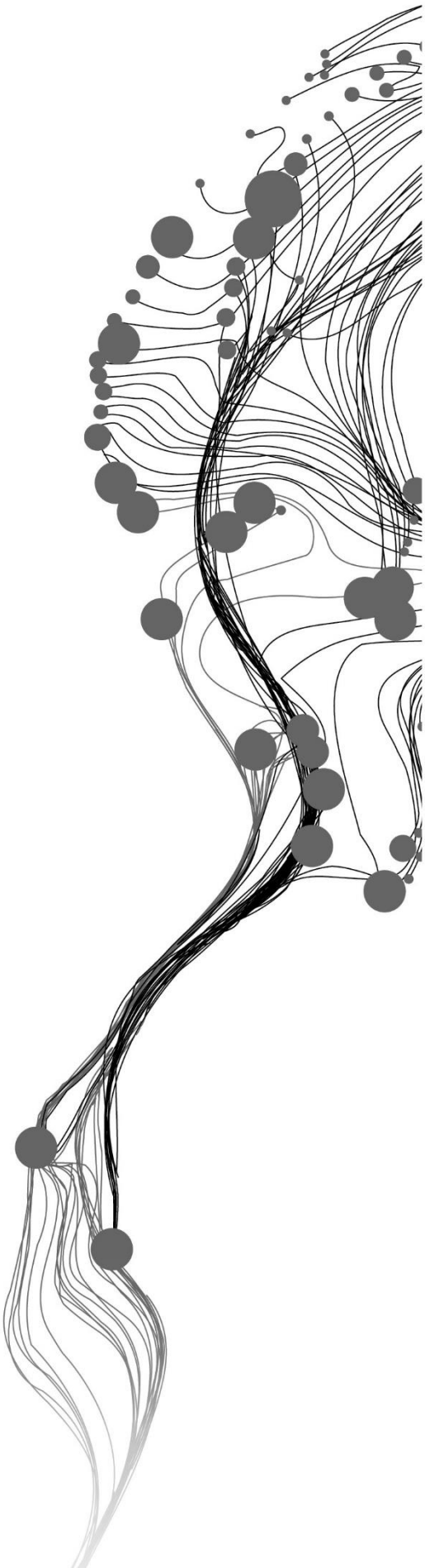
[BUILDING OUTLINE DELINEATION AND ROOFLINE EXTRACTION: A DEEP LEARNING APPROACH]

[MINA GOLNIA]
[August, 2021]

SUPERVISORS:
[dr. M.N. Koeva]
[dr. C. Persello]

External supervisor:
[C. Valk]

ADVISOR:
[W. Zhao]



BUILDING OUTLINE DELINEATION AND ROOFLINE EXTRACTION: A DEEP LEARNING APPROACH

MINA GOLNIA
Enschede, The Netherlands, August, 2021

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.
Specialisation: Urban Planning and Management

SUPERVISORS:
dr. M.N. Koeva
dr. C. Persello
External supervisor:
C. Valk, NEO B.V.
ADVISOR:
W. Zhao

THESIS ASSESSMENT BOARD:
prof. dr. R.V. Sliuzas
dr. C.M. Gevaert

DISCLAIMER

This document describes work undertaken as part of a programme of study at the Faculty of Geo-Information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author and do not necessarily represent those of the Faculty.

ABSTRACT

Nowadays, many authorities are attempting to address complex urban and environmental issues through digital technology. Following the achievements of deep learning and the availability of remote sensing data, the interest in developing automatic and robust techniques to generate accurate and up-to-date building mapping models, which are fundamental for constructing 3D models or urban digital twins, is rapidly increasing. Deep learning proved helpful in recognising urban objects and structures and extracting the buildings' geometrical characteristics. Having all in mind, in this research, we propose a methodology to automatically extract the building rooflines, namely, Eave, Ridge and Hip lines, which are the prerequisites for 3D building models with Level Of Detail 2 (LOD2) using a CNN-based deep learning technique. Our strategy combines two stages; first, predicting a binary building mask that will be added as an input layer in the second stage- roofline extraction. In both stages, the Unet-Resnet network architecture with 51 and 101 layers are adopted and fine-tuned to find the optimal solutions. The proposed method is tested in Enschede, the Netherlands, using the 25cm orthorectified aerial (RGB) images in 2018. Both networks are also tested using the normalised Digital Surface Model (nDSM) to improve the results. Unet-Resnet 101 performs better in both stages, reaching an average F1-score (the harmonic mean between precision and recall) of 0.68 for binary building mask prediction and 0.55 for rooflines extraction. The results improve to 0.85 and 0.66 for binary building mask prediction and roofline extraction, respectively. A class-wise evaluation is also applied to clearly understand the model's behaviour for each class of rooflines. Accordingly, an average F1-score of 0.81, 0.55 and 0.32 is achieved for eave, ridge and hip lines, correspondingly. The precision (correctness) and recall (completeness) values for eave lines prediction do not deviate much (0.82 and 0.81). In contrast, the ridge and hip classes have a higher recall (0.61 ridges, 0.51 hips) than the precision value (0.49 ridge, 0.23 hip). Having predicted the lines, they are then simplified using the Douglas-Peucker simplification algorithm, with a tolerance of 0.5m. Regarding our investigation results, the proposed method can be effectively used to automatically extract building roof structures and linear elements, which can be generalised to any type of roof. Besides, the model is able to extract inner walls, which is a big challenge in the building segmentation field. However, it is recommended to use higher resolution images and a larger amount of training data with more variety in building types in future studies.

Keywords: Deep learning, Remote sensing, LOD2, Segmentation, Rooflines extraction

ACKNOWLEDGEMENTS

In the light of this, I would like to express my deepest gratitude to the Faculty of Geo-information Science and Earth Observation (ITC) for the ITC Excellence Scholarship that provided financial support during my MSc tenure.

I extend my earnest gratitude to my supervisors, dr. M.N. Koeva and dr. C. Persello for their relentless support and sheer patience. Their amicable and critical approach helped me grow as a researcher as well as an individual. I would also like to thank W.Zhao, whose support as my advisor was also of great help. It has been a privilege to be able to work under their supervision.

I would also like to express my gratitude to NEO B.V. and my internship supervisor C. Valk, for allowing me to pursue my internship in their organisation while growing and learning a plethora of new things.

I am highly indebted to my beautiful family – my loving parents, my brother, my husband for believing in me and providing unconditional support throughout my M.Sc. journey and beyond.

Reaching the shore after a long voyage through troubled waters at times gives me the feeling of accomplishment and joy, but it also puts me in some melancholy. I am thankful to each and everyone who made this journey a lifetime experience and an eternal memory.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

1. INTRODUCTION

1.1. Background and justification

Achieving sustainable cities is one of the main objectives of many authorities and governments (Billen et al., 2014) following the establishment of Sustainable Development Goals (SDGs) targeted by 2030 around the world (United Nations, 2015a). It is estimated that 66% of the world's population will be living in cities by 2050 (UNEP, 2018). This shift will result in a considerable expansion of current urban areas and might lead to the need for building new cities. As a result, cities will be experiencing unprecedented challenges related to growth, competitiveness and performance. Following the climate and energy targets set by the United Nations (2015) and the European Commission (2014), developing smart solutions to overcome the current urbanisation issues is urgent (Estevez, Lopes, & Janowski, 2016).

The concept of "Smart Cities" has emerged to address these issues (Estevez et al., 2016). In practice, smart cities share similar goals as sustainable cities (Ahvenniemi, Huovila, Pinto-seppä, & Airaksinen, 2017). In the European Union's (2011) view, the concept of smart cities supports the idea of environmental sustainability, intending to reduce greenhouse gas emissions using innovative technologies. Recently, smart cities' concept has been transformed into urban "Digital Twins", which are established to integrate virtual and real-world elements of the smart city (Hämäläinen, 2020). Digital twins enable comprehensive data exchange to explore real-world features and behaviours by developing models, simulations, and algorithms (Dembski, Wössner, Letzgus, Ruddat, & Yamu, 2020).

One form of digital representation of the urban environment which provides fundamental building blocks for digital twins is the 3D city model (Biljecki, Ledoux, & Stoter, 2016a), consisting of green space models, street space models, digital terrain models (DTMs) and building models derived from building mapping techniques (Buyukdemircioglu, Kocaman, & Isikdag, 2018). Among all the urban environment components, buildings drew more attention due to their predominancy and significance in urban life. Objects like buildings are very likely to change over time as a result of new constructions or developments. It is, therefore, necessary to produce accurate models of buildings promptly (Qin et al., 2019). However, generating such models is demanding, time-consuming and costly as it requires a lot of manual work (Sugihara & Shen, 2017).
Consequently, the automation of this process is essential to reduce labour and enhance accuracy (Agoub, Schmidt, & Kada, 2019).

The recent achievements of Deep Learning and computer vision (Ibrahim, Haworth, & Cheng, 2020) and the availability of open geospatial data, such as very high-resolution aerial images and LiDAR (Light Detection and Ranging) point clouds in developed countries, reduce the costs and labour of generating large-scale 3D city models (Park & Guldmann, 2019). The integration of building footprints and surface elevation data can also be used to construct semantic 3D building models (Zhu et al., 2015). Such models pave the way for more detailed urban analyses such as population distribution (Qiu, Sridharan, & Chun, 2013), housing prices (Hamilton & Morgan, 2010), and energy efficiency (Chen, Hong, Luo, & Hooper, 2019).

Despite all the improvements, 3D building modelling has remained challenging, specifically for demonstrating greater details. In the literature, these details are recognised as the Level of Detail (LOD), which defines the building model's similarity to its real-world equivalent (Biljecki et al., 2016a). The primary elements of 3D Building Reconstruction (3DBR) are building outlines (footprints) and height data (Zhu et al., 2015).
Consequently, recent studies in remote sensing and computer vision are mainly centred on the extraction of building outlines (Li & Wegner, 2019; Zhao, Ivanov, Persello, & Stein, 2020; Girard, Smirnov, Solomon, & Tarabalka, 2020) that only fulfils the 3DBR requirements at a basic level of detail. However, to have a higher LOD, more information about roof details such as inline roof contours or planes is required. Recently, Convolutional Neural Networks (CNNs), as a leading representative of the Deep Learning family, performed impressively in segmentation tasks that can be utilised to extract both roof outlines and inlines. (Alidoost, Arefi, & Tombari, 2019). Even so, the initial outputs of segmentation lack the sharp corners and edges required for 3DBR purposes. Therefore, these outputs need a post-processing procedure such as shape refinement, simplification and vectorisation (Zuoyue Li & Wegner, 2019).

Having all in mind, this study is taking one step ahead toward automatic 3DBR using convolutional neural networks and remote sensing data. The focus of the study remains on developing a method for extraction of building roof outlines and inlines, to first: addressing the issue with non and semi-automatic 3DBR methods, and second: taking upon this opportunity for increasing the details of 3D building models by one level (from LOD1 to LOD2).

1.2. Research problem

Regarding the necessity of 3D building models for tackling urban issues and the complex and changing nature of buildings, creating an automatic method that reduces the costs, time, and human intervention is of great importance. Following the achievements of neural networks in segmentation tasks, the burden for automation of 3DBR can be remarkably decreased. However, most studies in this field could not go further than LOD1 3D building models based on images as they only predict building outlines. As a result, automatic extraction of both outline and inline elements of roofs as the primary elements of 3DBR at LOD2 forms the main problem of this research.

1.3. Research objectives and questions

The overall objective of this research is to automatically extract building roof outlines and inlines as the primary elements of 3DBR with LOD2.

1.3.1. Research objectives

Objective 1: To prepare the data for building roof structure extraction

Objective 2: To develop a methodology for automatic roof structure extraction as a prerequisite for 3D building reconstruction at LOD2

Objective 3: To evaluate the developed method and the created model

1.3.2. Research questions

Objective 1:

1. What are the suitable datasets to be used?
2. What is the quality of the reference data?
3. How to prepare the required data?
4. How to design training and testing datasets?

Objective 2:

1. What are the methods for 3D building reconstruction?
2. What is the state-of-the-art DL methods in building delineation?
3. How to further develop the existing methods to move toward automatic 3D building reconstruction?

Objective 3:

1. What metrics can be applied to evaluate this research's outputs?
2. What is the performance of the developed model for roof outlines and inlines extraction toward 3DBR?

1.4. Conceptual framework

Figure 1 shows the interrelations between major concepts of the research. Different approaches might be considered for 3D building roof outlines and inline reconstruction. In the context of this research, Deep Learning (DL) will be used to automate the process. The commonly used input dataset to feed DL networks is remote sensing data that provides aerial and satellite imagery, height data, and reference datasets. 3D building models can have different applications according to their corresponding level of detail. The majority of previously done studies focus on LOD1, which shows the buildings as a simple cubic form. Therefore, in this study, the focus remains on optimally modifying DL algorithms to increase the details to LOD2, which is necessary for some applications that are sensitive to the roof structures/shapes, such as solar panel installation or energy estimation.

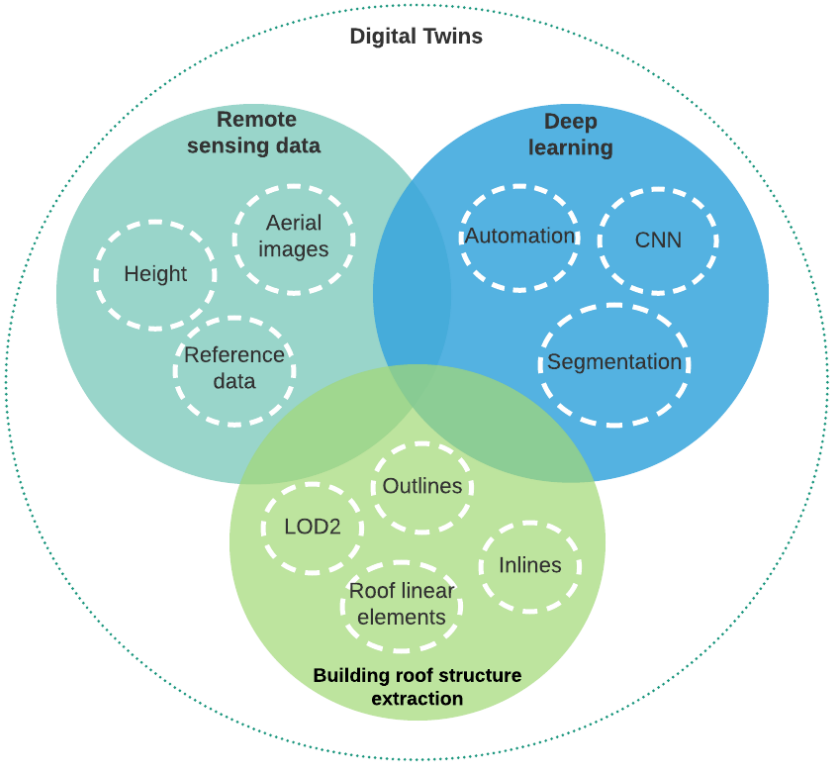


Figure 1-Conceptual framework

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. The main concepts and the underlying relations are indicated in a conceptual framework.

Chapter 2. Literature review

Related concepts for 3D building reconstruction and state-of-the-art feature extraction techniques are reviewed in this chapter. Former relevant scientific literature is also reviewed in this part.

Chapter 3. Methodology

An overview of the research methodology and the study area is introduced in this chapter, followed by a detailed description of each step, including data preparation, outline and inline roof element extraction toward automatic 3DBR and accuracy assessment.

Chapter 4. Results and analysis

The experimental results are presented here with a brief explanation.

Chapter 5. Discussion

In this chapter, an elaborate discussion of the obtained results is presented.

Chapter 6. Conclusions and recommendations

This chapter closes the thesis with concluding remarks of the entire research and recommendations for future study.

1.6. Summary

This chapter elaborates on the background of the research core aspects, leading to the main problems and objectives. It also justifies the overall structure of the thesis in the following chapters. In summary, this research aims to extract roof outlines and inlines using CNNs to take a step toward automating the process of generating a 3D building model at LOD2.

2. LITERATURE REVIEW

2.1. Concepts related to 3D building reconstruction (3DBR)

There is a broad range of studies on 3D building reconstruction, which can be characterised based on the concept of Levels of Details (LODs). LODs differentiate between the various levels of geometric and semantic objects complexity, focusing on buildings. Open Geospatial Consortium (OGC) (2012) has defined five LODs (Fig.2) in a standardised data format called CityGML 2.0. CityGML can store the semantic information and geometries of the available objects in 3D city models such as buildings (Donkers, 2013).

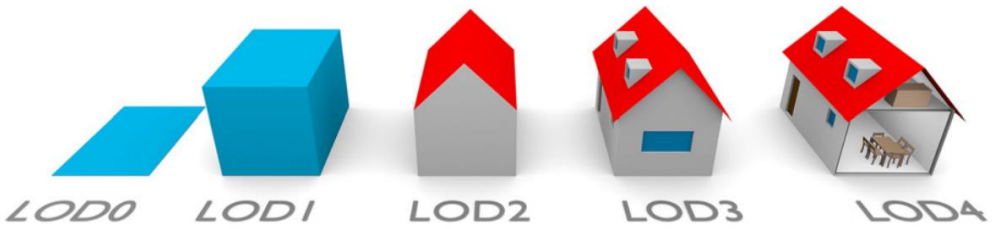


Figure 2- Five LODs in CityGML (Biljecki, Ledoux, & Stoter, 2016, p. 26)

According to OGC (2012), LOD0 is a 2.5D representation of object footprints or the roof edges polygons. LOD1 is an extrusion of the object's roof polygons created from the LOD0 model. LOD2 contains more details of the semantic classes of the building, like roof structures or shapes. LOD3 goes a step further by visualising the building's architectural details, such as windows and doors. LOD4 adds indoor features to the model provided by LOD3. Although the LOD1 is adequate for many environmental analyses, some applications require more details (Ziqi Li, Zhang, & Davey, 2015). For example, solar energy analyses are sensitive to the precise orientation and angle of the roof because it directly affects the amount of absorbed solar energy (Sugihara & Shen, 2017). Therefore, this study aims at achieving LOD2.

To create such models, various approaches can be used, such as GIS-based procedures (Zheng, Weng, & Zheng, 2017; Sugihara & Shen, 2017). In their study, Pollino et al. (2015) proposed a 3D building model using the CityEngine platform for modelling and creating a virtual city and in an ArcGIS environment for data edition and analysis. In addition, thanks to the availability of high-resolution LiDAR point clouds, most of the previous models are built upon the datasets provided by this technology (Zhu et al., 2015; Teo, 2019; TU Delft, 2020a).

2.2. Building segmentation techniques

Nowadays, following the popularity of deep learning and its impressive influence on remote sensing, the problem of 3DBR using aerial imagery and segmentation techniques turned into an exciting field for many scholars (Wu, Filippovska, Schmidt, & Kada, 2019). First, unlike LiDAR point clouds, aerial images are relatively more available worldwide (Kadhim, 2018); second, this approach increases the automation levels to a great extent (Partovi, Fraundorfer, Bahmanyar, Huang, & Reinartz, 2019). As a result, this research is also taking advantage of publicly available aerial imagery (in the Netherlands) and segmentation techniques.

As regards the segmentation techniques, Deep Convolutional Neural Networks (CNNs) turned into state-of-the-art (Persello & Stein, 2017; Qin et al., 2019) due to their impressive performance in recognising patterns in a large set of input data (Ma et al., 2019). The output of CNNs is only a class label. Therefore, from 2015, a more intuitive form of CNNs called FCN was developed that could perform notably well in semantic segmentation tasks. SegNet, DeconvNet, U-net and Resnet are some of the most commonly used architectures (Ji, Wei, & Lu, 2019).

The majority of studies using DL, are dedicated to building footprint extraction that can be used in producing 3D models with LOD1. Boonpook et al. (2018), developed a DL network using Segnet architecture to cover multi-dimension urban settlement appearances. In another study, Qin et al. (2019) presented an automatic pipeline for building roof segmentation over large areas in China using DCNN. Girard et al. (2020) added a frame field output to a fully convolutional network to extract building footprints. They tested their model with two different architectures, U-net and Resnet. However, regarding the predefined objectives of this research, achieving the LOD2, in addition to building footprints, requires semantic information of roof types and structures, as addressed in the following sections.

2.3. Roof structure extraction

As mentioned before, extraction of roof structures from overhead images is a fundamental task for 3DBR with LOD2. In the majority of previous studies, rooftops are identified by building appearance criteria like uniform colours (Cote & Saeedi, 2013), regular shapes (Inglada, 2007), and shadows (Femiani, Li, Razdan, & Wonka, 2015). Afterwards, an algorithm was being designed to identify the objects that satisfy the criteria.

Zhang, Wang, Chen, Yan, and Chen (2014) used the RANSAC algorithm to extract roof segments from LiDAR point clouds and aerial ortho-photos. Then the 3D building model was generated by minimising the distance between the reconstructed model and point clouds based on a predefined library of five standard roof primitives. Castagno and Atkins (2018) focused on the augmented classification accuracy resulted from integrating both LiDAR and satellite image data. They manually labelled the processed LiDAR and satellite images to create a diverse annotated roof image dataset for small to large urban cities. They then applied DL for feature extraction and random forest algorithm for roof shape classification.

In another study, Zheng et al. (2017) developed a multi-stage approach for 3DBR with LOD2 using high-resolution ortho-photos, nDSMs (normalised Digital Surface Model) and building footprints. First, a Canny-based line segmentation was employed to split building footprints into main plane-based partitions. Next, different types of roofs were classified using a rule-based technique based on the slope and orientation values of the planes. Besides, a watershed analysis algorithm was utilised to extract ridgelines of roofs.

The proposed algorithm by Partovi et al. (2019) is a multi-stage method including building boundary extraction and decomposition, image-based roof type classification, and initial roof parameter computation. In other words, buildings are decomposed into simple parts. A library of roof types was defined. A pre-trained Resnet architecture followed by an SVM classifier was used to classify the type of each building. The best-fitted roof type to each decomposed section of the building is identified.

All above-mentioned studies lack the generalisation ability since they are all limited to a predefined library of roof primitives and types to classify the roof segments. In a more recent study, Alidoost et al. (2019) presented an automatic framework for 3DBR using two optimised MSCDNs trained for height prediction and roofline segmentation tasks. In their study, they developed a knowledge-based 3D building model using the inherent and latent features from a single RGB image. Unlike previous studies, they utilised CNNs to extract linear elements of roofs in three classes of eave, ridge and hip lines instead of fitting the roof shapes into a predefined library. They tried to further improve this approach by designing a y-shaped CNN with one encoder and two decoders to predict the height and rooflines simultaneously (Alidoost, Arefi, & Hahn, 2020). Intrigued by their framework, this research also aims to extract linear elements of roofs to improve the generalisation of 3DBR models at LOD2.

2.4. Summary

In this chapter, relevant concepts and prior studies related to 3DBR has been reviewed. Two major components were identified for this research context; Building outlines and roof inlines. Accordingly, the challenges and gaps were recognised to be addressed in this research; Automation in reaching LOD2 3DBR, generalisation ability by using publicly available input data and independence of predefined roof type libraries. Overall, extraction of the linear elements of roofs using CNNs was selected for generating the perquisites of 3DBR at LOD2, as shown in Figure 3.

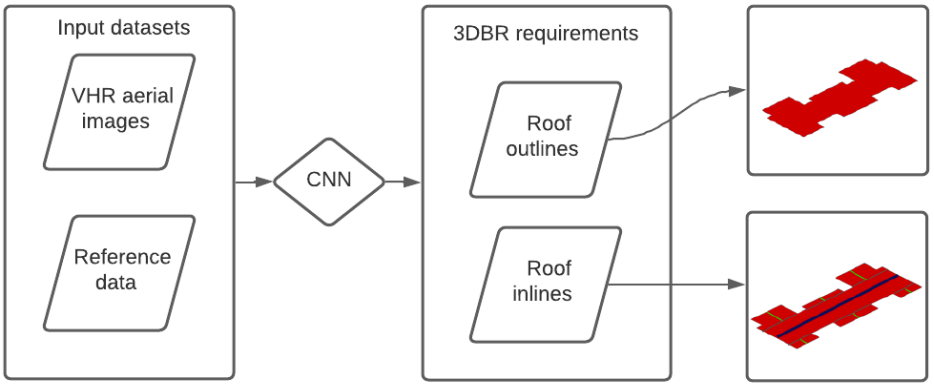


Figure 3- The adapted framework based on literature review

3. METHODOLOGY

3.1. Overall methodology

Figure 4 shows the overall methodology, demonstrating three major steps of this research, data preparation, roof outline and inline extraction and evaluation, corresponding to the predefined objectives in chapter one. To achieve the main goal, a multi-stage model is defined. The first stage (A) is a segmentation model, which outputs a binary building mask. This mask will be included as the fifth band to the input data for the next stage. As visualised in Figure 4, the input labels/reference data for the first stage are the building outlines derived from the National portal (PDOK[1]) in a polygon form. On the other side, the desired reference data geometry for the second stage is a line form. To feed the network in stage B, three types of rooflines, namely, eave, ridge and hip, are defined (see section 3.2, Fig.6). Due to the unavailability of such data, this data must be manually digitised. In response to the second objective, another CNN-based segmentation network is employed to output the non-regularised roof outlines and inlines. Next, a post-processing algorithm is applied to the segmented lines to obtain the regularised rooflines. Each stage is followed by an accuracy assessment step, including precision, recall, and F1-score (see section 3.4).

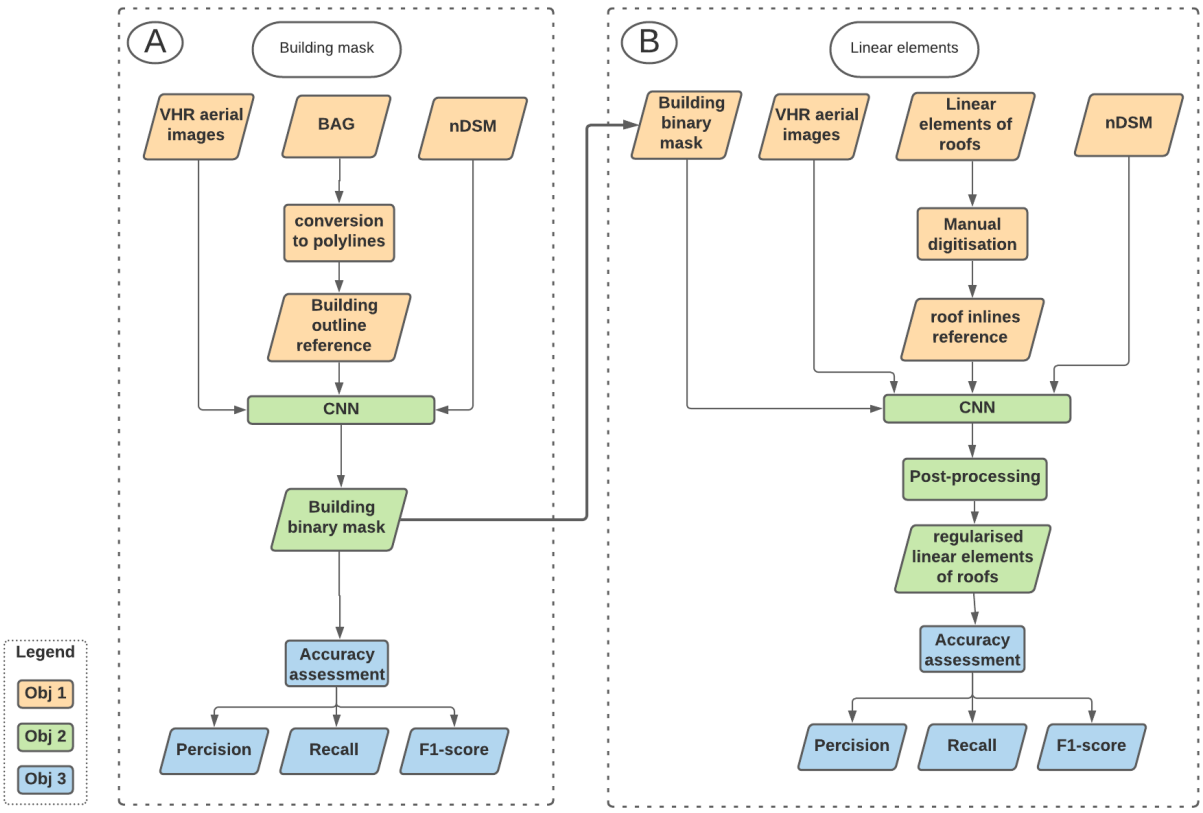


Figure 4- Overall methodology stages

[1] https://www.pdok.nl/

3.2. Data preparation

The reference building footprints (for stage A), VHR (Very High Resolution) aerial orthophoto image, and nDSM used in this research are provided by Kadaster[2] in PDOK[3] open platform from 2018. The images have three bands (RGB) with a spatial resolution of 25 cm. Subsequently, a detailed description of data usage in each step is presented.

In Stage A, the available building footprints (BAG), RGB image for the entire city of Enschede, the Netherlands, as the input. In order to improve the results, nDSM is added as the fourth band to the input images. This area is divided into 16641 tiles with a size of 256*256 pixels. Among them, 11641 tiles (about 70%) were chosen for training and 5000 tiles (about 30%) for testing, as shown in Figure 5. The validation set is randomly selected using 20% of the training set.

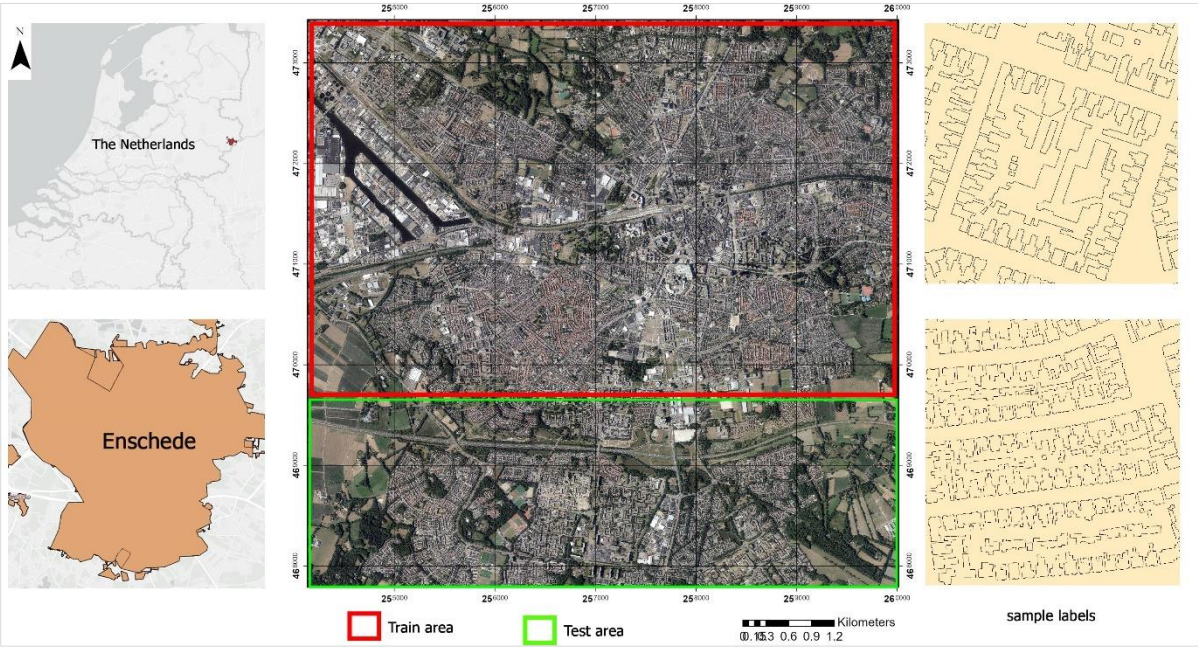


Figure 5- Study area- Stage A

In stage B, a smaller part of Enschede (1.63 km2), covering about 3700 buildings, was selected (corresponding to the primary goal of this research. The area is chosen based on two desired characteristics; the area includes both detached and attached buildings, buildings have a variety of roof shapes (flat, gable, complex). Due to the lack of reference data for this stage, the roof outlines and inlines must be manually digitised in ArcMap.

The shapefile represents three types of lines as illustrated in Figure 6: eave, ridge, and hip lines, defined by binary codes as follows:

a) [1, 0, 0, 0] if it belongs to eave
b) [0, 1, 0, 0] if it belongs to ridge lines,
c) [0, 0, 1, 0] if it belongs to hip lines
d) [0, 0, 0, 1] if it belongs to background

[2] https://www.kadaster.nl/zakelijk/datasets/open-datasets
[3] https://www.pdok.nl/

Figure 6-Different types of rooflines

It was observed that the provided building outlines from BAG do not fully match the image. Therefore, adjustments based on nDSM, Google Map[4] , and Google Earth 3D[5] have been made to create more accurate input data. First, the individual buildings sharing the same roof were merged (Fig.7-a). Second, the building polygons with areas smaller than 20 m2 were masked because they mostly show bicycle sheds, which are out of this research interest. Third, the polygons were reshaped to flat and shaped parts (Fig.7-b). After all, the inline shapefiles were delineated manually.

Figure 7-Modifications to BAG- a) Merging the buildings sharing the same roof, b) Splitting the flat and shaped parts

The study area was then divided into 483 tiles of 256*256 pixels. RGB layers and building outline and inline labels were prepared for each tile for the segmentation task. Among them, 363 tiles (75%) were used for training and 120 tiles (25%) for testing the model. The validation set is randomly selected using 20% of the training sets. Figure 8 visualises some samples of digitised labels.

[4] https://www.google.com/maps/Enschede
[5] https://earth.google.com

Figure 8- Samples of manually digitised rooflines

3.3. Model development

3.3.1. Multi-stage segmentation approach

As mentioned before, the intended model is characterised by two different stages(Figure 4). The first stage aims to segment the area of interest (AOI) to focalise the building area as our subsequent analysis. In the second stage, the segmentation of the eave, ridge and hip lines is carried out on the image sub-portions extracted thanks to the binary building mask from stage one.

Unet is a breakthrough in computer vision (Wu et al., 2019) that is proven effective for segmentation tasks where a similar size and resolution of the input and output is desired. However, when using Unet, the results are likely to lack fine details due to up/downsampling steps, specifically when networks get deeper (Thomas, 2019). As the winner architecture in ILSVRC 2015, Resnet, drawn from a simple deep CNN, solves this problem by taking advantage of skip connections (Wu et al., 2019). Skip connections overcome the vanishing gradient issue and thus enable the model to achieve higher accuracy in deeper networks (Tsang, 2018). Since the original Resnet can not perform image segmentation, Unet architecture encoder-decoder is combined to transform the output to an image of the same size. Traditionally, more layers result in better network and outputs; thus, Unet-Resnet50 and 101 are employed as the selected candidates to achieve the objectives.

Resnet initially starts with a convolution layer followed by max-pooling with kernel sizes of 7*7 and 3*3, respectively. In stage 1, there are three residual blocks, including three layers. An identity connection fits the input from the previous layer to the next layer without any modification relates all residual blocks to fit the input from the previous layer to the next layer without any

modifications. The convolution operation in residual blocks is performed with stride two, which decrease the image height and width to half and doubles the width of the channel as we progress through stages. In deeper networks like Resnet50 and 101, a bottleneck design is added to reduce the parameters without degrading the network's performance. This technique takes three layers of 1*1, 3*3, 1*1 convolution for each residual function stacked over the other. The last layer is an average pooling layer. Resnet101 is similarly built with more layers.

Our adopted network architecture is shown in Figure 9. It consists of 50 and 101 layers for our network candidates. To speed up the training, a BatchNormalisation step along with 1 MaxPool have also been added to reduce spatial dimensions.

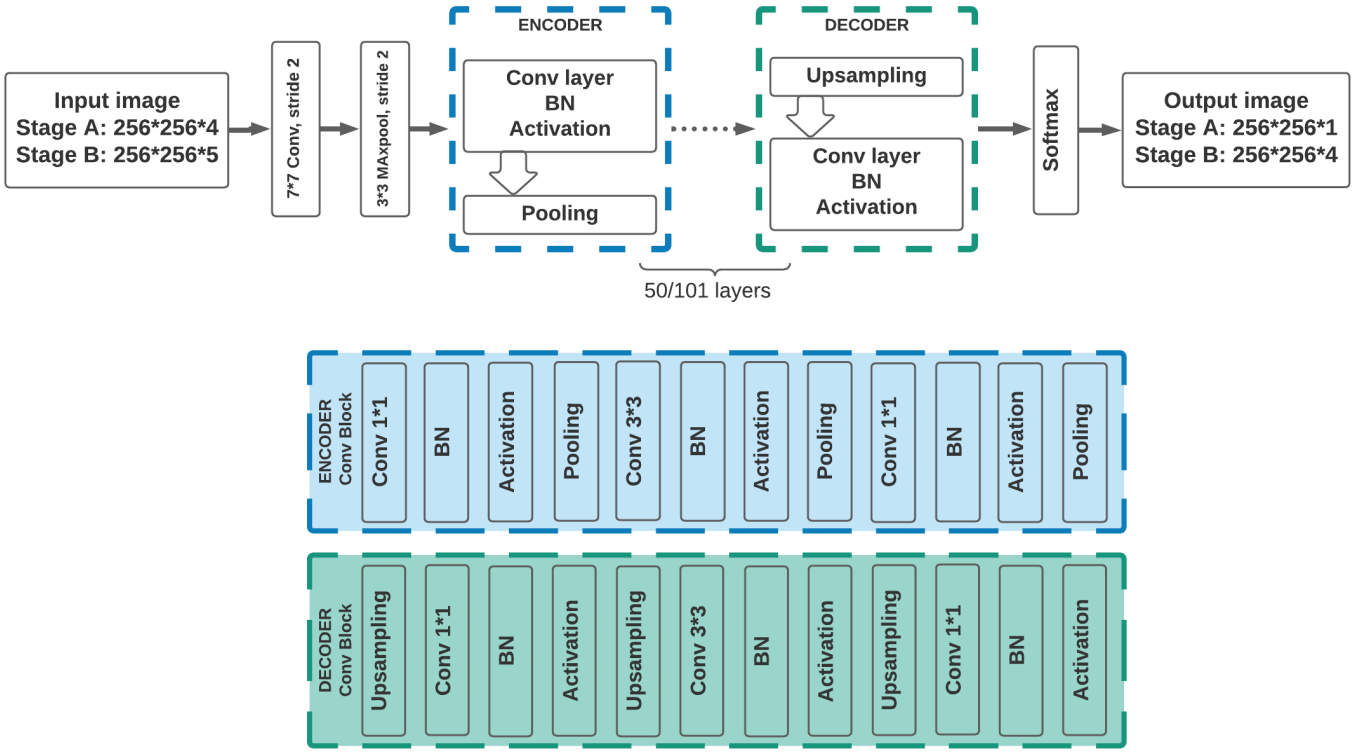


Figure 9- The overview of the adapted network

There are several parameters within each network that affect the performance of the results. The core component of the network is a convolutional layer followed by a Batch normalisation step that helps to improve the convergence process and keeps the network weight under control. In addition, it can handle the internal covariate shift issue by normalising each layer's inputs, which reduces the number of required epochs and increases the learning stability (ElGhany & Ibrahim, 2019).

The next component in each residual block is a transfer function identified as the activation function, which is added to each layer's output. The most commonly used activation function is Rectified Linear activation Unit (ReLU) that returns the values using $ReLU(x)=max(0,x)$ equation. When the values are greater than 0, it returns 1, and for values less than or equal to 0, it is set to 0. Due to its simplicity, ReLU allows the model to learn faster while avoiding the vanishing gradient problem. However, in some cases, ReLU introduces the 'dying ReLU' problem, where the network's components are never updated to a new value (Hansen, 2019). Therefore, although ReLU is sufficient for many applications, specifically with shallower

networks, a newer branch of activation functions called Scaled Exponential Linear Unit (SELU) proposed by Klambauer et al. (2017) was also investigated, which uses the following equation (Equation 1):

Equation 1- SELU activation equation

$$SELU(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha x^{x} - \alpha, & x \leq 0 \end{cases}$$

SELU is a self normalising layer added to neural networks which outperform the commonly used ReLU since it prevents the Dying ReLU problem as its derivative are not equal to 0 for negative values. Furthermore, SELU uses two parameters, and its function operation keeps the mean and variance of the outputs at all the network's layers close to normal distribution resulting in better performance (Moon, Park, Rho, & Hwang, 2019). Figure 10 shows the function curves of ReLU and SELU.

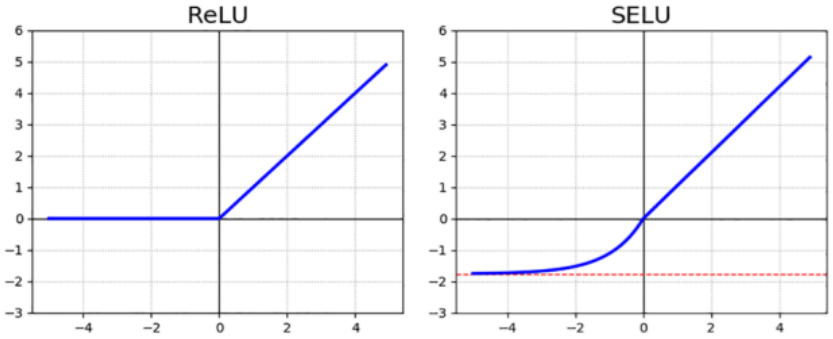


Figure 10- ReLU and SELU function curves (Moon et al., 2019, p.10)

After defining the initial setups of the network, the hyper-parameters then need to be determined to structure the model and learning strategy. In order to obtain a more accurate model, a manual optimisation strategy is used to select the optimal combination of hyperparameters. The initial values to start the optimisation are derived from the previous related studies. This research focuses on tuning the Batch size, number of Epochs, Learning rate, Loss function and implements the optimal values in the two proposed network architectures. The adam optimiser parameters are also tested with different values. Table 1 shows an overview of all the hyperparameters to be studied in this research.

Table 1- Selected Hyper-parameters

PARAMETER	VALUES
Batch size	4, 8, 16
Number of Epochs	100, 150, 200
Learning rate	schedular
Loss function	Binary/Categorical Cross entropy, Focal Tversky, Dice loss
Activation function	ReLU, SELU
Adam optimiser parameters	Beta1, Beta2, Epsilon
Network depth	Unet-Resnet50, 101

The Dice coefficient is a widely used metric in computer vision to compute the similarity between two images. Later, it has been adapted as a loss function known as Dice Loss (Equation 2). 1 is added in the numerator and denominator to ensure that the function is not undefined in edge case scenarios (Jadon, 2020).

Equation 2- Dice loss equation

$$DL(y, \hat{p}) = 1 - \frac{2y\hat{p} + 1}{y + \hat{p} + 1}$$

Focal Tversky loss (Equation 3) attempts to learn hard-examples such as with small region of interest with the help of γ coefficient as shown below; where T I specifies Tversky index, and γ can range from [1,3]

Equation 3- Focal Tversky equation

$$FTL = \sum_{c} (1 - TI_c)^{\gamma}$$

Before running the model in stage B, a data augmentation stage is also added to improve network accuracy by randomly transforming the original data during training. Data augmentation adds variety to the training data without increasing the number of labelled samples. Thus, it is specifically helpful while a vast amount of labelled data is not available such as in our case. However, it should be noted that data augmentation is only applied to training sets since the test and validation data should be representative of the original data (Jafar & Myungho, 2020). Our selected operations are flip, rotation and scaling.

The final prediction is carried out with a softmax function first to segment the building blocks (stage A) and second to assess whether pixels belong to the eave, ridge, hip or background (stage B). In the second stage, the network is supposed to learn some rules built upon the binary mask. Every pixel outside the mask is set to zero. Therefore, the inlines should be all within the building mask, and the outlines can also use the mask to predict the borders.

3.3.2. Post-processing

The initial output of CNN networks are rasters which will be polygonised using a code script. These converted outputs are irregular polygon/polyline features that can not be directly utilised in some applications such as 3D modelling. Therefore these output features need to be modified using regularisation and simplification techniques to obtain fine edges and lines. One of the most well-known simplification techniques is Douglas-Peucker that identifies and removes redundant vertices based on a user-defined tolerance value to simplify a given feature (Douglas & Peucker, 1973). This technique can work optimally with line features. Nevertheless, because it only measures the geometric deviation from an initial configuration of a complex polygon, the polygon output may easily drift from the real object, leading to considerable accuracy loss in practice (Tarabalka, 2018).

To enhance the polygon regularisation outputs, more sophisticated and complex operations should be performed. However, these methods require applying many steps to achieve accurate

regularised polygons, such as finding the main orientation of the buildings using the standard Hough transform (SHT) to avoid mismatching angles, minimum bounding rectangle (MBR)-based technique, minimum bounding triangle (MBT)-based technique for non-rectangular buildings approximation (Alidoost et al., 2020).

Considering this research's main objective, which is roofline extraction, although a network is trained and fine-tuned in our multi-stage approach to obtain optimal results for binary building masks, the computationally expensive regularisation procedures at this stage are avoided. Therefore, it is assumed that the output of the polygonal building segmentation model is perfect. Here, the ground truth digitised data is used instead of our building segmentation model to ensure that the binary masks are in their best condition. Subsequently, the building binary mask is added as the 5th band to initial RGB nDSM images to feed into the roofline segmentation model.

3.4. Accuracy assessment

A commonly accepted way to evaluate the performance of a model is the standard quality measures, including recall, precision, and F1-score, computed based on a pixel-based confusion matrix (Alidoost et al., 2019). **Precision**, also known as correctness, is the ratio of correctly predicted positive observations to the total predicted positive observations. While **recall,** also known as completeness, is the ratio of correctly predicted positive observations to all observations in the actual class. **F1-score** represents the geometric mean of Precision and Recall (Joshi, 2019). As this score takes both false positives and false negatives into account, it can be considered an overall quality measure. The equation for each measure is as follows (Equation 4):

Equation 4- Evaluation metrics equations

$$Comp. = \frac{TP}{TP+FN};\ Corr. = \frac{TP}{TP+FP};\ F1-score = 2.\frac{Corr.\times Comp.}{Corr.\times Comp.}$$

Where:

True Positives (TP) are the correctly predicted positive values, **True Negatives (TN)** are the correctly predicted negative values, **False Positives (FP)** are the incorrectly predicted positive values, and **False Negatives (FN)** are the incorrectly predicted negative values. Table 2 illustrates the evaluation confusion matrix.

Table 2- Confusion matrix of the classification

		PREDICTED CLASS	
		Class = Yes	Class = No
ACTUAL CLASS	Class = Yes	TP	FP
	Class = No	FP	TN

In the context of this research, pixels labelled as the outline and inline classes in both prediction and reference are addressed as True Positive (TP). In contrast, pixels labelled as outlines and inlines in prediction while they do not belong to line classes in reference data are called False

Positive (FP). The False Negative (FN) and True Negative (TN) pixels are similarly determined. Accordingly, the network performance will be analysed at both Global and Class levels.

3.5. Summary

In this chapter, the methods to fulfil the research objectives are elaborated in detail. Two study sites are selected; the Entire city of Enschede for the building segmentation task and a smaller part of the city for rooflines extraction. The first reference dataset is obtained from PDOK, and the second set is manually digitised. A multi-stage workflow of Unet-Resnet50 and 101 is designed to execute the roofline segmentation. Douglas-Peucker simplification technique will be applied to regularise segmented lines. Finally, a pixel-based accuracy assessment will be carried out by commonly-used Precision, Recall, and F1-score measures both at Global and Class levels.

4. RESULTS AND ANALYSIS

In order to find the optimal combination of the hyper-parameters, a single hyper-parameter is changed sequentially, while the rest are kept constant. The decision for keeping a hyperparameter is made based on the average F1-score on test sets. Once a value is selected, it will be kept fixed in the following experiments until all parameters are determined. In the following subsections, the results are organised.

4.1. Building segmentation: Hyper-parameter optimisation

As mentioned before, a binary building mask is first extracted to facilitate achieving our primary goal of roofline extraction. Therefore, less considerable effort is put into the optimisation of the building segmentation. The optimisation is carried out on 50 % of the entire data. The process starts with varying the batch size, followed by changes in the learning rate and loss function with different epochs. ReLU activation function and constant adam optimiser parameters were used for all the experiments. Besides, as the input image tile size was initially set to 256 to facilitate the network's data processing, it is kept 256 in all our experiments, and a new patch size will not be assigned. Optimal values will then be implemented with the two candidate network depths.

Table 3- Hyperparamter optimisation

NO. EXPERIMENT	BATCH SIZE	LEARNING RATE	LOSS FUNCTION	NO. EPOCHS	AVG F1-SCORE
1	4	1e-2	BCE*	100	0.64
2	8	1e-2	BCE	100	0.69
3	16	1e-2	BCE	100	0.69
4	8	1e-4	BCE	100	0.72
5	8	1e-6	BCE	100	0.70
6	8	1e-4	FT**	100	0.73
7	**8**	**1e-4**	**FT**	**150**	**0.78**
8	8	1e-4	FT	200	0.72

* Binary Cross entropy **Focal Tversky

As shown in Table 3, the values for optimising the building segmentation network are determined through eight sequential experiments. F1-score increased by changing the batch size from 4 to 8, but the increase from changing the value to 16 was too slight to make 16 the excellent choice. Additionally, a smaller batch size increases the training speed as it requires lower RAM, therefore 8 was selected as the best Batch size value.

As the network converges too fast using a 1e-2 learning rate which results in a suboptimal output, a learning rate of 1e-4 was used, which trained the network at a reasonable pace. To ensure that the lower learning rate enhances the training procedure or not, the model was also run using a 1e-6 learning rate. However, it did not remarkably affect the training up to the operated epochs. Accordingly, the network required more epochs and time to converge to obtain a similar result as 1e-4.

Since it is expected to solve a binary issue in this stage, the commonly used Binary Cross Entropy was first utilised. The urban structure of Enschede is not dense; subsequently, the model is likely to suffer from data imbalance between building and non-building classes. Considering the

characteristics of the study area, the model was run using Focal Tversky Loss which can handle the data imbalance issue. It improved the F1-score to 0.73 in 100 iterations.
Finally, to find the optimal number of iterations, the model was run for 50 more epochs with no signs of over/underfitting. However, adding more epochs turned out to cause an overfitting issue.

4.2. Roofline extraction: Hyper-parameter optimisation

Similar to the first stage, once the right fit of each parameter is found, it is kept constant during the next parameter's optimisation. Throughout the optimisation procedure in this stage, the patch size is set to 256 in all experiments. In addition, due to the availability of a relatively small area, the entire datasets are taken into account for optimisation. Besides, the selected optimiser is Adam. Instead of using a fixed number of epochs, the early stopping command was used.

4.2.1. Activation function

Following the recent success of the SELU as a new type of activation function, the model's performance was evaluated by comparing the results between ReLU and SELU. The F1-score values in Table 4 confirm that the model converges better and giving higher accuracy using SELU.

Table 4- Comparing the model's behaviour with ReLU and SELU activations

ACTIVATION FUNCTION	RELU	SELU
F1-score	0.48	0.53

4.2.2. Batch size

Similar to our experiment with batch size in the first stage, the best result was achieved using the batch size of 8. As shown in Table 5, an F1-score of 0.55 was achieved using a batch size of 8.

Table 5- Batch size optimisation

BATCH SIZE	4	8	16
F1-score	0.46	0.55	0.52

4.2.3. Learning rate

Despite the consensus on the adaptivity of Adam's learning rates, it is supposed that an explicit learning schedule could be beneficial to the model's convergence behaviour to avoid too low or too high values as the initial learning rate. As shown in Table 6, three combinations of learning rates are scheduled to drop at epoch 30, 60, 90, and above.

Table 6- Leaning rate optimisation

LEARNING RATE	F1-SCORE
1e-03, 1e-04, 1e-05,1e-06	0.56
1e-04, 1e-05, 1e-06, 1e-07	0.58
Exponential decay	0.61

A schedule was also directly passed into a Keras optimiser as decay every 100000 steps with a base of 0.96 with an initial rate of 1e-3. The best result was achieved using exponential decay equals 0.61 F1-score value.

4.2.4. Adam optimiser parameters

Drawing from Several studies (Dozat and Manning, 2017; Laine and Aila, 2017), lower β values work better than Adam's default values of 0.9 and 0.999. Thus, different values were also applied in our experiments. Additionally, a better F1-score was also achieved by changing the default Epsilon value, as seen in Table 7.

Table 7- Adam optimiser parameter tuning

BETA1	BETA2	EPSILON	F1-SCORE
0.9	0.999	1.00E-08	0.61
0.98	0.9	1.00E-08	0.62
0.98	0.9	1.00E-09	0.64

4.2.5. Loss functions

Initially, the procedure started with commonly-used Categorical Cross Entropy rewarded with over 90% accuracy. However, in addition to the expanded urban structure of Enschede, it was noticed that the roofline pixels to be segmented accounted for a tiny percentage of the total pixels in the image. As a result, all the model had to do was predict an entirely black image where the background class is much larger than the other classes, explaining the high obtained accuracy. To cope with this issue, experimenting with sensitive losses to class imbalance was initiated, as indicated in Table 8.

Table 8- Loss function optimisation

	CATEGORICAL CROSS-ENTROPY	DICE LOSS	FOCAL TVERSKY
F1-score	0.64	0.66	0.69

Due to its natural ability to focus on harder examples, mainly small-scale segmentations such as this research's case, Focal Tversky turned out to be the best candidate resulting in the highest model's performance of 0.69.

Having selected the optimal values, in the following section, they will be implemented in our two candidate networks to achieve the final results.

4.3. Model Implementation

The results include two parts, binary building mask generation and roofline extraction. It has been already elaborated on the data usage for each stage and evaluation metrics in chapter 3. The visual and numerical results of the final model are demonstrated in the following figures and tables.

4.3.1. Building segmentation

Table 9 presents the segmentation accuracy of the test sets. A recall of 0.65, precision of 0.63 and an F1-score of 0.64 are achieved by Unet-Resnet50. Moreover, there is a considerable increase in F1-score after applying nDSM, raising to 0.68. A similar trend can also be witnessed from Resnet50 to Resnet101, and the F1-score increased from 0.68 to 0.85. Comparing these two tables, the results of both Resnet50 and 101 in all the metrics get a higher value with nDSM data. Besides, in both networks, Unet-Resnet50 manifests lower precision and higher recall than Unet-Resnet101.

Figures 11 and 12 depict the final output maps of both networks, with and without including nDSM. Visual quality assessment of both models confirms that the deeper Unet-Resnet101 outperforms the Unet-Resnet50 network. It is also evident that utilising height data as an input layer improves the network's performance to F1-score 0.85.

Table 9- Evaluation metrics of the trained building segmentation task

INPUT	*NETWORK*	*PRECISION*	*RECALL*	*F1-SCORE*
Without nDSM	Unet Resnet50	0.63	0.65	0.64
	Unet Resnet101	0.68	0.69	0.68
With nDSM	Unet Resnet50	0.77	0.79	0.78
	Unet Resnet101	0.85	0.85	0.85

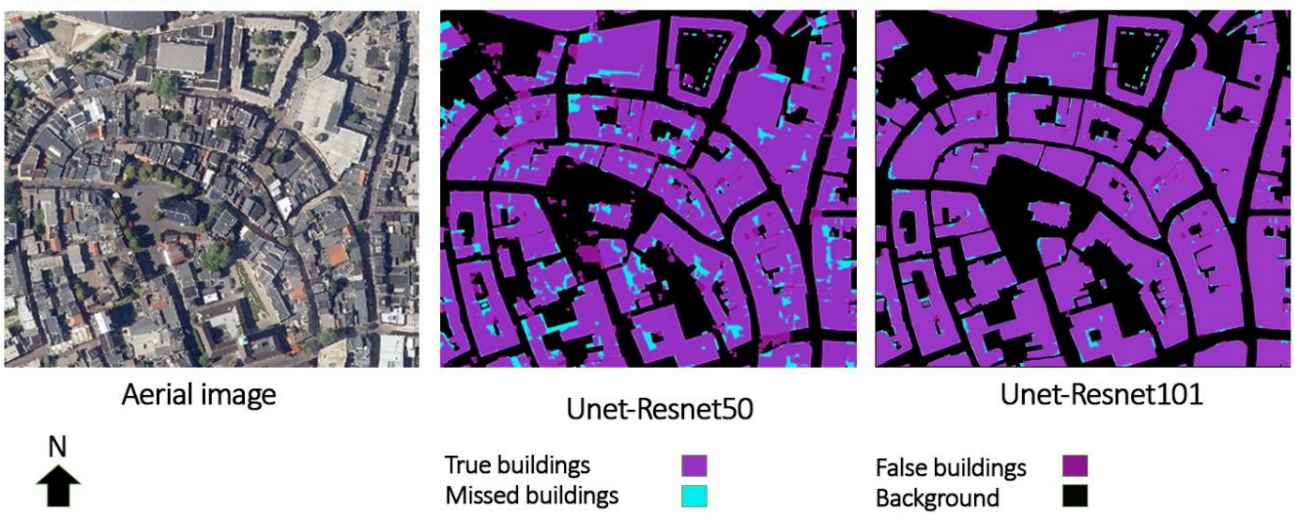


Figure 11-Implementation of the optimised values in the two candidates networks without using nDSM

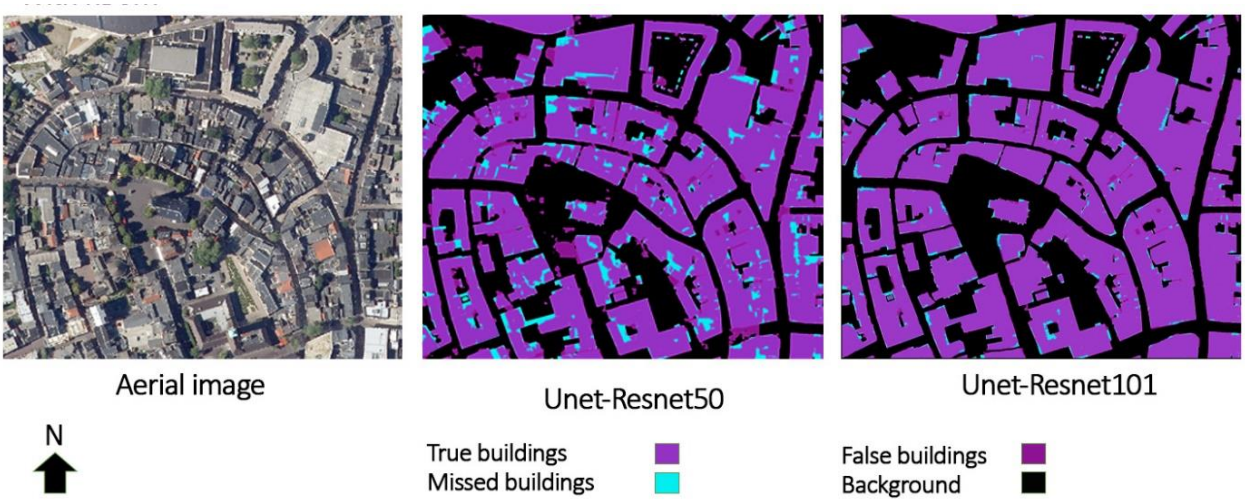


Figure 12- Implementation of the optimised values in the two candidates networks using nDSM

Although building blocks have been finely extracted with relatively clean boundaries, the model performed poorly where two buildings are so close (Fig.13-a), or there is an empty area between buildings (Fig.13-b). Additionally, these models cannot detect individual buildings and building partitions in attached and semi-attached buildings or within the building blocks. These issues are the most significant challenges in building segmentation models and require complex procedures to achieve. As a result, it was decided to feed the next model with the binary buildings mask from the digitised reference buildings instead of the output of our building segmentation model to ensure the perfect quality of input data used in the roofline extraction stage.

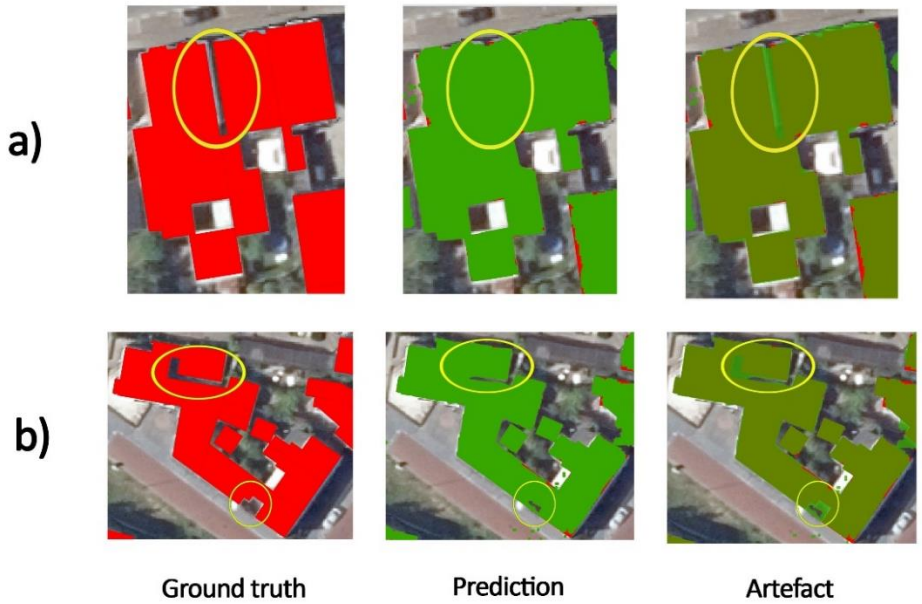


Figure 13- Model prediction artefacts- a) misclassification of two close buildings, b) misclassification of empty spaces

4.3.2. Roofline extraction

Table 10 shows the evaluation metrics of the developed roofline extraction model. As expected, the inclusion of nDSM has a remarkable influence on the performance of both networks. It increased by about 0.18 in Unet-Resnet50 and by 0.11 in Unet-Resnet101 using the nDSM. Furthermore, the deeper 101 layer network achieved better results with a 0.66 F1-score.

Table 10- Accuracy assessment of roofline extraction model

INPUT	*NETWORK*	*RECALL*	*PRECISION*	*F1-SCORE*
Without nDSM	Unet Resnet50	0.45	0.42	0.43
	Unet Resnet101	0.59	0.52	0.55
With nDSM	Unet Resnet50	0.62	0.60	0.61
	Unet Resnet101	0.72	0.62	0.66

Since the best results were achieved through Unet-Resnet101, a closer look was taken into the class accuracy assessment of the model shown in Table 11. The eave class has the highest F1-score value of 0.81, while the hip class with an F1-score of 0.32 has the lowest value. The precision and recall values do not deviate much from each other in eave lines prediction, meaning that the model is strict enough to detect both false negatives and false positives. However, this tradeoff follows a different trend in the rest of the classes. Higher recall values were witnessed in ridge and hip line predictions, meaning that almost every positive instance was correctly classified. Nevertheless, there were more members of the negative class classified as positive, which explains the low precision values.

Table 11- Class-wise accuracy assessment

CLASS	*PRED. EAVE*	*PRED. RIDGE*	*PRED. HIP*	*OTHER*	*TOTAL*	*PRECISION*	*RECALL*	*F1-SCORE*
Ref. Eave	488934	33562	21962	62148	606607	0.82	0.81	0.81
Ref.Ridge	13378	51560	15191	3760	83889	0.49	0.61	0.55
Ref. Hip	4374	5953	14908	4092	29327	0.23	0.51	0.32
Other	87526	14119	12621	2435424	2549632	0.97	0.96	0.96
Total	594212	105194	64682	2505425	3269513	0.63	0.72	0.66

The building segmentation output was used as a mask for two tasks. First, all the line segments out of the building mask area were set to zero so that model would not learn irrelevant objects. However, this led to removing some line segments of our interest because the predicted lines do not fully overlap with the edges of the building mask. Figure 14 shows the three major steps of prediction of the final Unet-Resnet101. After converting the predictions to vector format, the lines

smaller than 0.5 m were removed. Although defining this rule will clean up the outputs from irrelevant lines, it results in data loss at some points.

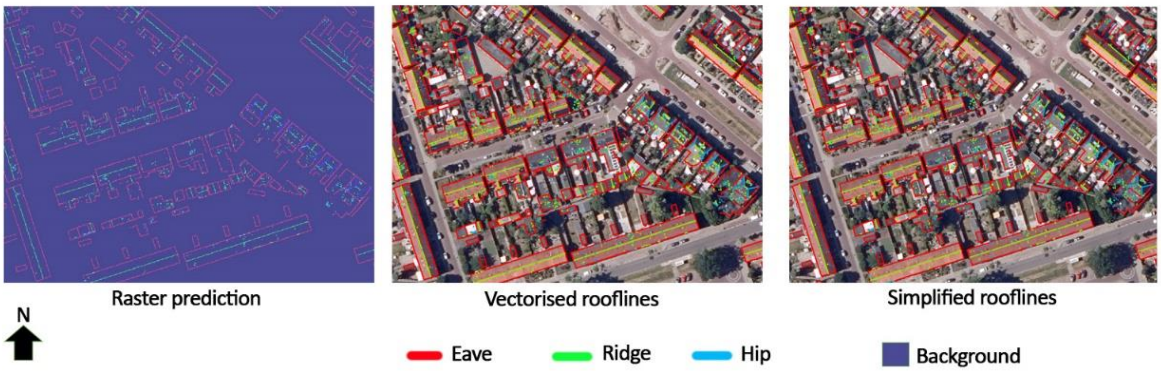


Figure 14- Initial raster prediction of the winner model (Unet-Resnet101)

Figure 15 depicts the overall framework to extract rooflines. Figures 15-a and b are the initial input RGB and nDSM, and c is the corresponding labels fed into the network. Figure 15-d is the ground truth binary mask (that is used instead of predicted binary mask) which is added as the fifth band to initial input datasets to improve the network's performance. 15-e shows the initial prediction of the network. This output suffers from some artefacts. First, the predicted lines do not intersect with their corresponding vertices. Second, the extracted lines are incomplete due to the fact that the network was not allowed to learn anything outside the binary building mask. In order to make regular lines, a Douglas-Peucker simplification with a 0.5 tolerance was applied using the line simplification toolbox in ArcGIS Pro, which is shown in Figure 15-f. Next, as shown in Figure 15-g, the incomplete lines were extended manually using the extend tool in ArcGIS Pro to meet their adjacent edges. Finally, the ground truth labels and the predicted lines are overlaid in Figure 15-h to understand how the model behaved clearly. In the end, the final output map of our proposed model in a larger area is shown in Figure 16.

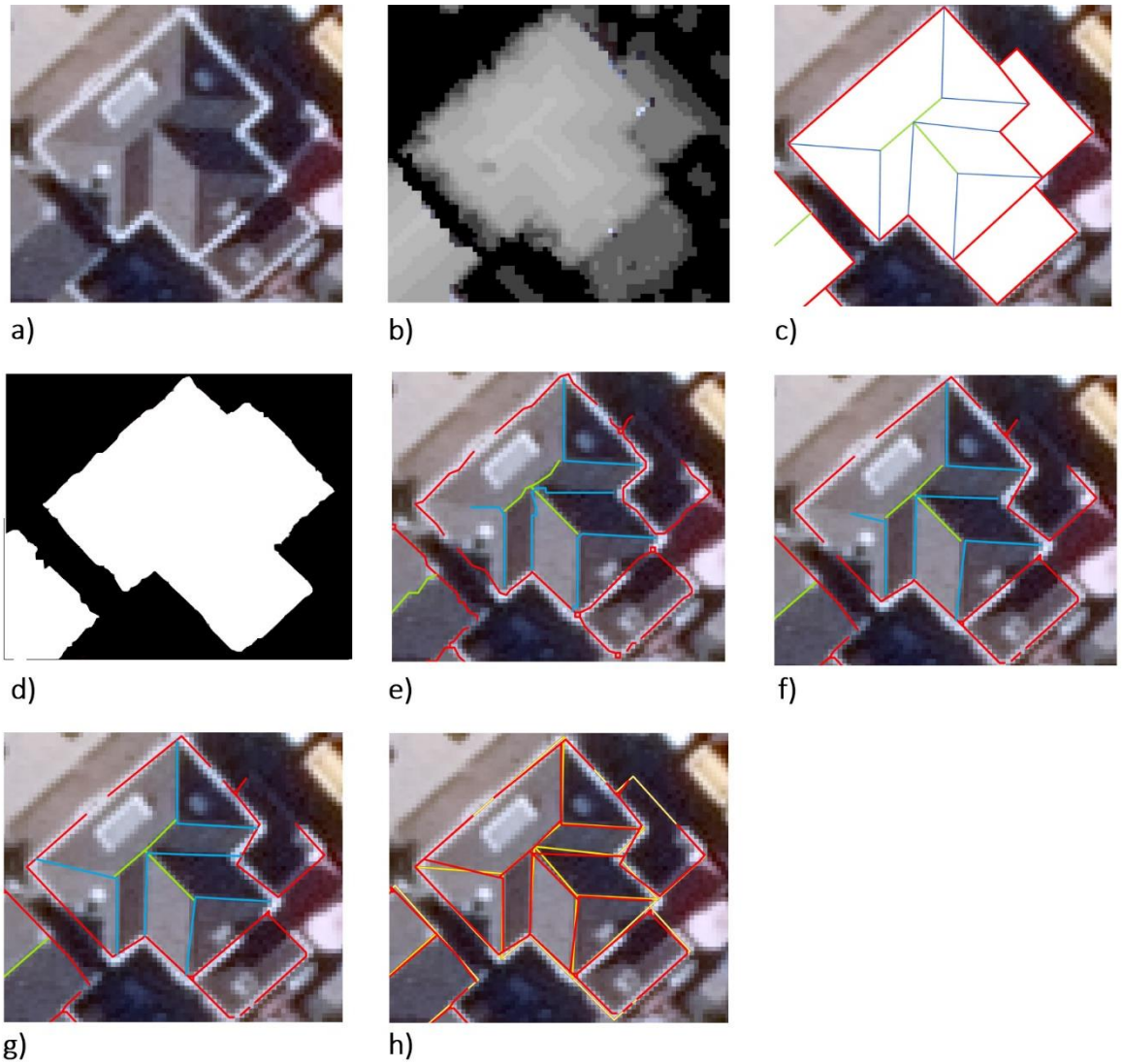


Figure 15- The roofline extraction workflow: (a) The input RGB image; (b) the input nDSM; (c) the labelled building polygon and corresponding rooflines; (d) the predicted binary building mask from stage A; (e) the predicted rooflines, Eave in red, Ridge in green and Hip in blue; (f) the simplified predicted rooflines; (g) the extended rooflines; (h) the difference between predictions in red and ground truth data in yellow.

Figure 16- Final output of our developed model

4.4. Summary

In this chapter, the optimal parameters to achieve this research's goal were studied and determined. Tables 12 and 13 give an overview of the selected values for each parameter.

Table 12- Optimal parameters for building segmentation

PARAMETER	ALTERNATIVES	OPTIMAL VALUE
Batch size	4, 8, 16	8
Learning rate	1e-3,1e-4,1e-5	1e-4
Activation function	ReLU	ReLU
Adam optimiser parameters	Beta1, Beta2, Epsilon	0.9, 0.999, 1e-8
Loss function	Cross entropy, Focal Tversky	Focal Tversky
Number of Epochs	100, 150, 200	150
Network depth	Unet-Resnet50, 101	101

Table 13- Optimal parameters for roofline extraction

PARAMETER	ALTERNATIVES	OPTIMAL VALUE
Batch size	4, 8, 16	8
Learning rate	Epoch scheduler, Exponential decay	Exponential decay
Activation function	ReLU, SELU	SELU
Adam optimiser parameters	Beta1, Beta2, Epsilon	0.98, 0.9, 1e-9
Loss function	Cross entropy, Focal Tversky, Dice loss	Focal Tversky
Number of Epochs	Early stopping	163
Network depth	Unet-Resnet50, 101	101

Having implemented the optimal values within the candidate networks, they were evaluated using predefined accuracy metrics. The network with the highest score is Unet-Resnet101, achieving an F1-score of 0.66. By taking a closer look into the class-wise evaluation, it was found out that the model's accuracy increases from eave lines to ridge and hiplines, respectively. In ridge and hip line predictions, recall values are higher than the precision values, which demonstrate the model's failure in classifying the line pixels correctly, which is confirmed by the visual evaluation of the model.

5. DISCUSSION

This research took advantage of deep learning to take a step toward 3D building reconstruction at LOD2. As such, a novel method was determined to extract roof structures in a linear format that can be generalised to any roof type. Unlike traditional segmentation models that are unable to detect inner walls, our proposed method is trained to spot both outer and inner walls (red lines called eaves throughout the document), which is a remarkable achievement in the field of building segmentation and mapping. However, there are some disadvantages to our proposed method and input datasets, as discussed below.

The resolution of aerial images plays a vital role in training the model. This research prioritised using the publicly available 25cm aerial images, which result in less accurate outputs. In addition, the nDSM was used as the fourth band to images to improve the results. Although the results showed promising improvements in segmenting both building masks and, more importantly, roofline segmentation, like where shadows cover the buildings, it also causes uncertainty to the network, such as Figure 17, where the borders between two buildings do not show any discontinuity in nDSMs. As a result, the simplification stage also fails, as the technique cannot generate a single line.

Figure 17- The prediction artefacts- left nDSM, middle initial prediction without simplification, right simplified lines

Another issue appears on roofs with solar panels or additional parts. The model misclassifies the solar panels as the ridgelines (Fig.18). This issue is somehow controlled with the help of nDSMs; however, where there is only a slight change of height, the nDSMs are unable to help.

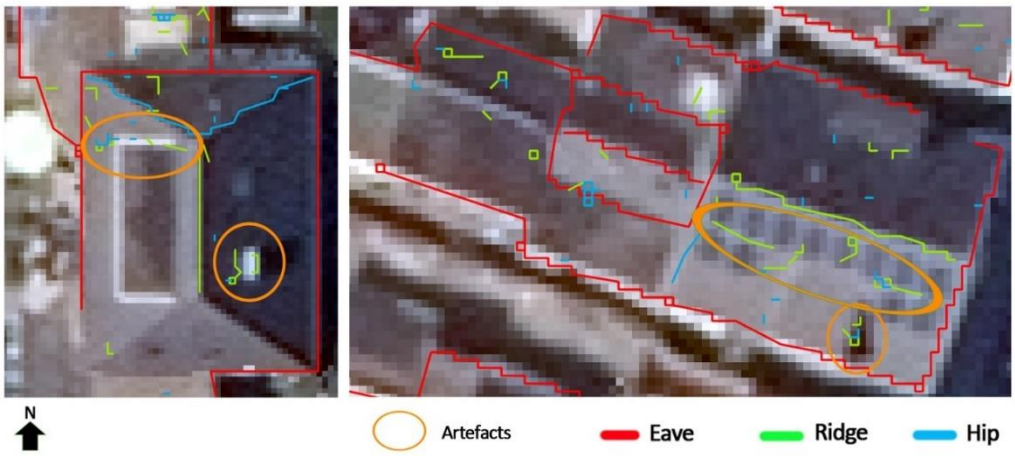


Figure 18- misclassification of solar pannels as inner lines

Another drawback of the predictions is the incomplete lines. Although the extend command can fix this issue at some points, it would not be of any help where the endpoints of the lines do not meet at any adjacent lines (Fig.19). This issue can be overcome by applying morphological filters or shape approximation stages to obtain closed boundaries. Besides, if the eave and ridgelines are predicted more accurately, predicting the hip lines is not precisely significant as they can only be extended from the endpoint of ridgelines to meet their adjacent vertices on eave lines.

Figure 19- Failure of extending command to fix incomplete lines

The next shortcoming of the proposed approach appears following the simplification stage, which considerably results in an accuracy drop. As shown in Figure 20, after applying the simplification technique, some vertices of our interest are wrongly eliminated. Optimising the tolerance for simplification or using other approaches such as frame field could be a possible solution to this issue.

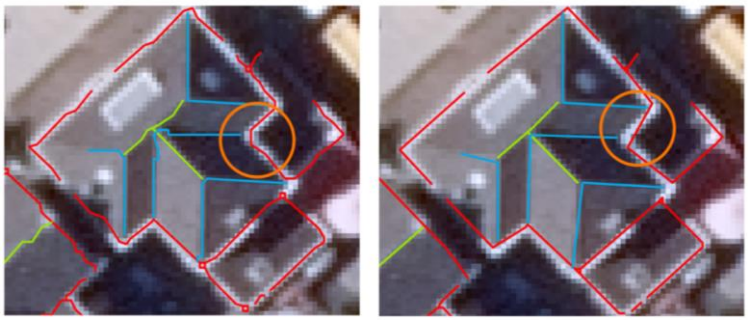


Figure 20- Loss of accuracy due to simplification

The next observed issue corresponds to tiny linear structures classified as ridge and hips, especially on flat roofs. Although the model was trained to distinguish between background and lines of interest, some misclassified pixels are scattered over the roofs, which might be caused by the nDSM layer, as shown in figure 21.

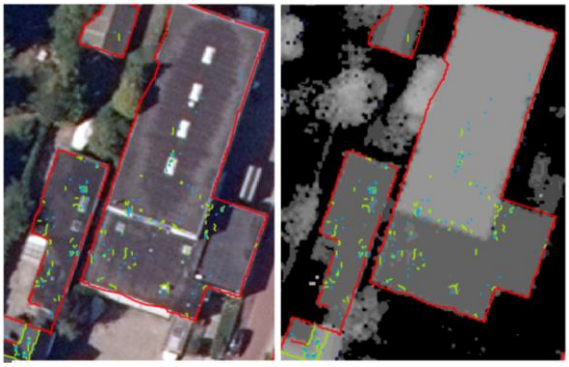


Figure 21- Misclassified pixels on flat roofs due to nDSMs

As previously discussed in chapter 4, quantitative evaluation metrics' value does not exceed the 0.66 average F1-score, which is reasonable, yet work has to be done to improve it. First of all, higher resolution aerial images (like 10 cm instead of 25cm) can improve the results to a great extent, specifically for ridge and hip lines that are more complex to be observed by the network. Furthermore, it is assumed that the small amount of data might be another culprit. In other words, if the amount of input data increases, perhaps the model will be able to learn more accurately. This hypothesis can be confirmed by looking into the evaluation metrics for each class. Achieving an F1-score of 0.81, the eave lines are the most successfully recognised class to which the majority of input pixel values belong. Moving toward ridge and hips, it becomes evident that the smaller input data results in a lower F1-score.

Through comparing the predictions and ground truth for different building shapes, it can be noticed that the model works better with simpler roof shapes. To be specific, the simpler the roof shape, the more accurate the prediction. It is also worth mentioning that the orientation of the buildings is a significant characteristic that majorly affects generating sharper and finer lines. In fact, the model behaves more accurately when predicting vertical and horizontal lines than those with a certain degree, such as 45°.

In order to use the proposed method for 3D building model generation, some complementary stages should be applied. First, the predicted rooflines need to be cleared of any noisy and small segments. Although a threshold of 0.5 m was defined initially to remove some irrelevant linear segments, it could not clear the predictions of all irrelevant segments. Increasing the threshold would eliminate some lines of interest as well. Therefore, a more sophisticated rule-based strategy is required to clear the data.
Additionally, some morphological operators such as closing and eliminating should be applied to connect the incomplete and fragmented lines. Next, the outline polygons generated from the corrected eave lines should be regularised by employing shape approximation techniques such as the Minimum Bounding Rectangle (MBR) or Minimum Bounding Triangle (MBT). Feature Manipulation Engine (FME) might also be handy in approximating the incomplete polygons. In addition to eave lines, some rules can also improve the ridgelines quality, for instance, a line crossing the centre of a polygon. Although the model was trained to predict the hip lines, this class can be independently obtained by connecting the endpoints of the ridgelines to the vertexes of the approximated polygons. Accordingly, generating accurate ridge lines is of greater importance. Finally, the height values of each class of regularised rooflines should be extracted from the nDSMs to reconstruct the 3D building model.

6. CONCLUSIONS AND RECOMMENDATIONS

6.1. Reflection to research objectives and questions

In this research, a methodology was proposed to automatically extract building rooflines which can be considered the primary components for 3D building reconstruction. To achieve this generic objective, the answers to three specific sub-objectives followed by several research questions were sought through this research. The answers are elaborated below based on the findings of our study.

Objective 1: To prepare the data for building roof structure extraction

a) What are the suitable datasets to be used?
b) What is the quality of the reference data?
c) How to prepare the required data?
d) How to design training and testing datasets?

In response to the first objective, Enschede, the Netherlands, was selected as our primary study area. The preference was on using the publicly available datasets. Therefore, the RGB images with 25 cm resolution and 50 cm nDSMs (resampled to 25) were obtained from the PDOK platform for 2018. The proposed method is a multi-stage approach; therefore, two subsets of labelled data are taken. For the first stage, the BAG building outlines, the RGB images and nDSMs, which are available nationwide in the Netherlands, were used to train the model. This stage took the entire Enschede for training and testing the model. On the other side, for the second stage, since the reference roofline labels for our proposed method have not been created so far, a part of Enschede was manually digitised in a line format (Eave, Ridge and hips) to fit our purpose. To be specific, the bases of eave lines are the BAG data which have been modified and adjusted based on nDSM, google maps and google earth 3D viewer to increase the accuracy. The other two classes were digitised from scratch. In this stage, the input data to train the model consist of the digitised labelled rooflines, RGB tiles and nDSM layers. For both stages within our model, 70% of the data was selected for training and validation, and 30% was used for testing purposes. Both training and testing areas were tiled with a size of 256*256 pixels.

Objective 2: To develop a methodology for automatic roof structure extraction as a prerequisite for 3D building reconstruction at LOD2

a) What are the methods for 3D building reconstruction?
b) What is the state-of-the-art DL methods in building delineation?
c) How to further develop the existing methods to move toward automatic 3D building reconstruction?

As a response to the second objective, two branches of Unet-Resnet architecture with 50 and 101 layers as a strong network recommended by other scholars for building mapping and segmentation were selected and tested. Unlike most studies that are limited to a predefined library of roof types, extraction of rooflines was proposed that can be generalised to any type of roof. Another advantage of this method is the ability to detect inner walls, which is a significant issue in conventional segmentation models. In order to further develop the existing methods, a multi-stage methodology was defined first to predict a building binary mask and then extract the rooflines in three classes of eave, ridge and hips. The idea was to use the binary mask added as a

5th band to the input image to facilitate the extraction of rooflines by limiting the network to learn within building objects only.

Objective 3: To evaluate the developed method and the created model

a) What metrics can be applied to evaluate this research?
b) What is the performance of the developed model for roof outlines and inlines extraction toward 3DBR?

To evaluate the model, the commonly used Precision, Recall and F1-score metrics were selected in both stages. In addition to the global accuracy assessment, a pixel-based class-wise evaluation was also utilised for the roofline extraction stage to understand the model's behaviour in predicting each class clearly. Considering the quantitative and qualitative performance of the model, we confirm that the model can be effectively used for rooflines extraction. However, there is room for our approach to be improved both in the acquired input data and network characteristics.

6.2. Conclusions

In this study, a multi-stage framework was proposed that employs the Unet-Resnet101 network to automatically extract rooflines (eave, ridge and hips), which can be taken as the perquisites for generating 3D building models with LOD2. It is claimed that this approach can be effectively used for building roof structure extraction by predicting linear elements of roofs, which can be generalised to any roof type. Furthermore, our method proved useful in detecting inner walls, unlike conventional segmentation methods. Besides, it was discussed that the post-processing stage to regularise lines is more straightforward than regularising polygons which is a major challenge in building segmentation tasks. Our developed model's evaluation results showed an average F1-score of 0.66, proving that the method is trustable and there is room for further improvements to progress with the proposed method. Comparing 0.72 precision and 0.62 recall values showed that our model is more successful in detecting positive instances. However, these predictions are not entirely precise, which is also confirmed by visually monitoring the outputs.
In addition, a class-wise evaluation was also run to have a better understanding of the model's limitations. The precision, recall and F1-score of each class are as follows, respectively: Eave (0.82, 0.81, 0.81), Ridge (0.49, 0.61, 0.55) and Hip (0.23, 0.51, 0.32). It was found that the hip class is the most difficult to be recognised by the network as it has the lowest frequency among the input dataset due to the relatively small study area.
At the current stage, complementary techniques, such as morphological operators, shape approximation, are necessary to make use of the predicted outputs of this research. Besides, it is recommended to use higher resolution input images and a larger set of data in future studies.

6.3. Recommendations

Based on current research, recommendations for future works are listed as follow:

The very first recommendation is to increase the resolution of images to higher resolutions such as 10 cm instead of 25 cm used in this research. In addition, due to the time limitation during this research, it was not possible to create a larger set of data to train the networks. Therefore, it is best to carry out future experiments with a larger dataset. Besides, to improve the generalisation capability of the proposed method, datasets should be enriched by a more variety of buildings,

including more complicated ones with non-parallel walls and rooflines over different city structures.

Another point to be considered in future works is developing a multi-task network that outputs both binary masks and rooflines within the same stage. Additionally, the building mask generation stage can be carried out using more up-to-date techniques that directly output the regularised vector formats of the building. Otherwise, integrating some other techniques such as shape approximation or using other platforms such as FME to create the sharp complete polygons is a must. Finally, more precise network optimisation and selection of networks characteristics would be effective.

LIST OF REFERENCES

Agoub, A., Schmidt, V., & Kada, M. (2019). Generating 3D City Models Based on the Semantic Segmentation of LiDAR Data using Convolutional Neural Networks. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*(4/W8), 3–10. https://doi.org/10.5194/isprs-annals-IV-4-W8-3-2019

Ahvenniemi, H., Huovila, A., Pinto-seppä, I., & Airaksinen, M. (2017). What are the differences between sustainable and smart cities ? *JCIT*, *60*, 234–245. https://doi.org/10.1016/j.cities.2016.09.009

Alidoost, F., Arefi, H., & Hahn, M. (2020). Y-SHAPED CONVOLUTIONAL NEURAL NETWORK FOR 3D ROOF ELEMENTS EXTRACTION TO RECONSTRUCT BUILDING MODELS FROM A SINGLE AERIAL. *Science, Computer Engineering, Geospatial*, *V*, 321–328. https://doi.org/10.5194/isprs-annals

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 Sensing*, *11*(19). https://doi.org/10.3390/rs11192219

Biljecki, F., Ledoux, H., & Stoter, J. (2016). GENERATION of MULTI-LOD 3D CITY MODELS in CITYGML with the PROCEDURAL MODELLING ENGINE RANDOM3DCITY. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*(4W1), 51–59. https://doi.org/10.5194/isprs-annals-IV-4-W1-51-2016

Biljecki, Filip, Ledoux, H., & Stoter, J. (2016). An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, *59*, 25–37. https://doi.org/10.1016/j.compenvurbsys.2016.04.005

Billen, R., Cutting-Decelle, A.-F., Marina, O., de Almeida, J.-P., M., C., Falquet, G., … Zlatanova, S. (2014). *3D City Models and urban information: Current issues and perspectives*. https://doi.org/10.1051/tu0801/201400001

Boonpook, W., Tan, Y., Ye, Y., Torteeka, P., Torsri, K., & Dong, S. (2018). A deep learning approach on building detection from unmanned aerial vehicle-based images in riverbank monitoring. *Sensors (Switzerland)*, *18*(11). https://doi.org/10.3390/s18113921

Buyukdemircioglu, M., Kocaman, S., & Isikdag, U. (2018). Semi-automatic 3D city model generation from large-format aerial images. *ISPRS International Journal of Geo-Information*, *7*(9). https://doi.org/10.3390/ijgi7090339

Castagno, J., & Atkins, E. (2018). Roof shape classification from LiDAR and satellite image data fusion using supervised learning. *Sensors (Switzerland)*, *18*(11). https://doi.org/10.3390/s18113960

Chen, Y., Hong, T., Luo, X., & Hooper, B. (2019). Development of city buildings dataset for urban building energy modeling. *Energy and Buildings*, *183*, 252–265. https://doi.org/10.1016/j.enbuild.2018.11.008

Dembski, F., Wössner, U., Letzgus, M., Ruddat, M., & Yamu, C. (2020). Urban Digital Twins for Smart Cities and Citizens : The Case Study of Herrenberg , Germany. *MDPI-Sustainability*, 1–17. https://doi.org/doi:10.3390/su12062307

Donkers, S. (2013). *Automatic generation of CityGML LoD3 building models from IFC model* (Delft University of Technology). Retrieved from http://resolver.tudelft.nl/uuid:31380219-f8e8-4c66-a2dc-548c3680bb8d

Douglas, D. H., & Peucker, T. K. (1973). ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *10*(2), 112–122. https://doi.org/10.3138/fm57-6770-u75u-7727

ElGhany, S., & Ibrahim, R. (2019). Diagnosis of Various Skin Cancer Lesions Based on Fine-Tuned ResNet50 Deep Network. https://doi.org/10.32604/cmc.2021.016102

Estevez, E., Lopes, N. V., & Janowski, T. (2016). *Smart Sustainable Cities. Reconnaissance Study*.

European commission. (2014). 2030 climate & energy framework | Climate Action. Retrieved October 7, 2020, from https://ec.europa.eu/clima/policies/strategies/2030_en#tab-0-0

European Union. (2011). *Cities of tomorrow- Challenges, visions, ways forward*. https://doi.org/10.2776/41803

Girard, N., Smirnov, D., Solomon, J., & Tarabalka, Y. (2020). *Polygonal Building Segmentation by Frame Field Learning*. 1–30.

Hämäläinen, M. (2020). SMART CITY DEVELOPMENT WITH DIGITAL TWIN TECHNOLOGY. https://doi.org/10.18690/978-961-286-362-3.20

Hamilton, S. E., & Morgan, A. (2010). Integrating lidar, GIS and hedonic price modeling to measure amenity values in urban beach residential property markets. *Computers, Environment and Urban Systems*, *34*(2),

133–141. https://doi.org/10.1016/j.compenvurbsys.2009.10.007
Hansen, K. (2019). Activation Functions Explained - GELU, SELU, ELU, ReLU and more. Retrieved June 29, 2021, from https://mlfromscratch.com/activation-functions-explained/#relu
Ibrahim, M. R., Haworth, J., & Cheng, T. (2020). Understanding cities with machine eyes: A review of deep computer vision in urban analytics. *Cities*, *96*(May 2019), 102481. https://doi.org/10.1016/j.cities.2019.102481
Jadon, S. (2020). *A survey of loss functions for semantic segmentation*. Retrieved from https://github.com/shruti-jadon/
Jafar, A., & Myungho, L. (2020). Hyperparameter Optimization for Deep Residual Learning in Image Classification. *Proceedings - 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion, ACSOS-C 2020*, 24–29. https://doi.org/10.1109/ACSOS-C51401.2020.00024
Ji, S., Wei, S., & Lu, M. (2019). Fully Convolutional Networks for Multisource Building Extraction from an Open Aerial and Satellite Imagery Data Set. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(1), 574–586. https://doi.org/10.1109/TGRS.2018.2858817
Joshi, R. (2019). Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog. Retrieved May 31, 2021, from https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/
Kadhim, N. (2018). *Creating 3D City Models from Satellite Imagery for Integrated Assessment and Forecasting of Solar Energy* (Cardiff). Retrieved from https://orca.cf.ac.uk/109232/1/Thesis%28Final Copy%29.pdf
Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, *2017-Decem*, 972–981.
Li, Ziqi, Zhang, Z., & Davey, K. (2015). Estimating Geographical PV Potential Using LiDAR Data for Buildings in Downtown San Francisco. *Transactions in GIS*, *19*(6), 930–963. https://doi.org/10.1111/tgis.12140
Li, Zuoyue, & Wegner, J. D. (2019). *Topological Map Extraction From Overhead Images*.
Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, *152*(April), 166–177. https://doi.org/10.1016/j.isprsjprs.2019.04.015
Moon, J., Park, S., Rho, S., & Hwang, E. (2019). A comparative analysis of artificial neural network architectures for building energy consumption forecasting. *International Journal of Distributed Sensor Networks*, *15*(9). https://doi.org/10.1177/1550147719877616
OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0. (2012). In G. Gröger, T. H. Kolbe, C. Nagel, & K.-H. Häfele (Eds.), *OGC Document No. 12-019*. Retrieved from https://portal.opengeospatial.org/files/?artifact_id=47842
Park, Y., & Guldmann, J. M. (2019). Creating 3D city models with building footprints and LIDAR point cloud classification: A machine learning approach. *Computers, Environment and Urban Systems*, *75*(November 2018), 76–89. https://doi.org/10.1016/j.compenvurbsys.2019.01.004
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 Sensing*, *11*(14), 1–38. https://doi.org/10.3390/rs11141660
Persello, C., & Stein, A. (2017). Deep Fully Convolutional Networks for the Detection of Informal Settlements in VHR Images. *IEEE Geoscience and Remote Sensing Letters*, *14*(12), 2325–2329. https://doi.org/10.1109/LGRS.2017.2763738
Pollino, M., Caiaffa, E., Carillo, A., La Porta, L., & Sannino, G. (2015). Computational Science and Its Applications -- ICCSA 2015. *Springer International Publishing Switzerland*, *9157*, 495–510. https://doi.org/10.1007/978-3-319-21470-2
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*(5). https://doi.org/10.3390/s19051164
Qiu, F., Sridharan, H., & Chun, Y. (2013). Spatial autoregressive model for population estimation at the census block level using LIDAR-derived building volume information. *Cartography and Geographic Information Science*, *37*(3), 239–257. https://doi.org/10.1559/152304010792194949
Sugihara, K., & Shen, Z. (2017). Automatic generation of 3D building models with efficient solar photovoltaic generation. *International Review for Spatial Planning and Sustainable Development*, *5*(1), 4–14.

https://doi.org/10.14246/irspsd.5.1_4
Tarabalka, N. G. and Y. (2018). *END-TO-END LEARNING OF POLYGONS FOR REMOTE SENSING IMAGE CLASSIFICATION Nicolas Girard and Yuliya Tarabalka Universit ´ e Cˆ ote d ' Azur , Inria , TITANE team , France Email : nicolas.girard@inria.fr*. 2087–2090.
Teo, T. (2019). *DEEP-LEARNING FOR LOD1 BUILDING RECONSTRUCTION FROM AIRBORNE LIDAR DATA*. 86–89.
Thomas, C. (2019). U-Nets with ResNet Encoders and cross connections | by Christopher Thomas BSc Hons. MIAP | Towards Data Science. Retrieved June 22, 2021, from https://towardsdatascience.com/u-nets-with-resnet-encoders-and-cross-connections-d8ba94125a2c
Tsang, S. (2018). Review: ResNet — Winner of ILSVRC 2015 (Image Classification, Localization, Detection) | by Sik-Ho Tsang | Towards Data Science. Retrieved June 22, 2021, from https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8
TU Delft. (2020). All 10 million buildings in the Netherlands available as 3D models. Retrieved April 27, 2021, from https://www.tudelft.nl/en/2021/bk/all-10-million-buildings-in-the-netherlands-available-as-3d-models
UNEP. (2018). The weight of cities. Retrieved October 7, 2020, from https://www.unenvironment.org/news-and-stories/story/weight-cities
United Nations. (2015a). Climate Action – United Nations Sustainable Development. Retrieved October 7, 2020, from https://www.un.org/sustainabledevelopment/climate-action/
United Nations. (2015b). What is the Paris Agreement? | UNFCCC. Retrieved October 7, 2020, from https://unfccc.int/process-and-meetings/the-paris-agreement/what-is-the-paris-agreement
Wu, Y., Filippovska, Y., Schmidt, V., & Kada, M. (2019). Application of Deep Learning for 3D building generalization. *Proceedings of the ICA*, *2*(July), 1–8. https://doi.org/10.5194/ica-proc-2-147-2019
Zhang, W., Wang, H., Chen, Y., Yan, K., & Chen, M. (2014). 3D building roof modeling by optimizing primitive's parameters using constraints from LiDAR data and aerial imagery. *Remote Sensing*, *6*(9), 8107–8133. https://doi.org/10.3390/rs6098107
Zhao, W., Ivanov, I., Persello, C., & Stein, A. (2020). Building Outline Delineation: From Very High Resolution Remote Sensing Imagery To Polygons With an Improved End-To-End Learning Framework. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLIII-B2-2*, 731–735. https://doi.org/10.5194/isprs-archives-xliii-b2-2020-731-2020
Zheng, Y., Weng, Q., & Zheng, Y. (2017). A hybrid approach for three-dimensional building reconstruction in indianapolis from LiDAR data. *Remote Sensing*, *9*(4). https://doi.org/10.3390/rs9040310
Zhu, L., Lehtomäki, M., Hyyppä, J., Puttonen, E., Krooks, A., & Hyyppä, H. (2015). Automated 3D Scene Reconstruction from Open Geospatial Data Sources: Airborne Laser Scanning and a 2D Topographic Database. *Remote Sensing*, *7*(6), 6710–6740. https://doi.org/10.3390/rs70606710