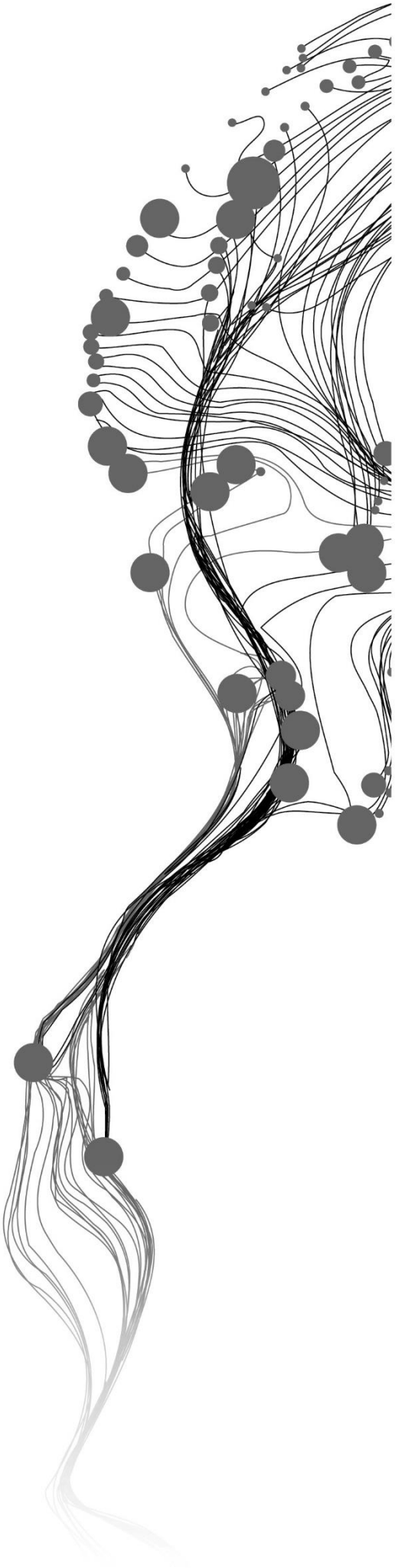
Planar roof structure extraction from Very High-Resolution aerial images and Digital Surface Models using deep learning

MERUYERT KENZHEBAY
June 2022

SUPERVISORS:
Prof. dr. Claudio Persello
dr. Mila Koeva

ADVISOR:
Wufan Zhao



Planar roof structure extraction from Very High-Resolution aerial images and Digital Surface Models using deep learning

MERUYERT KENZHEBAY
Enschede, The Netherlands, June 2022

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

Specialization: Geoinformatics

SUPERVISORS:
Prof. dr. Claudio Persello
dr. Mila Koeva

ADVISOR:
Wufan Zhao

THESIS ASSESSMENT BOARD:
Prof. dr. ir. Alfred Stein (Chair)
dr. Ronny Hänsch (External Examiner, DLR)

ABSTRACT

Roof structure reconstruction is one of the more recent and active research directions in urban-related studies. Roof geometry information is needed for the generation of 3D models, which are used for applications such as solar potential estimation and telecommunication installation planning, wind flow simulations for pollutant diffusion analysis, etc. Given the advance in remote sensing technologies and the machine learning field, particularly deep learning, the prospects of deriving the roof structure information accurately and efficiently are promising. Many approaches for extracting roof structure have been proposed; however, there are still issues with output regularization, false detection and misclassification, and low computational efficiency, which leaves room for further improvement.

In our study, we attempt to address these issues by proposing deep learning FCN-based methods for extracting roof structure from aerial imagery and Digital Surface Models (DSM) in the form of joined inner and outer rooflines directly in a regularized vector format. We develop and compare two roof structure extraction methods. The methodology and implementation details of both models are identical, with the exception that one of them has frame field learning branches for inner rooflines and outer rooflines. Frame field is a 4-D PolyVector field that helps to extract more regularized building boundaries with the correctly detected corners. The methodology is comprised of outer and inner rooflines segmentation, vectorization and post-processing. The approach was evaluated using pixel-level IoU metric and line-level PoLiS, $\text{Precision}_{\text{PoLiS}\leq 0.5}$, $\text{Recall}_{\text{PoLiS}\leq 0.5}$ and $\text{F-score}_{\text{PoLiS}\leq 0.5}$ metrics on both outer and inner rooflines. The experimental study area is the Stadsveld – 't Zwering neighbourhood of Enschede, Netherlands.

According to our experiments, both models showed quite good performance in extracting building roof structures. The frame field learning model slightly outperformed the no-field model on inner rooflines segmentation with an IoU value of 0.35 and a little worse than the no-field model on outer rooflines, 0.37. However, the no-field model performed better than frame-field learning on PoLiS distance with values of 3,5 m and 1,2 m for outlines and inner rooflines, respectively. Besides, the no-field model scored higher on PoLiS-thresholded F-score for outlines and inner rooflines, having, 0.31 and 0.57 respectively. The no-field model produced better visual results, with straighter walls and fewer missed inner roofline detections. It can predict buildings with common walls thanks to the skeleton graph computation. To summarize, the frame field had little impact on the findings, and the proposed no-field method is suitable for urban applications and has the potential to be improved further.

Keywords: image processing, image analysis, deep learning, roof structure extraction, roof vectorization, frame field learning

ACKNOWLEDGMENTS

I wish to express my deepest appreciation to ITC, University of Twente, for awarding me an ITC Excellence Scholarship, which allowed me to obtain an incredible amount of knowledge and experience within their walls and spend two years in the Netherlands full of fun, joy, and memorable moments.

I am extremely grateful to both of my supervisors, prof.dr. Claudio Persello and dr. Mila Koeva, for sharing their great insights and experience in this field, guidance, and encouragement during my master's thesis journey. Also, many thanks to PhD candidate Wufan Zhao for consulting me and always being available to answer my queries. I was able to complete a thorough project under the supervision of this esteemed team.

I would also like to thank my committee members prof.dr.ir. Alfred Stein and dr. Mariana Belgiu for their valuable feedback and suggestions, which helped me improve my work.

My appreciation also extends to the friendly and supportive ITC community, to the teachers and friends who made my time in Enschede so enjoyable. Special thanks to my dear friend Tsaqif Wismadi for his unwavering moral support when I worked on my thesis.

Last but not least, I would like to express my gratitude and affection to my parents, Azhar Kenzhebay and Kairat Baidilov, who have always believed in me and supported me in all my undertakings. Their faith and outlook on life propelled me to where I am today, and everything I have accomplished in my life is thanks to them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background and justification

Buildings are essential attributes of an urban environment. Extraction of building contour is widely performed for topographic mapping, cadastral purposes, urban planning, disaster management and population density analysis (Sun et al., 2021). Other applications, such as solar radiation potential assessment to plan solar panel installation, wind flow simulations for pollutant diffusion analysis in the built environment and mobile telecommunication installations necessitate more detailed building geometry information including the roof shape knowledge (Macay Moreia et al., 2013). And thus, to generate 3D building models, reconstruction of the building roof structure is needed.

As buildings are likely to change over time, there is a need for producing accurate models efficiently (Qin et al., 2019). Given the availability of decimetre-resolution aerial images and elevation data, it is possible to extract more detailed information of building outlines and their roof geometry (Alidoost and Arefi, 2016). In this regard, the progress in machine learning gives a great opportunity to develop building extraction methods that consume less time and human labour resources(Luo et al., 2021). Furthermore, recent approaches based on deep learning (DL) algorithms, e.g., Convolutional Neural Networks (CNN), Fully Convolutional Networks (FCN), and Recurrent Neural Networks (RNN), showed high potential to recognize and extract detailed building features (Alidoost et al., 2019; Girard et al., 2020; Nauata and Furukawa, 2020; Qin et al., 2019; Zhang et al., 2020). Nonetheless, there are still remaining problems such as false detection and misclassifications, low computational efficiency, and the fact that the majority of the methods produce output in the raster format, which leaves the scope for further improvement (Hang and Cai, 2020).

The research done in the building extraction field can be divided into two categories based on the output format, which can be either raster or vector. The raster-based output usually tends to have over-smoothed corners and imprecise and irregular contours. Thus, methods with vector-based output are preferred since they address the above-mentioned problems with a regularization process. Besides, the vector-based output is more widely used in Geographic Information System (GIS) applications (Girard et al., 2020; OpenStreetMap contributors, 2017). In recent years, more attention in building polygons extraction was given to the methods based on DL, a subfield of machine learning, which allows neural networks with multiple layers to learn data features at different scales. DL models are currently used in various tasks such as object detection, speech recognition, language processing and others (Lecun et al., 2015). These methods are not new in image classification and segmentation, but relatively novel in building polygons extraction. In the past few years, several valuable techniques have been proposed. For instance, Girard et al.(2020), Zhao et al., (2021a) and Sun et al. (2021) proposed DL-based methods to extract building footprints in vector format. Two successful models, and the basis for further improvement, are frame field learning (Girard et al., 2020) and Polymapper (Li et al., 2019). The idea of the frame field learning framework is to learn building edge directions that are useful to extract regularized building outlines in vector format. It improves segmentation performance and recognizes different types of buildings in size (small and big) and structure (regular and with inner holes). Polymapper, based on CNN-RNN architecture and regularization of graph structures, extracts topological features such as road networks and building footprints. Both methods facilitate straightforward vectorization from remotely sensed (RS) images and present higher performance than Mask R-CNN (Zhou et al., 2019) and PANet (Liu et al., 2018) based methods.

A further step in building information extraction is to obtain the roof structure (*Figure 1*) of the building for a 3D model generation. The roof structure is comprised of outer and inner rooflines connected at their

vertices. Recent methods to fulfil this aim were proposed by Alidoost et al., 2019; Zhang et al., 2020; Zhao et al., 2021. The state-of-the-art methods presented are either end-to-end or consist of two-step approaches. End-to-end techniques output the building rooflines directly in vector format, while the two-step approach first generates output in raster format and then goes through the vectorization step. End-to-end approaches use Graph Neural Networks(GNN) to infer the relationships of the feature lines of the building (Zhang et al., 2020; Zhao et al., 2022). With our work, we propose a two-step approach in which the resulting raster information from the DL framework continues with an efficient vectorization step.

Figure 1. Roof structures of individual buildings and corresponding VHR image

1.2. Research problem

Given the importance of 3D building models in addressing urban issues and the complex and changing nature of buildings, developing an automatic method that reduces costs, time, and human effort is critical. Up to now, there are only limited studies on automatically extracting building roof geometry in vector format. Such studies face problems such as false detection and misclassifications, low computational efficiency and limited to image patches with single buildings(Zhang et al., 2020; Zhao et al., 2021b). To contribute to the progress of roof structure extraction research, we design a deep learning-based method to extract building roof structures directly in a vector format. As mentioned previously, roof structure (*Figure 2*-c) consists of building outlines(*Figure 2*-a), external edges of the building roof, and inner rooflines (*Figure 2*-b), internal intersections of the main roof planes. Besides, in the context of this research thesis, we test the frame field learning idea further building on top of the work done by (Girard et al. 2020). We also take advantage of the study of (Sun et al., 2021b) which proved that height information can improve the building segmentation results. Thus, we aim to generate not only the polygons of the building outlines but also the inner rooflines in a vector format.

Figure 2. a) Building outlines; b) Inner rooflines ; c) Building roof structure

1.3. Research objectives

1.3.1. General objective

The general objective of the research is to design a DL-based method to extract building roof structures in vector format.

1.3.2. Specific objectives

The main objective of the proposed research thesis is to jointly extract the building outlines and inner rooflines in a regularized vector format from VHR images using deep neural networks. To achieve the objective, we set the following specific objectives (SO) and corresponding research questions:

SO 1: To acquire knowledge in frame field learning for building segmentation (Girard et al., 2020);
1. What is the framework of the segmentation process?
2. How was the frame field learning implemented?

SO 2: To prepare the dataset;
1. What input data is needed for the approach?
2. Do the inputs (e.g., roofline vector file) need correction? If yes, what needs to be corrected?

SO 3: To design a DL approach to jointly extract building outline and inner rooflines;
1. How to adapt the Frame Field Learning framework to extract inner rooflines?
2. What backbone is to be used for building outlines and inner rooflines extraction?
3. What loss functions need to be introduced to align and regularize rooflines?

SO 4: To evaluate the accuracy of the proposed approach.
1. What metrics are to be used to assess the accuracy of the approach?
2. How accurate is the result of the approach?
3. What are the strengths and limitations of the approach and how can this be improved?

1.4. Conceptual framework

The conceptual framework *(Figure 3)* depicts the interrelationships between the three main concepts of the research. Our research aims to extract building roof structures in the form of interconnected roof inner lines and outer lines. Sensor technology advancements and the growing availability of large amounts of Earth observation data can help to answer the need for a more precise and scalable roof structure extraction method. And among the latest methods of data analysis, DL algorithms outstand with their state-of-the-art performance. Their key benefit is learning abstract hierarchical representations of data which enable networks to uncover hidden spatial, spectral, and temporal patterns. Modern DL algorithms are filling the

gap between the performance of automated workflows and the demand for accurate and reliable information mandated by real-world applications(Persello et al., 2022). Thus, we use remote sensing data such as very high-resolution aerial images and normalized Digital Surface Model (nDSM) as an input dataset and DL to automize the workflow and achieve cutting-edge performance.

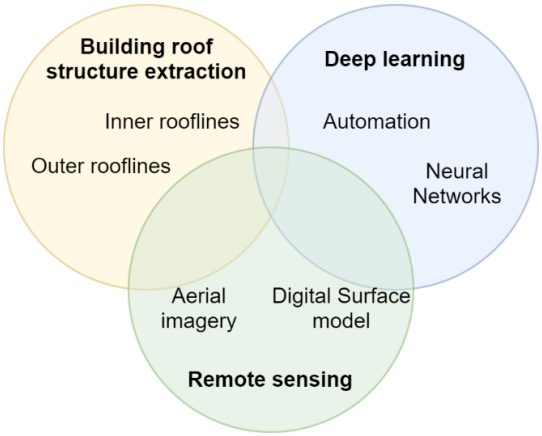


Figure 3. Conceptual framework

1.5. Thesis structure

The structure of this thesis is as described below:

Chapter 1. Introduction
This chapter gives the background and justification of the research, clarifying the research problem, objectives and questions.

Chapter 2. Literature review
Related literature for building roof structure extraction is reviewed in this chapter. Different state-of-the-art techniques are presented in this part.

Chapter 3. Materials and methodology
An overview of the research methodology and used materials is introduced in this chapter, followed by a detailed description of each step, including data preparation, outlines and inner rooflines segmentation, vectorization and post-processing. The details of evaluation metrics are also presented in this part.

Chapter 4. Results
The quantitative and qualitative analyses are presented in this chapter.

Chapter 5. Discussion
This chapter presents a broad discussion of the acquired results and recommendations for further improvements.

Chapter 6. Conclusion
The final remarks of the research and answers to the research questions are given in this concluding chapter.

1.6. Summary

This chapter gives information on the background of the research following with the main problem, general and specific objectives of the study. To summarize, the goal of the research is to jointly extract outer and inner rooflines using deep learning in a regularized vector format for the generation of 3D building models.

2. LITERATURE REVIEW

Building roof structure extraction is essential for many applications such as urban planning, manufacturing and solar potential assessment. Over the last decade, the research in this field has taken different directions from the perspective of data sources, methods and output formats. The overview of the recent studies is given below.

2.1. Data sources for building roof structure extraction

Roof structure extraction has been performed using different Earth observation data. The two main data sources for roof structure reconstruction are Light Detection and ranging (LIDAR) point clouds and remote sensing (RS) imagery. LIDAR point clouds are a suitable data source to reconstruct roof structures (Wang and Chu, 2009) due to their high accuracy (Novacheva, 2008). However, it also has drawbacks on data availability and affordability, outdatedness and the inability to differentiate boundaries with nearby objects (Hang and Cai, 2020). On the other hand, RS imagery, particularly very high resolution (VHR) satellite and aerial images, contains a huge amount of textural and spatial information and, given the lower/no cost, can be obtained for different areas and scales (Hang and Cai, 2020; Wang et al., 2021). Another option is to fuse different datasets, which was proposed in Alidoost et al., (2019); Awrangjeb et al., (2013). However, fusion also has challenges as different characteristics of data sources for the registration process, different spatial resolution and simultaneous availability (Liu et al., 2020). In our research, we focus on using an open-access aerial imagery dataset and nDSM for developing our method. This fusion was performed by Sun et al., (2021) for building outline delineation with a frame field learning framework, which showed higher performance than using solely aerial images.

2.2. State-of-the-art methods in building roof structure extraction

In recent years, there has already been research on methodologies for the recognition and extraction of roof structures in a raster format using DL algorithms, including the works of Alidoost and Arefi (2016), Castagno and Atkins (2018), Partovi et al. (2017), Muftah et al.(2021). Alidoost and Arefi (2016) designed a model-based method that can recognize and label different roof structures using CNN from LiDAR and aerial images. Similarly, Castagno and Atkins (2018) proposed a roof-type classification approach which performs feature extraction using CNN and classification with Random Forest from LiDAR data and satellite imagery. Partovi et al. (2017) designed a hybrid multiple steps method which consists of building contour extraction and refinement, image-based roof type classification using CNN, initialization and enhancement of geometric parameters of the roof models as prior knowledge for the 3D model fitting. The approach performs well on simple buildings but cannot handle complex roof types.

The above-mentioned methods produce results in raster format. However, for urban applications, the main interest lies in vectorized output. Simple vectorization of the raster output is not sufficient to obtain a vectorized output of decent quality for real world applications. Therefore, regularization and simplification must be introduced to obtain straight edges and corners. Since automatically extracting building roof geometry in the regularised vector format is a challenging task, there are only limited studies that address this problem (Alidoost et al., 2019; Zhang et al., 2020; Zhao et al., 2021; Nauata and Furukawa, 2020; Partovi et al., 2017). The state-of-the-art methods presented are either end-to-end or consist of post-vectorization approaches.

Alidoost et al.(2019) proposed an approach to reconstruct 3D model details such as height and rooflines from a single aerial RGB image. Based on an optimized multi-scale convolutional–deconvolutional network (MSCDN), their framework consists of multiple steps for feature extraction and subsequent prismatic and

parametric model generation. The MSCDN outputs line segments (eaves, ridges, hips) which then go through multiple steps. First, they use to create the initial primitive of the building model using eaves. Next, they take advantage of standard Hough transform (SHT) to generate regularized and simplified boundaries(eaves). It calculates the main orientation of the building. Then they use the minimum bounding rectangle (MBR)-based and the minimum bounding triangle (MBT)-based techniques to approximate the polygons and use ridges and hips to divide the roof into building parts. According to the quality metric, the accuracy of linear elements extraction accounted for 91% and 83.4% for two different manually digitized datasets.

Nauata and Furukawa (2020) proposed an algorithm that uses CNN to detect geometric primitives (lines, corners and regions) and integer programming (IP) which collects the information as a planar graph. Similarly, Zhang et al.(2020) propose a method which extracts building features as geometric primitives which form planar graphs from RGB images utilizing their new architecture Convolutional Message Passing Network. The method is highly dependent on pre-processing, computationally inefficient and does not show high accuracy.

Wang et al. (2021) presented an approach for autonomous vectorization and 3D reconstruction using a single-channel photogrammetric DSM and a panchromatic (PAN) image. They start by filtering away non-building objects and enhancing the building shapes of the input DSM with a conditional generative adversarial network (cGAN). A semantic segmentation network is utilized to detect edges and corners of building rooftops using the revised DSM and the input PAN image. Following that, a series of vectorization algorithms for building roof polygons is performed. Lastly, the corrected DSM height information is processed and provided to the polygons to generate a vectorized level of detail (LoD)-2 building model. This method is superior to another similar method (Partovi et al., 2019) by accurately reconstructing most of the building models, however, still has limitations such as missed line segments detection and incompleteness of building models due to the loss of building components.

Gui and Qin (2021) suggest a model-driven approach for reconstructing LoD-2 building models using the "decomposition-optimization-fitting" paradigm. Building detection results are first vectorized into polygons using a "three-step" polygon extraction method, then decomposed into densely connected basic building rectangles prepared to fit primitive building models using a novel grid-based decomposition method. To further enhance the orientation of the 2D construction rectangle, they added OpenStreetMap (OSM) and Graph-Cut (GC) labelling as options. Building-specific parameters are used in the 3D modeling process to maximize the flexibility of employing a small number of basic models. Eventually, building roof types are updated, and nearby building models in one building segment are integrated into a complex polygonal model. Since the proposed strategy has limited model types in their library, it may not be applicable for some types of structures, such as those with dome roofs, as may over-partition building segments with complex shapes.

Zhao et al. (2021) introduced an end-to-end roofline extraction approach using an integrally attracted wireframe parsing (IAWP) framework to generate a planar graph from VHR imagery. In this work, they also incorporated geometric line priors using Hough Transform into deep networks. The results showed that the method outperforms the Conv-MPN architecture proposed by Zhang et al. in F-score metrics by 0.7% for corner points and 8.8% for edges. Besides, the method has higher computational efficiency taking half time and only 0.6 of GPU memory. Nevertheless, this method only works with image patches containing a single building. It thus cannot map entire urban areas from images with the city coverage. Moreover, the approach still results in several missing detections and incorrect roof structure models.

In their most recent work, Zhao et al. (2022) proposed the Roof Structure Graph Neural Network (RSGNN) method that has 2 components: 1) a Multi-task Learning Module (MLM) to extract and match geometric primitives, 2) a Graph Neural Network (GNN) based Relation Reasoning Module (RRM) for roof structure reconstruction. It outperforms state-of-the-art models but still faces similar issues to IAWP, which are missing line detections and single building extraction per patch.

2.3. Summary

This chapter gives an overview of the main data sources and state-of-the-art methods for roof structure extraction. LiDAR pointclouds as input data have high accuracy but can be unavailable, outdated or unaffordable. VHR images, on the other hand, have rich spectral information and can be obtained for large areas at low/no cost. The fusion of different data sources is also a common practice. The recent methods reach good performance however still have disadvantages such as a significant amount of missed and false detections, low computational efficiency, single building extraction per patch, dependence on predefined models library and others.

3. MATERIALS AND METHODOLOGY

3.1. Polygonal Building extraction by Frame Field Learning

The method of Girard et al.(2020) is used to test the frame field learning for roof structure extraction task. The original method extracts regularized building polygons using the DL approach in 3 main steps: building segmentation, frame field learning and Active Skeleton Model (ASM) polygonization. The central idea of the method is to use frame field learning for polygonization to have regularized building outlines with correct corners. The frame field is a 4-PolyVector field that defines each point on the place with 4 vectors {u, -u, v, -v} where two of them are restricted to be opposite to the other two. Since the frame field is used to detect the corners of the building, two directions define the frame with u; v ∈ C. To avoid sign change and relabelling ambiguity, the directions are converted to coefficients $\{c_0,c_2\}$ using the polynomial given in equation 1. The properties of the frame field are followings: 1) at least one frame field direction aligns with the polygon tangent direction of the building along the building wall; 2) both field directions align with the tangent direction at building corners.

$$f(z) = (z^2 - u^2)(z^2 - v^2) = z^4 - c_2 z^2 + c_0 \quad (1)$$

The model consists of the backbone and two branches that output building probability map in two channels (edges and interior) and a frame field with four channels (four vectors representing two directions: ±u, ±v). The backbone architectures used in this paper are UNet16 and UResNet101. More details about the segmentation and frame field losses are given in 3.8.

ASM polygonization step is inspired by the Active Contour Model (ACM) optimization. In this method, the contour of the interior probability map is optimized using an energy function that fits the contour points to the optimal position. In the case of ASM optimization is performed on the skeleton, which, in short, is the graph representation of the edge probability map.

The proposed method significantly outperforms other state-of-the-art methods, e.g., PolyMapper, PANet and others(Li et al., 2019; Liu et al., 2018; Zorzi et al., 2020) in building polygon extraction both in accuracy and computational efficiency. The frame field does not add any cost to the inference while yielding more regular building edges and correct sharp corners. Therefore, in our research, we decided to attempt to modify the Frame Field Learning model to our task and examine if frame field and ASM vectorization will be beneficial for roof structure extraction.

3.2. Overall methodology

In our study, we develop and compare two models for roof structure extraction. Both models are identical in their methodology and implementation details except one of them has frame field learning branches for building outlines and inner rooflines. It is expected that the frame fields will aid in the extraction of more regularized inner rooflines and building outlines with correctly detected corners.

As illustrated in *Figure 4*, the proposed (no-field) method consists of the following steps:

1. Data preparation. This step includes reference data correction, input data tile generation and distribution;

2. Feature map extraction with one of the backbones(Unet16, UResNet101);

3. Building outlines and inner rooflines segmentation. These tasks are performed in separate blocks and simultaneously;

4. Vectorization. The main steps include skeletonization, regularization and simplification of the segmentation output;

5. Post-processing. This includes automated merging and correction of the building outlines and inner rooflines;

6. Method evaluation consisting of pixel-level and line-level accuracy assessments.

In the case of the frame field learning model, we add two frame field learning blocks, one for the outlines and the other for the inner rooflines. They are later used in the vectorization step for regularization of the lines and corner detection of the building.

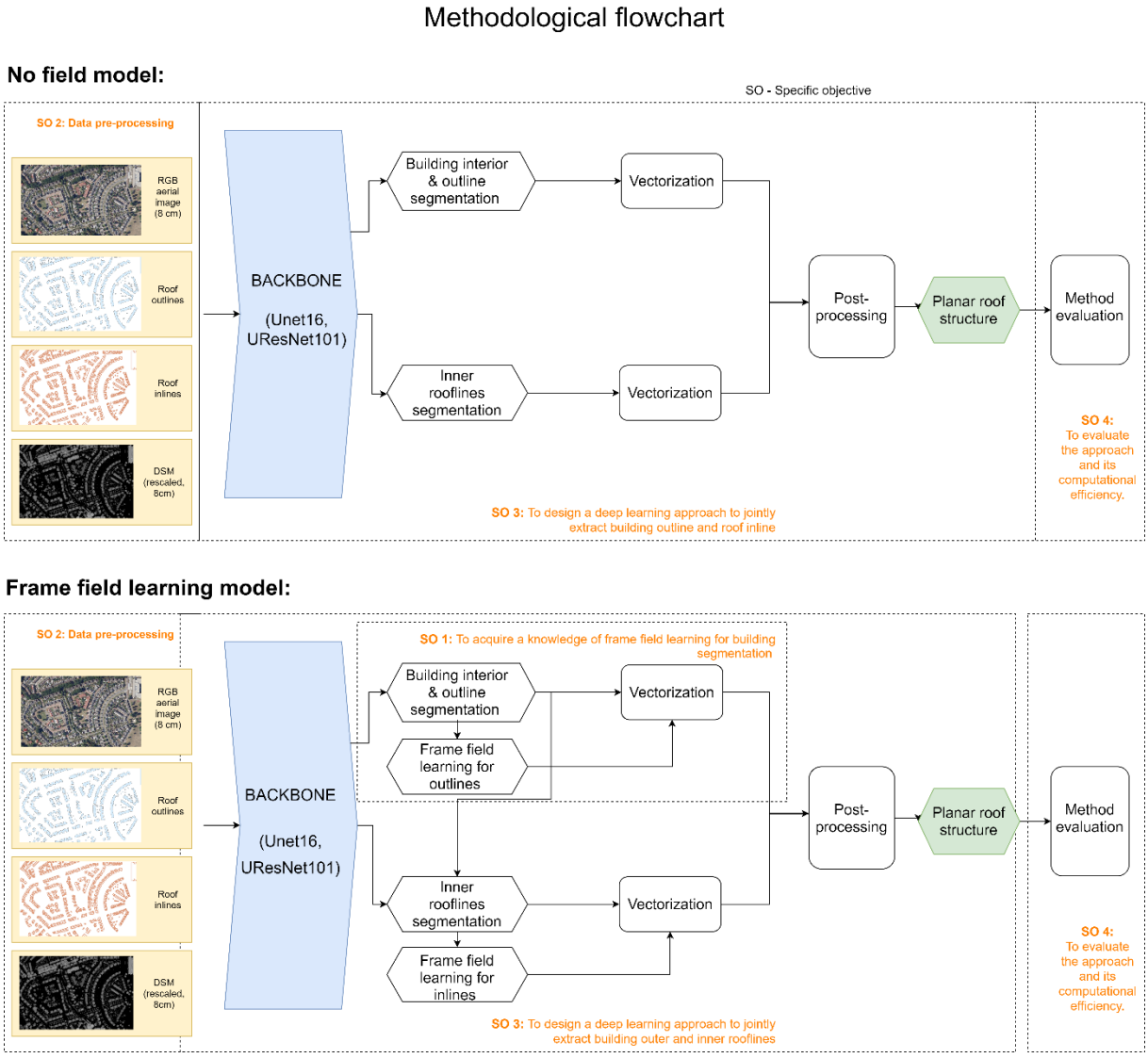


Figure 4 . Methodological flowchart

3.3. Study area

The study area selected for this research is the neighbourhood Stadsveld – t Zwering, a residential area, in Enschede city, Netherlands ((*Figure 5*). The choice of the study area was made due to the availability of the labelled dataset.

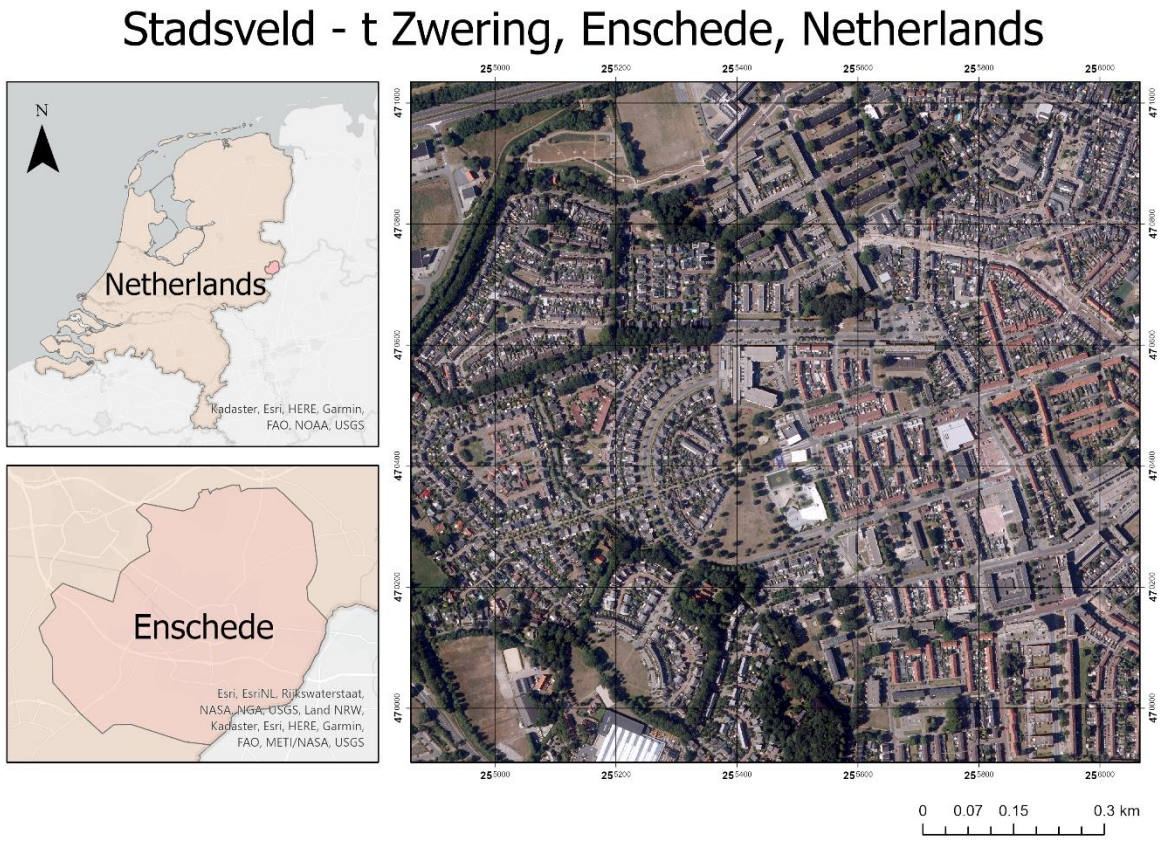


Figure 5. Study area

The dataset contains files mentioned in *Table 1* below.

Table 1. Dataset content

Data	Source
BAG building footprints (vector format)	Public Services On the Map (PDOK)("PDOK," 2013)
Roofline (Eave, Ridge, Hip) (vector format)	Produced by ITC Master's degree graduate Mina Golnia
Orthophoto (8 cm) from aerial imagery, 2021	PDOK
nDSM (50 cm), 2019	PDOK

3.4. Data preparation

The input for our method is an RGB aerial orthophoto of 0.08 m spatial resolution and an nDSM of 0.5 m resolution, building footprint and inner lines reference data. nDSM was resampled to 0.08 m resolution using bilinear interpolation. RGB bands and nDSM were stacked as a 4-band input raster. Since originally reference data for outer and inner rooflines were manually digitized over the 0.25 m resolution image and had some minor mismatches or lack of features (*Figure 6*), it is modified in accordance with the resolution of the 0.08 m. Both 4-band raster and reference data were split into tiles of 1000x1000 size. The size was chosen based on the idea of having multiple buildings in the same tile for later evaluation.

Figure 6. Mismatches of the reference data with 0.08 m resolution RGB image

Tiles distribution. The dataset was divided into training, validation and testing tiles in the proportion 7: 1: 2 respectively (*Figure 7*, *Table 2*). The distribution of the tiles was random to have most of the roof types represented in training. Training and validation tiles were split again into patches of 500x500 pixels size to fit GPU memory and still cover more than one building in one patch.

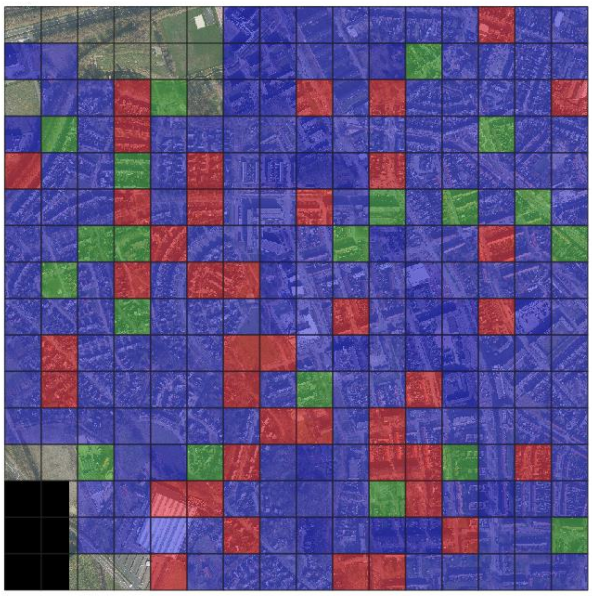


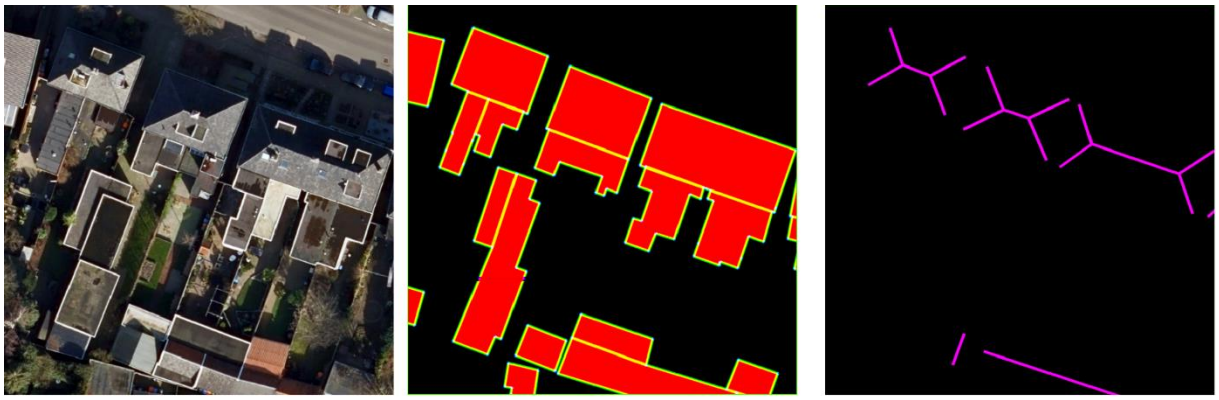
Figure 7. Tiles distribution(1000x1000): train(purple), validation(green), test (red)

Table 2. Dataset tiles distribution

Type	Tile size	Number of tiles
Training	500x500	584
Validation	500x500	84
Testing	1000x1000	42

The tiles were generated using "Create Fishnet" in ArcGIS and were used to clip 4-band raster and shapefile of reference data. The "Clip Raster" tool clips a 4-band raster with a previously generated fishnet. The shapefile of the reference data both for inner rooflines and outlines was divided with the "Split" tool and converted to geoJSON format, the accepted format for the input to the method. The annotation file format geoJSON is an open standard geospatial data interchange format based on JavaScript Object Notation (JSON). It contains the information on the type, coordinate reference system, geometry type of the feature, coordinates and properties.

In the preprocessing step implemented in the method, the building polygons and inner rooflines are rasterized for supervised learning in building interior & outline and building inner roofline segmentation branches. Polygons are rasterized into two bands – building interior and outlines(*Figure 8-b)*, while inner rooflines are rasterized into one band(*Figure 8-c)*. For frame field learning, building contours' and inner rooflines' angles of the unsigned tangent vector were computed.



a) Image patch

b) Building polygons rasterised to interior(red) and outlines(yellow)

c) Rasterized inlines

Figure 8. Pre-processing: rasterization of the reference data

3.5. The backbone of the model

In this study, we will use lightweight UNet16 and pre-trained UResNet-101, the backbone that provided the highest performance in (Girard et al., 2020). The feature extractor has the replaced downsampling section of U-Net with ResNet-101 (He et al., 2016), 101-layer Residual Network architecture and has been pre-trained on ImageNet (Deng et al., 2010). U- Net (*Figure 9*) is a network architecture that is built upon FCN (Ronneberger et al., 2015). It was originally designed for biomedical image segmentation. The architecture consists of downsampling and upsampling paths to output the feature map of the size similar to the input.

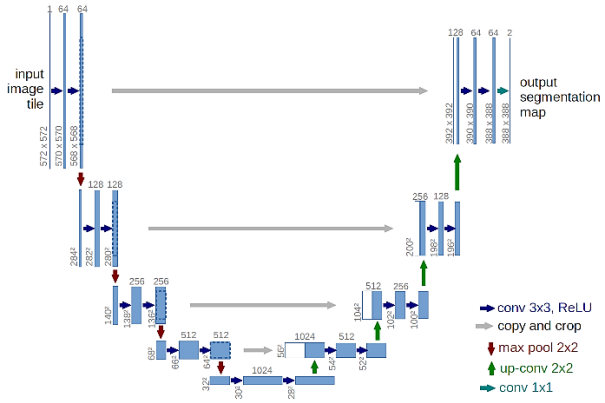


Figure 9. U-Net architecture (Ronneberger et al., 2015)

The backbone takes an image tile with 4 channels (RGB+nDSM) as an input and produces an F-dimensional feature output (F=number of extracted feature maps) with the height and width of input size, which is used in further steps.

3.6. Building outlines and inner rooflines extraction

3.6.1. Building interior & outlines segmentation map

An F-dimensional feature map undergoes a fully convolutional block that outputs a building segmentation map. The block has the structure represented in *Figure 10*. This block consists of a 3x3 convolutional layer, a batch normalization layer, an Exponential Linear Unit (ELU) activation function, another 3x3 convolution, and a sigmoid nonlinearity. The output consists of 2 maps: interior mask and edges (building outlines). The interior mask is used to enforce the edges of the buildings to align their contour and later used in in vectorization to correct the building outlines mask. Tversky loss (Salehi et al., 2017) or the combination binary cross-entropy and dice loss {BCE+Dice} is used for both losses L_{int} and L_{edge} applied on the interior and edge outputs respectively.

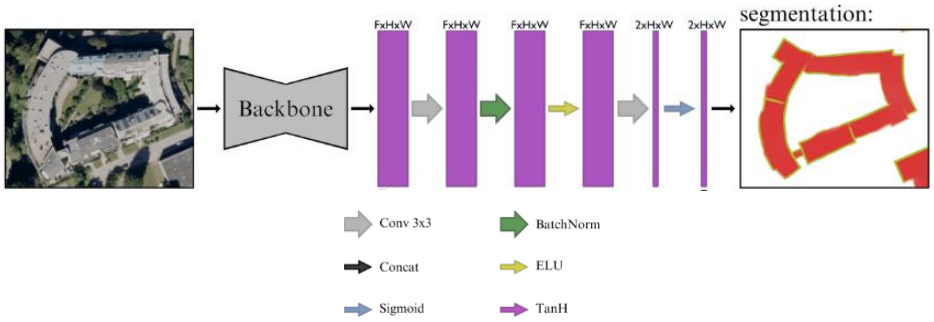


Figure 10. Building segmentation (Girard et al., 2020)

3.6.2. Inner rooflines segmentation map

The F-dimensional feature map and building segmentation map are inputs for a fully convolutional block that will output inner rooflines. The building segmentation map is used as input to guarantee building inner rooflines be inside the building interior. The block has the same structure as for building outline segmentation. The output consists of one channel – an inner roofline segmentation probability map.

3.7. Frame field learning for outlines and inner rooflines

Frame field for outlines. Building segmentation map and F-dimensional map will be further fed to the sub-head for frame field learning. This block consists of a 3x3 convolutional layer, a batch normalization layer, an ELU nonlinearity, another 3x3 convolution, and a tanh nonlinearity, as shown in *Figure 11*. The frame field for outlines will consist of 4 channels that correspond to c_0 and c_2 coefficients which recover 2 directions comprising spatial information. Having 2 directions instead of 1 will facilitate corner detection.

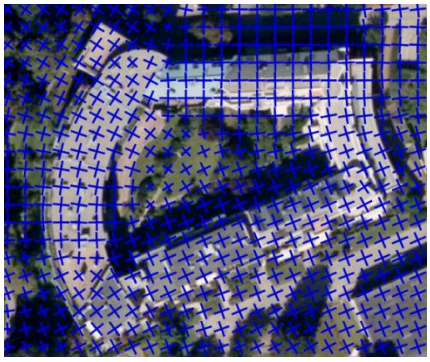


Figure 11. Frame field with learned outlines directions (Girard et al., 2020)

Frame field learning for inner rooflines. The sub-head takes an F-dimensional feature map and inner roofline output to generate a frame field for inner rooflines. The block structure is the same as for outlines. The output consists of 4 channels corresponding to {u, -u, v, -v} vectors as in frame field learning for inner rooflines.

3.8. Loss functions

The total loss function is comprised of multiple loss functions used for different learning branches: 1) outline edge and interior segmentation; 2) inner roofline segmentation; 3) frame field for outlines; 4) frame field for inner rooflines; 5) coupling losses.

Building outline, interior and inner roofline losses. Building outline, interior and inner roofline segmentation tasks use the Tversky (Salehi et al., 2017) or a combination of Cross-Entropy and Dice {BCE +Dice} loss functions for learning.
The {BCE+Dice} loss function is defined by equations 2-4 below:

$$L_{BCE}(y,\hat{y}) = \frac{1}{HW}\sum_{x\epsilon I} y(x)\cdot log(\hat{y}(x)) + (1 - y(x)) \cdot log(1 - \hat{y}(x)), \quad (2)$$

$$L_{Dice}(y,\hat{y}) = 1 - 2 \cdot \frac{|\hat{y}\cdot y|+1}{|\hat{y}+y|+1}, \quad (3)$$

$$L_{\{BCE+Dice\}}(y,\hat{y}) = \alpha \cdot L_{BCE}(y,\hat{y}) + (1-\alpha)\cdot L_{Dice}(y,\hat{y}), \quad (4)$$

where H and W are the height and the width of the image, ŷ is the predicted probability of the pixel being interior/outline/inner roofline, and y is ground truth equal to 1. L_{BCE} is the cross-entropy loss and L_{Dice} is the Dice loss, the combination of which is applied to the interior, outline and inner roofline output of the model. The α is a hyperparameter which is set to 0.25 as it showed good results in a similar application (Girard et al., 2020; Sun et al., 2021b).

Tversky loss is based on the Tversky index which handles the problem of data imbalance and gives a better trade-off between precision and recall. In our case, the number of pixels which contribute to building interior or edges is less than the non-interior or non-edge pixels. The equation of the Tversky index is defined below:

$$S(P,G;\alpha,\beta) = \frac{|PG|}{|PG|+\alpha|P\Gamma|+\beta|G\backslash P|} \quad (5)$$

where P and G are the set of predicted and ground truth binary labels, α and β control the extent of penalties for False Positives and False Negatives, respectively. And Tversky loss is given by the equations below:

$$T(\alpha,\beta) = \frac{\sum_{i=1}^{N} p_{0i}g_{0i}}{\sum_{i=1}^{N} p_{0i}g_{0i} + \alpha\sum_{i=1}^{N} p_{0i}g_{1i} + \beta\sum_{i=1}^{N} p_{1i}g_{0i}} \quad (6)$$

$$L_{Tversky} = 1 - T(\alpha, \beta) \quad (7)$$

where the p_{0i} is the probability of pixel i being an edge/interior and p_{1i} is the probability of a pixel being a non-building. Also, g_{0i} is 1 for ground truth edge/interior for pixel i and 0 for a non-building pixel and vice versa for the g_{1i}.

Frame field losses for building outlines and inner rooflines. The frame field learning is performed separately on outlines and inner rooflines. For frame field for inner rooflines, the reference label is an angle $\theta_\tau \in [0, \pi)$ of the unsigned tangent vector of the inner rooflines, while for outlines it is an angle $\theta_\tau \in [0, \pi)$ of the unsigned tangent vector of the polygon contour. τ is the tangent direction. The three losses are used for learning the frame field for inner rooflines/outlines are computed using the following equations:

$$L_{align} = \frac{1}{HW}\sum_{x\epsilon I} y_{edge}\,(x)\left|f(e^{\theta\tau}i;\ \hat{c}_0(x), \hat{c}_2(x))\right|^2, \quad (8)$$

$$L_{align90} = \frac{1}{HW}\sum_{x\epsilon I} y_{edge}\,(x)\left|f(e^{\theta\tau\perp}i;\ \hat{c}_0(x), \hat{c}_2(x))\right|^2, \quad (9)$$

$$L_{smooth} = \frac{1}{HW}\sum_{x\epsilon I}(||\nabla\hat{c}_0(x)|| + \nabla\,||\hat{c}_2(x))||^2), \quad (10)$$

where H and W are the height and the width of the image, $\mathbf{y_{edge}}$ is outline edge segmentation map/inner roofline segmentation map, c_0,c_2 are output coefficients of the frame field, and $\tau\perp = \tau - \pi/2$.
Each loss computes a specific feature of the field: 1) L_{align} ensures that the frame field is aligned with the tangent directions; enforces alignment of the frame field to the tangent directions; 2) • $L_{align90}$ facilitates the frame field to align with $\tau\perp$ to avoid it from collapsing into a line field; 3) L_{smooth} is a Dirichlet energy that measures how smooth $\hat{c}_0(x)$ and $\hat{c}_2(x)$ are as functions of image location x.
Coupling losses. To ensure mutual integrity between the predicted outputs, we apply coupling losses:

$$L_{interior\ align} = \frac{1}{HW}\sum_{x\epsilon I} f(\nabla\hat{y}_{int}(x);\ \hat{c}_0(x),\ \hat{c}_2(x))^2, \quad (11$$

$$L_{inline\ align} = \frac{1}{HW}\sum_{x\epsilon I} f(\nabla\hat{y}_{inline}(x);\ \hat{c}_0(x),\ \hat{c}_2(x))^2, \quad (12)$$

$$L_{outline\ align} = \frac{1}{HW}\sum_{x\epsilon I} f(\nabla\hat{y}_{edge}(x);\ \hat{c}_0(x), \hat{c}_2(x))^2, \quad (13)$$

$$L_{int\ align} = \frac{1}{HW}\sum_{x\epsilon I} max(1 - \hat{y}_{int}(x), ||\nabla\hat{y}_{int}(x)|\ |_2) \cdot |\,||\nabla\hat{y}_{int}(x)|\ |_2 - \hat{y}_{edge}(x)| \quad (14)$$

$L_{interior\ align}$, similarly to L_{align}, enforces the spatial gradient (SG) of the predicted interior map y_{int} to align with the frame field. $L_{outline\ align}$ aligns the SG of the predicted outline map $y_{outline}$ with the frame field. $L_{inline\ align}$ aligns the SG of the predicted inner roofline edge map y_{inline} with the frame field for inner rooflines. $L_{int\ edge}$ gets the projected edge map equal to the norm of the predicted interior map's spatial gradient.

Final loss. Since the losses have different units, we calculate a normalization coefficient for each loss by averaging its value over a random portion of the training dataset using a randomly-initialized network. The losses are linearly combined after being normalized by this coefficient. The goal of this normalization is to rescale losses to make them easier to balance.

3.9. ASM vectorization of inner rooflines and outlines

Active Skeleton Model vectorization is a framework introduced by Girard et al. (2020). It adapts the Active Contour Model (ACM) method (Kass and Witkin, 1988) to perform optimization on skeleton graphs instead

of contours. The ACM method uses energy minimization functions to fit the vertices to an optimal position. This energy is usually comprised of a term that fits the contour to the image and extra terms that restrict the degree of stretch and/or curvature. The main difference between the two methods is that ACM can only be used for vectorization of isolated buildings, while ASM, thanks to skeleton graph representation, can differentiate the individual buildings with adjacent walls. Skeleton graph (example *Figure 12*) is a collection of paths, polylines, connected with junction nodes, vertices. In the example below, skeleton graph represents two flat-roof buildings with an adjacent wall. Purple and orange polylines consisting of chain of vertices represent non-adjacent parts of the buildings, while cyan color represents common wall. The vertices 0 and 4 are junction nodes shared between three polylines.

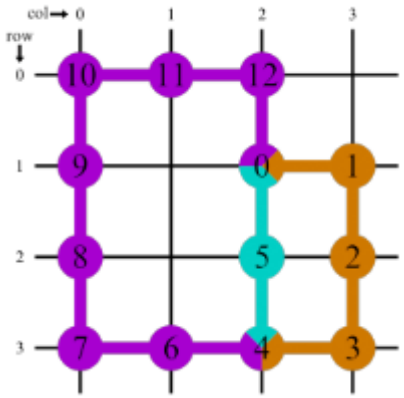


Figure 12. The data structure of a skeleton graph representing two buildings with a shared wall (Girard et al., 2020)

The vectorization for building outlines and inner rooflines follows the same procedure (*Figure 13*) with the differences in a few steps. In general, the framework consists of the thinning method, skeletonization, Active skeleton Model optimization, corner detection, non-corner vertices simplification and post-processing.

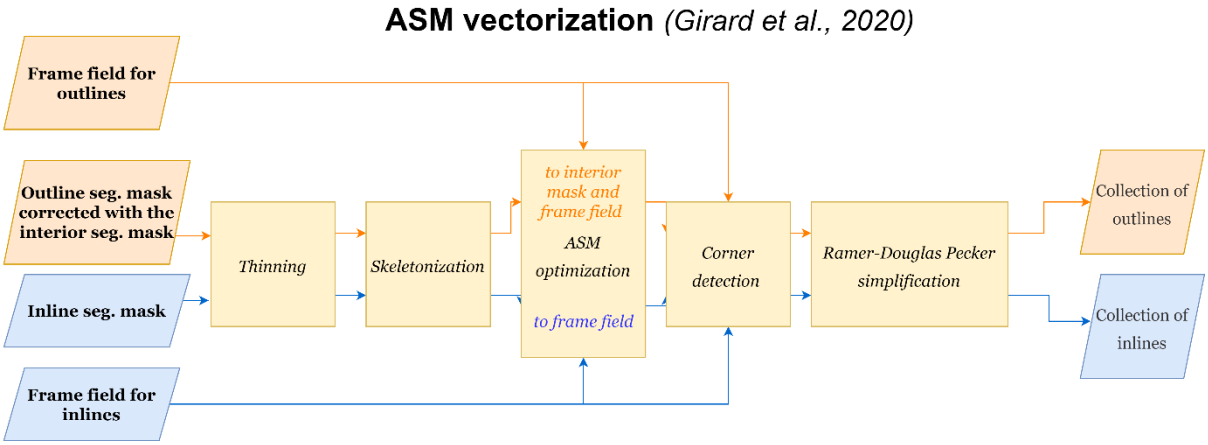


Figure 13. ASM vectorization workflow

First, an inner/outer rooflines segmentation mask is computed from the predicted probability map with a segmentation threshold≥0.5. Second, the mask is converted to a one-pixel wide representation using thinning method (Zhang and Suen, 1984). Third, the skeleton graph, which connects those pixels, is generated with Skan Python library. Third, Active Skeleton Model optimization (*Figure 14*) is performed on the graph to fit vertices to the optimal position by using energy minimization functions: (1) adjusting skeleton paths to the contour of the building interior mask (in case of outlines) and (2) aligning to the frame field for both tasks. Then, with corner detection operation and previously computed frame field vectors,

the paths are split and converted into sub-paths of polylines where each sub-path represents a single wall of individual buildings or inner roofline. Finally, the non-corner vertices are simplified with the Ramer-Douglas-Peucker (RDP) algorithm (Ramer, 1972) which allows to tune the complexity-to-fidelity ratio with the tolerance value of 5 and filtered with the IOU>=0.5 per feature.

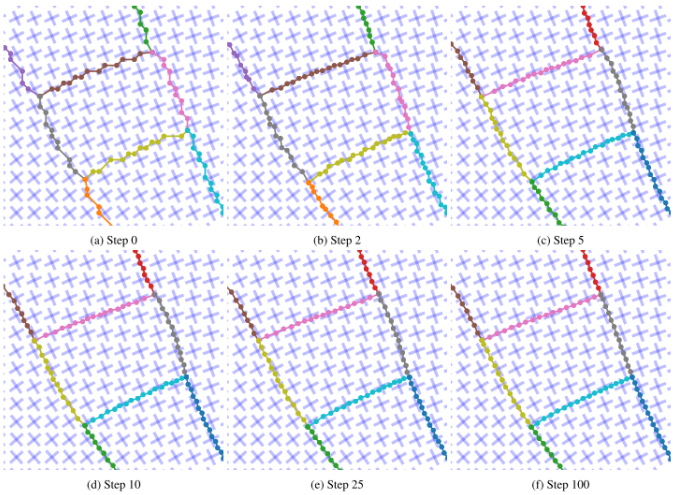


Figure 14. ASM optimization: an iterative process of adjusting polylines using energy function(Girard et al., 2020)

3.10. Simple skeleton vectorization

In the no-field model, since we do not have a frame field for optimization, the simple skeleton vectorization *Figure 15*) is performed. It is comprised of the thinning method, computation of the skeleton graph, conversion to polylines and RDP simplification which result in the collections of outlines and inner rooflines in vector format.

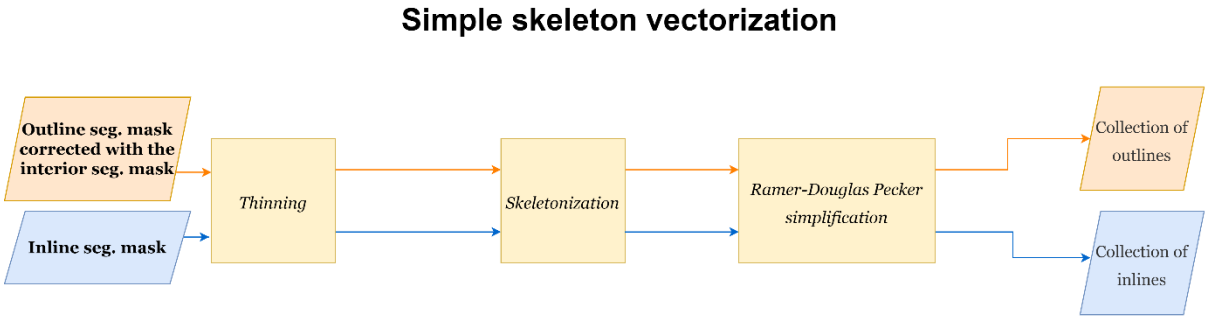


Figure 15. Simple skeleton vectorization

3.11. Post-processing: joining inner rooflines and outlines

Finally, the predicted inner rooflines are matched with predicted building footprint polygons to compose the whole planar roof in the form of intersecting line segments. First, building outlines and inner rooflines are merged in one feature class polylines (Figure 16-a). Then ArcGIS Extend Lines tool is used to automatically correct inner lines that do not reach the building contours (Figure 16-b) and Trim Lines tool for those that go beyond them (Figure 16-c).

Figure 16. Post-processing: a) raw predicted output; b) post-line extension output; c) post-line trimming output

3.12. Implementation details

For our experiment we set the maximum number of epochs to 350, training batch size 4, Adam optimizer with starting learning rate 10^{-3}. GPU used is NVIDIA Titan X (Pascal).

The maximum number of epochs was chosen based on our previous experiments and the best epoch used for the test is set using the validation loss trend. The effective batch size that GPU memory can perform with is four. The optimizer and initial learning rate are selected based on studies by (Girard et al., 2020; Sun et al., 2021a).

3.13. Method evaluation

3.13.1. Pixel-level metric

Intersection over Union is used for pixel-wise evaluation of the results. As given in equation 15, IoU is calculated by division of intersection area by union area of predicted segmentation mask (p) and ground truth (g). All the predicted pixels with the probability value≤0.5 make up the predicted segmentation mask. This metric is used to evaluate the accuracy of predicted interior, outline and inner roofline segmentation masks.

$$IOU = \frac{Area\ (p \cap g)}{Area\ (p \cup g)} \qquad (15)$$

3.13.2. Line-level metric

To evaluate the similarity of predicted lines to ground truth, polygons and line segments measurement (PoLiS) was computed on predicted outlines and inner rooflines. The metric originally calculates the distance between the predicted polygon and ground truth. It takes into account both positional and shape changes by treating polygons as a series of connected edges rather than just point sets. For our output, we did minor changes to perform the procedure on the line segments. So, a PoLiS distance between predicted line segments A and ground truth B is calculated with equation 16, by taking the average of the distances between each vertex $a_j \in A$, j = 1,..., q of line A and the nearest point b ∈ ∂B (not necessarily a vertex) on line B plus the average of the distances between each vertex $b_k \in A$, k = 1,..., r of line B and the nearest point a ∈ ∂A on line A. Normalization factors (1/2q) and (1/2r) are used to calculate the total average dissimilarity per pair of predicted and reference polygons. The PoLiS distance units are the same as the line segment vertices unit.

$$p(A,B) = \frac{1}{2q}\sum_{a_j \in A} \min_{b \in \partial B} \| a_j - b \| + \frac{1}{2r}\sum_{b_k \in B} \min_{a \in \partial A} \| b_k - a \| \qquad (16)$$

The PoLiS distance between two objects is illustrated in *Figure 17*. Similarly, the distance is computed between two line segments. The black line depicts the distance between a vertex of one polygon to the closest point of another polygon. The predicted polygon is represented with the orange color. The bold blue

and dotted blue lines show two different ground truth representations of the same building with the same vertices. However, the PoLiS distance calculated for them is different since the distance to the closest point differs in the upper right corner. Thus, PoLiS distance considers shape changes.

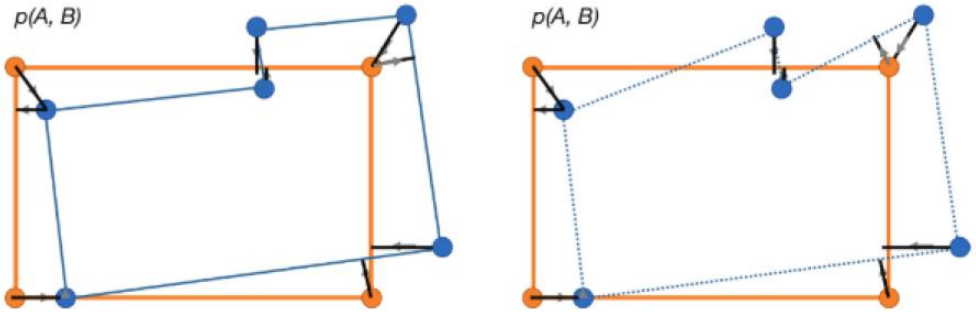


Figure 17. PoLiS distance between predicted building A (orange) and reference building (blue) marked with a black line (Zhao et al., 2021a)

For our task, to ensure that the metric is evaluating the roof structure per building, we calculate PoLiS distance per line segment, then average it for individual building and then on the test set.

To further evaluate the accuracy of our model, we introduce Precision, Recall and F-score with the specific PoLiS tolerance value. Precision indicates the fraction of the predicted outer/inner rooflines being real outer/inner rooflines of the building on the ground. Recall indicates the fraction of the reference outer/inner rooflines being predicted by the model. F-score combines precision and recall in a form of harmonic mean. Since not all the predicted line segments can be correct, we set the tolerance value for geometric precision PoLiS≤0.5 m and consider the line segments with PoLiS distance below this value as correctly predicted. There is no reference to rely on for setting this tolerance, thus we derive it from the resolutions of our input data. The resolution of the RGB image is 0.08 m which can be considered a too strict threshold, so we use the original resolution of nDSM which is 0.5 m as PoLiS tolerance. The tolerance value is set with the consideration of results applicability. Then, we compute Precision, Recall and F-score using the given equations 17-19, where True Positive is the number of predicted line segments with the PoLiS≤0.5 m, False Positive is the rest of the predicted line segments and False Negative is the correct line segments that were not predicted by the model.

$$Precision_{PoLiS\leq 0.5} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (17)$$

$$Recall_{PoLiS\leq 0.5} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (18)$$

$$F1 - score_{PoLiS\leq 0.5} = 2 \times \frac{Precision\ x\ Recall}{Precision + Recall} \quad (19)$$

3.14. Summary

In this chapter, a detailed step-by-step explanation of the two proposed methods is given. Both methods share data preparation, inner and outer roofline segmentation and post-processing steps. However, one of the models has additional frame field learning branches for both of the rooflines and ASM vectorization step where frame fields are utilized. Another no-field model undergoes a simple skeleton vectorization procedure. The methods are evaluated using pixel-level IoU metric, line-level PoLiS and Precision, Recall and F-score with the PoLiS≤0.5 m. Both models also have the same implementation details. The maximum number of epochs is 350, the training batch size is 4 and the used optimization algorithm is Adam with starting learning rate 10^{-3}.

4. RESULTS

Throughout our study, we performed experiments with multiple models:

1) No-field model with UResNet101 backbone pre-trained on ImageNet dataset and Tversky loss;
2) Frame field learning models with UResNet101/UNet16 backbone pre-trained on ImageNet dataset and Tversky/{BCE+Dice} losses;
3) Frame field learning UResNet101 models pre-trained on the whole Enschede city BAG building with Tversky/{BCE+Dice} losses.

However, in the results section, we will only look at three of the best performing models that use the same pre-trained UResNet101 backbone but have different losses – two frame field learning models with Tversky and {BCE+Dice} losses and the no-field model with Tversky loss. Since we do not have a frame field needed for ASM vectorization in the last model, we perform simple skeleton vectorization. Other models, frame field learning model using the UNet16 backbone and frame field learning model UResNet101 with pre-trained weights on the Enschede footprint dataset, produced lower quality results, possibly because the former model is too simple for our task and not pre-trained as UResNet101, and the latter model did not significantly improve after pre-training for segmentation of building interiors and outlines and performed worse on inner rooflines segmentation.

4.1. Quantitative analysis

We analyse our models' outcomes from pixel-level and line-level perspectives. As has been outlined in 3.13, the pixel-wise evaluation is performed using IoU for building outlines and inner rooflines segmentation maps, while to evaluate line-level accuracy we compute the PoLiS metric. To assess inner and outer rooflines detection accuracy, Precision, Recall and F-score with the PoLiS≤0.5 m threshold are calculated.

Table 3 shows the IoU achieved on the predicted building interior, outer and inner rooflines segmentation map. The frame field learning UnetResnet-101 model with Tversky loss function has a higher IoU on almost all building elements, with values of 0.85, 0.37, and 0.35 for the building interior, outer and inner rooflines, respectively compared to the same model with {BCE+Dice} loss. This indicates that the model predicts better than the frame field learning model with {BCE+Dice} loss function and almost the same as the model without frame field learning. All models perform much better when it comes to predicting the interior of the building since the building interior is made up of all the pixels that correspond to the footprint. Predicting outlines and inner rooflines, on the other hand, is a more difficult task as it requires predicting line elements with far fewer pixels than the building footprint (*Figure 8*).

Table 3. IoU of the predicted interior, outlines and inner rooflines probability maps

Model	$IoU_{interior}$	$IoU_{outlines}$	$IoU_{inner\ rooflines}$
UnetResnet-101, Tversky	0.85	0.37	0.35
UnetResnet-101, {BCE+Dice}	0.81	0.22	0.22
UnetResnet-101, Tversky, no field	0.85	0.38	0.32

Line-level evaluation (*Table 4*). The model with Tversky loss and no field outperforms the other models, with an average PoLiS distance of 3.5 m for outlines and 1.2 m for inner rooflines. The frame field learning model with {BCE+Dice} loss performs worse than the frame field learning model with Tversky loss, with the PoLiS distance for outlines (4.2 m) larger by 0.6 m and inner rooflines (3.9 m) by 1.9 m.

Table 4. PoLiS distance of outlines and inner rooflines

Model	$PoLiS_{outlines\ (m)}$	$PoLiS_{inner\ rooflines\ (m)}$
UnetResnet-101, Tversky	3.6	2
UnetResnet-101, {BCE+Dice}	4.2	3.9
UnetResnet-101, Tversky, no field	3.5	1.2

Using the PoLiS threshold for defining our true positive predictions, we calculated the Precision, Recall and F-score for both building outlines and inner rooflines. According to *Table 5,* the model without a frame field outperforms in almost all the metrics, constituting 0.28, 0.34, 0.31 for Precision $_{PoLiS\leq0.5}$, Recall $_{PoLiS\leq0.5}$ and F-score $_{PoLiS\leq0.5}$ for building outlines respectively. For the inner rooflines, the Precision $_{PoLiS\leq0.5,}$ Recall $_{PoLiS\leq0.5}$ and F-score $_{PoLiS\leq0.}$ are 0.72, 0.47 and 0.57 respectively. *Table 5* also demonstrates that the models' performances are not yet sufficient for outlines, but that they perform better for inner rooflines. This mostly happens due to two reasons. First, the model occasionally misses the shared wall between the adjacent buildings (*Figure 18-a*) in which case the computed PoLiS distance will have a high value. Second, the model predicts very small buildings (*Figure 18-b*), e.g., storage sheds, which are not included in the reference data. In this case, the PoLiS will be calculated to the closest line segment that does not actually correspond to the predicted building and this also results in a high value.

Table 5. Precision, Recall and F-score for the inner rooflines and outlines with PoLiS≤0.5

Model	Outlines			Inner rooflines		
	Precision $_{PoLiS\leq0.5}$	Recall $_{PoLiS\leq0.5}$	F-score $_{PoLiS\leq0.5}$	Precision $_{PoLiS\leq0.5}$	Recall $_{PoLiS\leq0.5}$	F-score $_{PoLiS\leq0.5}$
UnetResnet-101, Tversky	0.26	0.34	0.29	0.59	0.39	0.47
UnetResnet-101, {BCE+Dice}	0.10	0.28	0.15	0.17	0.46	0.25
UnetResnet-101, Tversky, no field	0.28	0.34	0.31	0.72	0.47	0.57

Figure 18. Cases with high PoLiS distance: a -missed detections of adjacent walls (PoLiS – 1.56 m); b – storage sheds on the backyard which are not included in the reference data (PoLiS – 3.56 m 7.01 m for both sheds)

4.2. Qualitative analysis

Figure 19 illustrates the post-processed results obtained with the three models, frame field learning UResNet101 models with Tversky and {BCE+Dice} losses and no field UResNet101 model with Tversky loss, as well as corresponding reference data. Intuitively, it can be seen that all models in general perform well on the test data. However, the no-field UResNet101 model with Tversky loss delivers considerably better results with straighter and accurately detected roof lines, more aligned to the roof outer and inner edges(*Figure 19.(1-5)-d)*. The frame field model with Tversky loss performs slightly worse than the no-frame field model by misdetecting (*Figure 19.4-c)* or missing inner rooflines (*Figure 19.3-c)*. Nonetheless, we can observe the contribution of the frame field to the corner detection procedure. Both models are better trained at predicting outer rooflines and ridges, the horizontal line on the intersection of two opposite roof slopes, because the reference dataset has much more of their examples. With more examples of roof hips and valleys, the outwards and inwards diagonal joints formed by the intersection of two roof slopes, it is certain that the model predictions will have substantial improvement on them too.

UResNet101 model with {BCE+Dice} loss fails to have straight walls and correct corners since the probability maps for inner rooflines and outlines do not have clear predictions for them, the example for which in comparison with ground truth and the other model can be seen in *Figure 20*. Besides targeted outer and inner rooflines the model has noisier results and detects with high probability other roof elements or installations such as chimneys(*Figure 19.1-b)*, solar panels (*Figure 19.5-b)* and skylights(*Figure 19.(2,4)-b)*, etc. At the same time, the model cannot partition well the adjacent buildings. All models misdetect trees as part of the buildings (*Figure 19.1-(b-c))* since some of the buildings in our training data are also covered by a tree, and secondly, perhaps the 4th band nDSM, in which trees sometimes have the same height as the near building, confuses the network. However, the UResNet101 model with {BCE+Dice} loss also has false positives on trees not surrounded by any building (*Figure 19.(2,4)-b)* indicating that the model sometimes cannot differentiate between building and tree. False positives for small buildings are observed among all models. Most of them are garden or storage sheds, and, as mentioned before, not all of them are digitized as ground truth data.

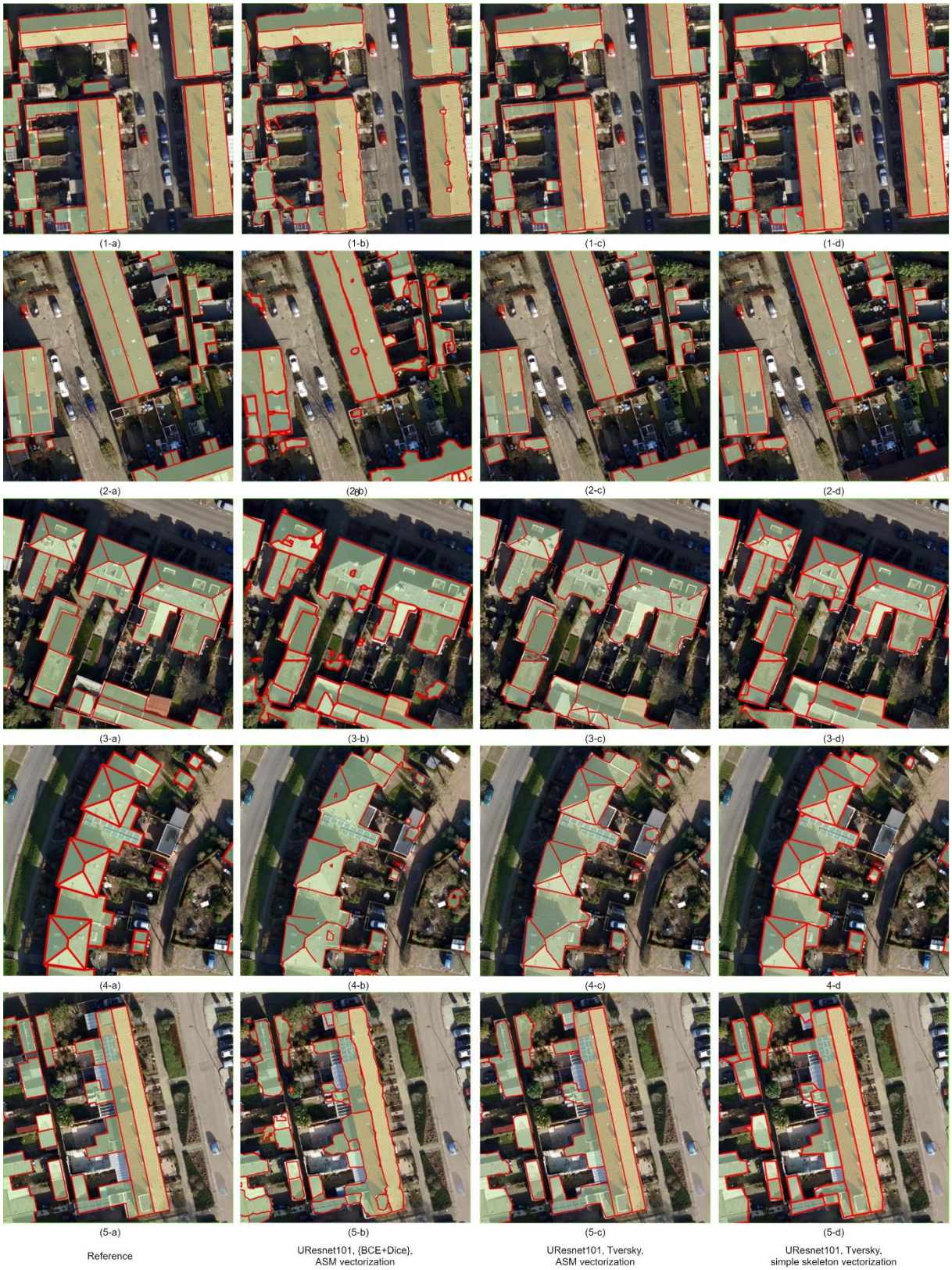


Figure 19. Results obtained with three models: UResNet101 with {BCE+Dice} loss, UResNet101 with Tversky loss, no-field UResNet101 with Tversky loss and corresponding reference data

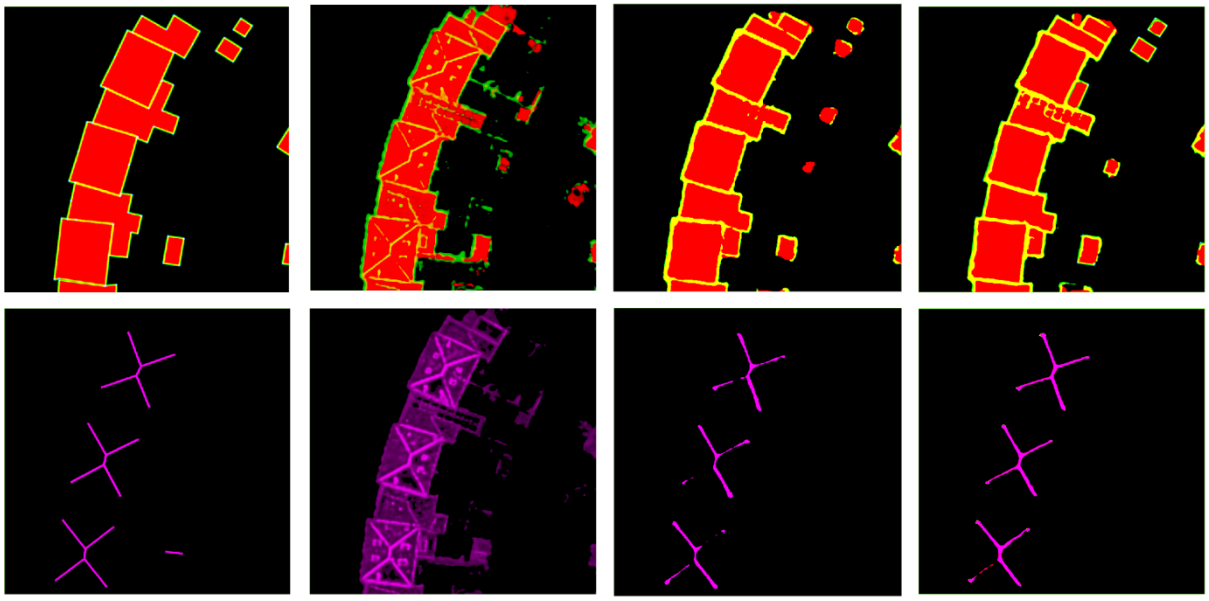


Figure 20. Segmentation results: first row - interior (red) and outline(yellow) probability maps; second row – inner roofline(purple) probability maps

Figure 21 shows the rooflines extracted by three models, as well as the corresponding reference data for comparison. The outlines and inner rooflines extracted by the model with Tversky loss have PoLiS distances of 0.16 m and 0.05 m, respectively, while the values for the model with {BCE+Dice} loss are 1.87 m and 4.2 m. As illustrated in the figure below, the latter (*Figure 21-b*) extracts roof elements outside of the scope of our interest and the true inner rooflines both in the predicted outline and inner roofline probability maps, resulting in two overlapping lines. The UResNet101 model without the frame field learning (*Figure 21-d*) obtained almost the same results as the identical model with Tversky loss but with a frame field learning branch(*Figure 21-c*).

Figure 21. Rooflines of the building extracted by the models with {BCE+Dice} and Tversky losses. UResNet with {BCE+Dice} :PoLiS outline – 1.87 m, PoLiS inner roofline – 4.2 m; UResNet101 with Tversky : PoLiS outline – 0.16 m, PoLiS inner roofline – 0.05 m; UResNet101 with Tversky, no field : PoLiS outline – 0.13 m, PoLiS inner roofline – 0.17 m

4.3. Summary

In this chapter, three models are quantitively and qualitatively compared. The models with Tversky loss with and without frame field perform better than the model with {BCE+Dice} loss at extracting roof structure. Between models with Tversky loss, the frame field learning model slightly outperformed the no-field model on inner roofline segmentation with the IoU value of 0.35 and performed a little worse on outlines, 0.37. The no-field model showed better results on PoLiS distance with the values of 3,5 m and 1,2 m for outlines and inner rooflines respectively. In addition, the no-field model had a higher PoLiS-thresholded F-score of 0.31 for outlines and 0.57 for interior rooflines.

5. DISCUSSION

5.1. Reflection on the performance of frame field learning model

In the previous section, we compared the performance of the three models – two frame field learning models with the {BCE+Dice} and Tversky loss respectively, and no field model with Tversky loss. Both models with Tversky loss show better and similar quantitative results for the roof structure extraction task. Qualitatively, the model without the frame field is better at extracting the inner lines of the roof. The key distinction between these two models is that the frame field learning model has two additional branches for learning frame fields for inner rooflines and outlines. The losses for frame field learning ensure the smoothness of frame field lines and not collapsing of directions into one. The latter means always having 90 degrees angle between two outlines/inner rooflines. While this rule can be useful for most of the buildings due to their right corners, it is not facilitating the extraction of inner rooflines as they do not have the right angle in between. However, according to the results (*Figure 19.(1-5)-b,c)*, the frame field learning for outlines still did not improve the results of outline vectorization and frame field learning for inner rooflines has a negative effect on the vectorization of inner rooflines. The frame field learning showed considerable improvement to the vectorization step in Girard et al. (2020). Perhaps, the contribution of the frame field becomes small when using the very high-resolution image and height information as they provide a rich amount of details. It was expected that the vectorization with the inner and outer rooflines orientation from frame fields would result in more regularized edges and correctly detected corners. However, as can be observed from *Figure 19*, the simple skeleton vectorization procedure without the frame field has similar output in terms of regularization and corner detection. Besides, even though, the segmentation of the frame field learning model has a slightly higher IOU, it is still predicting worse in some cases of inner rooflines (*Figure 22-b*). There are two reasons for this: 1) not enough examples of the inner roof elements in the training data compared to outlines; 2) having a combined loss for four tasks can be confusing for the model during training. Overall, this implies that the frame field's influence is negligible. However, learning about and developing this method contributed significantly to the acceptable performance of another model, UResNet101 with Tversky loss but no frame field:

1) testing the model with the pre-trained UResNet101 backbone. As can be observed from the performance of our no-field model, this backbone alone gives quite a good performance for segmentation of rooflines;
2) branching the tasks of building footprint segmentation and frame field learning which inspired us to branch the task of interior & outline and inner roofline segmentation;
3) using interior & outline segmentation output as a 5th band input to the inner roofline segmentation task. Thanks to this, we were able to restrict the prediction of inner rooflines within the building footprint;
4) using interior building mask to correct the prediction of outlines;
5) thinning method and the following computation skeleton graph, which are the core steps of the ASM vectorization procedure. The thinning method converts the predicted segmentation into a one-pixel wide representation that is practical for the next step. The skeleton graph is extremely beneficial in the case of adjacent buildings with common walls.

5.2. Benefits and drawbacks of the roof extraction approach

Since the best roof extraction approach is a no-field model with a pre-trained UResNet101 backbone and Tversky loss, we outline the advantages and disadvantages of this model. The benefits of the method are given as followings:

1) no handcrafted features were used. To improve the segmentation, vectorization and corner detection we tried to implement frame field learning. Even though it adds only a small cost to the training and inference time, it does not improve the results.
2) the model can perform roof structure extraction for multiple buildings in one image patch compared to other state-the-art roof structure extraction methods. This advantage facilitates both training and prediction since for training there is no need to selectively generate an image patch and reference having one building, and during prediction, we can output multiple roof structures at once;
3) the model can detect the shared walls of the buildings with the usage of outline segmentation and computation of the skeleton graph;
4) the method outputs the closed building outlines thanks to the separate branching for building interior & outline segmentation and correction of outlines with the contour of the building interior.

The proposed method, however, has several drawbacks. First, the output can have missed detections for which we apply post-processing with the extend/trim lines tools. Though, we still have missed inner rooflines *(Figure 22-c)* since the extension is limited to the distance of up to 4 m and will only work if there is at least some part of the line segment predicted. Second, since the extension is automatic and attempts to join the closest endpoints, it can also lead to odd results, as can be observed in *Figure 23*.

Figure 22. Limitations of the method: missed predictions of the inner rooflines(yellow circle) and odd results of the extension(purple circle)

Figure 23. Odd results of the extension procedure

In contrast, the model is better at predicting simpler roofs consisting of only outlines and ridges, as shown in *Figure 24*. The majority of the inner rooflines in our dataset belong to the ridges from what we can deduce that the model prediction could improve if the reference dataset had more examples of other roof elements such as hips and valleys.

Figure 24. Predicted roof structures with straight walls and correct corners

Lastly, the model predicts objects such as trees as part of the building if they stand at a near distance since they may have a similar height to the building (*Figure 25*).

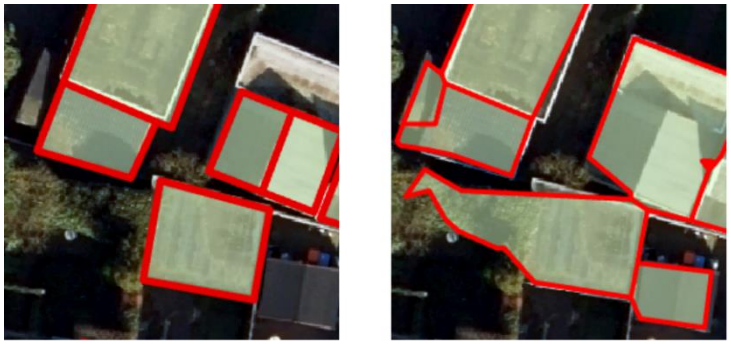


Figure 25. False-positive example - a tree extracted as part of a building

The main causes of the method's drawbacks are the insufficiency of training data and mismatching of the buildings in training data with their real-life appearance. Even though the pre-processing was performed on the reference dataset, due to the time limitations the correction was only made on the outlines and inner rooflines with severe mismatches. Another cause is mismatching between the RGB image and nDSM as they have been generated in different years, 2021 and 2019 respectively.

5.3. The applicability of the method and recommendations for improvement

Having in mind the previously discussed advantages and disadvantages of the method, the proposed method can be considered useful for urban applications. It can be improved using the following recommendations which can be seen as suggestions for the further studies:

1) The collection of a larger amount of the training data; this can definitely improve the performance of the model, particularly for the inner rooflines;
2) Performing nDSM refinement before using an input to the network. This has been done in the proposed method of (Wang et al., 2021). Using refined nDSM will facilitate the elimination of trees in the predictions of the model.
3) Incorporation of an additional final block based on Graph Neural Networks will help to ensure the connectivity of the rooflines, which was done in (Zhang et al., 2020; Zhao et al., 2022). For this task besides predicting rooflines, we can add an extra branch for predicting vertices and take advantage of existing skeleton graph computation. This will be practical as it can possibly substitute the imperfect post-processing step of extension and trimming of rooflines.

We trained and tested our model in a typical Dutch residential neighbourhood with a variety of roof types. When the same method is used in different geographical locations, the results may vary. Our pre-trained model can have good spatial transferability to the other residential area in the Netherlands or another country if the area has a similar residential architecture design. However, if the new test area has non-repeating buildings with complicated roof structures (e.g., New York, Tokyo, Sofia), the pre-trained model may produce poor results. Even if we retrain the model on a subset of building roofs in the selected area, the test results may be unsatisfactory because the roof types are complex and unique to the training and test sets. On the other hand, if most of the buildings in the test area have flat roof types and are not connected, the model will perform better because there will be no need to detect the inner rooflines and outlines of the buildings with adjacent walls. Furthermore, as the model can detect the building with shared walls, this method can also be used for slum areas where the built-up environment is very dense. Besides, with some modifications, the method can be used for the extraction of road maps, cadastral or agricultural boundaries.

6. CONCLUSION

The roof structure is compulsory information for many 3D modelling applications. 3D models with roof geometry information are used for various purposes such as solar radiation potential assessment to plan solar panel installation, wind flow simulations for pollutant diffusion analysis in the built environment and mobile telecommunication installations planning. Besides, the buildings can get constructed, reconstructed or demolished throughout time. Thus, there is a need for an efficient way of extracting and updating roof structure information. On that account, the thesis is focused on developing a method to extract building roof structures in a regularized vector format.

In this study, we use open-access very high-resolution aerial imagery for spectral information and nDSM for 3D information as the input data and DL as the main tool. We primarily create two DL models comprised of segmentation and vectorization procedures, one of which also learns the frame fields. Frame field learning aids segmentation by imposing additional losses that force inner rooflines/outlines to align with the frame field, and it is used in vectorization to detect corners and regularize line segments. Another no-field model is utilized to evaluate the contribution of the frame field to the method's performance.

According to our experiments, both models showed quite good performance in extracting building roof structures. The frame field learning model slightly outperformed the no-field model on inner roofline segmentation with the IoU value of 0.35 and performed a little worse on outlines, 0.37, while the no-field model showed better results on PoLiS distance with the values of 3,5 m and 1,2 m for outlines and inner rooflines respectively. Besides the no-field model scored higher on PoLiS-thresholded F-score for outlines and inner rooflines, having, 0.31 and 0.57 respectively. Visually, the no-field model obtained better results with straighter walls and fewer missed detections of inner rooflines. Thanks to the computation of the skeleton graph, it can predict buildings with common walls. However, it still has limitations such as predicting trees as false positives, extracting building shapes inaccurately, and having an imperfect post-processing procedure that can lead to odd outcomes.

To conclude, in this study, the frame field did not improve the performance model. Perhaps, in the case of using the limited amount of data and height information, the contribution of the frame field becomes insignificant. Secondly, the better-performing no-field model can be applied for roof structure extraction task with some improvements to be made. In further studies, collecting more training data will benefit the performance. Besides, preliminary nDSM refinement can remove the false positive predictions of objects other than buildings. Lastly, adding another block based on the GNN can help with the retaining connection within building roof elements. Moreover, with some changes to the segmentation block, the proposed method could also be used for the extraction of road maps, cadastral or agricultural boundaries and slum studies.

6.1. Answers to research questions

SO 1: To acquire knowledge in frame field learning for building segmentation (Girard et al., 2020);

1. **What is the framework of the segmentation process?**

The framework of the segmentation process consists of feature extraction and building interior and outline segmentation blocks. For feature extraction pre-trained UResNet101 backbone or lightweight UNet16 can be used. The interior and outline segmentation block consist of a 3x3 convolutional layer, a batch normalization layer, an Exponential Linear Unit (ELU) activation function, another 3x3 convolution, and a

sigmoid nonlinearity. The output consists of 2 maps: interior mask and edges (building outlines). The interior mask is used to enforce the edges of the buildings to align their contour. Tversky loss (Salehi et al., 2017) or the combination of binary cross-entropy and dice loss {BCE+Dice} is used for losses applied on the interior and edge outputs.

2. **How was the frame field learning implemented?**

Frame field is a 4-D PolyVector field that helps to extract more regularized building boundaries with the correctly detected corners. Building segmentation map and F-dimensional map are further fed to the sub-head for frame field learning. This block consists of a 3x3 convolutional layer, a batch normalization layer, an ELU nonlinearity, another 3x3 convolution, and a tanh nonlinearity The frame field for outlines will consist of 4 channels that correspond to c_0,c_2 coefficients which recover 2 directions comprising spatial information. Having 2 directions instead of 1 facilitates corner detection.

SO 2: To prepare the dataset;

1. **What input data is needed for the approach?**

As an input, the 0.08 m resolution RGB image and 0.5 m resolution nDSM resampled to the RGB image resolution were used. We train the model using reference data for building outlines in polygon shapefile format and for building inner rooflines in a form of polylines in shapefile format which are rasterized in pre-processing step.

2. **Do the inputs (e.g., roofline vector file) need correction? If yes, what needs to be corrected?**

Yes, the nDSM needed to be resampled to 0.08 m resolution and reference data required manual correction. The ground truth data had mismatches with the RGB image such as non-existent buildings, reconstructed buildings, new buildings and not correctly digitized inner rooflines and outlines.

SO 3: To design a deep learning approach to jointly extract building outlines and inner rooflines;

1. **How to adapt the Frame Field Learning framework to extract inner rooflines?**

The sub-head takes an F-dimensional feature map and inner roofline output to generate a frame field for inner rooflines. The block structure is the same as for outlines. The output consists of 4 channels corresponding to {u, -u, v, -v} vectors as in frame field learning for inner rooflines.

2. **What backbone is to be used for inner rooflines extraction?**

The same backbones are shared between the outline, inner roofline segmentation branches and frame field learning branches for outlines and inner rooflines. We used lightweight Unet16 and deeper UResNet101 with pre-trained weights on the ImageNet dataset.

3. **What loss functions need to be introduced to align and regularize rooflines?**

Tversky and the combination of BCE and Dice losses are used interchangeably for building outlines and inner rooflines segmentation. Besides, there are smoothing, frame field aligning and coupling losses used for training and incorporation of the frame fields.

SO 4: To evaluate the approach accuracy.

1. **What metrics are to be used to assess the accuracy of the approach?**

The results were assessed using pixel-level and line-level metrics. For pixel level, the IoU for predicted building interior, outlines and inner rooflines was computed. At the line level, we used the PoLiS metric that computes the distance that considers shape and positional changes between line segments and polygons. Using the PoLiS threshold for defining our true positive predictions, we subsequently calculated the Precision, Recall and F-score for both building outlines and inner rooflines.

2. **How accurate is the result of the approach?**

According to our experiments, both models showed quite good performance in extracting building roof structures. The frame field learning model slightly outperformed the no-field model on inner roofline segmentation with the IoU value of 0.35 and performed a little worse on outlines, 0.37, while the no-field model showed better results on PoLiS distance with the values of 3,5 m and 1,2 m for outlines and inner rooflines respectively. Besides the no-field model scored higher on PoLiS-thresholded F-score for outlines and inner rooflines, having, 0.31 and 0.57 respectively. Visually, the no-field model obtained better results with straighter walls and fewer missed detections of inner rooflines. Thanks to the computation of the skeleton graph, it can predict buildings with common walls.

3. **What are the strengths and limitations of the approach and how can this be improved?**

The benefits of the method are given as followings:

1) no handcrafted features were used. To improve the segmentation, vectorization and corner detection we tried to implement frame field learning. Even though it does not add any cost to the training and inference, it does not improve the results either.
2) the model can perform roof structure extraction for multiple buildings in one image patch compared to other state-the-art roof structure extraction methods. This advantage facilitates both training and prediction since for training there is no need to selectively generate an image patch and reference having one building, and during prediction, we can output multiple roof structures at once;
3) the model can detect the shared walls of the buildings with the usage of outline segmentation and computation of the skeleton graph;
4) the method outputs the closed building outlines thanks to the separate branching for building interior & outline segmentation and correction of outlines with the contour of the building interior.

The proposed method, however, has several drawbacks:

1) the output can have missed detections for which we apply post-processing with the extend/trim lines tools. Though, we still have missed inner rooflines since the extension is limited to the distance of up to 4 m and will only work if there is at least some part of the line segment predicted.
2) since the extension is automatic and attempts to join the closest endpoints, it can also lead to odd results such as an overextension.
3) The model can fail to predict roof lines such as hips and valleys (*Figure 19.(3,4)-d).* In contrast, the model is better at predicting simpler roofs consisting of only outlines and ridges. The majority of the inner rooflines in our dataset belong to the ridges from what we can deduce that the model prediction could improve if the reference dataset had more examples of other roof elements such as hips and valleys.
4) the model predicts objects such as trees as part of the building if they stand at a near distance since they may have a similar height to the building.

The main causes of the method's drawbacks are the insufficiency of training data and mismatching of the buildings in training data with their real-life appearance. Even though the pre-processing was performed on the reference dataset, due to the time limitations the correction was only made on the outlines and inner rooflines with severe mismatches. Another cause is mismatching between the RGB image and nDSM as they have been generated in different years, 2021 and 2019 respectively.

In further studies, collecting more training data will benefit the performance. Besides, preliminary nDSM refinement can remove the false positive predictions of objects other than buildings. Lastly, adding another block based on the GNN can help with the retaining connection within building roof elements. Moreover, besides improving this method for roof structure extraction, with some modifications, it could be used for the extraction of road maps, cadastral or agricultural boundaries and slum studies.

LIST OF REFERENCES

Alidoost, F., Arefi, H., 2016. Knowledge Based 3D Building Model Recognition Using Convolutional Neural Networks From Lidar and Aerial Imageries. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLI-B3, 833–840. https://doi.org/10.5194/isprs-archives-xli-b3-833-2016

Alidoost, F., Arefi, H., Tombari, F., 2019. 2D image-to-3D model: Knowledge-based 3D building reconstruction (3DBR) using single aerial images and convolutional neural networks (CNNs). Remote Sens. 11. https://doi.org/10.3390/rs11192219

Awrangjeb, M., Zhang, C., Fraser, C.S., 2013. Automatic extraction of building roofs using LIDAR data and multispectral imagery. ISPRS J. Photogramm. Remote Sens. 83, 1–18. https://doi.org/10.1016/J.ISPRSJPRS.2013.05.006

Castagno, J., Atkins, E., 2018. Roof shape classification from LiDAR and satellite image data fusion using supervised learning. Sensors (Switzerland) 18. https://doi.org/10.3390/s18113960

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, Li Fei-Fei, 2010. ImageNet: A large-scale hierarchical image database 248–255. https://doi.org/10.1109/cvpr.2009.5206848

Girard, N., Smirnov, D., Solomon, J., Tarabalka, Y., 2020. Polygonal Building Segmentation by Frame Field Learning, in: ArXiv. pp. 1–30. https://doi.org/10.1109/IGARSS39084.2020.9324080

Gui, S., Qin, R., 2021. Automated LoD-2 model reconstruction from very-high-resolution satellite-derived digital surface model and orthophoto. ISPRS J. Photogramm. Remote Sens. 181, 1–19. https://doi.org/10.1016/J.ISPRSJPRS.2021.08.025

Hang, L., Cai, G.Y., 2020. CNN based detection of Building Roofs from High Resolution Satellite Images, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. International Society for Photogrammetry and Remote Sensing, pp. 187–192. https://doi.org/10.5194/isprs-archives-XLII-3-W10-187-2020

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

Kass, M., Witkin, A., 1988. Snakes: Active Contour Models. Int. J. Comput. Vis. 321–331.

Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

Li, Z., Wegner, J.Di., Lucchi, A., 2019. Topological map extraction from overhead images, in: Proceedings of the IEEE International Conference on Computer Vision. Institute of Electrical and Electronics Engineers Inc., pp. 1715–1724. https://doi.org/10.1109/ICCV.2019.00180

Liu, K., Ma, Hongchao, Ma, Haichi, Cai, Z., Zhang, L., 2020. Building extraction from airborne lidar data based on min-cut and improved post-processing. Remote Sens. 12, 1–25. https://doi.org/10.3390/rs12172849

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path Aggregation Network for Instance Segmentation. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 8759–8768. https://doi.org/10.1109/CVPR.2018.00913

Luo, L., Li, P., Yan, X., 2021. Deep learning-based building extraction from remote sensing images: A comprehensive review. Energies 14. https://doi.org/10.3390/en14237982

Macay Moreia, J.M., Nex, F., Agugiaro, G., Remondino, F., Lim, N.J., 2013. From DSM To 3D Building Models: a Quantitative Evaluation. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XL-1/W1, 213–219. https://doi.org/10.5194/isprsarchives-xl-1-w1-213-2013

Muftah, H., Rowan, T.S.L., Butler, A.P., 2022. Towards open-source LOD2 modelling using convolutional neural networks. Model. Earth Syst. Environ. 8, 1693–1709. https://doi.org/10.1007/s40808-021-01159-8

Nauata, N., Furukawa, Y., 2020. Vectorizing World Buildings: Planar Graph Reconstruction by Primitive Detection and Relationship Inference, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 711–726. https://doi.org/10.1007/978-3-030-58598-3_42

Novacheva, A., 2008. Building roof reconstruction from LiDAR data and aerial images through plane extraction and colour edge detection. Int. Arch. Photogramm. … 53–58.

OpenStreetMap contributors, 2017. Open Street Map [WWW Document]. URL https://planet.osm.org/

(accessed 10.5.21).
Partovi, T., Fraundorfer, F., Azimi, S., Marmanis, D., Reinartz, P., 2017. Roof type selection based on patch-based classification using deep learning for high resolution satellite imagery, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. International Society for Photogrammetry and Remote Sensing, pp. 653–657. https://doi.org/10.5194/isprs-archives-XLII-1-W1-653-2017
Partovi, T., Fraundorfer, F., Bahmanyar, R., Huang, H., Reinartz, P., 2019. Automatic 3-D building model reconstruction from very high resolution stereo satellite imagery. Remote Sens. 11, 1660. https://doi.org/10.3390/rs11141660
PDOK [WWW Document], 2013. URL https://www.pdok.nl/introductie/-/article/basisregistratie-adressen-en-gebouwen-ba-1 (accessed 11.24.21).
Persello, C., Wegner, J.D., Hansch, R., Tuia, D., Ghamisi, P., Koeva, M., Camps-Valls, G., 2022. Deep Learning and Earth Observation to Support the Sustainable Development Goals: Current Approaches, Open Challenges, and Future Opportunities. IEEE Geosci. Remote Sens. Mag. 30. https://doi.org/10.1109/MGRS.2021.3136100
Qin, Y., Wu, Y., Li, B., Gao, S., Liu, M., Zhan, Y., 2019. Semantic segmentation of building roof in dense urban environment with deep convolutional neural network: A case study using GF2 VHR imagery in China. Sensors (Switzerland) 19, 1164. https://doi.org/10.3390/s19051164
Ramer, U., 1972. An iterative procedure for the polygonal approximation of plane curves. Comput. Graph. Image Process. 1, 244–256. https://doi.org/10.1016/S0146-664X(72)80017-0
Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 9351, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
Salehi, S.S.M., Erdogmus, D., Gholipour, A., 2017. Tversky loss function for image segmentation using 3D fully convolutional deep networks. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 10541 LNCS, 379–387. https://doi.org/10.48550/arxiv.1706.05721
Sun, X., Zhao, W., Maretto, R. V., Persello, C., 2021a. Building polygon extraction from aerial images and digital surface models with a frame field learning framework. Remote Sens. 13, 4700. https://doi.org/10.3390/rs13224700
Sun, X., Zhao, W., Maretto, R. V., Persello, C., 2021b. Building outline extraction from aerial imagery and digital surface model with a frame field learning framework. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 43, 487–493. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-487-2021
Wang, L., Chu, C.H.H., 2009. 3D building reconstruction from LiDAR data, in: Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics. pp. 3054–3059. https://doi.org/10.1109/ICSMC.2009.5345938
Wang, Y., Zorzi, S., Bittner, K., 2021. Machine-learned 3D building vectorization from satellite imagery, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 1072–1081. https://doi.org/10.1109/CVPRW53098.2021.00118
Zhang, F., Nauata, N., Furukawa, Y., 2020. Conv-MPN: Convolutional message passing neural network for structured outdoor architecture reconstruction, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 2795–2804. https://doi.org/10.1109/CVPR42600.2020.00287
Zhang, T.Y., Suen, C.Y., 1984. A Fast Parallel Algorithm for Thinning Digital Patterns. Commun. ACM 27, 236–239. https://doi.org/https://doi.org/10.1145/357994.358023
Zhao, W., Persello, C., Stein, A., 2022. Extracting planar roof structures from very high resolution images using graph neural networks. ISPRS J. Photogramm. Remote Sens. 187, 34–45. https://doi.org/10.1016/J.ISPRSJPRS.2022.02.022
Zhao, W., Persello, C., Stein, A., 2021a. Building outline delineation: From aerial images to polygons with an improved end-to-end learning framework. ISPRS J. Photogramm. Remote Sens. 175, 119–131. https://doi.org/10.1016/j.isprsjprs.2021.02.014
Zhao, W., Persello, C., Stein, A., 2021b. End-To-End Roofline Extraction From Very-High-Resolution Remote Sensing Images, in: International Geoscience and Remote Sensing Symposium (IGARSS). pp. 2783–2786. https://doi.org/10.1109/IGARSS47720.2021.9554162

Zhou, K., Chen, Y., Smal, I., Lindenbergh, R., 2019. Building segmentation from airborne vhr images using mask r-cnn. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 42, 155–161. https://doi.org/10.5194/isprs-archives-XLII-2-W13-155-2019

Zorzi, S., Bittner, K., Fraundorfer, F., 2020. Machine-learned regularization and polygonization of building segmentation masks, in: Proceedings - International Conference on Pattern Recognition. Institute of Electrical and Electronics Engineers Inc., pp. 3098–3105. https://doi.org/10.1109/ICPR48806.2021.9412866