DEEP LEARNING-BASED BUILDING EXTRACTION USING AERIAL IMAGES AND DIGITAL SURFACE MODELS

XIAOYU SUN
JUNE, 2021

SUPERVISORS:
Dr. C. Persello
Dr. R V. Maretto

DEEP LEARNING-BASED BUILDING EXTRACTION USING AERIAL IMAGES AND DIGITAL SURFACE MODEL

XIAOYU SUN
Enschede, The Netherlands, June, 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.
Specialization: Geoinformatics

SUPERVISORS:
Dr. C. Persello
Dr. R V. Maretto

THESIS ASSESSMENT BOARD:
Prof.dr.ir. A. Stein (Chair)
Dr. R. Hänsch (External Examiner, DLR Germany)

ABSTRACT

Building information is essential in multiple applications. The emerging of very high-resolution remote sensing imagery made the recognition of small-scale objects like buildings possible. However, manually extracting buildings from images is time-consuming. Therefore, different automatic or semi-automatic approaches have been developed for building extraction. With the rise of deep learning, Convolutional Neural Networks (CNNs) have outperformed traditional methods based on handcrafted features and become the dominant approach in image analysis. As the most popular CNN type for semantic segmentation, fully convolutional networks (FCNs) are widely used in building extraction.

Most deep learning-based building extraction methods produce building masks in raster format, which cannot be directly integrated in geographic information systems (GIS) applications. Hence, some deep learning-based semantic segmentation models have been adapted for focusing on extracting building footprints as polygons directly. However, these models face the challenge of producing precise and regular building outlines. Recently, a building delineation method based on frame field learning was proposed by Girard et al., (2020) to extract regular building footprints as vector polygons directly from aerial RGB images. An FCN is trained to learn simultaneously the building mask, contours, and frame field followed by a polygonization method.

Optical imagery has some limitations. The normalized digital surface model (nDSM) derived from Light Detection and Ranging (LiDAR) data can provide 3D information, which can serve as complementary information to help overcome these limitations. Hence, we introduce 3D information into the framework and explore the data fusion of different combinations of aerial images (RGB), Near-infrared (NIR) and nDSM to extract precise and regular building polygons. The results are evaluated at pixel-level, object-level and polygon-level, respectively. Moreover, we performed an analysis to assess the statistical deviations in the number of vertices per building extracted by the proposed methods compared with the reference polygons. The comparison of the number of vertices focuses on finding the output polygons easier to be edited by human analysts in operational applications. This analysis can serve as guidance to reduce the post-processing workload for obtaining high accuracy building footprints.

The experiments were conducted in Enschede, the Netherlands. The results demonstrate 3D information provided by the nDSM overcomes the aerial images' limitations and contributes to distinguishing the buildings from the background more accurately. The method benefited from the data fusion and achieved better results using the composite images (RGB + nDSM) than those achieved using RGB and nDSM only, considering both quantitative and qualitative criteria. The height information could reduce the false positives and prevent missing the real buildings on the ground. In addition, the nDSM improves positional accuracy and shape similarity, resulting in better-aligned building polygons. The additional NIR information further improves the results. Compared with the alternative method, the method outperformed the PolyMapper in all coco metrics, which shows that the investigated model can predict more precise and regular polygons for the study area.

Keywords: Building Outline Delineation, Convolutional Neural Networks, Regularized Polygonization, Frame Field

ACKNOWLEDGEMENTS

I express my gratitude to all of those who helped me along the way…
…my parents are peasants who live most of their lives in a remote village. They insisted on supporting me to receive education while most of the teenagers are dropout of school. Their love and sacrifice change my destiny.

…my sister plays the parents, sister, friend roles in my life. Whenever I need her, she is always there. I feel so lucky to have her as my sister.

…my husband and my son. Quit our jobs and came to the Netherlands is a big decision for my family. My husband and I support each other to complete our master's degrees here while we live far away from my son. I hope I can be a role model for my son. The world is so big, go around to view more sceneries.

…my supervisor Dr. C. Persello, with his extensive experience in machine learning, always has great insights into the thesis and patiently inspired and motivated me, hoping that I can continue to challenge myself and think deeper and be innovative. I realize my deficiencies in critical thinking and innovation. What I learned from him will be of great help to my future work and study.

…my supervisor Dr. R V. Maretto, due to the pandemic, I have not met him in person, but he is very patient and always welcomes me to contact him online whenever I need help. In addition, he spent time analyzing specific technical details with me and give suggestions of solutions.

…now Ph.D. Wufan Zhao from the EOS department is a good advisor and provides a lot of technical assists. With his help, the data processing, which is the most time-consuming part, is largely shorted. It gives much more time to study the method itself.

…my advisors Vera Liem, Vincent van Altena, Marieke Kuijer from Kadaster, Thanks for guidance about the data pre-processing and preparation. Thanks to Vera Liem for the technical guidance of the design of the experiments. Thanks to Kadaster for supporting this academic research.

…my undergraduate teacher Mr. Jiejun Huang. He is always willing to help his students. Although I graduated for ten years, he still remembers me and wrote the recommendation letter for my application. He is a role model for me both in academic and normal life.

…my Chinese colleagues, for their friendship, accompany and food. They make my life in Enschede more colorful.

I am also grateful to the University of Twente for providing me with the ITC Excellence Scholarship to study this master course.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background and research problem

Buildings are an essential element of cities and information about them is needed in multiple applications. With the fast speed of urbanization, an increasing number of people live in cities. 55% of the world's population is living in urban areas, and this proportion is expected to increase to 68% by 2050 (United Nations,2019). Buildings serve as a shelter for humans living and activities, making building information more and more critical. Hence, the information is needed for many applications, such as urban planning, risk, and damage assessment of natural hazards, 3D city modeling, and environmental sciences (Nahhas et al., 2018). Precisely extracting the building boundaries is of utmost importance for applications like urban management, reconstruction (Z. Zhao, Duan, Zhang, & Cao, 2016). Especially for 3D modeling, footprint and height are two basic elements to generate building models (K. Zhang, Yan, & Chen, 2006).

Building extraction has been active for decades due to the availability of a large amount of high-quality remote sensing data and the need for buildings information in multiple applications. The emerging of Very High Resolution (VHR) remote sensing imagery made the recognition of small-scale objects like buildings possible, making object detection and object extraction become active fields to obtain building information. Object detection aims to extract the locations of objects in an image (Z. Q. Zhao, Zheng, Xu, & Wu, 2019). Object extraction, on the other hand, involves the detection of the object of interest and the extraction of its geometric boundary (Sohn & Dowman, 2007). Traditional building detection and extraction need human interpretation and manual annotation, which is highly labor-intensive and time-consuming, making the process expensive and inefficient (Sohn & Dowman, 2007). Therefore many researchers have been developing automatic or semi-automatic approaches for building detection and extraction in the last decades. Automated building detection and footprint extraction are essential instruments for map updating, 3D city modeling, and the identification of unregistered buildings. They speed up processing and reduce the costs of building detection and footprint extraction (Nex et al., 2013).

With the rise of deep learning, CNN-based models have outperformed traditional methods and become the dominant approach in building extraction. The traditional machine learning classification methods are usually based on spectral, spatial and other handcrafted features. The creation and selection of features depend highly on the experts' experience of the area, which results in limited generalization ability (W. Zhao, Persello, & Stein, 2021). The CNNs can extract spatial features from images and demonstrate excellent pattern recognition capabilities, making it the new standard in the remote sensing community for semantic segmentation and classification tasks. As the most popular CNN type for semantic segmentation, FCNs are widely used in building extraction. The pooling and convolution in FCN often result in the loss of the location information, making it difficult to extract buildings of different sizes, especially large buildings. Liu et al., (2019) propose a novel FCN to solve the problem. The convolution kernel size is enlarged and dilated convolution is used to capture more context information. They modified the ResNet-101 encoder to generate multi-level features and used a new proposed spatial residual inception module in the decoder to capture and aggregate these features. The network can extract buildings of different sizes.

Most GIS applications need building information in vector format. The conventional deep segmentation results in a raster format could not be used directly in multiple applications, which still need further

processing to obtain the building in a polygon format before use. The output building masks are often produced with over-smoothed corners and irregular edges, mainly caused by shift and spatial invariant characteristics of a CNN architecture. The imbalance between building content and boundary label pixels also results in irregular edges (W. Zhao et al., 2021). The conventional deep segmentation is often not able to produce sharp corners, which results in undesired artifacts. These methods need expensive and complicated post-processing procedures to refine the results (Girard et al., 2020). Due to these problems, traditional semantic segmentation methods are not able to produce accurate and regular buildings. In contrast, building delineation could obtain more regularized building outlines that are ready for most GIS applications.

With the need to automatically delineate objects in polygons, Li, Wegner, & Lucchi (2019) proposed an end-to-end deep learning architecture named PolyMapper, which skips the semantic segmentation step, takes aerial images as input and output polygons directly. The network is composed of a CNN to extract corners and an RNN to connect them to generate polygons. Manual interpretation and annotation to produce object boundaries in vector format are time-consuming and expensive. Their work makes the whole annotation process automatic. The PolyMapper could automatically delineate the building boundaries, but it performed worse on large buildings than Mask R-CNN (Li, Wegner, & Lucchi, 2019). Moreover, it could not deal with the polygons with holes (Girard, Smirnov, Solomon, & Tarabalka, 2020).

Deep learning-based semantic segmentation models for building delineation such as PolyMapper face the challenge of producing precise and regular building outlines. Recently, Girard et al., (2020) proposed a building delineation method based on frame field learning to extract regular building footprints as vector polygons directly. An FCN is trained to learn simultaneously the building mask, contours and frame field followed by a polygonization method. The FCN is a multi-learning model, the corner sharpness and wall straightness of segmentation are increased by learning the related the frame field (Girard et al., 2020). With the direction information of the building contours stored in the frame field, the polygonization algorithm can detect the corners more accurately and preserve them in the simplification; the edges of the polygon can also be iteratively adjusted to be more aligned to the ground truth in the optimization. Hence the method can produce more precise and regular building contours. In addition, the method can predict polygons with holes.

Despite the recent progress made in this research field, accurately extracting buildings from optical images is still challenging and LiDAR data or its derivatives could help overcome some problems. Two main reasons may be pointed out why the building extraction is difficult: (i) Buildings have varied sizes and spectral response across the bands. Trees or shadows often obscure them. (ii) The high intra-class and low inter-class variation of building objects in high-resolution remote sensing images make it complex to extract the spectral and geometrical features of buildings (Huang, Zhang, Xin, Sun, & Zhang, 2019). Building objects can be extracted from many data sources like aerial imagery and airborne LiDAR scanning. Due to shadows and low contrast, using only optical imagery to extract buildings in densely built-up areas does not perform well. Therefore, LiDAR data can serve as complementary information to help overcome these problems (Awrangjeb & Fraser, 2014). The fusion of LiDAR point cloud and aerial images could improve the quality of building detection (Nahhas et al., 2018).

Many studies have been done to detect or extract buildings from the fusion of images and point clouds. The information of point clouds could be transformed into images and used directly in the networks. Rizaldy, Persello, Gevaert, & Oude Elberink, (2018) converted the LiDAR point clouds data into a large image, then used an FCN to classify pixels into the ground and non-ground classes. To better differentiate ground pixels from non-ground, dilated convolutions were used to capture features of a large area in their network. Their

method is efficient and achieved good accuracy. In addition to point clouds, Digital Surface Model (DSM) and nDSM derived from LiDAR point clouds or stereo imagery are the most commonly used models that encapsulate 3D information. The DSM represents the elevation information of the terrain surfaces while the nDSM represents the height information of the objects on the terrain surfaces. Nahhas et al., (2018) combined LiDAR-derived DSM, DEM, nDSM; the number of returns and spectral bands from orthophoto. The spectral and texture geometry and shape features are extracted from them and then fed into a CNN. The results show that their work outperforms traditional machine learning methods based on Support Vector Machine (SVM) and achieved a higher accuracy.

By integrated elevation from point clouds with the spectral information from images, fusion-based deep learning networks achieved good results. Bittner et al., (2018) proposed a Fused-FCN4s network that used three parallel branches to take the RGB, nDSM, and the panchromatic (PAN) band as input for each branch separately. The three networks are concatenated, and three convolutional layers are applied at the end. The network can extract buildings correctly, even the small ones. It was successfully applied to other cities, demonstrating a good generalization ability. Instead of using nDSM, Schuegraf & Bittner, (2019) take the low-resolution multispectral images and DSM as inputs directly. Compared with Fused-FCN4s, they found that the multispectral information only slightly increases the overall network parameters but led to more complete building footprints. Their Hybrid-PS-Unet can segment complex and tiny building structures accurately with fewer parameters and higher inference speed.

The building delineation is a promising direction that produces building polygons ready for multiple GIS applications. However, the state-of-art delineation method (Girard et al., 2020) only takes the aerial images (RGB) as input. Therefore, its performance is affected by the problems derived from the optical imagery. Optical imagery has some limitations, and data fusion can integrate different information of 2D and 3D to overcome these limitations. We combined the data fusion with the frame field learning method to overcome the drawbacks of the optical imagery, taking advantage of the height information provided by LiDAR point clouds derivatives. By integrating the aerial images with the nDSM as a single dataset and feed it into the frame field learning method, we combined the method with data fusion to delineate the building boundaries. This study aims to further improve accuracy by learning multiple characters from the data fusion.

1.2. Research objectives and questions

1.2.1. General objectives

This study aims to develop a deep learning strategy, which adapts the state-of-the-art method to take the additional 3D information to extract building contour in a vector format. Combining different features from both optical imagery and nDSM, we aim to improve the accuracy of building footprints. The general scheme of the adopted methodology is shown in Figure 1.

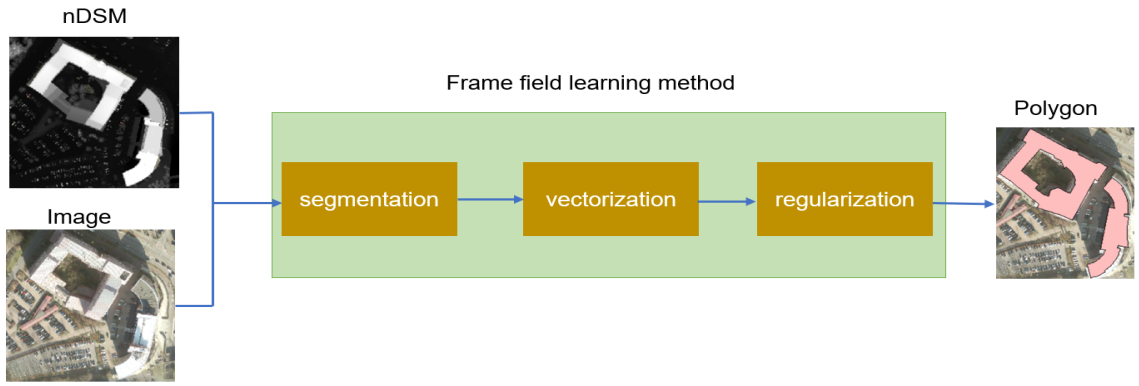

Figure. 1. *The general objective.*

1.2.2. Specific objectives

Objective 1: Prepare the data for building boundary delineation.
Objective 2: Develop a deep learning strategy that takes 2D and 3D data as input to generate building boundaries in polygon format.
Objective 3: Evaluate the results by assessing both qualitative and quantitative criteria. Compare the proposed methods with alternative strategies.

1.2.3. Research questions

Objective 1:
a. How is the quality of the available reference polygon data?
b. Are there any systematic or random shifts in building polygons from the corresponding boundaries?

Objective 2:
a. What relevant deep learning-based models exist, and what are their disadvantages and advantages?
b. When the resolution of nDSM is different from imagery, how can we perform the data fusion? For example, to fuse the data directly or adapt the network to take multi-resolution data as input?

Objective 3:
a. What are the advantages and disadvantages of the proposed model?
b. Does 3D information help to improve the results? Is the improvement significant?

2. LITERATURE REVIEW

2.1. Imagery-based building footprint extraction

In recent years, deep learning methods have become widespread in object extraction. The CNNs can extract spatial features from images and demonstrate excellent pattern recognition capabilities. Convolutional layers can extract high-level features like building boundaries (Wang, Yan, Mu, & Huang, 2020). Due to their ability to learn high-level features, CNNs are widely used in image classification and object segmentation. Using information from the neighborhood rather than considering the pixel in isolation enables it to learn contextual features, which is essential in the segmentation of buildings. The CNNs' ability to learn specific spatial and contextual features makes it suitable for detecting buildings (Griffiths & Boehm, 2019). Traditional CNNs consist of three types of layers: convolutional layer, pooling layer, and fully connected layer. The convolutional layer is where filters are applied to extract features from the input. The pooling layer applies a filter to downsample the input, increasing the receptive field of the next layer and reducing the dimensionality, reducing consequently the computational cost. It can also improve the robustness of the network to the exact location of the features. The fully connected layers are layers in which all neurons are connected to all neurons in the previous layer.

Shelhamer, Long, & Darrell, (2015) proposed FCN for semantic segmentation by replacing the fully connected layer of the CNNs with a 1x1 convolutional layer, enabling the network to accept arbitrary size input and produce a pixel-wise prediction. This feature makes it become the most popular CNN structure. FCN is widely used in semantic segmentation, including applications in building footprints extraction. To solve the incomplete and inaccurate problems when extracting buildings from VHR remotely sensed images, Shao et al., (2020) proposed an FCN named Building Residual Refine Network (BRRNet), which comprises the prediction module and the residual refinement module. To include more context information, they used atrous convolution in the prediction module. By adding zero paddings into the normal convolution kernel, the atrous convolution expands the kernel size to include more context information. The prediction module take images as input and output preliminary result. The residual refinement module takes the preliminary result as input. By comparing it with the ground truth and learned from the residual, the method can output more accurate results.

Mask R-CNN uses a small FCN as the mask branch and achieved good performance in instance segmentation. It detects the object by generating the bounding box of the individual objects and produces segmentation masks for the objects precisely (He, Gkioxari, Dollár, & Girshick, 2017). By removing the branch for category detection, L. Zhang, Wu, Fan, Gao, & Shao, (2020) adapted Mask R-CNN to building extraction. To solve the poor edge recognition and incomplete extraction of CNN, they refined the results by the Sobel edge detection algorithm. Although Mask R-CNN performed well in building segmentation, Wei et al., (2020) found that the details of the building were lost when small feature maps were up-sampled to the same size of the input, Hence when compared with FCN, the boundary of Mask R-CNN is over-smooth and less accurate. To thoroughly use the multiscale information, Wei et al., (2020) choose the feature pyramid network(FPN) as their backbone. Combining feature maps of different scales to extract buildings makes the method more robust and compact than other FCN-based methods.

2.2. Fusion-based building footprint extraction

Building objects can be extracted from many different data sources. With the rapid increase in the availability of multisource data, a lot of work has been done to explore the data fusion of multisource data. The optical sensors have some limitations, such as sensitivity to clouds and illumination, which influence the image quality. Buildings are also obscured by shadows or trees in imagery. In contrast to optical sensors, LiDAR

sensors have a different imaging mechanism that can penetrate clouds and sparse vegetation. Hence it could help to alleviate the performance degradation caused by the optical sensor (Hong et al., 2020). The information of point clouds could be transformed into images and used directly in the networks(Aldino Rizaldy, Persello, Gevaert, Oude Elberink, & Vosselman, 2018). In addition to point clouds, DSM and nDSM are two popular options that are widely used in data fusion to extract buildings.

Many studies have been done to detect or extract buildings by fusing spectral information of optical images with 3D information from point clouds derivatives. Different data combinations are tested on a variety of CNNs variants to extract building footprints. Since the information in this channel is strongly correlated with the red and green channels, containing many redundancies Griffiths & Boehm, (2019) used the 3D information to replace the blue channel in the RGB image. They feed the fused data into RetinaNet and Mask-RCNN. The result shows that RetinaNet is better than Mask R-CNN. The dataset has the issue of large class imbalance between the foreground and easy to detect background. The focal loss function used by RetinaNet could make the model focus on the hard example. Huang, Zhang, Xin, Sun, & Zhang, (2019) feed four channels (NIR, Red, Green, nDSM) into their GRRNet. They introduced a new gated feature labelling unit to solve the issues related to feature selection and feature transmission in the encoder-decoder network architecture. Compared with alternative approaches, they achieved a competitive performance in building extraction when tested in public datasets.

In addition to proposing fusion network architecture and testing different integrations of multisource data, some studies also focus on finding optimized deep learning fusion strategies. Audebert, Le Saux, & Lefèvre, (2018) investigated early and late fusion of LiDAR and multispectral imagery and found that late fusion could recover errors from the ambiguous data. Early fusion can better perform joint feature learning but is sensitive to missing data. Hong et al., (2020) systematically discussed issues about the fusion of multimodal data, like when to fuse data and how to fuse data. They tested different fusion modules and found that middle fusion and late fusion tend to yield better classification results, especially middle fusion. For fusion strategies, they found compactness-based fusion networks (including encoder-decoder fusion strategy and newly proposed cross fusion) show their superiority in blending multimodal features than others. Compared with other fusion strategies, cross fusion can transfer the information across modalities more effectively.

2.3. Building footprint delineation

Even though building polygons are ready for most GIS applications, manual interpretation and annotation are time-consuming and expensive. One kind of approach to extract building contour is based on the active contour models (ACM), also known as the snakes(Kass & Witkin, 1988). Mayunga, S. D., Zhang, Y., & Coleman, (2005) proposed a semi-automatic method that uses a radial casting algorithm and snakes to extract building outlines. However, The model has many parameters that need to be set experimentally, making it unsuitable for the large area (Nguyen, Daniel, Gueriot, Sintes, & Caillec, 2020). Nguyen, Daniel, Gueriot, Sintes, & Caillec, (2020) proposed an automatic method based on the ACM. They solved the problem by learning some important parameters by CNNs and achieved a high accuracy result. But their model needs complicated data pre-processing to extract preliminary building boundaries from LiDAR and create high-resolution LiDAR-based elevation images.

Other methods rely on deep learning-based methods to obtain building boundaries. There are different deep learning-based strategies to obtain building footprints in vector polygons. Most widely used pixel-based segmentation methods that output building masks, which need multiple steps to obtain polygons. First, a binary segmentation map is produced by the deep-learning method, then a boundary extraction and a polygonization are applied on the map to get building footprint delineation (Wei et al., 2020). The conventional deep segmentation often could not produce sharp corners, which results in undesired artifacts.

These methods need expensive and complicated processing procedures to refine the results (Girard et al., 2020). Wei et al., (2020) proposed a multiscale aggregation FCN to extract building pixels. The building segmentation results are further refined and then converted to vector form to get polygons of buildings.

Instead of predicting segmentation for each pixel, deep learning-based delineation methods are trying to predict polygons directly. To facilitate the manual annotation process and provide a faster annotation tool, Castrejón, Kundu, Urtasun, & Fidler, (2017) proposed Polygon RNN, which integrating a CNN and a Recurrent Neural Network (RNN) enables semi-automatic annotation. The RNN can make predictions that not only relying on the current input but also using the information of previous outputs. As the vertices of a polygon are related to each other, the RNN is used to predict the vertices of the polygon. The CNN is used to extract features of the object. Then Acuna, Ling, Kar, & Fidler, (2018) further improved their work and proposed a network named Polygon RNN++. It is a semi-automatic segmentation method that allows delineating an object boundary within a manmade bounding box iteratively. They provided an interactive segmentation method that produces the polygon vertices and connects them to outline the object. The semi-automatic way needs to draw a bounding box of the target object manually. It also allows the annotator to supervise and correct the vertices (e.g., drag) during the delineation process. Their work speeds up the annotation process and achieves a good performance compared with manual annotation. Although the method could output buildings in polygons, the process still requires a human inference to provide the approximate spatial extent of the interest object and refine the boundary.

Despite the excellent performance achieved by Polygon RNN++, it still requires human intervention. Li, Wegner, & Lucchi, (2019) proposed a deep learning architecture named PolyMapper which makes the whole process automatically. Instead of providing the bounding box by manual annotation, the method could derive the bounding box of object instance by FPN. Hence it makes the whole annotation process automatic without human intervention. The model relies on a CNN to take aerial imagery as input to find the vertices of the building and connected them by RNN to generate polygons. While most existing deep learning research focusing on building detection output the result of building segmentation in raster map, their end-to-end architecture skips the semantic segmentation step and outputs the building object in vector format directly (Li, Wegner, & Lucchi, 2019).

Zhao, Persello, & Stein (2021) upgrade the feature extractor and detection module of PolyMapper and improve its performance. Finding suitable features is extremely important for neural networks. They introduced two improvements to the feature extractor. One improvement is introducing a boundary refinement block to amplify the distinction of features, which helps differentiate buildings from their complex background in VHR remotely sensed images. Another improvement is introducing a global context block. The block can include the long-distance pixels in filters to effectively use global information. In addition, a stacked conv-GRU is introduced to the RNN to replace the conv-LSTM to simplify the RNN and improve the performance. It can preserve the geometric relationship of the previous prediction but with fewer gates. Their method outperformed PolyMapper in all mAP and mAR metrics, demonstrating that it can predict more buildings correctly. Furthermore, their method performed better for medium and small buildings.

Despite the advantages of automation, these deep segmentation methods suffer from several disadvantages. They are difficult to train, and their output topology is restricted to simple polygons without holes. These methods also cannot deal with buildings with shared walls (Girard et al., 2020). To solve these problems and produce regular building polygons, Girard et al., (2020) trained an FCN to learn pixel-wise segmentation and a frame field aligned with the object tangents. The FCN is a multi-learning model, the corner sharpness and wall straightness of segmentation are increased by learning the related the frame field (Girard et al., 2020). The outputs of the model are inputs to a followed polygonization algorithm to obtain the building boundaries in polygons. The frame field is the key element in this method, and at least one field direction is

aligned to the tangent direction of the contour when it locates along the building edges. Therefore it stores the direction information of the tangent of the building outlines. The frame field is learned at every pixel of the image and used in the polygonization algorithm later. With the direction of the building contour, the edges of polygons are iteratively adjusted to be more aligned to the ground truth in the ACM. With the frame field, the corner can be detected and preserved during the simplification process. In this way, the method can produce regular and precise building outlines, especially for complex buildings with slanted walls.

3. METHODS

3.1. Overall workflow

The whole study process and experiment design are shown in Figure 2. We feed the network with fused aerial images and nDSM to improve the segmentation and frame field produced by the network. The process involves a series of procedures to train and optimize the network.

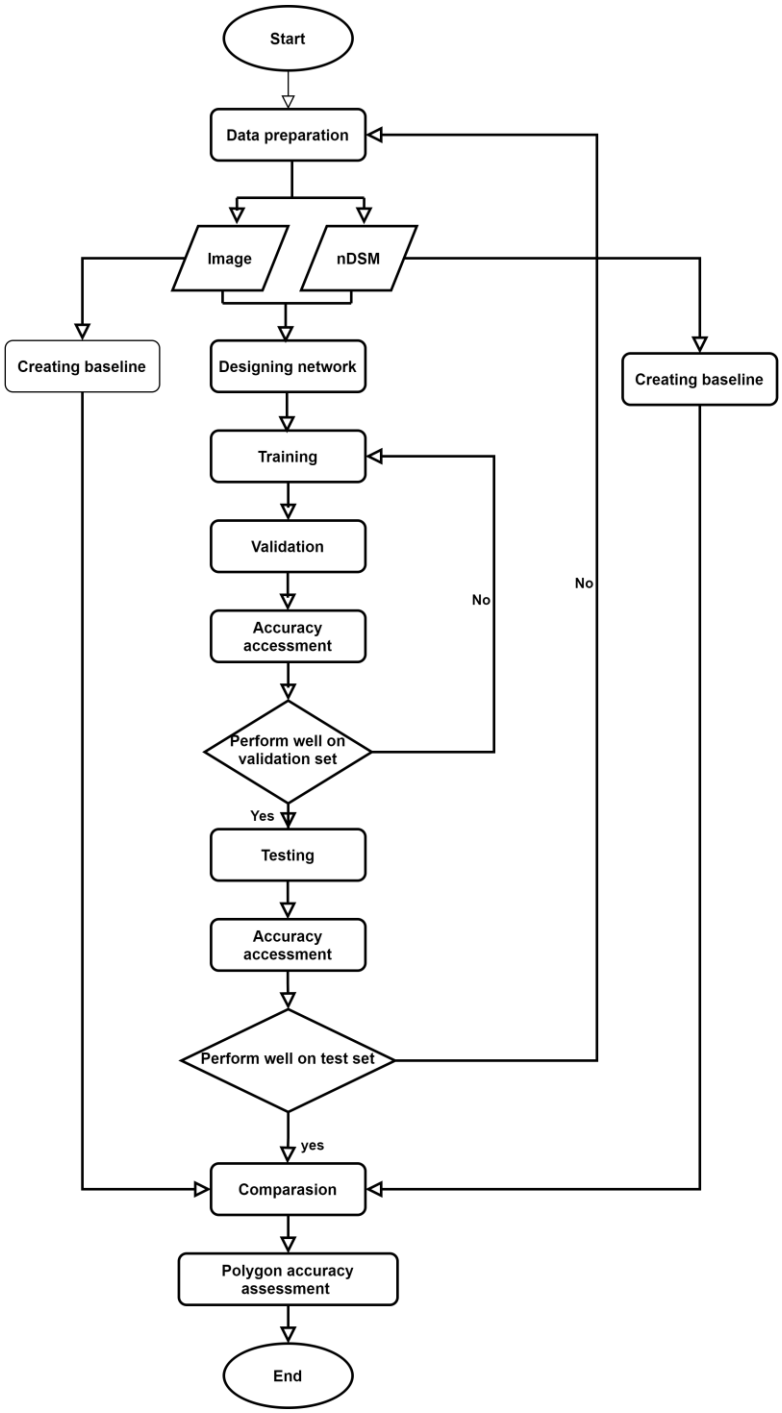

Figure. 2. The overall workflow.

The frame field learning method originally introduced in Girard et al. (2020) takes only the aerial images as input. However, the optical imagery has limitations, and the elevation data can help to overcome them. This thesis introduces the fusion of aerial images and 3D information (nDSM) into the framework to optimize the extraction of building polygons. We expect the data fusion can improve the accuracy and the geometrical regularity of the extracted building outlines. Figure 2 shows that two baselines are created for comparison to examine the difference caused by data fusion. One baseline takes as input the nDSM only; another one only analyses the aerial images. To be a fair comparison, all tiles of the different datasets are obtained with the same size and location; the setting of the networks are also kept the same. By comparing the results obtained from data fusion with the two baselines, we can evaluate the improvements due to the data fusion, especially the role of 3D information.

For the accuracy assessment, we evaluate our results at the pixel-level, object-level, and polygon-level, respectively. Furthermore, we analyzed the deviations in the number of vertices per building extracted by the proposed methods compared with the reference polygons. This is an additional accuracy metric capturing the quality of the extracted polygons that is not considered in standard metrics. It allows us to estimate the additional cost in human editing, which is generally still required for operational applications (e.g., cadastral mapping or the generation of official national geo data sets).

3.2. Boundary delineation with convolutional network

To better regularize the complex building, such as buildings with holes, Girard et al., (2020) trained an FCN to learn the interior map, edge map, and frame fields aligned with the building outline tangents. Then the frame field and interior map are used in the following polygonization algorithm. The architecture of the model is shown in Figure 3. The outputs of the model are the input of the polygonization algorithm shown in Figure 4. The polygonization algorithm is composed of several phases to delineate the building boundary. It can produce low-complex polygons almost without missing the corner vertices with the direction information from the frame field.

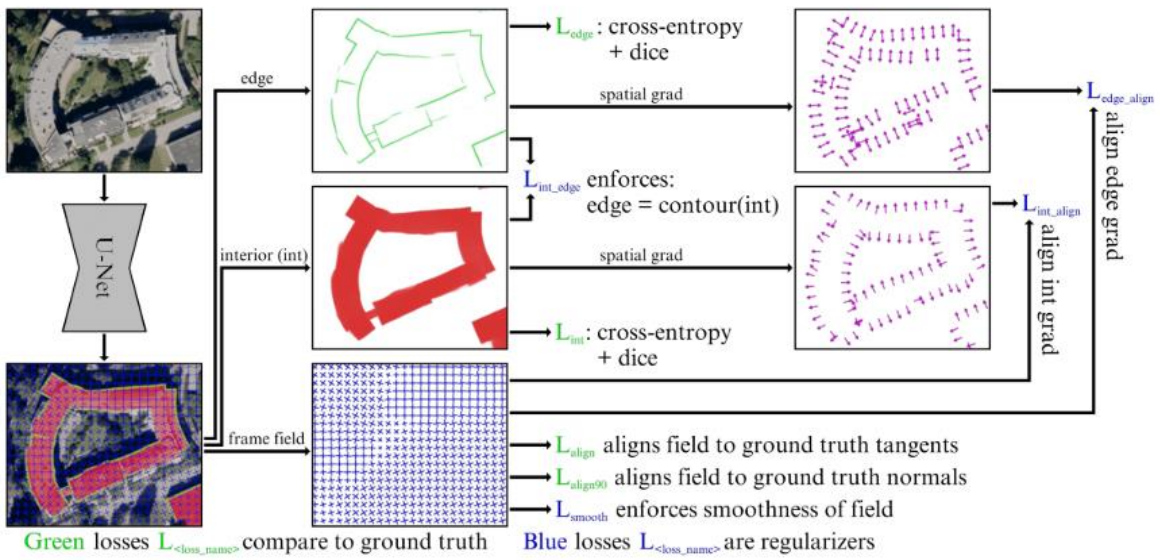

Figure. 3. Training procedure of adapted U-Net (Source: Girard et al., 2020)

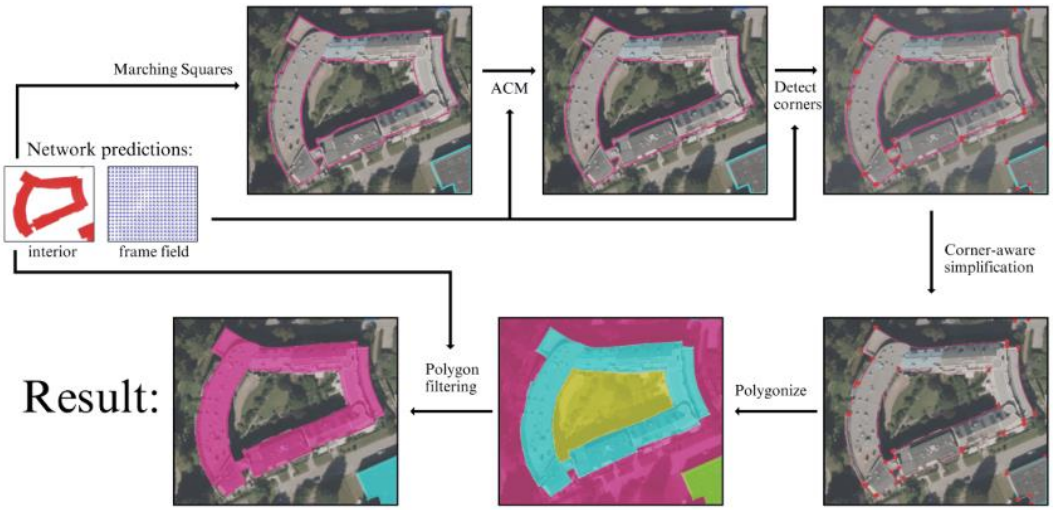

Figure. 4. Post-processing polygonization algorithm (Source: Girard et al., 2020)

Figure 5 shows the extended network, which takes images and nDSM as input data. We expected the additional height information could improve the intermediate frame field and building segmentation produced by the network. Because frame field is used to detect corners and optimize the polygon edges, and the segmentation is used to extract boundaries. We anticipated the predicted polygons would be improved too.

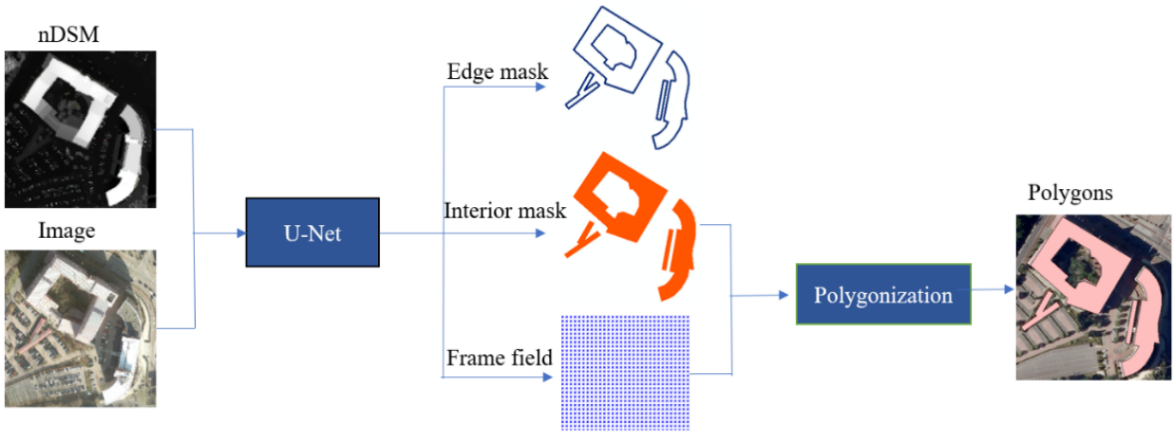

Figure. 5. The workflow of the investigated frame-field method for building delineation fusing nDSM and RGB data. Adapted from Girard et al., (2020).

3.2.1. Frame field learning

A frame field is comprised of two pairs of vectors with π symmetry each (Vaxman et al., 2016). It is a 4-PolyVector fields comprising two coupled 2-RoSy fields (Diamanti et al., 2014). "N-RoSy fields" are Rotationally-Symmetric fields, which are special vector sets comprising N unit-length vectors related by a rotation of an integer multiple of $2\pi/N$. An N-RoSy is the root set of the polynomials of the form $z^n - u^n$(Diamanti et al., 2014). If we denoted the two coupled 2-RoSy fields as u, v and the frame field as a (u, v) pair where $u, v \in C$, it has an order-invariant representative, which is the coefficients C_0, C_2 of the polynomial function in the equation 1 (Girard et al., 2020).

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

Where $C_0, C_2 \in C$. The frame field is the key element in this method, and at least one field direction is aligned to the tangent direction of the polygon when it locates along the building edges. Therefore it stores the direction information of the tangent of the building outlines. Instead of learning a (u, v) pair, a (C_0, C_2) pair was learned per pixel because it has no sign or ordering ambiguity.

A multitask learning model is designed to learn the frame fields and segmentation masks of buildings. These related tasks help the model to focus on the important and representative features of the input data. U-Net16 is used as the backbone. It is a U-Net backbone with 16 starting hidden features (Girard et al., (2020). The input layer of the backbone is extended to support taking input images with four or five channels. Then the output features of the backbone are fed into two branches with a shallow structure. The specific structure is shown in Figure 6. The edge mask and interior mask are produced by one branch as two channels of an image. The frame field is produced by another branch that takes the concatenation of the segmentation output and the output features of the backbone as input and outputs an image of four channels. The output frame field of the model is an image with four channels.

The model is trained in a supervised way. In the pre-processing part of the algorithm, the reference polygons are rasterized to generate reference edge masks and interior masks. For a frame field, the reference is an angle of the tangent vector calculated from an edge of a reference polygon. Then the angle is normalized to a range of $[0,255]$ and stored as the value of the pixel where the edge of the reference polygon locates. For other pixels where there is no edge, the value is zero. The reference data for the frame field is an image with the same extent as the original input image.

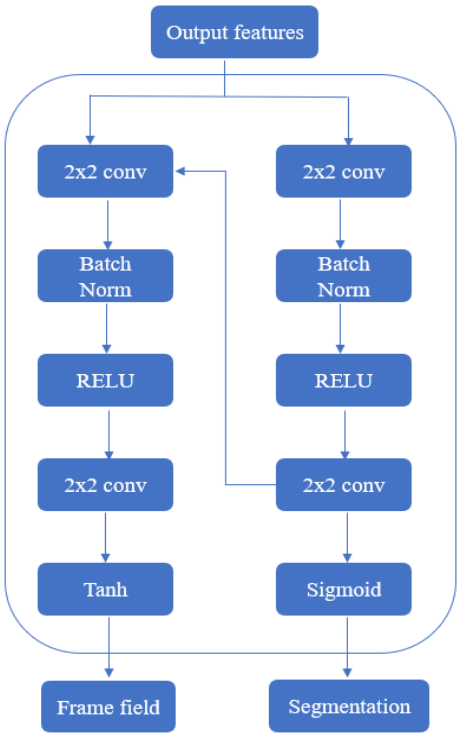

Figure. 6. The two branches produce segmentation and frame field.

3.2.2. Polygonization algorithm

The polygonization algorithm is composed of several steps. It takes the interior map and frame field of the neural network as inputs and output polygons corresponding to the buildings. First, an initial contour is extracted from the interior map by marching squares (Lorensen and Cline, 1987). Second, the initial contour is optimized by an ACM to make the edges more aligned to the frame field. Third, a simplification procedure is applied to the polygons to produce a more regular shape. Finally, polygons are generated from the collection of polylines from the simplification, and the polygons with low probabilities are removed.

ACM is a framework used for delineating an object outline from an image. A snake is a deformable spline influenced by external constraints that put the snake near the desired local minimum and image forces that pull it towards object contours and internal forces that resist deformation(Kass & Witkin, 1988). The image forces are related to features of the image like intensity and edge, use to adjust the contour to conform with the object in the image. The internal energy of the snake is related to the contour itself, use to control the continuity and smoothness. By energy minimization, snakes match a deformable model to features of interest in an image. The frame field and the interior map reflect different aspects of the building. In our method, the initial contour is produced by the marching square method from the interior map. The energy function is designed to constrain the snakes to stay close to the initial contour and aligned with the direction information stored in the frame field. Iteratively minimizing the energy function forces the initial contour to adjust its shape until it reaches the lowest energy.

The simplification is comprised of two steps. First, the corners are found with the direction information of the frame field. Each vertex of the contour corresponds to a frame field comprised of two 2-RoSy fields and two connected edges. If two edges are aligned with different of 2-RoSy fields, the vertex is considered as a corner. Then the contour is split at corners into polylines. The Douglas-Peucker algorithm further simplifies the polylines to produce a more regular shape. All vertices of the new polylines are within the tolerance distance of the original polylines. Hence the hyperparameter tolerance could be used to control the complexity of the polygons.

3.2.3. Loss function

The total loss function combines multiple loss functions for the different learning tasks: 1) segmentation, 2) frame field, and 3) coupling losses. H and W are the height and width of the input image, respectively. Different loss functions are applied to the segmentation. Besides combining cross-entropy loss (BCE) and Dice loss (Dice), Tversky loss is also tested for edge mask and interior mask. Tversky loss is proposed to mitigate the issue of data imbalance and achieve a better trade-off between precision and recall (Hashemi et al., 2018).

The BCE is given by equation 2.

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

where L_{BCE} is the cross-entropy loss, which applied to the interior and the edge output of the model, respectively.

The Dice loss is given by equation 3.

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

$$L_{int} = a \cdot L_{BCE}(\hat{y}_{int}, y_{int}) + (1 - a) \cdot L_{Dice}(\hat{y}_{int}, y_{int}) \tag{4}$$

$$L_{edge} = a \cdot L_{BCE}(\hat{y}_{edge}, y_{edge}) + (1 - a) \cdot L_{Dice}(\hat{y}_{edge}, y_{edge}) \tag{5}$$

Where L_{Dice} is the Dice loss, combined with the cross-entropy loss applied to the interior and the edge output of the model, respectively shown in equation 4 and 5. The a is the hyperparameter and was set to 0.25.

The Tversky loss is given by the equation 6 and 7.

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

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

Where p_{0i} is the probability of pixel i be a building (edge or interior), p_{1i} is the probability of pixel i is non-building. g_{0i} is the ground truth training label that 1 for pixel be a building and 0 for a non-building pixel, and vice versa for the g_{1i}.

A vector in a two-dimensional tangent space can be represented using Cartesian coordinates or equivalently as complex numbers. It is related to the angle-based representation via trigonometric functions or the complex exponential in equation 8 (Vaxman et al., 2016).

$$v = \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix} = e^{i\phi} \tag{8}$$

The output frame field contains four channels, each two for the two complex coefficients $C_0, C_2 \in \mathbb{C}$ They define an equivalence class corresponding to a frame field. The reference is an angle $\theta_\tau \in [0, \pi)$ of the tangent vector of the building contour. The following losses are used to train the frame field.

$$L_{align} = \frac{1}{HW} \sum_{x \in I} \hat{y}_{edge}(x) f\left(\mathrm{e}^{\mathrm{i}\theta_\tau}; C_0(x), C_2(x)\right)^2 \tag{9}$$

$$L_{align90} = \frac{1}{HW} \sum_{x \in I} \hat{y}_{edge}(x) f\left(\mathrm{e}^{\mathrm{i}\theta_\tau \perp}; C_0(x), C_2(x)\right)^2 \tag{10}$$

$$L_{smooth} = \frac{1}{HW} \sum_{x \in I} (\|\nabla C_0(x)\|^2 + \|\nabla C_2(x)\|^2) \tag{11}$$

From equation 8, we know that $e^{i\phi}$ is a vector tangent. In equation 9 and 10, the $\mathrm{e}^{\mathrm{i}\theta_\tau}$ represents a vector tangent to the building contour. θ_τ is the direction of vector τ, and $\tau^\perp = \tau - \frac{\pi}{2}$ .The L_{align}

makes the frame field more aligned with the tangent of the line segment of a polygon. L_{align} is small when the polynomial $f(\cdot; C_0, C_2)$ has a root near $e^{i\theta_\tau}$, meaning that one field direction is aligned with the direction of tangent τ. $L_{align90}$ prevents the frame field from collapsing into a line field. L_{smooth} produces a smooth frame field. Because these outputs are closely related and represent different information of the building footprints, there are the following functions, depicted in equations 12, 13 and 14, to make them compatible with each other.

$$L_{int\ align} = \frac{1}{HW}\sum_{x\in I} f\left(\nabla y_{int}(x); C_0(x), C_2(x)\right)^2 \tag{12}$$

$$L_{edge\ align} = \frac{1}{HW}\sum_{x\in I} f\left(\nabla y_{edge}(x); C_0(x), C_2(x)\right)^2 \tag{13}$$

$$L_{int\ edge} = \frac{1}{HW}\sum_{x\in I} max(1 - y_{int}(x), \|\nabla y_{int}(x)\|_2) \cdot \left|\|\nabla y_{int}(x)\|_2 - y_{edge}(x)\right| \tag{14}$$

Where $L_{int\ align}$ and $L_{edge\ align}$ constrain interior mask y_{int} and edge mask y_{edge} aligned with the frame field. The $L_{int\ edge}$ is to make the interior and edge mask compatible with each other.

3.3. Accuracy assessment

Pixel-level metrics. For evaluating the results, we used the mean Intersection over Union (IoU). IoU is computed by dividing the intersection area by the union area of a predicted segmentation (p) and a ground-truth (g) at the pixel level.

$$IoU = \frac{area(p \cap g)}{area(p \cup g)} \tag{15}$$

Object-level metrics. Building delineation is closely related to object segmentation, Average Precision (AP), Average Recall (AR) in MS COCO measures are introduced to evaluate our results. AP and AR are calculated based on multiple Intersection over Union (IoU) values. IoU is the intersection of the predicted polygon with the ground truth polygon divided by the union of the two polygons. There are 10 IoU thresholds range from 0.50 to 0.95 with 0.05 steps. As illustrated in Table 1, for each threshold, only the predicted results with IoU above the threshold will be count as true positives(tp). The rest will be denoted as false positives(fp). The ground truth with an IoU smaller than the threshold is a false negative(fn)(Girard et al., 2020). Then we could use equations 16 and 17 to calculate corresponding precision and recall. AP and AR are the average value of all precisions and recalls calculated over 10 IoU categories and can be denoted as mAP and mAR. AP and AR could also be further calculated based on the size of the objects: small (area < 32^2), medium (32^2 < area < 96^2), and large (area > 96^2). The area is measured as the number of pixels in the segmentation mask. They can be denoted as AP_S AP_M AP_L for the precision and AR_S AR_M AR_L for the recall. We followed the same metric standards but applied them to building polygons directly. To be specific, the IoU calculation is based on polygons. For the alternative method PolyMapper, as the input data is in COCO format, the evaluation is based on segmentation in raster format.

$$precision = \frac{tp}{tp + fp} \tag{16}$$

$$recall = \frac{tp}{tp + fn} \tag{17}$$

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="3">Reference</td></tr>
<tr><td colspan="2">Building</td><td>Non-building</td></tr>
<tr><td rowspan="3">Predicted</td><td rowspan="2">Building</td><td>IoU≥ threshold</td><td>True Positive</td><td rowspan="2">False Positive</td></tr>
<tr><td>IoU< threshold</td><td>False Negative</td></tr>
<tr><td>Non-building</td><td colspan="2">False Negative</td><td>True Negative</td></tr>
</table>

Table 1. Matrix of MS COCO measures

With the average precision and average recall calculated based on COCO metrics standards, the F1 score that is the weighted average of Precision and Recall can also be calculated by equation 18.

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{18}$$

For the polygons shown in Figure 7, when the threshold is 0.5, the left predicted polygon is almost fully overlapping with its reference, and the IoU is bigger than the threshold. Therefore, it is a true positive. For the polygon next to it, apparently, the predicted polygon is much smaller than its reference, and the IoU is smaller than 0.5. Therefore, the reference polygon will not be considered to be found and extracted. Hence, it is a false negative, the reference polygon in blue that does not intersect with any predicted polygon, demonstrating it is not found, which means it is considered non-building by the method. It is another false negative. The predicted polygon in red that does not intersect with any other reference polygon is a false positive. Hence the precision and recall for the examples in Figure 7 are 50% and 33%, respectively. The F1 score calculated based on them is 0.4.

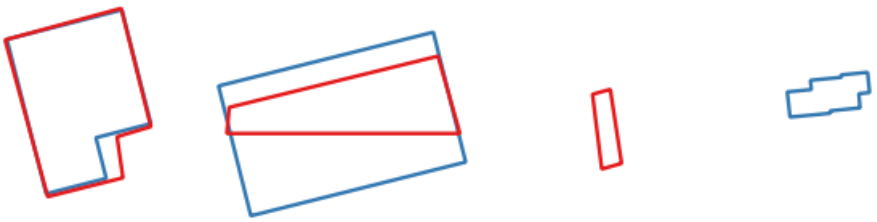

Figure. 7. Example predicted polygons(red) and the corresponding reference polygons(blue).

Polygon-level metrics. Besides the COCO metrics, polygons and line segments measurement (PoLiS) is introduced to evaluate the similarity of the predicted polygons with corresponding reference polygons. It accounts for positional and shapes differences by considering polygons as a sequence of connected edges instead of only point sets (Avbelj, Muller, & Bamler, 2015). We used this metric to evaluate the quality of the predicted polygon. We filter the polygons with IoU ≥ 0.5 to find the prediction polygons and the corresponding reference polygons. The metric is computed as depicted in equation 19.

$$p(A,B) = \frac{1}{2q}\sum_{a_j \in A} \min_{b \in \partial B} \|a_j - b\| + \frac{1}{2r}\sum_{b_k \in B} \min_{a \in \partial A} \|b_k - a\| \quad (19)$$

where $p(A,B)$ is defined as the average of the distances between each vertex $a_j \in A$,j = 1,…,q, of A and its closest point $b \in \partial B$ on polygon B, plus the average of distances between each vertex $b_k \in B$, k = 1,...,r, of B and its closest point $a \in \partial A$ on polygon A. The closest point is not necessarily a vertex, and it can be a point on edge. (1/2q) and (1/2r) are normalization factors to quantify the overall average dissimilarity per point.

Figure 8 shows the PoLiS distance between A and B. A black line indicates the distance from the vertices of a polygon to another polygon, and its arrow shows the direction. The distance between a vertex and polygon could be a distance from a vertex to another vertex or a point on the edge of another polygon. The dotted light-blue lines demonstrate one alternative way to connect point set B into a polygon. Even though it has the same vertices as the polygon connected by solid blue lines, the distance for the upper right corner of polygon A to polygon B is different. The shortest distance now points to another edge of polygon B, demonstrating polygon shape changes influence the distance calculation.

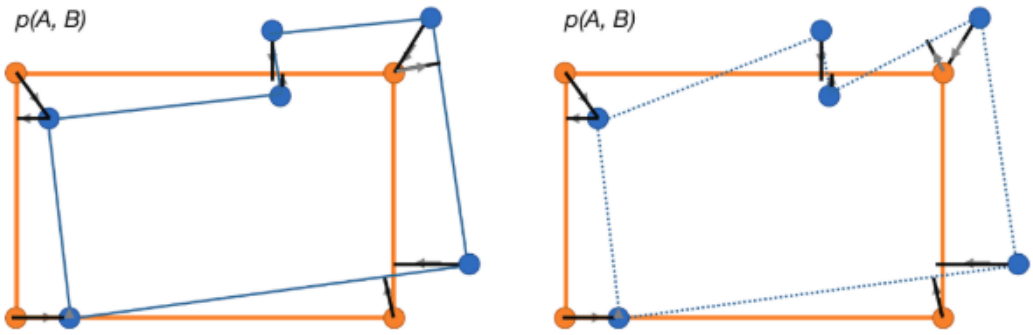

Figure. 8. PoLiS distance p between extracted building A (orange) and reference building B (blue) marked with solid black lines (Source: W. Zhao, Persello, & Stein, 2021).

To analyze the correlation between the number of vertices in the predicted polygon and their reference, we introduce the *average ratio of vertices number* and the *average difference of vertices number*. We first filter the polygons with IoU ≥ 0.5 to find the prediction polygons and the corresponding reference polygons. The *average ratio of vertices number* is computed by dividing the number of vertices of the predicted ones by that of their reference, then calculating the average value for all polygons as shown in equation 20. The *average difference of vertices number* is calculated by subtracting the number of vertices of the predicted ones by their references, then calculated the average value for all polygons shown in equation 21. Root Mean Square Error (RMSE) is also calculated by using the number of vertices of predicted polygons and their reference ones for all polygons, as shown in equation 22.

$$Average\ ratio = \frac{1}{n}\sum_{i=1}^{n} \frac{\hat{y}_i}{y_i} \quad (20)$$

$$Average\ difference = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i) \tag{21}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{22}$$

Where $\hat{y}_i$ is the number of the vertices for the predicted polygon and y_i is the number of the vertices for the corresponding reference polygon.

4. EXPERIMENTS SETUP

4.1. Study area and data

In this study, the municipality of Enschede was selected as the study area. It covers an area of 142.7 square kilometers and has 160,000 permanent residents. Specifically, it belongs to the province of Overijssel, right at the border between the Netherlands and Germany. A general view of the study area showed in Figure 9. The area in the red polygon is the Enschede.

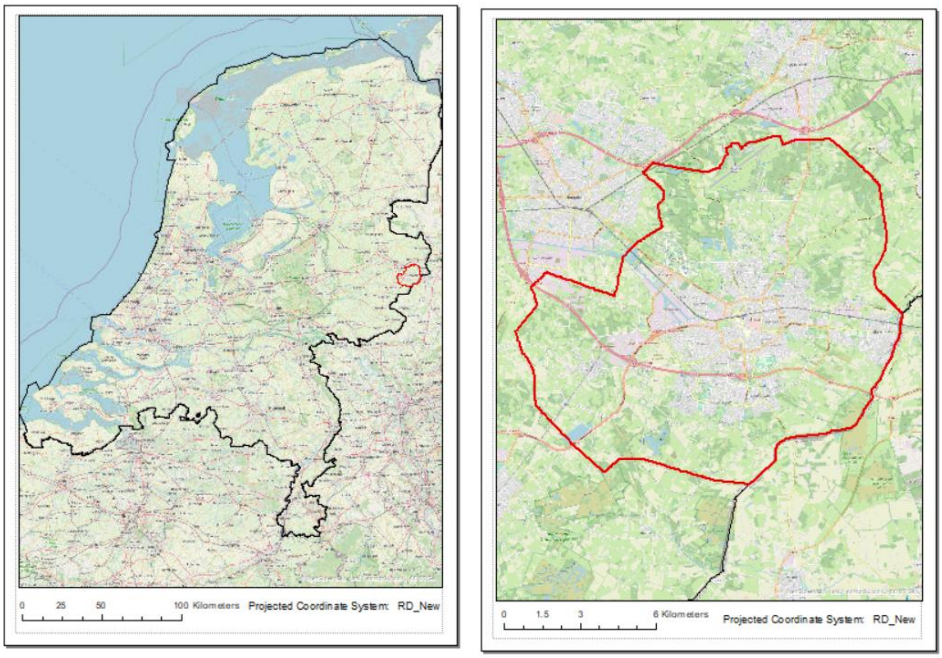

Figure. 9. The municipality of Enschede, study area. The area in the red polygon is the Enschede.

This research only includes collecting secondary data such as the available aerial images, DSM, and Digital Terrain Models (DTM). The aerial images are provided by Kadaster[1]. The 3D data are published on PDOK[2]. The PDOK is a portal website for obtaining open datasets from the government with current geo-information.

1) Aerial images

   Aerial images should provide a clear interpretation of visible building boundaries as much as possible. Hence, we choose VHR true orthophotos with 0.25 meter spatial resolution. The image of the study area is part of the nationwide summer flight in 2019. The sample data is shown in Figure 11.

2) Near-infrared (NIR) image

   The NIR image is an orthophoto, which was also acquired in the same nationwide summer flight in 2019 with a 0.25 meter spatial resolution.

3) nDSM

[1] Kadaster (The Netherlands' Cadastre, Land Registry and Mapping Agency)
[2] PDOK (the Public Services On the Map), https://www.pdok.nl/

An nDSM is obtained by subtracting the digital terrain model (DTM) from the DSM, then resampled to the same resolution as the images. The Current Elevation File Netherlands (AHN[3]) is the digital elevation map for the Netherlands. AHN3 dataset was acquired in the 3rd acquisition period (2014-2019). The mean point density of AHN3 is 8-10 points/m2. The DTM and DSM are derived from AHN3 based on the Squared IDW method with 0.5 m spatial resolution. The LiDAR point clouds and DSM are shown in Figure 10.

Figure. 10. Sample data of LiDAR point clouds(left) and the derived DSM with 0.5 meters of spatial resolution(right).

4) Building footprints

Building footprints are from the BAG[4] dataset, which is part of the government system of key registers. It is captured by a municipality and subcontractors, and data qualities may vary for different areas. The conclusions obtained for Enschede are not necessarily applicable to BAG data from another region. Example images and the corresponding building footprints are shown in Figure 11.

[3] AHN ((Het Actueel Hoogtebestand Nederland)
[4] BAG (Basisregistratie Adressen en Gebouwen)

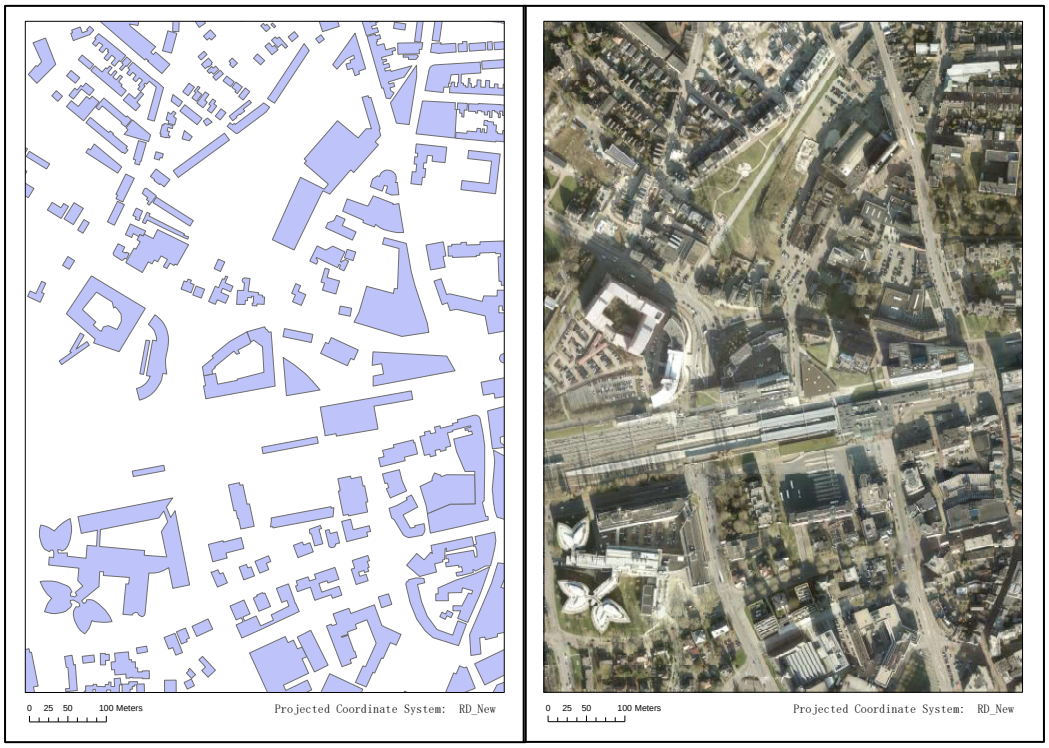

Figure. 11. Sample data of the reference data (left) and an aerial image of the represented area (right).

4.2. Data pre-processing

4.2.1. Building footprints

To check the accuracy of the BAG, we first used the polygons of OpenStreetMap as a reference. Through the spatial analysis in ArcMap, 826 polygons are found that do not exist in OpenStreetMap. Then these polygons were compared with the aerial images. The polygons that were different from the ground truth were manually edited. The details are shown in Table 2. The buildings which do not exist were removed. From the building status shown in Table 2, human activities cause these discrepancies, such as buildings not being constructed or demolished. The buildings with a shared wall are difficult to be distinguished by the network. The "dissolve" operation in QGIS was applied to BAG's original polygons to merge them into one. The dissolve results are shown in Figure 12.

Building status	Number of polygons before edit	Number of polygons after edit	Operation	Attributes in Dutch
Construction started	331	109	Remove the building where is a tree, grassland, or bare soil	Bouw gestart
Property out of order	1	1	Keep it	Pand buiten gebruik
Building in use	450	450	most of them exist in the aerial images. Keep all of them	Pand in gebruik
Building in use (not measured)	11	8	Remove the building where is a tree, grassland, or bare soil	Pand in gebruik (niet ingemeten)
Demolition permit granted	33	2	Remove the building that already demolition	Sloopvergunning verleend

Table 2. Edit operation of the polygons of BAG. Building status is an attribute for each polygon in the BAG.

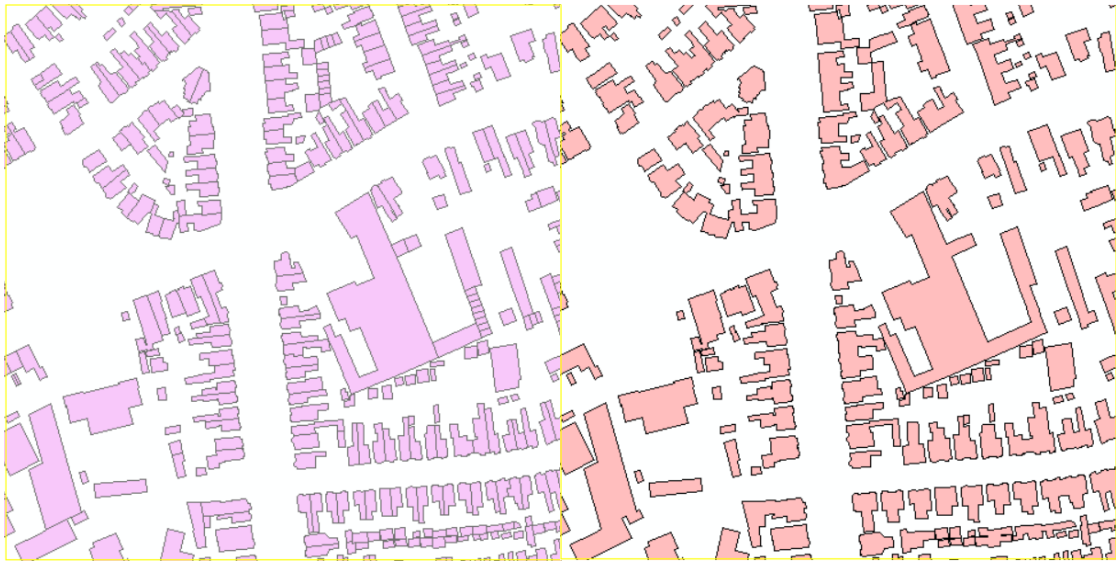

Figure. 12. Sample polygons of BAG dataset (left), sample polygons of BAG dataset after dissolve(right).

4.2.2. nDSM

The nDSM is produced by subtracting DTM from DSM. The DTM and DSM tiles were downloaded from PDOK and merged by the "Mosaic To New Raster" tool in ArcMap. Then they were resampled to the same resolution as the aerial images. Since there are no data values in built-up areas in DTM. They were filled by using the QGIS' fill nodata' tool with a maximum distance of 1000 pixels.

4.2.3. Datasets for deep learning

Table 3 shows the dataset produced based on BAG. The extent and distribution of tiles are the urban areas shown in Figure 13. Tiles are extracted from the aerial image (RGB), composite image (RGB + nDSM) and composite image (RGB+NIR+nDSM) with the same location and size. The composite image (RGB + nDSM) was produced by stacking the nDSM with the original aerial image as the 4th band. The composite image (RGB+NIR+nDSM) was produced by stacking the NIR as the 4th band and nDSM as the 5th band with the original aerial image.

Dataset	Number of tiles	Number of buildings	Ratio
training	579	29194	0.7
validation	82	4253	0.1
test	165	8531	0.2

Table 3. Information of the training set, validation set, and test set for the urban area using BAG reference polygons. The size of each tile is 1024×1024 pixels

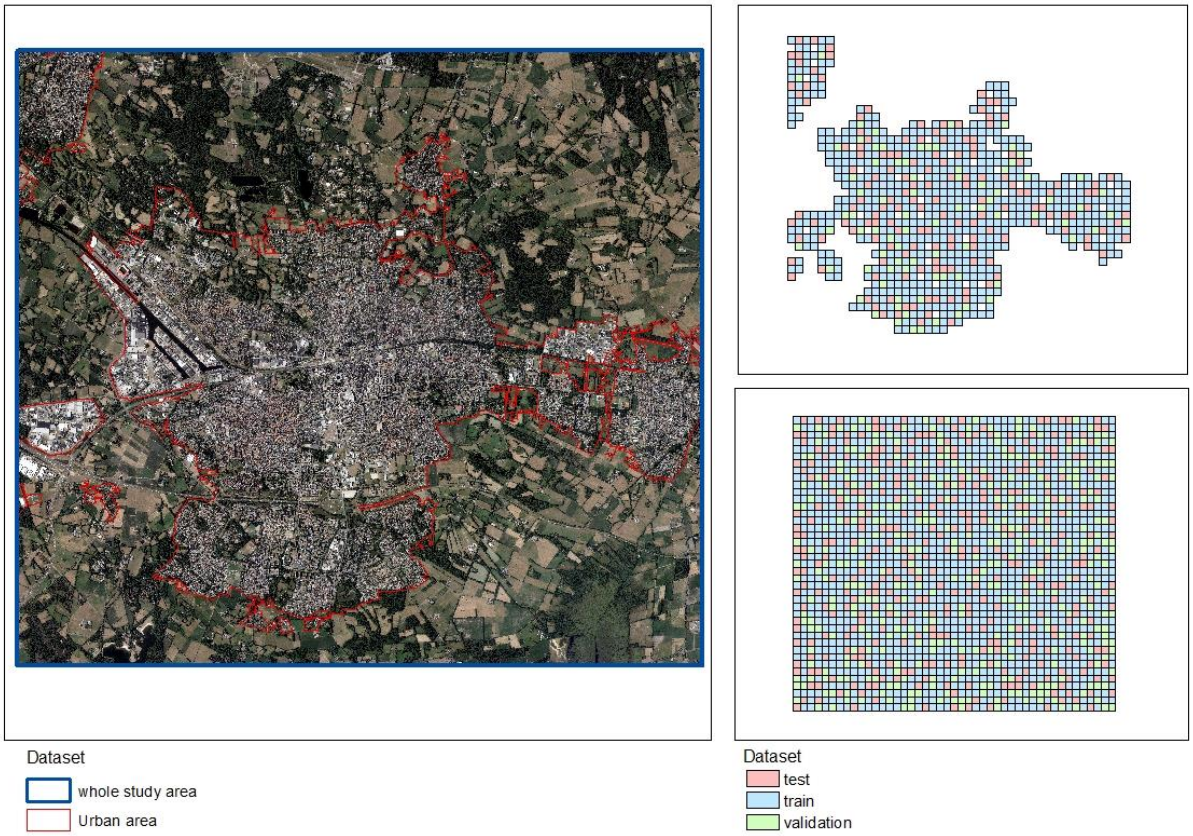

Figure. 13. The entire study area is the whole image of the city of Enschede; the urban area is denoted by the red polygons (right). The right side shows the tile distribution for the urban area (upper right) and the entire study area (lower right).

4.3. Implementation details

The model was trained with the following settings: Adam optimizer with a batch size b = 4 and an initial learning rate of 0.001. It applies exponential decay to the learning rate with a decay rate of 0.99. The max epoch is set to 200. The network is implemented using PyTorch 1.4. The training and testing are performed on a single NVIDIA Tesla P100 GPU. We set several values (1,3,5,7,9) for the tolerance parameter in the polygonization method. For each tile in the test set, the method produces polygons with a certain tolerance value. As we set multiple tolerance values, multiple polygons with different tolerance will be produced for each tile in the test set.

5. RESULTS AND DISCUSSION

This section will introduce the results achieved on the urban area dataset and discussion. Besides the results presented here, we also performed other experiments. However, the quality of the results obtained was considerably below those present in the thesis. Because of that, these experiments will not be detailed in the scope of this thesis. These experiments include:

1. Building footprints are obtained using publicly available geodata combining small buildings from the BAG with larger ones from BRT[5]. The BRT is a collection of digital topographical data on different scales. Buildings from the TOP10NL product were used in this experiment, which is topographical data suitable for the scales 1:5000-1:25000. We created two study areas: one is the urban area, and the other is the whole municipality (including the urban and rural areas), as shown in Figure 13. The results on the urban area are considerably better than that those achieved on the whole municipality. Girard et al., (2020) used the Inria dataset, which covers a larger extent and has all tiles extracted from urban settlements such as cities and towns(Maggiori, Tarabalka, Charpiat, & Alliez, 2017). Based on the difference between our dataset and the Inria dataset, we may hypothesize that the model needs a higher density of polygons in the training set to better learn the buildings' characteristics and perform well outside the city centers. The results achieved in the urban areas are considerably better but still worse than the results achieved using BAG as the reference building footprints only.

2. Take the ResNet-101 as the backbone for the FCN. It can achieve comparable results as Unet16, but Unet16 is more light-weighted. Therefore, the experiments present in this thesis all use the Unet16 as the backbone.

[5] BRT (Basisregistratie Topografie)

5.1. Quantitative analysis

Table 4 shows the quantitative results obtained using the composite images (RGB + nDSM), the single aerial images (RGB) and nDSM. The mean IoU achieved on the composite image is the highest, demonstrating that the method benefited from the data fusion and performed best on the fused data than the individual data source. The mean IoU achieved on the composite image (RGB + nDSM) test set was 80%, against 57% achieved for the test set of RGB image. The addition of the nDSM led to an improvement of 23% on the mean IoU. Compared with the results obtained only using nDSM, the mean IoU achieved on the composite image (RGB + nDSM) is 3% higher, which shows that the addition of spectral information only led to a slight improvement of the mean IoU. Hence we deduced that nDSM contributes more than aerial images in the building extraction. Moreover, the results obtained only with nDSM achieved a comparable accuracy, which is close to best the results obtained using the composite images (RGB+NIR+nDSM).

The same trend could also be found from the mAP and mAR of the composite image and two baselines. The mAP and mAR achieved on the composite images are considerably higher than those achieved on aerial images(RGB) only and slightly higher than those achieved on nDSM. Hence height information contributes more than spectral information in the building extraction. The higher average precision shows that height information help to reduce false positives, and higher average recall shows it helps prevent missing the real buildings on the ground. The composite image achieved higher precision and recall for all building sizes, demonstrating that it outperformed the individual source in all sizes of the buildings. In terms of size, buildings with medium size have the highest precision and recall. The small buildings have the lowest precision and recall, which means the model performs best for the medium building and worst for the small building. Fewer small buildings are correctly extracted, and more false positives are polygons of small size.

Comparing the results obtained on two composite images, the mean IOUs obtained with the BCE and Dice loss were almost the same, but the average precision and recall achieved on composite images (RGB + NIR + nDSM) were slightly higher, which means the NIR information helps to reduce false positives and prevent missing the real buildings on the ground. Tversky loss achieved the highest mean IOU (81.4%) and the highest average precision (43%) on the composite image (RGB+NIR+nDSM) among all the experiments. High precision means among the prediction polygons, most of them corresponding to real buildings on the ground. High recall means among the reference buildings, most of them are find and delineated correctly. The F1-score achieved on the same dataset with BCE and Dice loss is the highest. The F1 conveys the balance between precision and recall. The higher F1 value means the BCE and Dice loss can predict buildings more correctly and avoid missing the real buildings. It achieved a better balance between precision and recall.

Bands	Loss function	Mean IoU	mAP	mAR	F1	AP_S	AR_S	AP_M	AR_M	AP_L	AR_L
RGB, NIR, nDSM	BCE+Dice	0.805	0.425	0.499	0.447	0.262	0.200	0.591	0.609	0.543	0.478
	Tversky	0.814	0.430	0.413	0.412	0.218	0.244	0.457	0.507	0.502	0.376
RGB, nDSM	BCE+Dice	0.800	0.410	0.488	0.433	0.255	0.198	0.576	0.593	0.534	0.465
	Tversky	0.776	0.371	0.399	0.373	0.204	0.197	0.441	0.482	0.464	0.650
RGB	BCE+Dice	0.568	0.067	0.253	0.102	0.139	0.024	0.285	0.261	0.248	0.232
nDSM	BCE+Dice	0.767	0.313	0.436	0.347	0.197	0.129	0.532	0.553	0.525	0.420

Table 4. Extraction results for the urban area dataset. The mean IoU is calculated on the pixel level. Other metrics are calculated on the polygons with 1 pixel tolerance for polygonization.

In terms of the similarity of the polygons, Table 5 shows that the PoLiS distance achieved on the composite image (RGB + nDSM) is 0.54, considerably smaller than 0.87 for the RGB image and slightly smaller than 0.62 for the nDSM. The PoLiS distance achieved on the composite image (RGB + nDSM) is the smallest among all. The smaller PoLiS distance means the smaller dissimilarity, showing that the data fusion achieved the best similarity than the individual data source. The PoLiS distance obtained on nDSM is smaller than that obtained on aerial images, which means the nDSM contributes more than aerial images in improving the similarity for results obtained on the composite images. The PoLiS distance achieved on the composite image (RGB + NIR + nDSM) is 0.52, which is smaller than 0.54 achieved on the composite image (RGB+nDSM), demonstrating that the additional NIR information further improves the similarity. Furthermore, for the same composite images, the PoLiS distance of the model with the BCE and Dice loss is smaller than that with Tversky loss, which means the polygons produced by the combination of BCE and Dice loss are more similar to their reference.

Bands	Loss	PoLiS
RGB, NIR, nDSM	BCE+ Dice	0.52
	Tversky	0.62
RGB, nDSM	BCE+ Dice	0.54
	Tversky	0.62
RGB	BCE+ Dice	0.87
nDSM	BCE+ Dice	0.62

Table 5. PoLiS results for the urban area dataset. The PoLiS are calculated on the polygons with 1 pixel tolerance for polygonization.

5.2. Qualitative analysis

Figure 14 compares the predicted polygons obtained on tiles in the test set with different bands and the corresponding reference. The polygons obtained using the composite images are more aligned with the reference data and with fewer false positives than those obtained from RGB images or nDSM only. The performance gain is particularly visible for big buildings with complex structures and the building with holes. Fewer false positives are observed for small buildings in the results obtained using composite images. Compared with the polygons obtained from RGB images, the polygons obtained from the nDSM have fewer false positives and are more aligned with ground truth. In addition, the polygons of large buildings are more regular than the small ones in dense urban areas. There are more false positives for small buildings in dense urban areas than in sparse areas. By observation, some of them are storage sheds or garden houses, which are not included in the reference footprints. Their similar spectral character and height make it difficult to differentiate them from residential buildings. In summary, the nDSM improved building outlines' accuracy, resulting in better-aligned building polygons and preventing false positives. The polygons obtained from different composite images are very similar to each other.

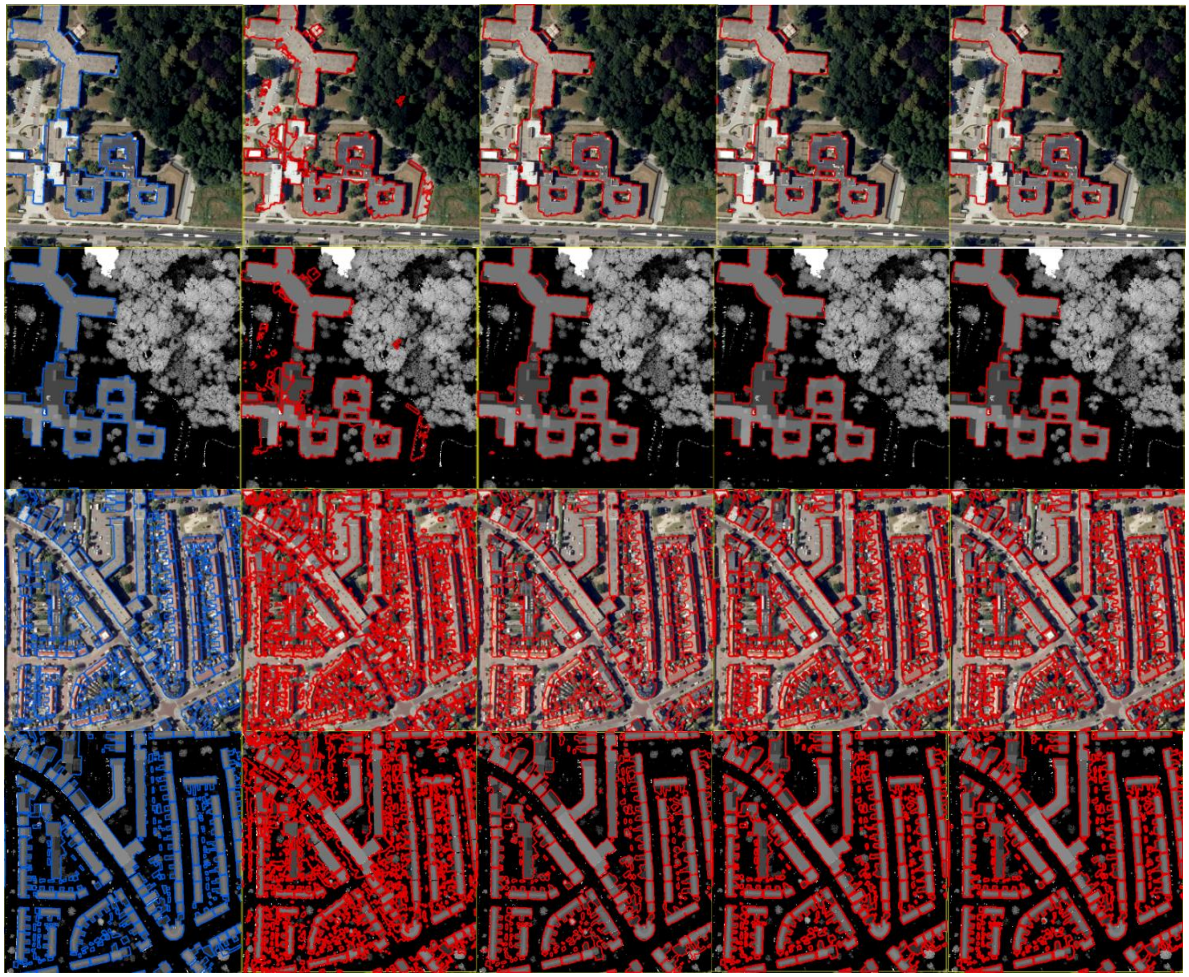

Figure. 14. Results obtained on two tiles of the test dataset for the urban area. The loss functions are cross-entropy and dice. The background is the aerial image and the corresponding nDSM. The predicted polygons are produced with 1 pixel for the tolerance parameter of the polygonization method.From left to right: (a) Reference building footprints, (b) Predicted polygons on aerial images (RGB), (c) Predicted polygons on nDSM, (d) Predicted polygons on composite images (RGB + nDSM), (e) Predicted polygons on composite images (RGB + NIR + nDSM)

Figure 15 shows the predicted polygon on different datasets. Compare the polygon obtained on the aerial image (RGB) with that on the composite image (RGB+nDSM), showing that the model cannot differentiate nearby buildings only with spectral information, which results in the predicted polygon on the aerial image (RGB) corresponding to several individual buildings. In addition, part of the road on the left side of the building is considered to be a building. Compare the polygon obtained on nDSM with that on the composite image (RGB+nDSM), showing that the model cannot differentiate closed buildings only with height information, which results in the upper right building is considered as part of the predicted building. Compares the predicted polygons on the composite image (RGB+nDSM) with that on the composite image (RGB+NIR+nDSM), the general shape is very similar to each other, the number of the vertex are almost the same, but the distributions are different. During the simplification in the polygonization process, the corners are kept while the other vertices are further simplified. Hence the corners are different too. The additional NIR affects the corner detection.

Table 6 shows the PoLiS distance of the example polygon. The polygon obtained on the composite image (RGB+NIR+nDSM) has the smallest distance, which is 0.39 against 0.47 for that on the

composite image (RGB+nDSM). Hence the additional NIR information help to improve the similarity between the predicted polygon with the reference polygon. The PoLiS distance achieved on nDSM is 0.81, which is considerably smaller than 5.32 obtained on aerial images only, demonstrating that the nDSM increased the similarity significantly.

(a) (b) (c) (d) (e)

Figure. 15. Results obtained on the urban area dataset. The predicted polygons are produced with 1 pixel for the tolerance parameter of the polygonization method. From left to right: (a) Reference building footprints, (b) Predicted polygon on aerial images (RGB), (c) Predicted polygon on nDSM, (d) Predicted polygon on composite images (RGB + nDSM), (e) Predicted polygon on composite images (RGB + NIR + nDSM)

Polygon	a	b	c	d	e
Data set	reference	*RGB*	*nDSM*	*RGB + nDSM*	*RGB + NIR + nDSM*
PoLiS		5.32	0.81	0.47	0.39
Vertices	74	612	44	112	111

Table 6. Example polygon produced with 1 pixel for the tolerance parameter of the polygonization method. The columns a, b, c, d,e correspond to the polygons (a), (b), (c), (d),(e) in Figure 15.

Figure 16 shows that the predicted polygons obtained on composited images (RGB+NIR+nDSM) with different losses. Compared with the reference polygons, the polygons obtained with Tversky loss function are much bigger, which means the non-building area close to the building is also be recognized as a building. Compared with polygons obtained with BCE and Dice loss, some buildings are connected to each other, which means it is hard to separate buildings close to each other with Tversky loss. The same problems also exist in the results with different losses obtained on composited images (RGB+ nDSM). It could be deduced that the combination of BCE and Dice loss help produce polygons that are more aligned with ground truth.

(a) (b) (c)

Figure. 16. Results obtained on the urban area test dataset (RGB+NIR+nDSM). The predicted polygons are produced with 1 pixel for the tolerance parameter of the polygonization method. (a) Reference building footprints, (b) Predicted polygons with cross-entropy and Dice as loss function, (c) Predicted polygons with Tversky as loss function.

5.3. Vertices number analysis

The last phase of the polygonization method is a simplification that produces more generalized polygons. Tolerance of simplification is an important parameter to balance the complexity and fidelity of polygons. We perform an analysis of the number of vertices per polygon by changing the tolerance value. We first filter the polygons with IoU ≥ 0.5 to find the predicted polygons and the corresponding reference polygons. Besides the RMSE, we introduced the *average ratio of vertices number* and *average difference of vertices number* to analyze the similarity of the vertices numbers. For the ratio, the best value is 1, which means the average vertices number is the same as its reference. The closer the ratio is to 1, the higher the similarity of the number of vertices. The best value is zero for the difference, which means the average vertices number of the predicted polygons is the same as their reference. The closer the difference value is to zero, the higher similarity of the vertices numbers. The negative difference value means the average vertices number of predicted polygons is smaller than that of their reference.

Table 7 shows vertices number analysis of results obtained on nDSM with BCE and Dice as the loss. Even though tolerance 1 has the smallest RMSE and PoLiS, the average vertices ratio and difference value are the biggest. The PoLiS distance represents polygon dissimilarity, and a smaller distance means a higher similarity. The polygons obtained with tolerance 1 have the most similar shape as their reference but contain more vertices than the reference. The ratios of tolerance 3,5,7,9 are close to each other, but tolerance 3 results in the difference most close to zero. Even though tolerance 9 has the smallest ratio, it results in the largest PoLiS value. The distance increases as the tolerance increases and tolerance 3 results in a second smallest PoLiS value. We may deduce that tolerance 3 is the best-generalized polygons that contain a similar number of vertices as the reference without losing too much positional and shape accuracy.

Tolerance	PoLiS	RMSE	Average ratio of vertices number	Average difference of vertices number
1	0.615	63.836	1.634	6.176
3	0.651	64.275	1.112	-1.768
5	0.670	64.849	1.039	-3.014
7	0.678	65.246	1.018	-3.422
9	0.684	65.501	1.010	-3.578

Table 7. Polygon obtained with different tolerance using the composite images (nDSM) for urban area dataset.

Table 8 shows vertices number analysis of results obtained on composited images (RGB + nDSM) with BCE and Dice as the loss. Even though the tolerance one has the smallest PoLiS, but the ratio is the biggest among all the results. The PoLiS distance represents polygon dissimilarity, and a smaller distance means a higher similarity. The polygons obtained with tolerance 1 have the most similar shape as their reference but contain more vertices than the reference. The ratios of tolerance 3,5,7,9 are close to each other, but tolerance 3 results in the difference most close to zero and the ratio most closest to one, demonstrating the number of vertices is most close to their reference. The distance increases as the tolerance increases and tolerance 3 results in a second smallest PoLiS value. Therefore, we may deduce that tolerance 3 is the generalized polygons that contain a similar number of vertices as the reference without losing too much positional and shape accuracy.

Tolerance	PoLiS	RMSE	Average ratio of vertices number	Average difference of vertices number
1	0.536	80.40	1.621	5.327
3	0.567	81.44	1.026	-3.588
5	0.588	83.07	0.935	-5.236
7	0.611	71.50	0.899	-5.426
9	0.636	72.96	0.872	-6.138

Table 8. Polygon obtained with different tolerance using the composite images (RGB + nDSM) for urban area dataset

Table 9 shows vertices number analysis of results obtained on composite images (RGB + NIR+nDSM) with BCE and Dice as the loss. Even though tolerance 1 has the smallest PoLiS value, the ratio is the biggest among all the results. The PoLiS distance represents polygon dissimilarity, and a smaller distance means higher similarity. The polygons obtained with tolerance 1 have the most similar shape as their reference but contain more vertices than the reference. Tolerance 3 results in a ratio of 0.969, which is most close to 1. It also results in the second most close to zero difference, which also proves that. The distance increases as the tolerance increases and tolerance 3 results in a second smallest PoLiS value. Therefore, we may deduce that tolerance 3 is the generalized polygons that contain similar vertex as the reference without losing too much positional and shape accuracy.

Tolerance	PoLiS	RMSE	Average ratio of vertices number	Average difference of vertices number
1	0.522	80.398	1.484	3.201
3	0.550	80.688	0.969	-4.427
5	0.571	82.563	0.902	-5.722
7	0.593	85.129	0.876	-6.513
9	0.618	87.477	0.859	-7.161

Table 9. Polygon obtained with different tolerance using the composite images (RGB +NIR+ nDSM) for urban area dataset

Tables 7, 8 and 9 show that as the tolerance increases, the ratio decrease and the PoLiS increase, demonstrating that while the polygons are simplified, the similarity between the predicted polygons and reference polygons decreases. To be specific, the positional accuracy and shape similarity decrease. The results obtained on two composite images shows tolerance 3 results in a ratio closest to one, which means the vertices number is most close to their reference. The vertices difference of polygons predicted with tolerance 3 also proved that. It usually has the closest or second most close to zero vertices difference.

Therefore, we deduced that 3 is an appropriate tolerance to obtain the best-generalized polygons without losing too much similarity for our dataset.

Figure 17 compares the predicted polygon with different tolerance levels. For sample building, the increase of tolerance results in the decrease of the number of vertices. Table 10 shows the PoLiS value increases as the tolerance increase, which means the dissimilarity of the predicted polygon and reference polygon increases. Compared to the polygon predicted with tolerance 1, changes happen to the shape of the polygons with bigger tolerance, such as the edge in the upper part of the polygons deviates from the ground truth.

Figure. 17. Example polygon obtained with different tolerance values using the composite images (RGB + nDSM): (a) Reference polygon, (b) Predicted polygon with tolerance 1 pixel, (c) Predicted polygon with tolerance 3 pixel, (d) Predicted polygon with tolerance 5 pixel, (e) Predicted polygon with tolerance 7 pixel, (f) and Predicted polygon with tolerance 9 pixel.

Polygon	a	b	c	d	e	f
Tolerance		1	3	5	7	9
PoLiS		0.472	0.537	0.628	0.697	0.763
Vertices	74	112	52	35	29	26
Ratio		1.514	0.703	0.473	0.392	0.351

Table 10. Example polygon with different tolerance and number of vertices. The columns a, b, c, d, e, f corresponding to the polygons (a), (b), (c), (d), (e), (f) in Figure 17.

For the applications that require high accuracy of the polygons, the predicted polygons still need post-processing to check the quality and increase accuracy. The predicted polygons with fewer vertices and comparable accuracy will facilitate this process and reduce the manual work. For the predicted polygons with 1 pixel tolerance, even though it has high positional accuracy, it contains the biggest number of vertices. Furthermore, most vertices are so close to each other that some of them are superfluous. If manually simplified the polygon, compared to the reference, a lot of vertices need to be removed, which increased the processing time and waste of human labor. For polygon predicted with bigger tolerance, outlines have a similar shape but fewer vertices. Thus, fewer editing operations need to be performed to move or delete vertices, which may simplify the post-processing procedure.

5.4. Comparison with an alternative method

We compared the frame field learning-based method to an end-to-end polygon delineation method PolyMapper. The experiments are performed on the original aerial images (RGB). The default setting of the PolyMapper method is adopted with the max iteration is 1600000 and the backbone is ResNet-101. Table 11 shows a quantitative comparison of two methods reported in COCO metrics. Frame field learning-based method achieves 6.7% mAP and 25.3% mAR, which outperforms PolyMapper in mAP and mAR metrics. It demonstrates that a higher proportion of buildings extracted by the frame field approach. In addition, the method works significantly better in delineating medium and large buildings and achieves higher precision at all scale levels.

Method	mAP	mAR	AP_S	AR_S	AP_M	AR_M	AP_L	AR_L
PolyMapper	0.009	0.017	0.001	0.001	0.004	0.028	0.014	0.065
Frame field	0.067	0.253	0.139	0.024	0.285	0.261	0.248	0.232

Table 11. Extraction results using aerial images (RGB) for urban area dataset. The metrics are calculated on the polygons with 1 pixel tolerance for polygonization for the Frame field learning-based method.

Figure 18 shows the results obtained on two tiles by different methods. PolyMapper only extracts part of the big building, and it cannot delineate the hole inside the building. The results obtained for the dense urban area by the PolyMapper cannot differentiate individual buildings from the surrounding road and trees. The method missed a lot of the real buildings on the ground. The low AR in Table 11 also proves that. The results predicted by the frame field learning method are much better, with more buildings be corrected extracted and more regular and aligned predicted polygons. But many false positives exist in the results obtained by the frame field method compared with the reference data. Some individual buildings in the densely urban areas are also connected, demonstrating it cannot differentiate buildings that close to each other only with the spectral information.

(a) (b) (c)

Figure. 18. Results obtained using aerial images (RGB) for the urban area dataset. From left to right: (a) Reference building footprints, (b) Predicted polygons with 1 pixel for the tolerance parameter of the polygonization method by frame field learning method, (c) Predicted polygons by PolyMapper.

6. CONCLUSION

In this thesis, we explored a building delineation method based on frame field learning. The overall framework of our method is based on an FCN architecture, which serves as an extractor of image segmentation and direction information of the building contour, followed by a polygonization method which takes the outputs of the model as inputs to generate the polygons. By learning the frame field, segmentation performance is increased. With the direction information stored in the frame field, the edges could be iteratively adjusted to be more aligned with ground truth in ACM, and corners can also be detected and further preserved during the simplification. Hence, it can produce more regular buildings and reduce the missing corners. The comparison with PolyMapper also indicates that, since PolyMapper missed a lot of corners and cannot delineate the buildings correctly. These regular building outlines in polygons can be directly used in most operational GIS applications. In contrast, buildings masks in a raster format still require complex and expensive post-processing to obtain building outlines in polygons. For public institutions like Kadaster, which are responsible for maintaining national cartography, building information needs to be updated regularly. Our approach can help these institutions to generate building footprints more effectively. Furthermore, the building polygons can also be used in other products like cadastral maps or building models.

The original method only takes aerial images (RGB) as input. We introduced the 3D information into the framework to overcome the limitations of the aerial image. To better evaluate the quality of buildings' polygons, we followed the same standards of COCO metrics and applied them directly to the output polygons. Moreover, the PoLiS distance metric was introduced to evaluate positional accuracy and shape differences between the predicted polygons and their reference ones. We also performed an analysis of the number of vertices and introduced the *average ratio of vertices number* and the *average difference of vertices number* to evaluate the agreement between the predicted polygon and their reference. By analyzing these two statistical metrics in combination with PoLiS distance for the polygons produced with different tolerances, we found out the best parameters for our model to produce simpler polygons while keeping high accuracy. The main advantage of producing simpler polygons is the need for fewer editing operations for adjusting the polygons in operational applications. They can serve as a reference for polygon generalization and guidance to reduce the workload of post-processing tasks for obtaining operational maps of building footprints.

Our method combined the original framework with data fusion by extending the model to take two different composite images (RGB+ nDSM and RGB+NIR+nDSM) as input. Compared with the results obtained on the two baselines, the polygons obtained on composite images were largely improved considering both quantitative and qualitative criteria. The method benefited from the additional nDSM and the height information contributes more than spectral information in building extraction. The 3D information provided by the nDSM overcame the aerial images' limitations and contributed to distinguish the buildings from the background more accurately. The nDSM also improved the accuracy of the building outlines, resulting in better-aligned building polygons and preventing false positives. A qualitative analysis of the results shows that our method can predict precise and regular polygons for large and complex structures.

6.1. Answer to research questions

For the three specific objectives of this study, research questions related to them are listed before. This section will present the answers to these research questions.

Research objective 1

a. How is the quality of the available reference polygon data?

We checked the data quality and found out that some reference polygons are different from the ground truth showing in the aerial images and manually edited them. We also found out that the reference polygons are not perfect. Some polygons are not aligned with the ground truth, and some do not exist on the ground. We managed to edit some of them, but there was not enough time to check every single polygon and make sure they are all precisely aligned with the ground truth showing in the aerial images. The results show that our method is invariant to this problem since it performed better than the reference data in some regions.

b. Are there any systematic or random shifts in building polygons from the corresponding boundaries?

There are no systematic shifts or random shifts. To be specific, the polygons do not shift as a whole from the ground truth, but some edges of polygons may not be totally aligned with the building boundaries.

Research objective 2

a. What relevant deep learning-based models exist, and what are their disadvantages and advantages?

The most relevant deep learning-based models are Mask R-CNN and PolyMapper. The advantage of Mask R-CNN is that it detects the object by generating the bounding boxes of the individual objects and produces segmentation masks for the objects precisely. One disadvantage is the predicted results of Mask R-CNN is a binary mask in a raster format that could not be used in multiple GIS applications directly, needing complex polygonization post-processing steps. Moreover, convert the raster format to vector are expensive and complicated and may also introduce errors in the conversion. Another disadvantage is that the details of buildings are lost when small feature maps are up-sampled to the same size as the input. Hence, when compared with the FCN, the boundaries of Mask R-CNN output are over-smooth and less accurate. The main advantage of PolyMapper is that it is an end-to-end network that takes the aerial image as input and directly outputs polygons. The disadvantage of PolyMapper is that it is hard to train and cannot properly delineate the buildings with holes. We found out that the PolyMapper performed worse than the frame field learning method considering both quantitative and qualitative criteria with the same dataset for our study area.

b. When the resolution of nDSM is different from imagery, how can we perform the data fusion? For example, to fuse the data directly or adapt the network to take multi-resolution data as input?

We resampled the nDSM to the same resolution as the aerial image and created two composite images. The composite image (RGB + nDSM) is produced by stacking the nDSM with the original aerial image (RGB) as the 4th band. The composite image (RGB+NIR+nDSM) is produced by stacking the NIR as the 4th band and nDSM as the 5th band with the original aerial image. Then the composite image was feed into the network as a whole. The frame field learning network was adapted to take these fused datasets as inputs.

Research objective 3

a. What are the advantages and disadvantages of the proposed model?

One advantage is the 3D information provided by the nDSM overcomes the limitations of the optical imagery and largely improved the accuracy of the predicted polygons. Another advantage is that, unlike the state-of-the-art method PolyMapper, this model has a light-weighted structure and is easier to train. Moreover, it could predict buildings with more complex structures and with holes.
Multiple disadvantages still need to be improved. The first disadvantage is that it performs worse for small buildings compared with medium and large buildings. There are more false positives for small buildings, especially in dense urban areas. The second disadvantage is that the regularity of the predicted polygons still needs to be improved. The predicted polygons usually contain more curves than their reference. The third disadvantage is that the method does not perform well for the building with an arc structure.

b. Does 3D information help to improve the results? Is the improvement significant?

Yes, the 3D information improved the results considering both quantitative and qualitative criteria. The height information largely improves the accuracy of the results. It also reduces the false positives and prevents missing the buildings on the ground. Furthermore, it improves the similarity between the predicted polygons and the corresponding reference polygons. The improvement was significant. The mean IoU achieved on the composite image (RGB + nDSM) test set was 80%, against 57% achieved for the test set of RGB image. The addition of the nDSM led to an improvement of 23% on the mean IoU. The mAP and mAR achieved on the composite images (RGB + nDSM) are 41% and 48.8% against 6.7% and 25.3% achieved on aerial images only. The addition of the nDSM led to 34.3% and 23.5% improvement on average precision and average recall, respectively. PoLiS distance achieved on the composite image (RGB + nDSM) is 0.54, considerably smaller than 0.87 for the RGB image showing nDSM improves the similarity. The improvement could also be seen in the qualitative analysis, which shows that the nDSM reduced the false positives and produced better-aligned polygons.

6.2. Suggestions for future works

For future work, we plan to improve the framework in the following directions:

- Use different branches to takes different data as input (multi-modal network), where one branch will take the nDSM as input, and another branch will take the aerial image (RGB) as input. Thus, it will perform the fusion in the feature level instead of in the image level, fusing the features from two data types in the network. Different fusion strategies could be applied, such as middle fusion or late fusion.

- Simplify the current model or design a new model. The model can reach high accuracy only with the height information of nDSM. The nDSM contains more straightforward and less complicated information than aerial images. We hypothesize that a lightweight model could extract buildings with comparable accuracy. Hence, simplify the current model or design a new model with a shallower structure could be a possible direction.

- Fine-tuning the method. The method is comprised of two parts: an FCN and a polygonization algorithm. The training strategy for the FCN could be refined. For polygonization, the energy function is critical for the ACM to optimizing the building contour. The default coefficients for energy items in the energy function are adopted. As our dataset is different from the open public datasets used by the original method, tuning these coefficients may result in better results.

- Test the generalization and transferability of the model in other cities.

LIST OF REFERENCES

Acuna, D., Ling, H., Kar, A., & Fidler, S. (2018). Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 859–868. https://doi.org/10.1109/CVPR.2018.00096

Audebert, N., Le Saux, B., & Lefèvre, S. (2018). Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. ISPRS Journal of Photogrammetry and Remote Sensing, 140, 20–32. https://doi.org/10.1016/j.isprsjprs.2017.11.011

Avbelj, J., Muller, R., & Bamler, R. (2015). A metric for polygon comparison and building extraction evaluation. IEEE Geoscience and Remote Sensing Letters, 12(1), 170–174. https://doi.org/10.1109/LGRS.2014.2330695

Awrangjeb, M., & Fraser, C. (2014). Automatic Segmentation of Raw LIDAR Data for Extraction of Building Roofs. Remote Sensing, 6(5), 3716–3751. https://doi.org/10.3390/rs6053716

Bittner, K., Adam, F., Cui, S., Körner, M., & Reinartz, P. (2018). Building Footprint Extraction From VHR Remote Sensing Images Combined With Normalized DSMs Using Fused Fully Convolutional Networks. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(8), 2615–2629. https://doi.org/10.1109/JSTARS.2018.2849363

Castrejón, L., Kundu, K., Urtasun, R., & Fidler, S. (n.d.). Annotating Object Instances with a Polygon-RNN. Retrieved from http://www.cs.toronto.edu/

Diamanti, O., Vaxman, A., Panozzo, D., & Sorkine-Hornung, O. (2014). Designing N-polyvector fields with complex polynomials. Eurographics Symposium on Geometry Processing, 33(5), 1–11. https://doi.org/10.1111/cgf.12426

Girard, N., Smirnov, D., Solomon, J., & Tarabalka, Y. (2020). Polygonal Building Segmentation by Frame Field Learning. Retrieved from http://arxiv.org/abs/2004.14875

Griffiths, D., & Boehm, J. (2019). Improving public data for building segmentation from Convolutional Neural Networks (CNNs) for fused airborne lidar and image data using active contours. ISPRS Journal of Photogrammetry and Remote Sensing, 154(May), 70–83. https://doi.org/10.1016/j.isprsjprs.2019.05.013

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. Proceedings of the IEEE International Conference on Computer Vision, 2961–2969.

Hong, D., Gao, L., Yokoya, N., Yao, J., Chanussot, J., Du, Q., & Zhang, B. (2020). More Diverse Means Better: Multimodal Deep Learning Meets Remote Sensing Imagery Classification. IEEE Transactions on Geoscience and Remote Sensing. https://doi.org/10.1109/TGRS.2020.3016820

Huang, J., Zhang, X., Xin, Q., Sun, Y., & Zhang, P. (2019). Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network. ISPRS Journal of Photogrammetry and Remote Sensing, 151(February), 91–105. https://doi.org/10.1016/j.isprsjprs.2019.02.019

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

Li, Z., Wegner, J. Di., & Lucchi, A. (2019). Topological map extraction from overhead images. Proceedings of the IEEE International Conference on Computer Vision, 2019-Octob, 1715–1724. https://doi.org/10.1109/ICCV.2019.00180

Liu, P., Liu, X., Liu, M., Shi, Q., Yang, J., Xu, X., & Zhang, Y. (2019). Building footprint extraction from high-resolution images via spatial residual inception convolutional neural network. Remote Sensing, 11(7), 830. https://doi.org/10.3390/rs11070830

Lorensen, W. E., & Cline, H. E. (1987). MARCHING CUBES: A HIGH RESOLUTION 3D SURFACE CONSTRUCTION ALGORITHM. Computer Graphics (ACM), 21(4), 163–169. https://doi.org/10.1145/37402.37422

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017, July). Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 3226-3229). IEEE.

Mayunga, S. D., Zhang, Y., & Coleman, D. J. (2005). Semi-automatic building extraction utilizing Quickbird imagery. Proc. ISPRS Workshop CMRT, Vol. 13, pp. 1–136.

Nahhas, F. H., Shafri, H. Z. M., Sameen, M. I., Pradhan, B., & Mansor, S. (2018). Deep Learning Approach for Building Detection Using LiDAR-Orthophoto Fusion. Journal of Sensors, 2018.

https://doi.org/10.1155/2018/7212307
Nex, F., Rupnik, E., & Remondino, F. (2013). Building Footprints Extraction from Oblique Imagery. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2(3W3), 61–66. https://doi.org/10.5194/isprsannals-II-3-W3-61-2013
Nguyen, T. H., Daniel, S., Gueriot, D., Sintes, C., & Caillec, J.-M. Le. (2020). Super-Resolution-based Snake Model -- An Unsupervised Method for Large-Scale Building Extraction using Airborne LiDAR Data and Optical Image. Undefined. Retrieved from http://arxiv.org/abs/2004.08522
Rizaldy, A, Persello, C., Gevaert, C. M., & Oude Elberink, S. J. (2018). FULLY CONVOLUTIONAL NETWORKS FOR GROUND CLASSIFICATION FROM LIDAR POINT CLOUDS. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4(2). https://doi.org/10.5194/isprs-annals-IV-2-231-2018
Rizaldy, Aldino, Persello, C., Gevaert, C., Oude Elberink, S., & Vosselman, G. (2018). Ground and Multi-Class Classification of Airborne Laser Scanner Point Clouds Using Fully Convolutional Networks. Remote Sensing, 10(11), 1723. https://doi.org/10.3390/rs10111723
Schuegraf, P., & Bittner, K. (2019). Automatic building footprint extraction from multi-resolution remote sensing images using a hybrid FCN. ISPRS International Journal of Geo-Information, 8(4), 191. https://doi.org/10.3390/ijgi8040191
Shao, Z., Tang, P., Wang, Z., Saleem, N., Yam, S., & Sommai, C. (2020). BRRNet: A fully convolutional neural network for automatic building extraction from high-resolution remote sensing images. Remote Sensing, 12(6), 1050. https://doi.org/10.3390/rs12061050
Shelhamer, E., Long, J., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440. https://doi.org/10.1109/TPAMI.2016.2572683
Sohn, G., & Dowman, I. (2007). Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction. ISPRS Journal of Photogrammetry and Remote Sensing, 62(1), 43–63. https://doi.org/10.1016/j.isprsjprs.2007.01.001
United Nations. (2019). World Urbanization Prospects: The 2018 Revision. In World Urbanization Prospects: The 2018 Revision. https://doi.org/10.18356/b9e995fe-en
Vaxman, A., Campen, M., Diamanti, O., Panozzo, D., Bommes, D., Hildebrandt, K., & Ben-Chen, M. (2016). Directional field synthesis, design, and processing. Computer Graphics Forum, 35(2), 545–572. https://doi.org/10.1111/cgf.12864
Wang, L., Yan, J., Mu, L., & Huang, L. (2020). Knowledge discovery from remote sensing images: A review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. https://doi.org/10.1002/widm.1371
Wei, S., Ji, S., & Lu, M. (2020). Toward Automatic Building Footprint Delineation from Aerial Images Using CNN and Regularization. IEEE Transactions on Geoscience and Remote Sensing, 58(3), 2178–2189. https://doi.org/10.1109/TGRS.2019.2954461
Zhang, K., Yan, J., & Chen, S. C. (2006). Automatic construction of building footprints from airborne LIDAR data. IEEE Transactions on Geoscience and Remote Sensing, 44(9), 2523–2533. https://doi.org/10.1109/TGRS.2006.874137
Zhang, L., Wu, J., Fan, Y., Gao, H., & Shao, Y. (2020). An efficient building extraction method from high spatial resolution remote sensing images based on improved mask R-CNN. Sensors (Switzerland), 20(5), 1465. https://doi.org/10.3390/s20051465
Zhao, W., Persello, C., & Stein, A. (2021). Building outline delineation: From aerial images to polygons with an improved end-to-end learning framework. ISPRS Journal of Photogrammetry and Remote Sensing, 175, 119–131. https://doi.org/10.1016/j.isprsjprs.2021.02.014
Zhao, Z., Duan, Y., Zhang, Y., & Cao, R. (2016). Extracting buildings from and regularizing boundaries in airborne lidar data using connected operators. International Journal of Remote Sensing, 37(4), 889–912. https://doi.org/10.1080/01431161.2015.1137647
Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, 30(11), 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865