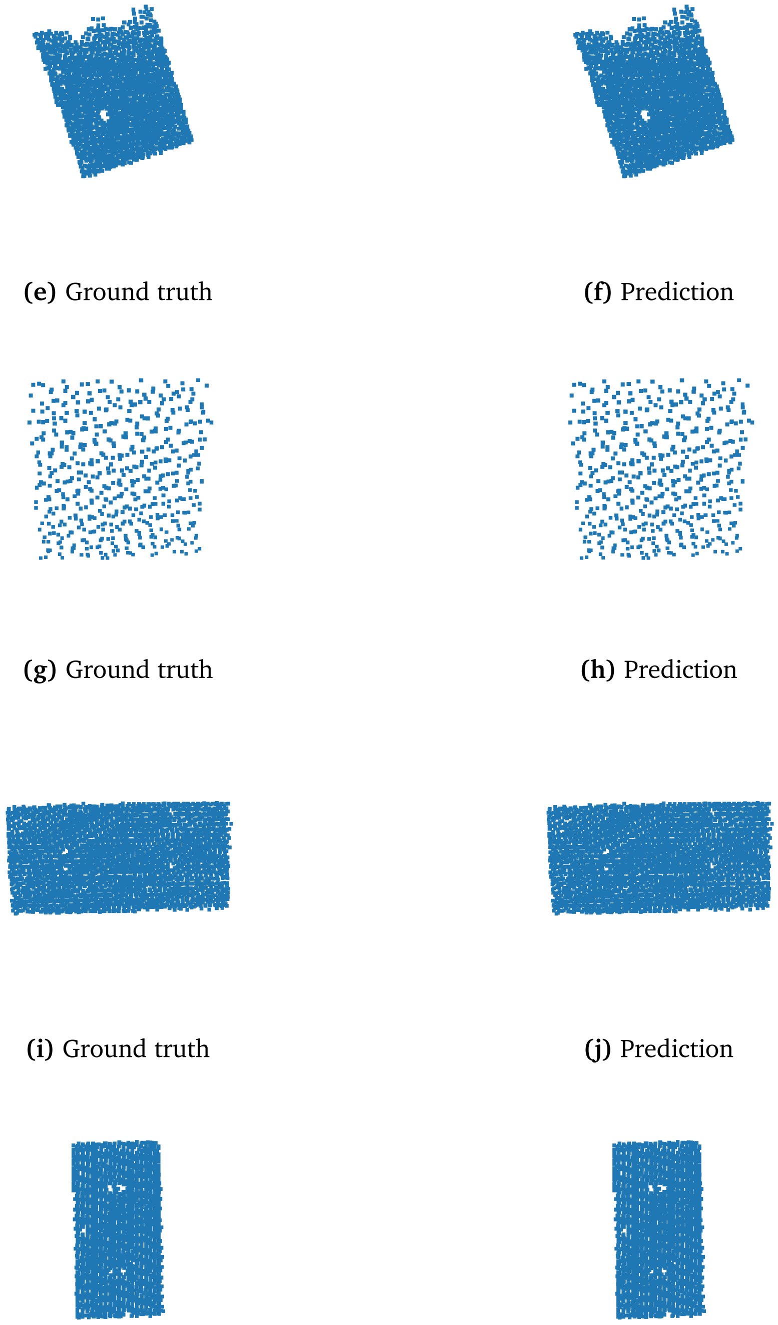
(b) Prediction



(c) Ground truth



(d) Prediction



(e) Ground truth

(f) Prediction

(g) Ground truth

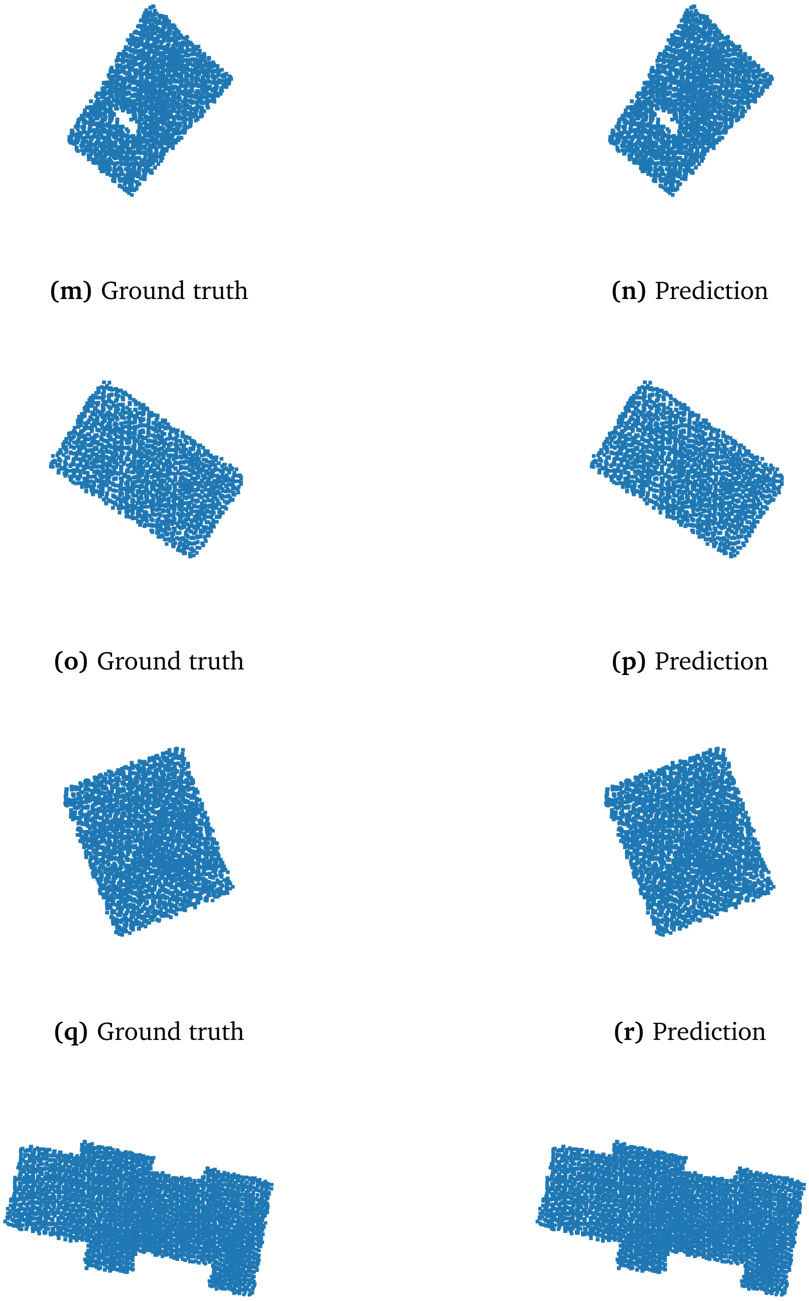
(h) Prediction

(i) Ground truth

(j) Prediction

(k) Ground truth

(l) Prediction



(m) Ground truth

(n) Prediction

(o) Ground truth

(p) Prediction

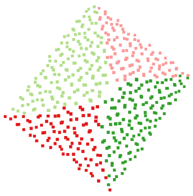
(q) Ground truth

(r) Prediction

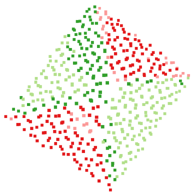
(s) Ground truth

(t) Prediction

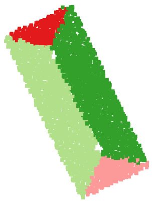
C.2 Type 2: *Hipped*



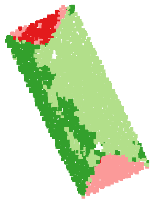
(a) Ground truth



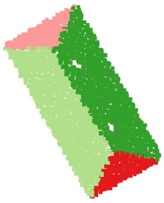
(b) Prediction



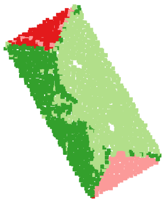
(c) Ground truth



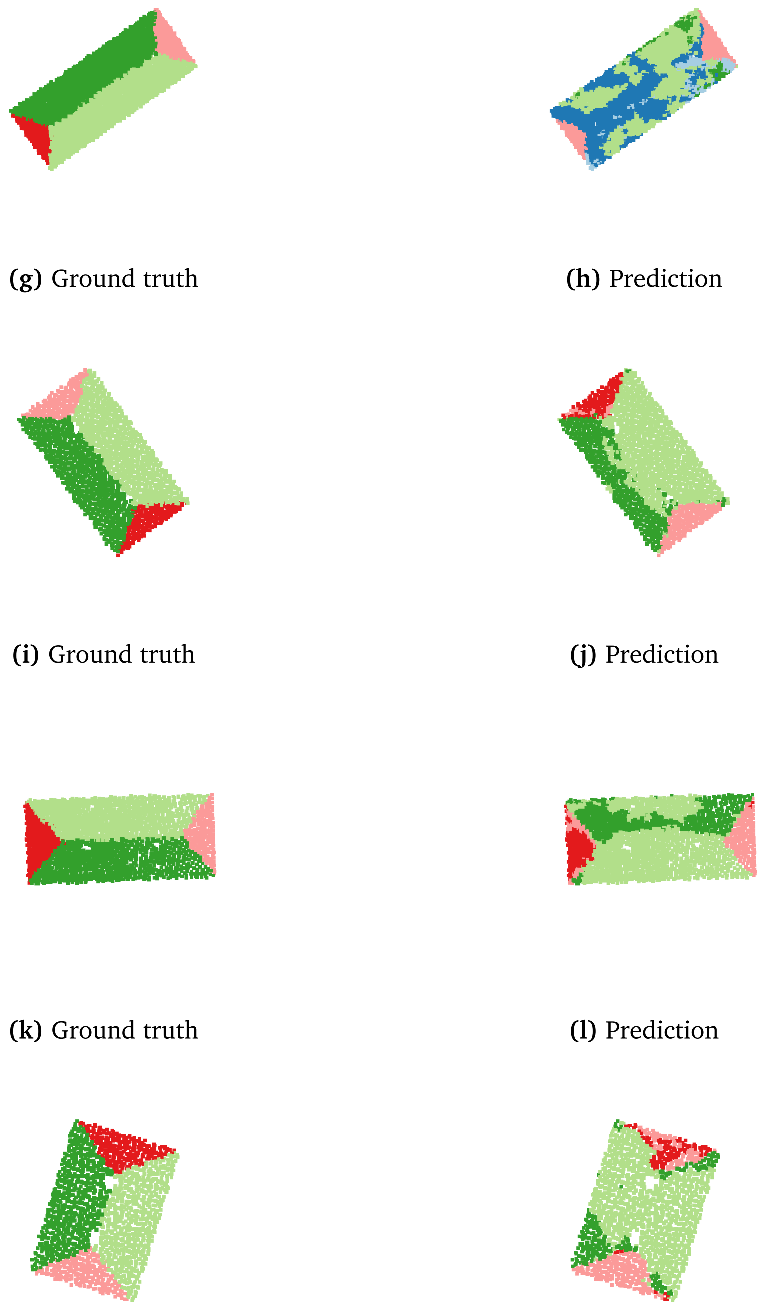
(d) Prediction



(e) Ground truth



(f) Prediction



(g) Ground truth

(h) Prediction

(i) Ground truth

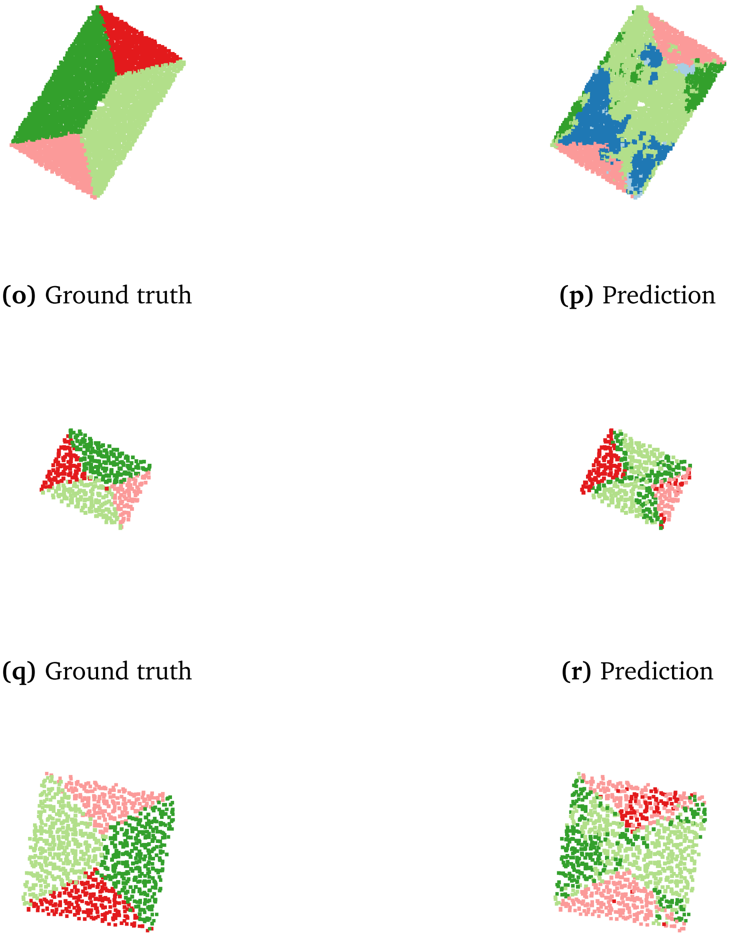
(j) Prediction

(k) Ground truth

(l) Prediction

(m) Ground truth

(n) Prediction



(o) Ground truth

(p) Prediction

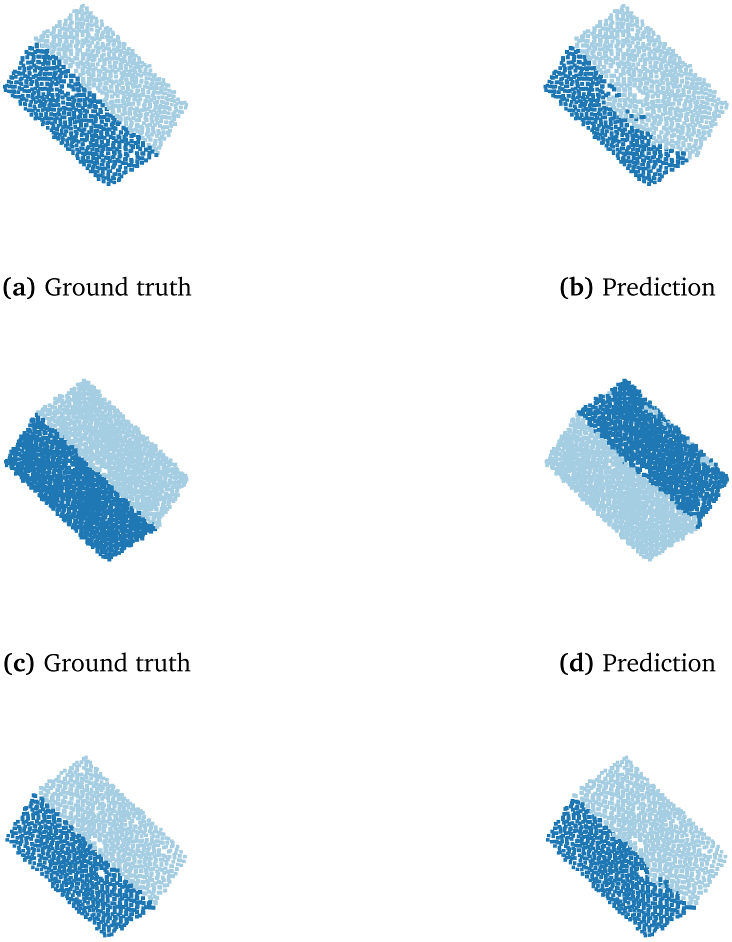
(q) Ground truth

(r) Prediction

(s) Ground truth

(t) Prediction

C.3 Type 3: *Gabled*



(a) Ground truth

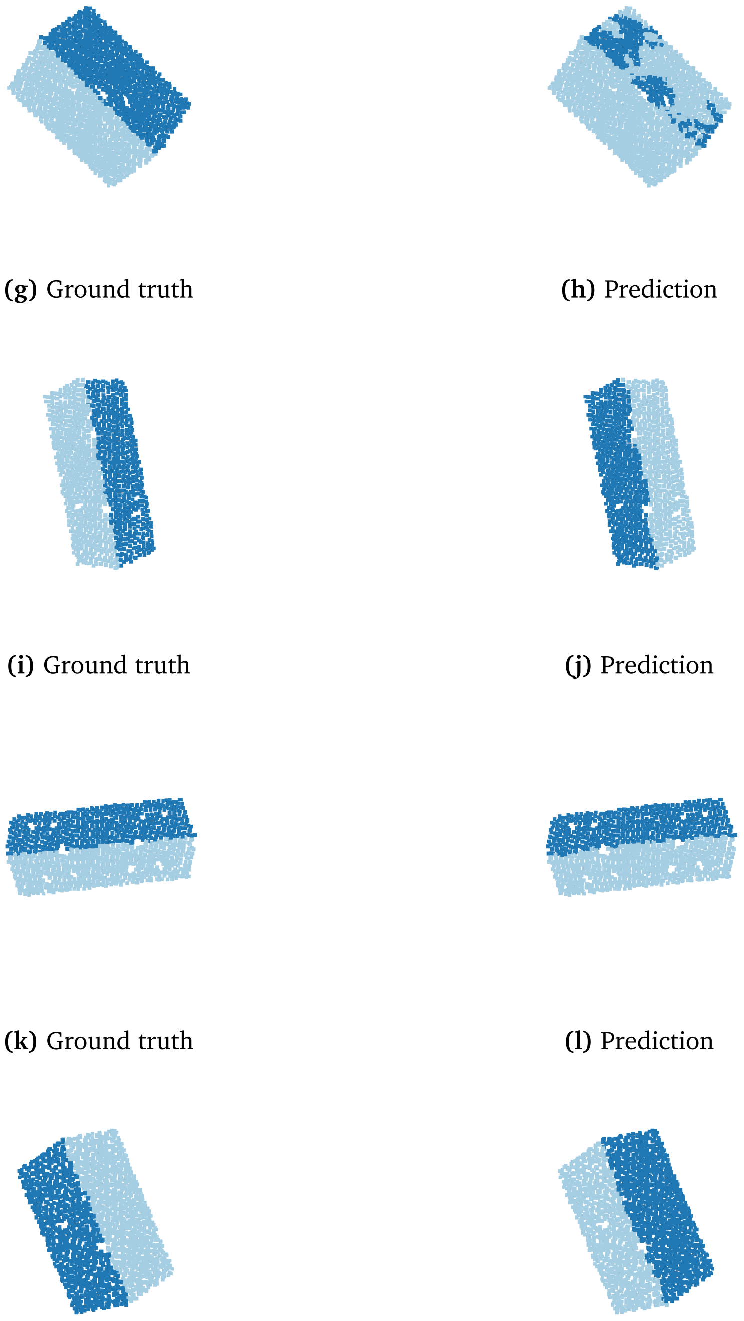
(b) Prediction

(c) Ground truth

(d) Prediction

(e) Ground truth

(f) Prediction



(g) Ground truth

(h) Prediction

(i) Ground truth

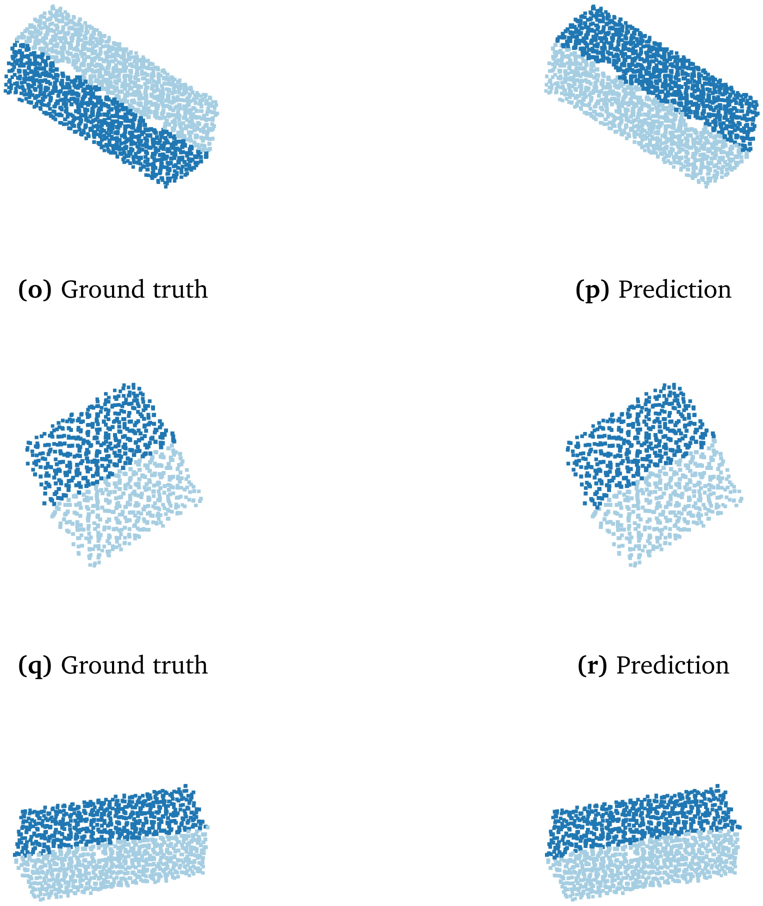
(j) Prediction

(k) Ground truth

(l) Prediction

(m) Ground truth

(n) Prediction



(o) Ground truth

(p) Prediction

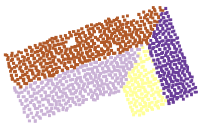
(q) Ground truth

(r) Prediction

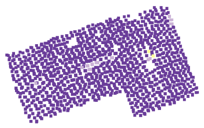
(s) Ground truth

(t) Prediction

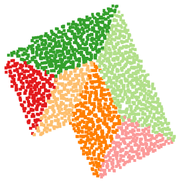
C.4 Type 4: *Corner Element*



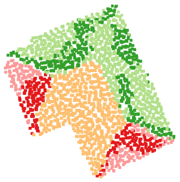
(a) Ground truth



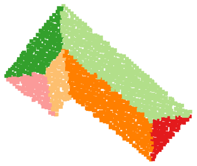
(b) Prediction



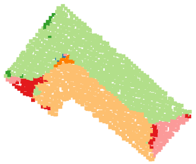
(c) Ground truth



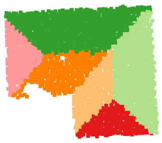
(d) Prediction



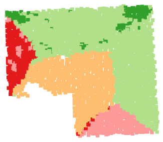
(e) Ground truth



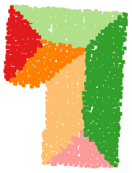
(f) Prediction



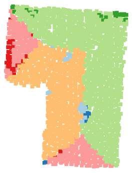
(g) Ground truth



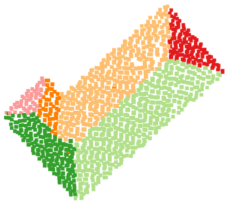
(h) Prediction



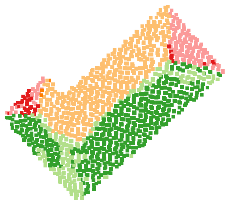
(i) Ground truth



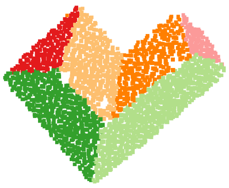
(j) Prediction



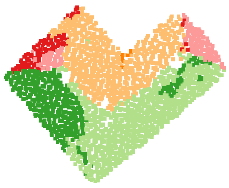
(k) Ground truth



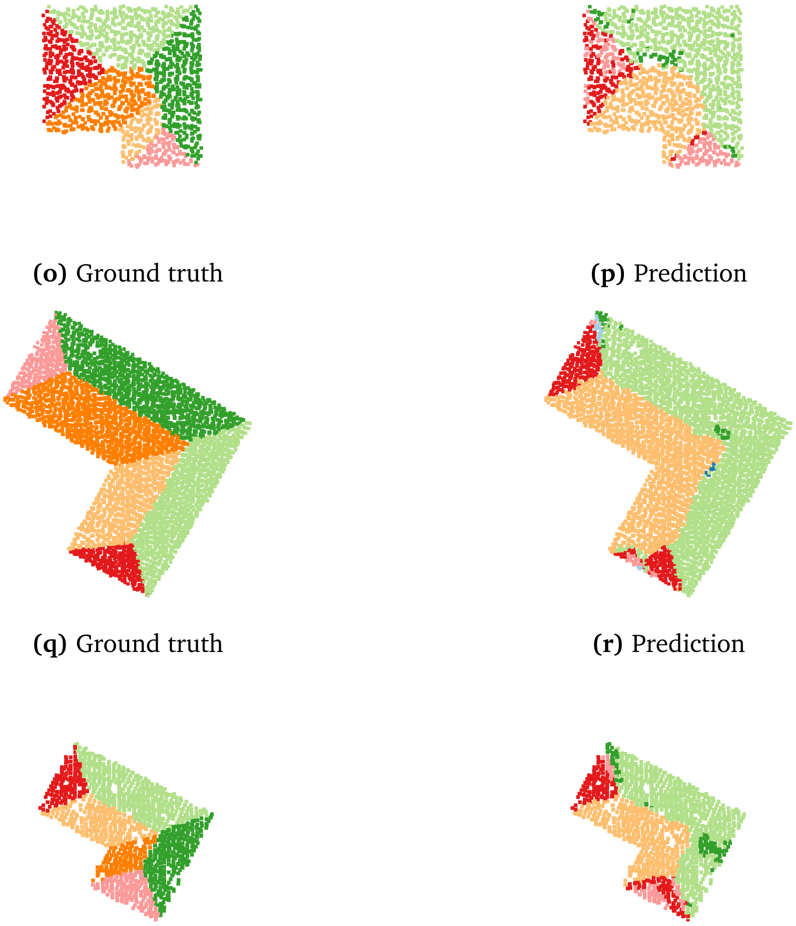
(l) Prediction



(m) Ground truth



(n) Prediction



(o) Ground truth

(p) Prediction

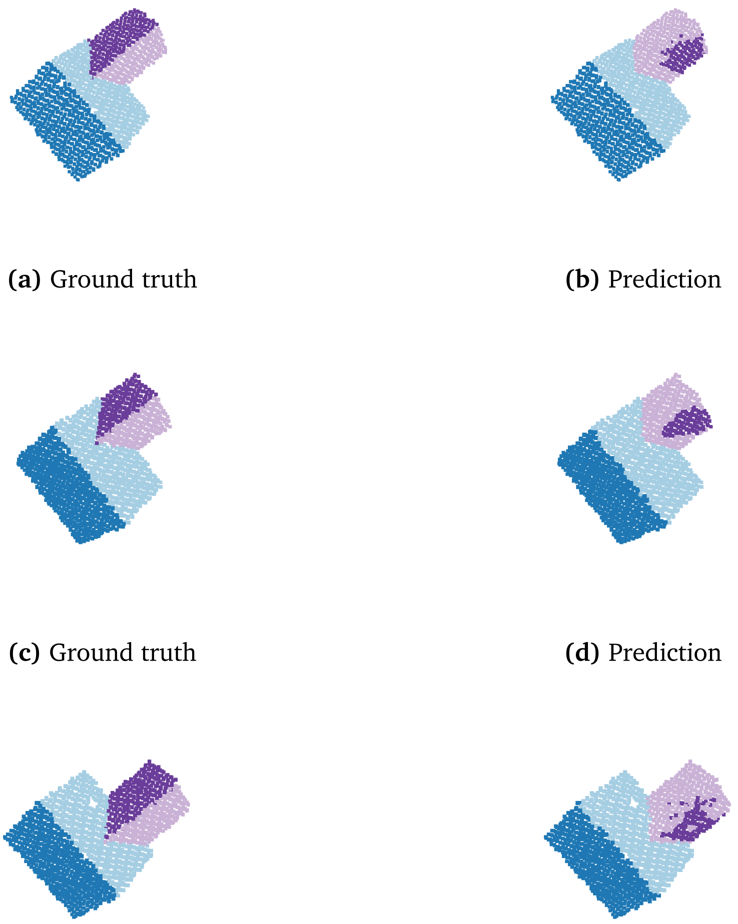
(q) Ground truth

(r) Prediction

(s) Ground truth

(t) Prediction

C.5 Type 5: *T-Element*



(a) Ground truth

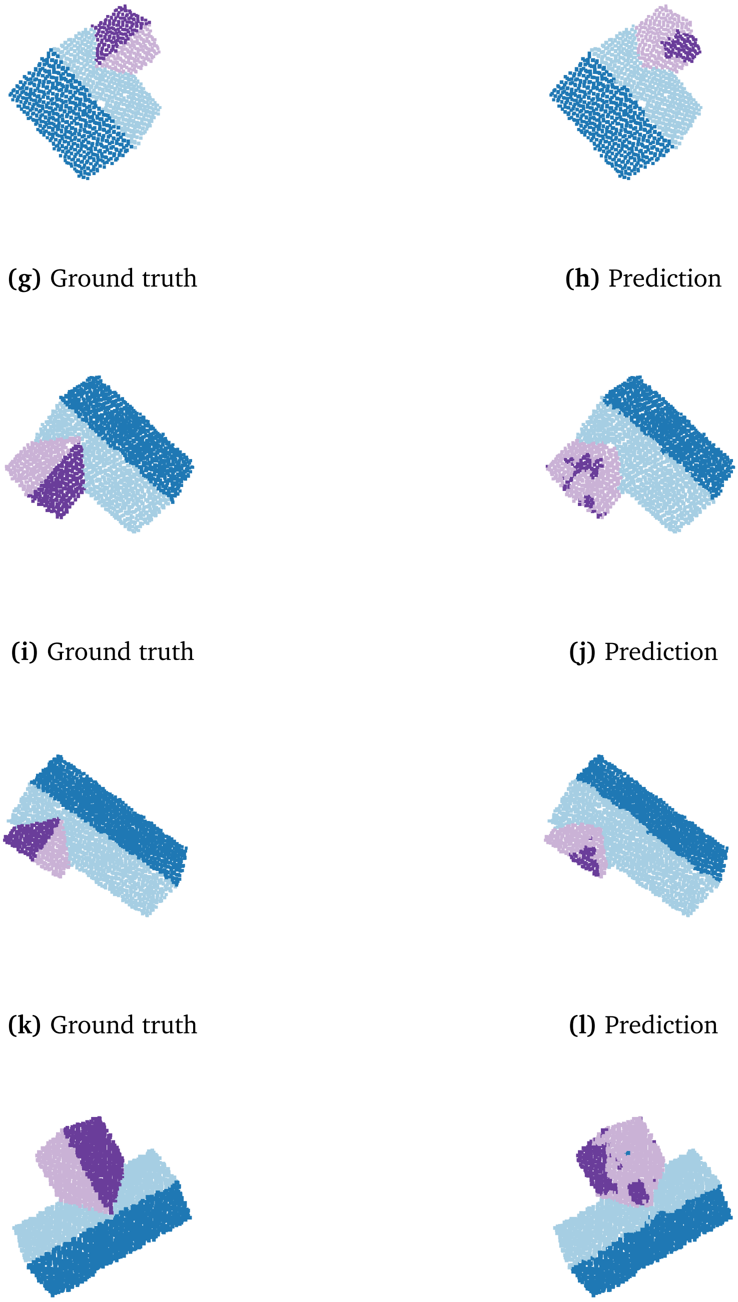
(b) Prediction

(c) Ground truth

(d) Prediction

(e) Ground truth

(f) Prediction



(g) Ground truth

(h) Prediction

(i) Ground truth

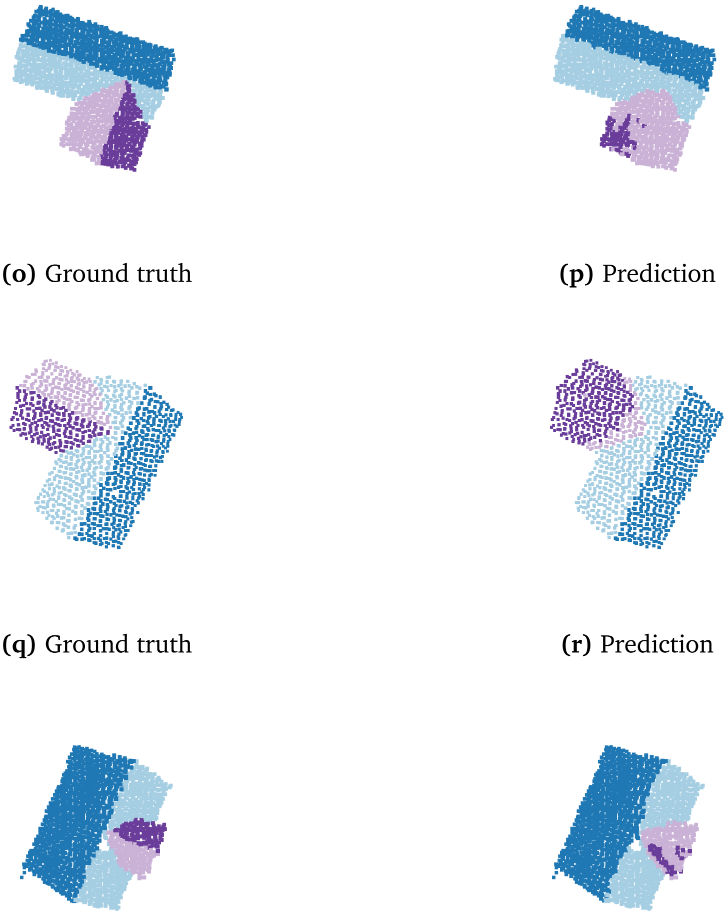
(j) Prediction

(k) Ground truth

(l) Prediction

(m) Ground truth

(n) Prediction



(o) Ground truth

(p) Prediction

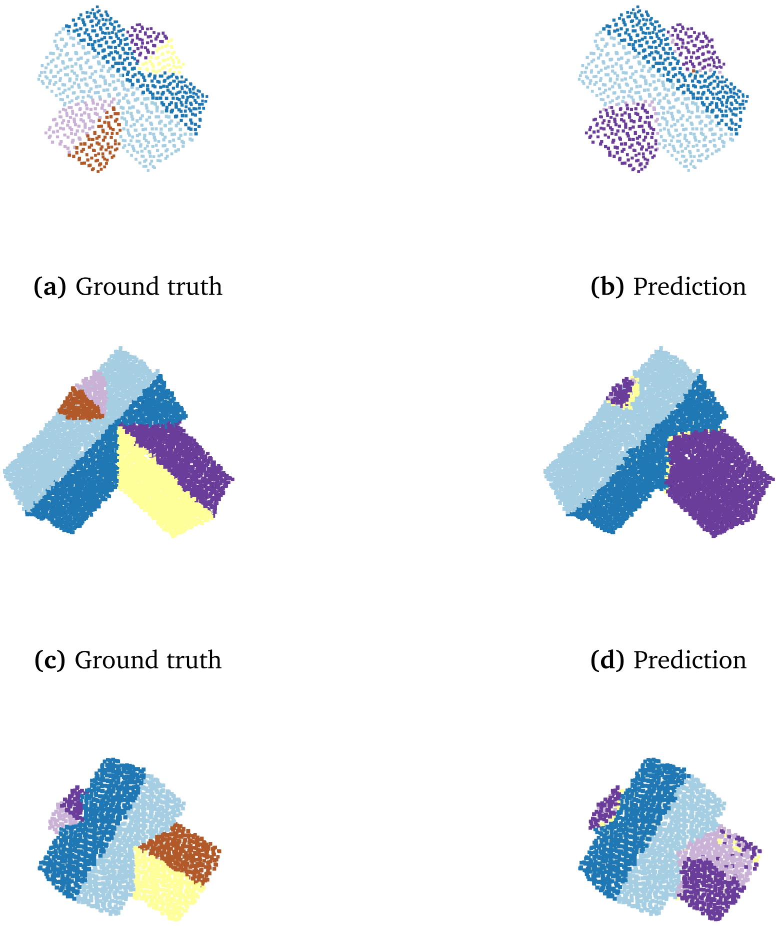
(q) Ground truth

(r) Prediction

(s) Ground truth

(t) Prediction

C.6 Type 6: *Cross-Element*



(a) Ground truth

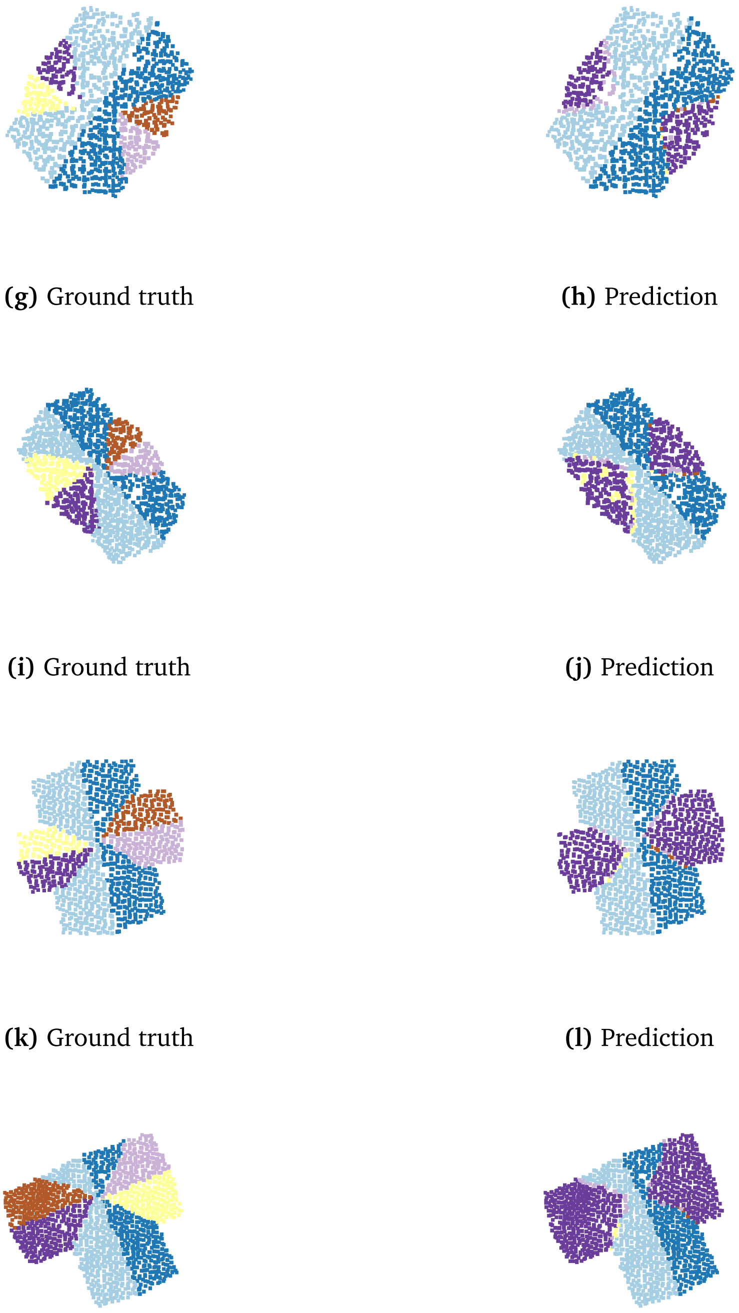
(b) Prediction

(c) Ground truth

(d) Prediction

(e) Ground truth

(f) Prediction



(g) Ground truth

(h) Prediction

(i) Ground truth

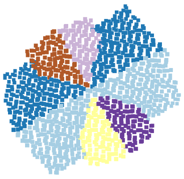
(j) Prediction

(k) Ground truth

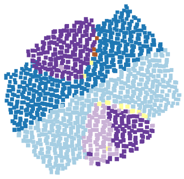
(l) Prediction

(m) Ground truth

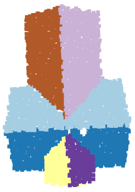
(n) Prediction



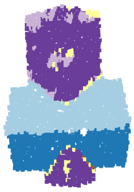
(o) Ground truth



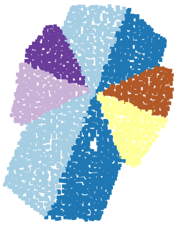
(p) Prediction



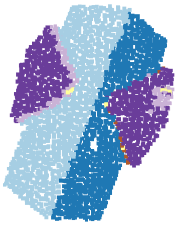
(q) Ground truth



(r) Prediction

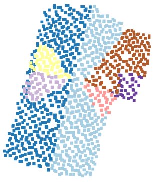


(s) Ground truth

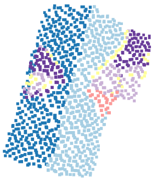


(t) Prediction

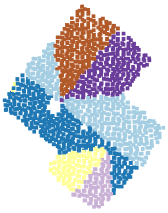
C.7 Type 7: *Combination*



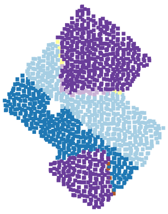
(a) Ground truth



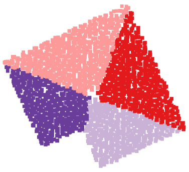
(b) Prediction



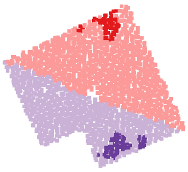
(c) Ground truth



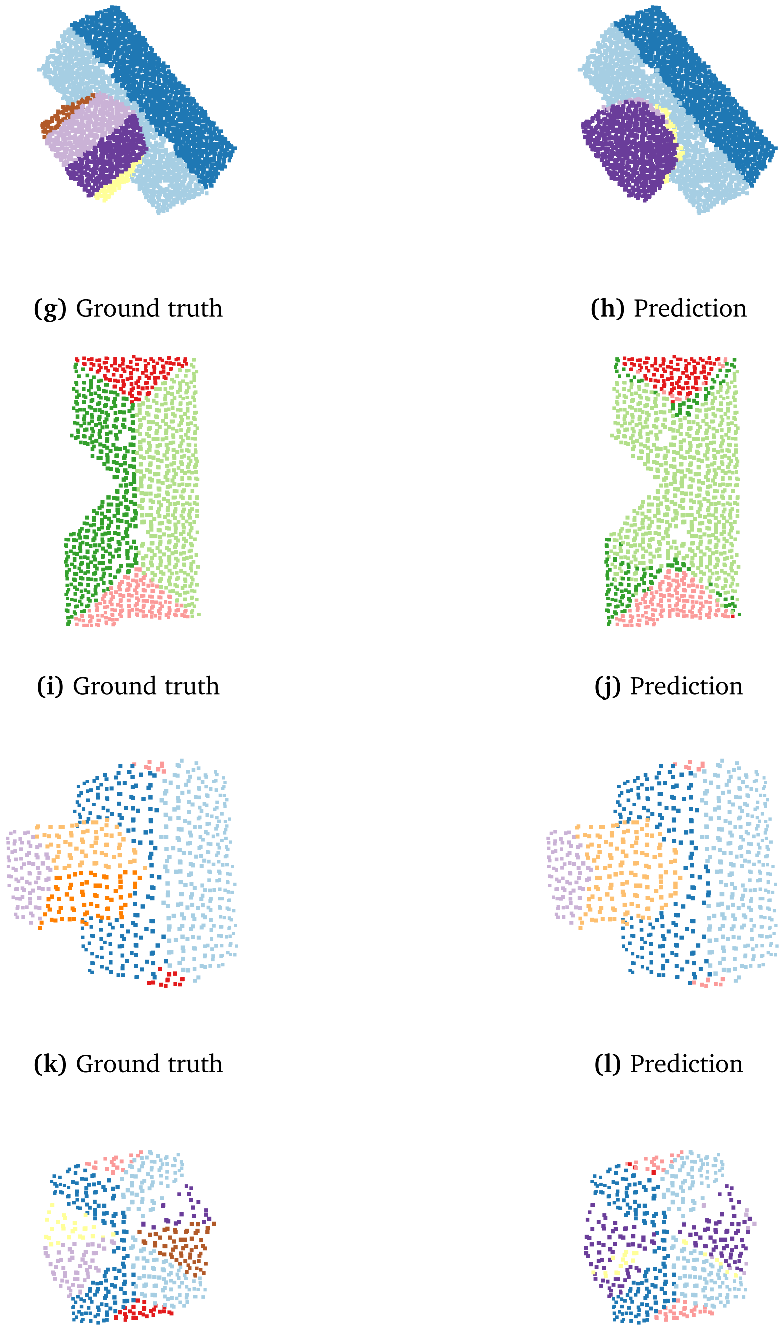
(d) Prediction



(e) Ground truth



(f) Prediction



(g) Ground truth

(h) Prediction

(i) Ground truth

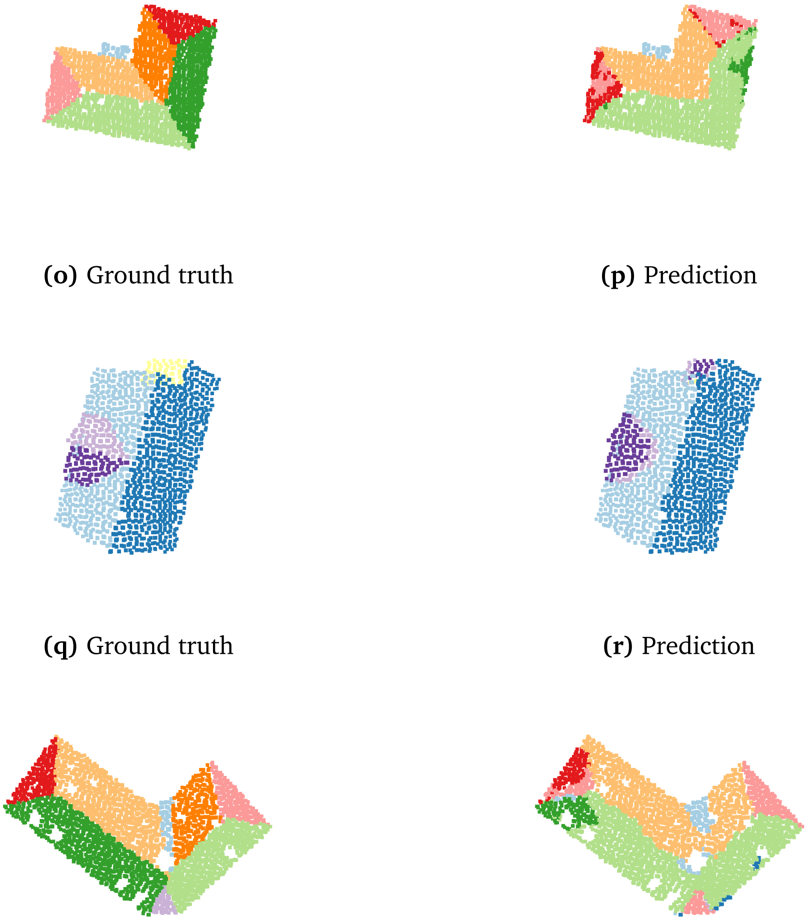
(j) Prediction

(k) Ground truth

(l) Prediction

(m) Ground truth

(n) Prediction



(o) Ground truth

(p) Prediction

(q) Ground truth

(r) Prediction

(s) Ground truth

(t) Prediction

Appendix D

Manual Evaluation on MSG_80

This appendix includes the metrics given in the manual evaluation of model MSG_80 for all 70 roofs.

Bid	Roof type	Part accuracy	Geometric accuracy
182224308	1	1.00	1.00
182279587	1	1.00	1.00
182284173	1	1.00	1.00
182287091	1	1.00	1.00
182380016	1	1.00	1.00
182397652	1	1.00	1.00
300118905	1	1.00	1.00
300287193	1	1.00	1.00
300455961	1	1.00	1.00
300557684	1	1.00	1.00
10456495-3	2	0.55	1.00
300079914-3	2	0.75	1.00
21070440-2	2	0.65	0.95
21074551-0	2	0.75	0.95
182142182-3	2	0.85	0.95
182142379-1	2	0.80	0.95
182215910-3	2	0.65	0.95
182181447-2	2	0.65	0.95
182142239-2	2	0.85	0.95
182143398-2	2	0.70	0.95
10519136	3	0.95	1.00
10519209	3	1.00	1.00
10519268	3	0.95	1.00

Table D.1 continued from previous page

Bid	Roof type	Part accuracy	Geometric accuracy
10519292	3	0.95	1.00
300279692	3	1.00	1.00
21104825	3	0.95	1.00
182152056	3	0.95	1.00
182130516	3	0.95	1.00
182213217	3	0.90	1.00
182214620	3	0.95	1.00
10470048-21	4	0.75	1.00
10473225-1	4	0.75	0.95
10477018-3	4	0.70	0.95
21084573-7	4	0.80	0.95
182149136-3	4	0.80	0.95
182150630-0	4	0.70	0.95
182213608-1	4	0.60	1.00
182211605-3	4	0.55	0.95
182245070-3	4	0.80	0.90
182274798-6	4	0.80	0.95
10486823-0	5	0.90	0.95
10486831-0	5	0.85	0.95
10474442-1	5	0.85	0.95
10477107-0	5	0.85	0.95
10477867-1	5	0.90	0.95
10478413-0	5	0.85	0.95
10498821-1	5	0.90	0.95
10505712-0	5	0.85	0.95
10517389-0	5	0.85	1.00
21048860-0	5	0.90	1.00
10465036-6	6	0.70	0.90
10457319-0	6	0.90	1.00
10557747-8	6	0.80	0.95
21021539-17	6	0.85	0.95
21062618-10	6	0.75	0.90
21088358-4	6	0.80	0.95
182210331-5	6	0.85	1.00
182210161-5	6	0.90	0.95
182280291-19	6	0.80	1.00
182294063-10	6	0.80	1.00

Table D.1 continued from previous page

Bid	Roof type	Part accuracy	Geometric accuracy
10463092-10	7	0.90	0.95
10498422-7	7	0.80	0.95
10541220-1	7	0.70	1.00
21022071-2	7	0.65	0.95
182173827-6	7	0.75	0.90
182177164-6	7	0.90	0.95
300504074-2	7	0.90	1.00
182280240-8	7	0.85	1.00
182278599-2-3	7	0.80	0.95
182279854-1	7	0.95	1.00

Appendix E

Manual Evaluation on MSG_100

This appendix includes the metrics given in the manual evaluation of model MSG_100 for all 70 roofs.

Bid	Roof type	Part accuracy	Geometric accuracy
182224308	1	1.00	1.00
182279587	1	1.00	1.00
182284173	1	1.00	1.00
182287091	1	1.00	1.00
182380016	1	1.00	1.00
182397652	1	1.00	1.00
300118905	1	1.00	1.00
300287193	1	1.00	1.00
300455961	1	1.00	1.00
300557684	1	1.00	1.00
10456495-3	2	0.65	1.00
300079914-3	2	0.80	0.95
21070440-2	2	0.45	0.70
21074551-0	2	0.80	0.95
182142182-3	2	0.80	0.95
182142379-1	2	0.80	1.00
182215910-3	2	0.80	0.95
182181447-2	2	0.80	0.95
182142239-2	2	0.85	1.00
182142298-2	2	0.30	0.50
10519136	3	0.90	1.00
10519209	3	0.95	1.00
10519268	3	0.95	1.00

Table E.1 continued from previous page

Bid	Roof type	Part accuracy	Geometric accuracy
10519292	3	0.65	1.00
300279692	3	0.95	1.00
21104825	3	0.95	1.00
182152056	3	0.95	1.00
182130516	3	0.95	1.00
182213217	3	0.95	1.00
182214620	3	0.95	1.00
10470048-21	4	0.40	1.00
10473225-1	4	0.70	0.95
10477018-3	4	0.70	0.90
21084573-7	4	0.65	1.00
182149136-3	4	0.65	0.95
182150630-0	4	0.60	0.95
182213608-1	4	0.70	0.90
182211605-3	4	0.80	0.95
182245070-3	4	0.65	1.00
182274798-6	4	0.60	0.95
10486823-0	5	0.80	0.95
10486831-0	5	0.85	0.95
10474442-1	5	0.90	0.95
10477107-0	5	0.90	1.00
10477867-1	5	0.85	1.00
10478413-0	5	0.90	1.00
10498821-1	5	0.75	0.95
10505712-0	5	0.85	0.95
10517389-0	5	0.85	1.00
21048860-0	5	0.90	1.00
10465036-6	6	0.65	0.85
10457319-0	6	0.80	1.00
10557747-8	6	0.85	0.95
21021539-17	6	0.85	0.95
21062618-10	6	0.65	0.90
21088358-4	6	0.70	0.95
182210331-5	6	0.80	1.00
182210161-5	6	0.80	1.00
182280291-19	6	0.75	0.95
182294063-10	6	0.70	0.95

Table E.1 continued from previous page

Bid	Roof type	Part accuracy	Geometric accuracy
10463092-10	7	0.85	0.95
10498422-7	7	0.65	1.00
10541220-1	7	0.60	1.00
21022071-2	7	0.60	1.00
182173827-6	7	0.60	0.90
182177164-6	7	0.85	0.95
300504074-2	7	0.80	0.95
182280240-8	7	0.85	1.00
182278599-2-3	7	0.90	0.95
182279854-1	7	0.90	1.00

Vilde Myren Mo and Marie Ting Falch Orre

TR3DRoofs: A Urban Roof Dataset

NTNU
Norwegian University of
Science and Technology