AUTOMATIC BUILDING ROOF PLANE STRUCTURE EXTRACTION FROM REMOTE SENSING DATA FOR LOD2 3D CITY MODELLING

CARLOS CAMPOVERDE
July, 2023

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

ADVISORS:
Ph. D. Candidate Konstantin Maslov
Ph. D. Candidate Weiqin Jiao

AUTOMATIC BUILDING ROOF PLANE STRUCTURE EXTRACTION FROM REMOTE SENSING DATA FOR LOD2 3D CITY MODELLING

CARLOS CAMPOVERDE
Enschede, The Netherlands, July, 2023

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: GEO-INFORMATION FOR LAND MANAGEMENT

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

ADVISORS:
Ph. D. Candidate Konstantin Maslov
Ph. D. Candidate Weiqin Jiao

THESIS ASSESSMENT BOARD:
Dr. Monika Kuffer (Chair)
Prof. Dessislava Petrova Antova(External Examiner)

DISCLAIMER

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

ABSTRACT

Roofs are fundamental parts of buildings, and mapping their structure represents an active and emerging research area in urban-related studies. Knowing roof characteristics can lead to more accurate and detailed 3D building models. Creating detailed 3D Building models involves more than just the basic shape of the building; it also includes the structural details of the roof. 3D Building models can derivate into highly detailed 3D city models, opening up a world of applications, such as enhanced urban planning, solar potential estimation, telecommunications planning, transportation, and creating digital twins for urban planning.

The rapid advancements in remote sensing technologies have allowed a variety of new datasets for geospatial analysis. In the same way, machine learning has experimented with similar growth, mainly deep learning. The potential for accurately and efficiently deriving object features from images is increasingly promising. While numerous deep learning approaches for extracting roof structures have been proposed, challenges persist. These include the regularization of output, false detections and misclassifications, and low computational efficiency. These ongoing issues highlight the need for continued research and innovation.

This study explores a cutting-edge deep learning method for planar graph reconstruction applied to building roof structure extraction. A framework has been designed to delineate and extract regularized building roof plane structures, using aerial imagery and the building footprint information across an entire area. The framework is built on top of the work developed by Chen et al. (2022) Holistic Edge Attention Transformer (HEAT), which harnesses the power of an attention-based neural network deployed to detect corners and classify interconnecting edges for planar graph reconstruction in RGB images. In addition, the generated roof plane structures have been tested for their applicability in generating a 3D city model by integrating information from the Digital Surface Model (DSM), Digital Terrain Model (DTM), and Normalized Digital Surface Model (nDSM). The process was performed by using 3D tools from Geographic Information Systems (GIS). The performance of the approach was evaluated using extensive quantitative metrics and qualitative visual analysis.

The research is founded on experiments conducted in three distinct geographical locations with different topologies of roof structures: the Stadsveld – 't Zwering neighborhood, and the Oude Markt area, in Enschede, The Netherlands, and the Lozenets neighborhood in Sofia, Bulgaria. The approach started by using the pre-trained HEAT model for outdoor architecture reconstruction to harness the model's learned planar graph reconstruction knowledge. Subsequently, the model underwent training on datasets created from the Stadsveld – 't Zwering neighborhood, Lozenets neighborhood, and a combination of both datasets.

The results show that the models tailored to specific study areas delineate building inner roof plane structures with the same performance as the model trained on a combined dataset. However, when the models were tested in the Oude Markt, the model trained with a combined dataset demonstrated superior performance with an F score value of 0.43 for building inner roof plane delineation against the F score value of 0.37 from the model trained only on the Stadsveld – 't Zwering neighborhood dataset, and 0.32 from the model trained only on the Lozenets dataset. The obtained building inner roof planes show substantial potential for urban applications and continuous improvement. Through this study, we explored new pathways for improving computational efficiency in detecting and extracting roof plane structures, thus contributing to advancing urban-related studies and a step forward in automated frameworks for digital twin cities.

Keywords: Image processing, Image analysis, deep learning, planar graph reconstruction, roof structure extraction, roof vectorization, Holistic Edge Attention Transformer, Python, GIS, 3D city modelling.

ACKNOWLEDGMENTS

I want to express my most profound appreciation to ITC, University of Twente, for trusting me and awarding me with a Scholarship, which allowed me to have this two-year experience full of knowledge, personal growth, and the best time of my life. The knowledge and relationships I have gained during this time are invaluable and will always be treasured.

I am deeply grateful to my supervisors, Prof. Dr. Claudio Persello (Grazie mille!) and Dr. Mila Koeva (Много благодаря!), for their invaluable guidance and encouragement throughout my thesis research. I also sincerely appreciate Ph.D. candidates Enzo Campomanes (Maraming Salamat!), Konstantin Maslov (Большое спасибо!), and Weiqin Jiao (非常感谢！) for their insightful teachings and consistent support. The opportunity to work under this extraordinary team has significantly enriched my academic journey.

Mi más profunda gratitud va para mi madre, cuya sabiduría y enseñanzas siempre han destacado el valor de la responsabilidad y la educación como caminos para trascender nuestras circunstancias. Extiendo un agradecimiento sincero a toda mi familia, cuyo apoyo financiero ha sido instrumental a lo largo de mi trayectoria académica. Su creencia en mis habilidades, destrezas y sacrificios me han traído hasta donde estoy ahora. Mis amigos en los Países Bajos y en Ecuador que siempre me alientan a seguir adelante y a que ningún desafío es demasiado grande para mí.

A special thanks to my closest ITC friends, Arturo Cauba and Dennis Ushiña. Their unwavering support has been invaluable, and our bond has grown so strong that I now consider them brothers. I extend my heartfelt gratitude to my other friends, Sam Holtzhuizer, Santiago Hidalgo, and Daksh Singh whose, with whom I shared unforgettable times and parties in The Netherlands. I also extend my thanks to Cham, Aulia, Archita, Arivia, Roy, Ahmed, Carolina, Jessica, and Finn. Each one of you has made a unique and significant impact on my journey. If I have forgotten to mention anyone, please accept my sincere apologies. Know that each one of you is a part of the cherished memories that will stay with me for a lifetime.

Finally, my appreciation extends to the whole ITC community at the University of Twente. The friendliness and support they have consistently demonstrated, regardless of your position or area of specialization, have created an inclusive and educational environment conducive to learning and growth. I am honored to be part of such a welcoming and supportive community.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background and justification

The rapid development of urban landscapes and the lack of available land in urban areas have promoted the development of infrastructures above and below the ground surface (Shojaei et al., 2018). With these developments, new challenges arise, such as 2D models and 2D cadastral registration systems are non-suitable for capturing the relationship between people and property (Van Oosterom, 2013). In this sense, 3D city models propose a solution to capture and record all these new vertical developments as 3D Geographic Information Systems (GIS) proposes new techniques for 3D geometric and semantic modelling of urban areas (Hajji et al., 2021). Moreover, 3D city models are helpful for various applications related to disaster management, energy, real-state, urban development, and tourism and as the first step toward Digital Twin Cities (Dimitrov & Petrova-Antonova, 2021).

3D city models provide a platform for performing analytical and simulated analyses considering scenarios revealing emergent patterns and behaviors (Peters et al., 2022). However, the intricacies associated with urban features pose considerable challenges for building end-to-end frameworks that span from building data extraction to the final stage of 3D model reconstruction (Soilán et al., 2018). The delineation of building roofs within the urban landscape is emerging as a focal point of contemporary research. (Sun, 2021). One of the critical steps in 3D building modelling is roof segmentation, which involves dividing the individual roof planes of buildings. This step is crucial as it forms the basis for the subsequent reconstruction of the 3D building models at a higher level of detail (J. Huang et al., 2022).

The research on automated extraction of roof plane structures in vector format remains relatively limited (Zhao et al., 2022). While manual extraction of roof structures is feasible, it is not practical for large-scale projects due to the significant involvement of time and cost (Ok, 2013). In this context, machine learning and deep learning methods have emerged as promising solutions to tackle this challenge (Ma et al., 2019). These methods enable the automation of object delineation, encompassing different features (e.g., buildings, roads, roofs, parcels) while facilitating efficient and accurate feature extraction (Qin et al., 2019). However, a key obstacle in computer vision research involves mimicking human-level perception to reconstruct comprehensive geometric structures from images, particularly in areas of complex topology. This complexity manifests in the conventional problem of pixel-grouped building extraction and the extraction of these structures (F. Zhang et al., 2020). Attempts to address this topic yielded results predominantly in raster format. Several techniques to extract building footprints in vector format have been proposed in the past few years. For instance, The Frame Field Learning method developed by Girard et al. (2020), and the end-to-end learning framework based on PolyMapper developed by Zhao et al.(2021), among others.

Recent research efforts are oriented to obtain more regularized (aiming to produce straighter edges in the output) vector format results (Zhao et al., 2022). By exploring and developing end-to-end frameworks to generate accurate planar graph reconstruction of buildings (Ye et al., 2019), considering the diversity of input remote sensing datasets and the complexity of different study areas. However, despite these advances, this problem remains considerably challenging (Zhao et al., 2021). Getting the capacity to perform holistic structure reasoning, such as graph reconstruction derivate from corners and edges, remains a challenging task for end-to-end neural networks (Chen et al., 2022).

A recurring challenge in the field is obtaining, processing, and preparing suitable datasets for developing automated feature extraction methods. Furthermore, the complexity of the processing areas can be derived into inaccuracies in the extracted features, such as occlusions, and imprecise borders, among other issues (Golnia, 2021). These challenges have been acknowledged in recent studies developed by Girard et al.(2020) using convolutional neural networks and the research on contour-based methods carried out by Marcos et al.(2018)

Understanding the configuration of building roof plane structures is essential for constructing accurate and detailed 3D models. Before starting the process of 3D model reconstruction, it is essential to define the desired level of detail (LOD) to be reached. (Kolbe et al., 2005) define the concept of "Levels of Detail" (LOD) as a hierarchical division of the geometric and semantic representation of objects in a 3D city model. As defined by Kolbe in the City Geography Markup Language (CityGML) standards, LODs have different levels going from 0 to 4, as explained in Table 1. Figure 1 illustrates the classification of LOD from level 1 to level 4. LOD2 3D building models incorporate the characteristics of building rooftops, such as pitch and geometry configuration, yielding a more comprehensive and appealing representation (Lee et al., 2009) than their counterpart LOD 1.

Table 1. Levels of Detail division proposed by Kolbe et al. (2005).

Levels of Detail (LOD)	**Description**
LOD 0	Representing the terrain without any digital building block
LOD 1	A digital block model of a building without any architectural details
LOD 2	A digital block building with a standard roof structure
LOD 3	A digital block building with a detailed roof structure
LOD 4	A digital block building with an interior structure

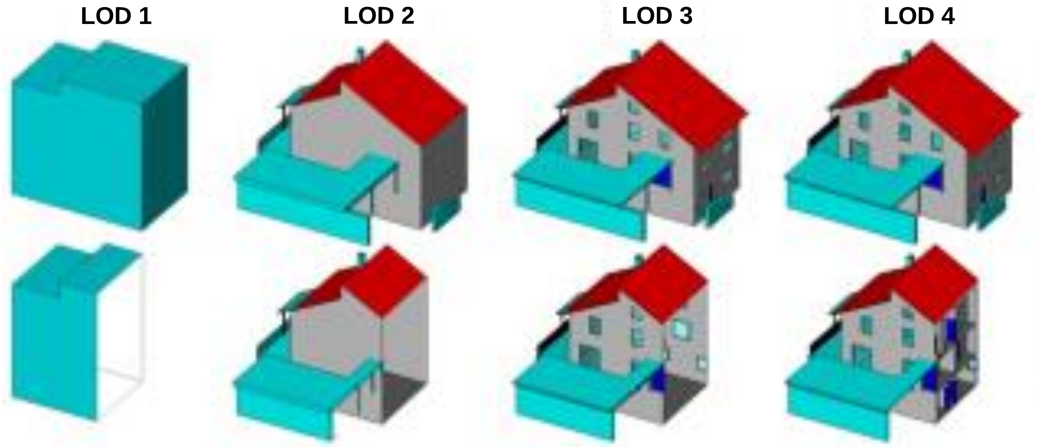


Figure 1. The four levels of detail defined by CityGML(Biljecki, 2013).

The new varied amount of remotely sensed datasets has allowed the development of new approaches for 3D construction. Nevertheless, there are challenges related to the quality of the available dataset, as not all remote sensing datasets are suitable for 3D model construction (Gui & Qin, 2021). For example, using satellite imagery with finer spatial resolution challenges the detection and extraction of different details for building 3D city models, particularly in areas with densely packed buildings, as the complexity of the building structures, shadows, and occlusion that can obscure part of the scene (P. Liu et al., 2019).

Nevertheless, reconstructing 3D models (LOD2) with diverse and complex roofs remains challenging. Recent research developed by Peters et al. (2022) shows the feasibility of developing automated 3D model construction by combining building footprint information with light detection and ranging (LIDAR) point cloud data to create

3D models subsequently upgraded by using the roof plane structure information. However, LIDAR is not universally accessible due to its expensive costs; hardware, software, high skills technicians, data volume, and maintenance (Lopac et al., 2022) make it impossible for everyone to obtain highly detailed 3D models.

Aligned in the path of 3D building models and 3D city models, the concept of digital twin cities has gained significant importance (Deng et al., 2021). Research around this topic is focused on: I) Exploring new ways to extract complex geometries of the urban environment, II) integrating massive point cloud information to complex objects, III) understanding relationships between urban features, IV) creating Digital Twin Cities at a high level of detail and V) exploring effective automated strategies for the creation of Digital Twin Cities that incorporates dynamic objects (F. Xue et al., 2020).

Based on the many challenges of obtaining information about the roof structures from remote sensing datasets. This research explores a state-of-the-art method for planar graph reconstruction applied to building roof structure reconstruction. An innovative framework for reconstructing and extracting regularized building roof plane structures from aerial imagery and the building footprint is proposed. In addition, the obtained outputs are used to generate LOD2 3D city models, thus contributing to advancing urban-related studies and applications.

1.2. Research problem

The key role that 3D building models can play in tackling urban issues underscores the need to develop approaches for automatically delineating and extracting building roof-plane structures to create LOD2 3D city models (Kenzhebay, 2022). While current research has made significant progress employing image segmentation techniques for roof structure extraction, these studies' outputs predominantly yield results in raster format (Sun, 2021). Few research efforts have focused on extracting building roof-plane structures in vector format (Zhao, 2022). Moreover, the lack of transferability in some deep learning methods limits their application to specific areas where they were developed, as changing their target areas could drastically affect model performance (Jiang et al., 2022)—presenting notable gaps in the current field of study.

In order to overcome the mentioned challenges, this research aims to develop a deep learning-based framework to automatically extract building roof plane structures in vector format from aerial images across a complete scene. The proposed framework aims to scale up the application of the work developed by Chen et al. (2022). In addition, this study aims to test the application of the generated buildings' roof planes structure by integrating them with Digital Surface Model (DSM), Digital Terrain Model (DTM), and Normalized Digital Surface Model (nDSM) datasets to generate LOD2 3D city models. Therefore, this research integrates deep learning, GIS, and remote sensing for automated building roof structure extraction for LOD2 3D city model creation.

1.3. Research objectives

1.3.1. General objective

The main objective of the research is to develop a deep learning-based framework to reconstruct and extract building roof plane structures in vector format from high-resolution remote sensing data and use it for building a LOD2 3D city model.

1.3.2. Specific objectives:

The proposed research thesis aims to use a deep learning (DL)-based method to extract building roof plane structures in a regularized vector format from aerial imagery. To achieve this aim, the following specific objectives (SO) and related research questions have been defined:

SO 1: To acquire knowledge in planar graph reconstruction from 2D raster images;
1. What is the process for planar graph reconstruction?
2. How to apply planar graph reconstruction for building roof plane structures extraction?

SO 2: To prepare the dataset for the further deep learning-based approach
1. What dataset and resources are needed to implement the selected approach?
2. What further data processing is needed?

SO 3: To design a deep learning-based framework to extract building roof plane structures in vector format from aerial images
1. How can the selected deep learning approach for planar graph reconstruction be adapted to extract building roof plane structures?
2. How to apply the developed framework in two different selected study areas?

SO 4: To develop a LOD2 3D city model from the obtained roof plane structures vector format dataset.
1. What is the level of detail of the obtained 3D models?
2. What are the further improvements for the obtained 3D model?

SO 5: To assess the performance of the developed approach
1. What are the performance differences between the two study cases?
2. What are the strengths and limitations of the developed approach?

1.4. Conceptual framework

The conceptual framework shown in Figure 2 presents the interrelationships between the main concepts of the present research. The recent availability of different earth observation technologies has unlocked an unprecedented array of diverse datasets creating new opportunities for numerous studies and applications. Nevertheless, the vast amount of data collected makes it necessary for automated technologies to speed up and simplify formerly manual, labor-intensive procedures. In response to this need, computer vision techniques, such as deep learning, are especially well-suited to assess the vast amounts of data from remote sensing technologies since they can understand complex patterns and correlations (Persello et al., 2022).

In the current research, a deep learning-based framework will be built to automatically delineate and extract building roof inner planes in vector format from aerial imagery and building footprints. Moreover, the obtained outputs will be combined with DSM, DTM, and nDSM to generate LOD2 3D models using GIS tools. Deep learning methods applied to remote sensing datasets present a cutting-edge path for data analysis, automated feature extraction, and 3D modelling.

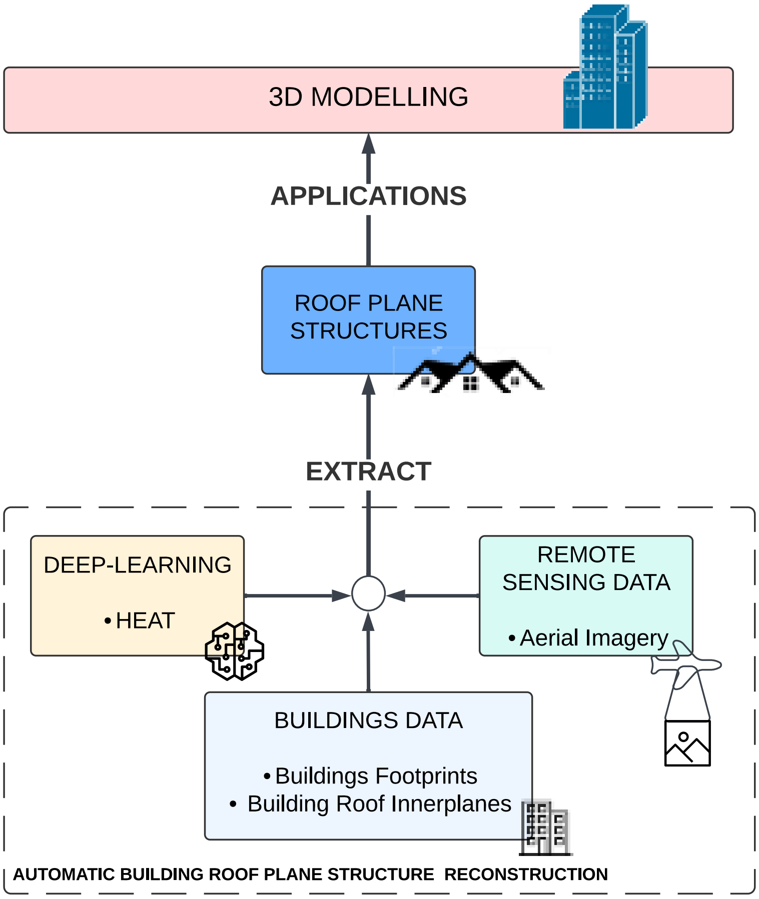


Figure 2. Conceptual framework.

1.5. Thesis structure

The structure of this thesis unfolds as follows:

Chapter 1. Introduction
This chapter provides the contextual background and justification of the study, presenting the research problem alongside the main goal and queries it seeks to address.

Chapter 2. Literature review
This chapter presents a review of pertinent literature related to building roof structure extraction. Different methods are presented in this section.

Chapter 3. Materials and methodology
This chapter presents an overview of the research methodology and materials employed in the study., followed by an explanation of each stage involved, including data preparation, planar graph reconstruction, vectorization, and subsequent post-processing. Further, the delineation of the evaluation metrics is also presented in this part.

Chapter 4. Results and analysis
This chapter presents qualitative and quantitative evaluations conducted in the study, along with an analysis of the obtained outcomes.

Chapter 5. Discussion
This chapter presents an extensive discussion of the obtained results, followed by thoughtful suggestions for further potential improvements on the scope of this research.

Chapter 6. Conclusion
This chapter presents the key findings of the research and answers to the research and provides answers to the research questions. Therefore, summarizing the aim of this research.

1.6. Summary

This chapter gives information on the background of the research, followed by the main problem, general and specific objectives of the study. To summarize, the research aims to employ a deep learning-based method to reconstruct and extract building roof planes in standardized vector format and thus could be used to generate 3D building models.

2. LITERATURE REVIEW

2.1. Data sources

Roof structure reconstruction from remote sensing datasets is an essential task in urban geospatial studies and has become a continuing field of study(Zhao, 2022). This task can be conducted using different remote-sensing datasets. LIDAR point clouds have gained popularity due to their highly dense information, which makes it possible to segment roof structures and obtain the roof planes with relatively high accuracy (Elberink & Vosselman, 2009). Despite this, LIDAR is not a technology accessible to many, and obtaining roof planes from LIDAR carried some difficulties in distinguishing boundaries adjacent to other objects (Hang & Cai, 2020).

In contrast, Remote Sensing imagery, particularly very high-resolution (VHR) imagery, provides vast amounts of spatial and textural information. VHR remote sensing data possesses several advantages; it is versatile, allowing acquisition at different scales and areas; it can be easily accessible to users due to the availability of numerous open data resources with relatively low or even zero cost depending on the area to be acquired (Lu & Weng, 2007).

An alternative strategy involves the fusion of various datasets, as suggested by (Alidoost et al., 2019) and (Awrangjeb et al., 2013). However, this approach comes with its challenges, such as compatibility between datasets, as different datasets may have different collection methods, acquisition times, precisions, accuracy, and resolutions among a diversity of characteristics depending on the origin of the data, which entails different integration process challenges (K. Liu et al., 2020).

2.2. State-of-the-art deep learning methods in structured geometry reconstruction

Following the classification presented by Chen et al. (2022), three groups of structured geometry reconstruction algorithms were identified.

2.2.1. Traditional methods

Structured geometry reconstruction is an active topic in computer vision research, focusing on transforming raster data into vectorized geometries. This conversion includes and is not limited: to wireframes, planes, room layouts, floorplans, and polygonal loops. Traditional methodologies are developed based on low-level image processing techniques such as the Hough transform (Cui et al., 2012) or superpixel segmentation (Bauda et al., 2015). Further development in the field has led to the proposal of more advanced methods, examples of which include graphical model inference based on graph cuts for planar reconstruction (Sinha et al., 2007), dynamic programming for floorplan recovery (Pintore et al., 2018), among others.

2.2.2. Hybrid systems that incorporate deep learning approaches

Deep learning has recently gained popularity for reconstructing vector geometries (Lundervold & Lundervold, 2019). Most cutting-edge systems adopt a two-stage process: First, neural networks are used to identify low-level primitives features such as corners, edges, and region segments. Then, optimization methods are used to assemble all components into the final models (K. Huang et al., 2018).

The works developed by (Nauata & Furukawa, 2020) and (Chen et al., 2019) are clear examples of the mentioned two-stage process. Their works rely on Mask R-CNN (He et al., 2020) to detect primitive features, followed by an optimization technique such as integer programming to assemble the planar graphs for outdoor building reconstruction and indoor floorplans. The MonteFloor method (Stekovic et al., 2021) employs a similar detection strategy but relies on Monte Carlo Tree Search to reconstruct planar graph structures. Despite their effectiveness, these optimization/search techniques require manual, domain-specific algorithm design and exhibit significantly slower testing times. The work developed by F. Zhang et al.(2021) focuses on exploration and classification steps to make up a novel strategy that results in a better solution.

2.2.3. End-to-end neural networks.

End-to-end neural architectures are gaining popularity owing to their ability to reduce the need for manual engineering and deliver rapid inference. The domain of wireframe parsing (K. Huang et al., 2018), L-CNN (Zhou, Qi, & Ma, 2019) demonstrates this by adapting a Convolutional Network (ConvNet) for junction detection, followed by an edge verification network responsible for the classification of each edge candidate.

Models such as PPGNet (Z. Zhang et al., 2019) and HAWP (N. Xue et al., 2020) propose more innovative designs while following the two-stage framework similar to L-CNN. (Zhou, Qi, Zhai, et al., 2019) extend the scope of the wireframe task to 3D by jointly estimating depths and vanishing points along with geometric primitives. However, these techniques treat edge candidates independently.

ConvMPN, another type of graph neural network explicitly designed for planar graph reconstruction, represents another approach (F. Zhang et al., 2020). The work developed by Zhao et al. (2022) proposed a novel design based on two components 1) multi-task learning for feature extraction and 2) Graph Neural Network. The approach is applied for planar reconstruction/extraction of rooftops in satellite imagery. The recent success of the Transformer-based object detector models, such as the DEtection TRansformers (DETR) model, has inspired its application to wireframe parsing by LinE segment Transformer (LETR) model. DETR/LETR employs "dummy nodes" as storage placeholders for detection responses and avoids heuristic-based steps like non-maximum suppression (Xu et al., 2021).

In contrast, to all the exposed methods above. The Holistic Edge Attention Transformer (HEAT) method developed by Chen et al.(2022) based on an attention-based neural network is a state-of-the-art end-to-end method for planar graph reconstruction. It infers an overarching structure by learning comprehensive structural patterns. In addition, the attention-based architecture allows the model to focus on specific areas (corners and edges) of the input data relevant for planar graph reconstruction; this has shown superior outcomes in comparison with the ConvMPN method according to the exposed in his work. On the other hand, DETR/LETR uses "dummy nodes" as placeholders to store detection results. These "dummy nodes" are essentially slots where the model can place predictions for where objects are in an image. If the model predicts fewer objects than the maximum number of slots, the remaining slots are filled with "dummy" predictions that indicate no object (Xu et al., 2021). In contrast, HEAT works on all "edge candidates" instead of dummy nodes throughout its designed decoders and learning strategy. By considering all edge candidates, this model may be able to more effectively understand the overall structure of the scene compared to DETR/LETR, which focuses on detecting individual features.

2.3. Summary

This chapter reviews the primary data sources and cutting-edge methodologies deployed for planar graph reconstruction. Although LIDAR point clouds serve as precise input data; however, they may be unavailable or cost prohibitive. Contrariwise, aerial imagery offers rich optical data and can be obtained for large regions at little cost compared with LIDAR, making them a beneficial alternative. Moreover, the fusion of diverse data sources has become a common strategy in this field. From all the revised approaches, the end-to-end approach developed by Chen et al. (2022)is the state-of-the-art method for planar graph reconstruction.

3. MATERIALS AND METHODOLOGY

3.1. Study areas and dataset

The presented research will be focused on two different areas, which are presented below:

1. Stadsveld – 't Zwering area, nestled in the central, southeastern area of Enschede, The Netherlands. (Figure 3). The area covers an area of approximately 153 hectares, containing a diverse blend of urban structures with various modern building types: residential, commercial, and public buildings. Notably, residential buildings emerge as the predominant structures in this area (Kenzhebay, 2022).

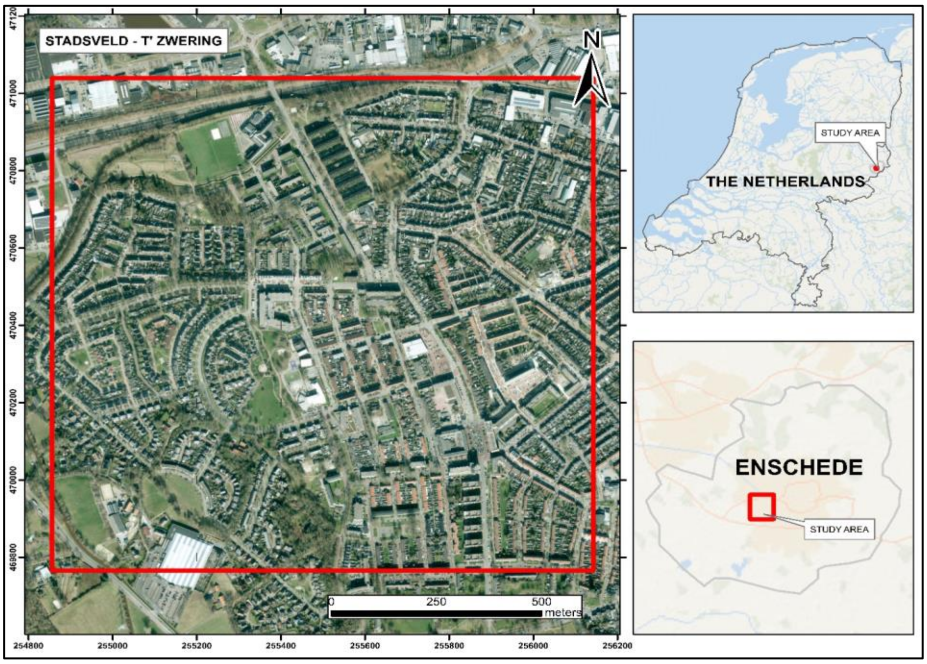


Figure 3. Location of the reference area Stadsveld – 't Zwering , in The Netherlands.

2. Oude Markt area, situated in the center of Enschede, The Netherlands. (Figure 4). covers an area of approximately 6 hectares. Moreover, it is encompassed by a diverse array of buildings, from historical to public, and plenty of commercial buildings (Van Melik, 2009).

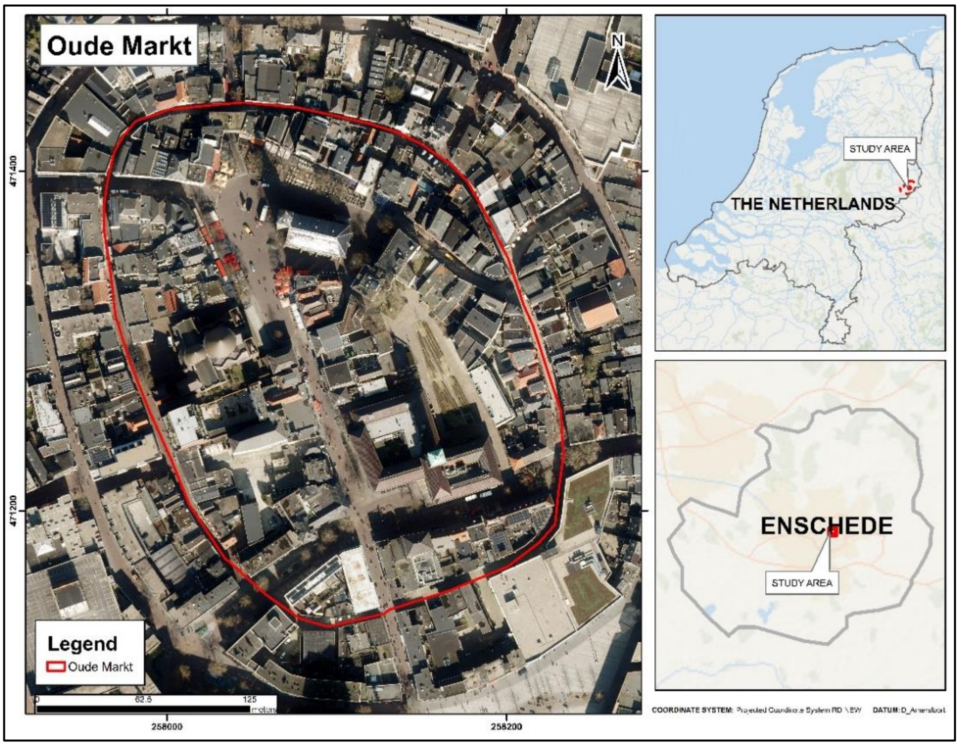


Figure 4. Location of the reference area Oude Markt, in The Netherlands.

3. Lozenets area, located in the south-central area of Sofia, Bulgaria. (Figure 5), covers an area of around 812 hectares. The area is divided into two main areas: Upper and Lower Lozenets. Lower Lozenets borders the center of Sofia and primarily consists of diverse and dispersed building block structures, commercial buildings, industrial structures, and many green areas. Upper Lozenets hosts several modern structures, communist-era apartment blocks, and historical buildings and currently undergoing regeneration, with many new constructions cropping up (Sandrini & Ii, 2022). This area represents a rich configuration of many different building structures and configurations.

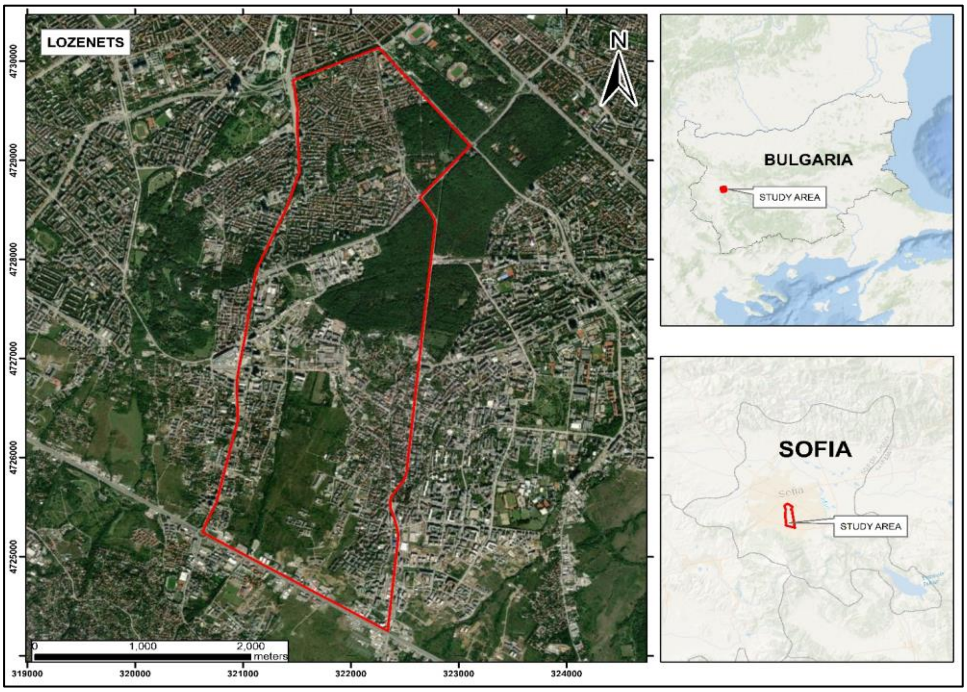


Figure 5. Location of the reference area Lozenets, Bulgaria.

The study area selection was guided by the ambition of cities to transition towards LOD2 3D modelling and digital twinning (Rezaei et al., 2023). This move aligns with the current research lines of "The Big Data for Smart Society Institute (GATE)[1]" from Sofia, Bulgaria, which collaborates with this research, providing us with access to quality datasets (aerial imagery, building footprints, cadastral information), as well as the service provided by the Public Services On the Map (PDOK), which is a service provided by the Dutch government that allows public geographic information in the Netherlands to be made freely available as open data. The datasets used in the current research are presented in Table 2.

Table 2. Dataset content.

Location	Data	Source
Stadsveld – 't Zwering, Enschede, The Netherlands	Orthophoto (8 cm) from aerial imagery, 2020	Public Services On the Map (PDOK)
	Buildings, inner roofs planes	Produced by the author
	Buildings footprints	Public Services On the Map (PDOK), edited by the author to reduce mismatches between the aerial images and the building footprint.
Oude Markt Enschede, The Netherlands	Orthophoto (8 cm) from aerial imagery, 2022	Public Services On the Map (PDOK)
	Buildings, inner roofs planes	Produced by the author
	Buildings footprints	Public Services On the Map (PDOK), edited by the author to reduce mismatches between the aerial images and the building footprint.
	DSM 0.2m resolution derived from AHN4 (Point Cloud), 2020	Water Board House
	DTM 0.2m resolution derived from AHN4 (Point Cloud), 2020	
	nDSM 0.2m resolution derived from AHN4 (Point Cloud), 2020	
Lozenets, Sofia, Bulgaria	Orthophoto (10 cm) from aerial imagery, 2020	GATE
	Buildings footprints	Produced by RMSI with the support of the author from a dataset provided by GATE
	Buildings, inner roofs planes	

*RMSI is a consulting company hired to digitize the buildings' inner roofs planes and buildings footprints in the study area of Lozenets, Sofia, Bulgaria.

Based on the presented dataset, three main stages for its use have been defined, as shown in Table 3.

[1] https://gate-ai.eu/en/home/

Table 3. Dataset and its use across the current research.

DATASETS	The training dataset for the further deep learning model	Building inner roofs planes delineation/extraction (Testing)	3D Modelling (Testing)
Stadsveld – 't Zwering, Enschede, The Netherlands dataset	X	X	
Oude Mark, Enschede, The Netherlands dataset		X	X
Lozenets, Sofia, Bulgaria dataset	X	X	

3.2. Overall methodology

This research uses a deep learning-based framework for automatic building roof plane extraction in vector format from remote sensing data (aerial images, vector shapefile dataset). The developed deep learning framework will undergo testing in two areas to analyze its performance across different scenarios. Subsequently, the roof vector outputs from the deep learning approach will be combined with Digital Elevation Models (DEMs) for 3D city models. The overall research consists of five consecutive phases, which are outlined as follows:

Phase I: Data preparation: This stage involves the generation of reference data for the selected study areas, automation of the data preparation process for its application to the chosen deep learning model, and the splitting of the dataset into training, validation, and testing subsets. For the current framework, the approach assumes the availability of the building footprint needed to create the samples and the required information to be used in the framework. In the current research, the building footprints for The Netherlands dataset were obtained from the PDOK portal and manually edited to correct mismatches with the aerial imagery. In the case of the dataset for the Bulgaria study area, the building footprints were manually digitalized.

Phase II: Training the deep learning model: This Phase is related to training the selected deep learning model using various training datasets (Enschede, Sofia, and a combination of Enschede and Sofia) to obtain the different models that can be employed in our study areas.

Phase III: Building roof planes extraction: This Phase involves converting the building roof planes, which have been generated and encoded in a Python dictionary during the previous phase, into vector-format building roof planes. In addition, several experiments in different areas were conducted to appraise the developed model's performance and identify potential avenues for refinement and challenges to the developed framework.

Phase IV: 3D Modelling: This Phase is related to using the extracted building roof planes for 3D modelling.

Phase V: Method evaluation: This Phase consists of defining and applying various predetermined and defined metrics at different stages of the research for evaluation.

A detailed overview of the proposed methodology is illustrated in Figure 6.

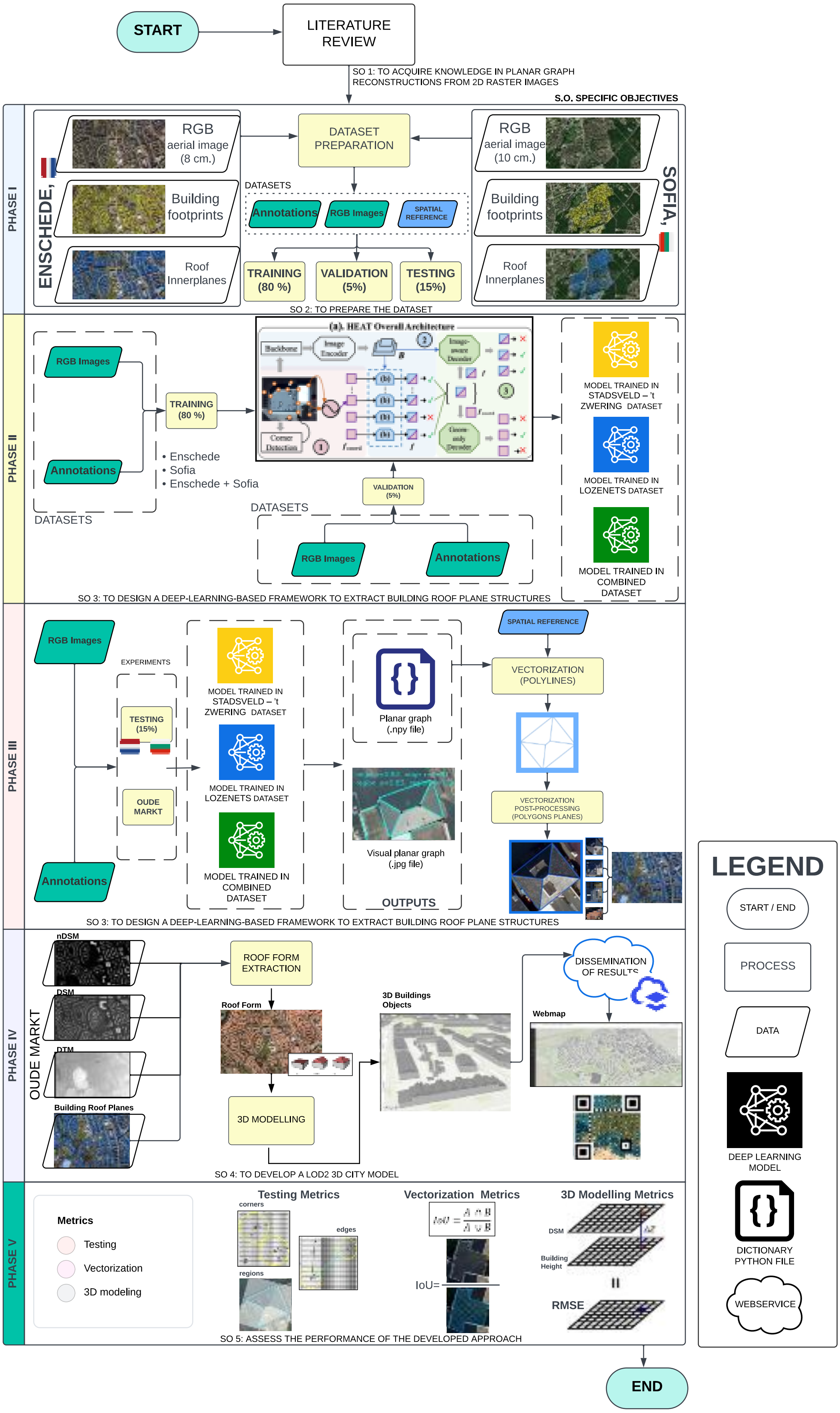


Figure 6. Methodological Flowchart of the research.

3.3. The architecture of the HEAT model

Based on the literature review, the Holistic Edge Attention Transformer (HEAT) was chosen to be adapted as the deep learning model used for the current research. Developed by Chen et al. (2022), HEAT is a novel attention-based neural network model for planar graphs reconstruction from 2D raster images using an end-to-end transformer-based neural architecture, depicting the underlying geometric structures. The architecture consists of several key components, which are detailed as follows:

Trigonometric positional encoding: The HEAT model uses a trigonometric positional encoding scheme to encode the spatial information of the input image. The process involves mapping the coordinates of the features to a continuous space and then applying the sine and cosine functions to generate the positional encodings. This strategy enables the network to learn the underlying geometric structures of the image by capturing the relative distance and spatial relationships.

Deformable attention: The HEAT model uses a deformable attention mechanism to attend its focus on different parts of the input image. The deformable attention is used to fuse multi-scale image features from the used backbone of each edge candidate. This allows the model to focus on the most relevant parts of the image to enhance its ability to reconstruct the planar graph.

Weight-sharing Transformer decoders: The HEAT model uses two weight-sharing Transformer decoders as the technical core of the approach. These decoders learn the geometric relationship patterns between the edges and corners and exploit image information. In that way, HEAT can classify each edge candidate as incorrect or correct. By using weight-sharing transformer decoders, the model can use the power of transformer architecture to capture the complex relationships of all features (corners and edges). The weight-sharing components allow the network to learn from images and the geometric pattern, allowing an accurate reconstruction of the planar graph on the raster information.

Masked learning strategy and iterative inference: The entire system is trained end-to-end with a masked learning strategy. The idea behind the masked learning strategy is to randomly mask some ground-truth labels for the edge candidates during training. Expressly, for each edge candidate, the ground-truth label is randomly set to one of three states: (T) true, (F) false, or (U) unknown. The network is then trained to predict the missing labels for the unknown states. This allows it to learn the underlying geometric structures and push the frontier of end-to-end neural architecture for structured reconstruction.

The testing phase uses an iterative label inference process. The inference is conductive iteratively, from the most confident to the harder. This process involves updating the model's predictions over three iterations, using a decoder that considers the image's content. All items labeled as (U) are in the state mask at the first iteration. In the second iteration, the model updates these labels based on its confidence predictions score. If the confidence prediction score for an edge is less than 0.01, the edge is updated and labeled as (F). If the confidence predictions score exceeds 0.9, the edge is updated and labeled as (T). All other edges retain the (U) label. In the final iteration, the model uses a confidence threshold of 0.5 to make its final predictions. This iterative inference process allows the model to refine its predictions over multiple passes, potentially leading to more accurate results.

The HEAT model is characterized by three stages. **The first stage** is edge node initialization. HEAT starts by detecting corners using a corner detector. The corner detector model is a variant of the edge classification HEAT model. In the corner detector model, pixels are considered corner candidates and become the nodes. Instead of treating every pixel in the image space as a candidate, each 4x4 “superpixel” is treated as a node to reduce the memory cost. Each pair of detected corners is considered a potential edge candidate. Then each edge candidate becomes a transformer node. Then each transformer node is initialized by a positional encoding strategy using equation 1, f_{coord}, which uses a 256 – dimensional trigonometric position encoding $\gamma(t)$ defined in equation 2, and w_i a central component of this function defined in equation 3. $\gamma(t)$ encodes the positional information of the corner in a structured manner.

$$f_{coord} = M_{coord}[\,\gamma(e_1^x), \gamma(e_1^y), \gamma(e_2^x), \gamma(e_2^y)], \quad \textbf{(1)}$$

$$\gamma(t) = [\, sin(w_0 t), cos(w_0 t), \dots sin(w_{31} t), cos(w_{31} t)\,], \quad \textbf{(2)}$$

$$w_i = \left(\frac{1}{10000}\right)^{\frac{2i}{32}} (i = 0,1, \dots ,31), \quad \textbf{(3)}$$

Where, e_1 and e_2 are the two corners (considered as an edge candidate). e_1^x and e_1^y denotes the x and y coordinates of the corner e_1 respectively. M_{coord} is a 256x256 learnable matrix for linear mapping. The function $\gamma(t)$ captures the information about the relative distances between coordinates. Encoding this information can improve the accuracy of corner detection by considering the spatial relationship of corners and capturing patterns that might occur at different scales within the images. Figure 7 shows a diagram that illustrates the edge initialization process.

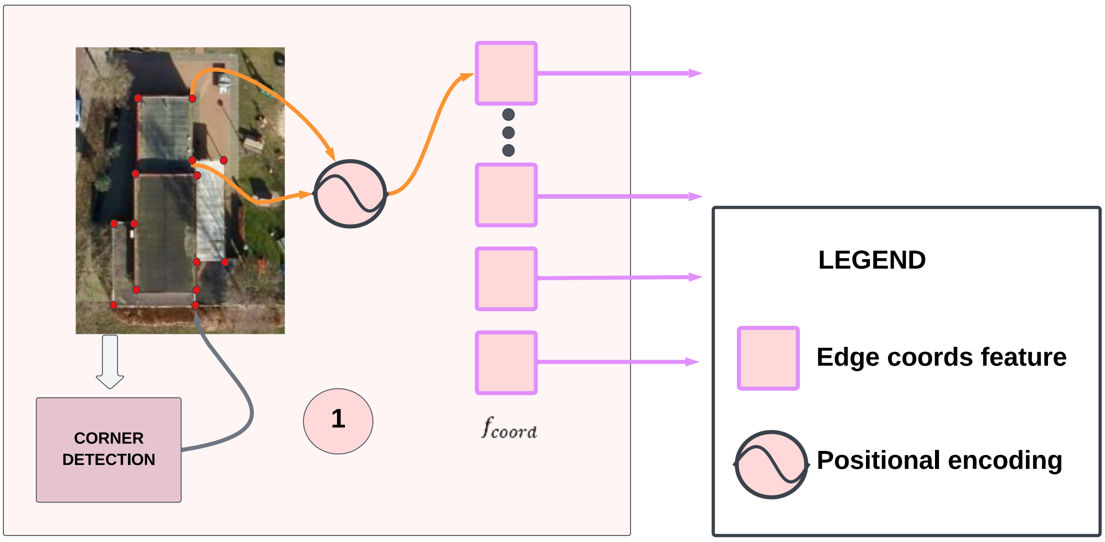


Figure 7. Edge Initialization (Chen et al., 2022).

The second stage of the model is the edge image feature fusion and edge filtering. First, edge nodes extract image features from the feature maps produced by a ConvNet backbone (The architecture of this ConvNet is composed of a ResNet backbone and a Transformer encoder borrowed from Deformable-DETR to build a 3-level image feature pyramid, with shapes of 64x64x256, 32x32x256, and 16x16x256). For each level (l=1,2,3), f_{coord} is used to generate sampling locations around an edge and the attached attention weight associated with them for aggregation, defined in equations 4 and 5.

$$\Delta^l = M_{loc}^l f_{coord}, \quad \textbf{(4)}$$

$$\omega^l = softmax(M_{agg}^l f_{coord}) \quad \textbf{(5)}$$

Where M_{loc}^{l}, and M_{agg}^{l} are learnable weights for the feature level l. Δ^{l} contains four relative positions from the center of an edge for a certain level of the feature pyramid. These positions are likely used as sampling locations for feature extraction, and ω^{l} stands for the corresponding attention weights, normalized, to sum up 1 using the softmax function. This determines the probability that the model should focus on the features extracted from different parts of the images. These values are used to calculate f_{img}(equation 6) at the edges, which represents the information the model extracts from the image at the edges.

$$f_{img} = \sum_{l=1}^{3} \sum_{i=1}^{4} \omega^{l}\ (i)[M_{img}^{l} \mathrm{B}^{l}(\frac{e_1+e_2}{2^{l+2}} + \frac{\Delta^{l}(i)}{2^{l+1}})] \quad \textbf{(6)}$$

Where f_{img} is the image feature at each edge, which is a 256-dimensional vector, M_{img}^{l} is a learnable weight matrix of level l (l=1,2,3). B^{l} stands for the feature map at level l of the feature pyramid. In resume, the model extracts feature from the image at each edge, then transforms this feature using a learnable weight matrix, and focuses on different parts of the images by using an 8-way multi-head attention strategy. The resulting image features are then combined with the positional encoding to create a final fused feature for each edge. This is done by obtaining a fused feature f (equation 7) as in the original transformer, by a standard add-norm layer, and a feed-forward network (FFN).

$$f = FFN\ (Add\&Norm(f_{img}, f_{coord}\ )) \quad \textbf{(7)}$$

This mechanism allows the network to selectively attend to different parts of the images and extract the relevant features of each edge candidate. This fused information is combined with the positional encoding of the edge nodes to obtain a refined feature representation for each edge candidate. The **second stage** involves filtering out edge candidates unlikely to belong to the reconstructed planar graph. By applying a Weight-Sharing Transformer Decoder, the model analyses the refined feature representation of each edge candidate. The transformer decoder is trained to identify geometrical patterns and learns to recognize arrangements between edges based on structural geometry patterns. The Transformer Decoder classifies each edge candidate as likely to belong to the planar graph structure or not. The HEAT framework can refine the edge nodes and produce a more accurate representation of the planar structure being reconstructed.

The corner detector model, the edge classification model, and the transformer decoder are trained jointly. This allows the different parts of the model to learn from each other and improve the learning/inference performance. Figure 8 illustrates the edge image feature fusion and edge filtering of HEAT.

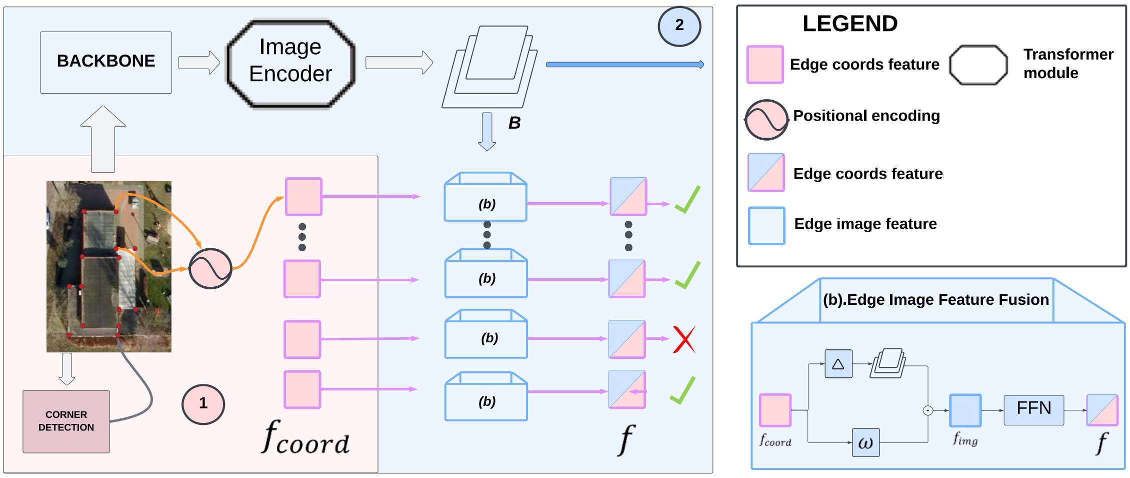


Figure 8.The edge image feature fusion process (Chen et al., 2022)

The **third stage** of the model is the final edge inference with weight-sharing transformer decoders. The "Image Aware Decoder" takes the fused features of the edge and conducts Holistic edge self-attention to understand the context in which each edge exists within the structure. The self-attention mechanism enables the model to attend to different edges within the images and learn their dependencies by assigning weights to each edge based on its relevance against others.

The second decoder. "The Geometry-only decoder" only takes the coordinates features from the node initialization step and has no access to the image information. It shares all the weights with the image-aware decoder and is only used during training as a regularization.

The total loss function used in HEAT framework for training is comprised of 4 binary cross-entropy (BCE) loss functions: one for corner prediction and three for edge classification, where the goal is to detect corners and classify each edge candidate as either correct or incorrect, the binary cross entropy function $L_{BCE}\ (y, \hat{y})$ is defined in equation 8, and the total loss function $Total\ _{loss}$ of the HEAT model is defined in equation 9.

$$L_{BCE}\ (y, \hat{y}) = -\frac{1}{HW}\Sigma_{x\in I} y(x) \cdot \log(\hat{y}(x)) + (1 - y(x)) \cdot \log(1 - \hat{y}(x)), \quad \textbf{(8)}$$

$$Total\ _{loss} = (L_{BCE}\ (y, \hat{y})_{S1})_{edge} + (L_{BCE}\ (y, \hat{y})\ ^{hb}_{S2})_{edge} + (L_{BCE}\ (y, \hat{y})^{rel}_{S2})_{edge} + (L_{BCE}\ (y, \hat{y}))_{corner} * \alpha \quad \textbf{(9)}$$

Where H and W are the image's height and width (256x256), ŷ is the predicted probability of the pixel being a corner/edge, and y is the ground truth. The corner loss is multiplied by the corner factor, α, which is a hyperparameter used for weighting the importance of the corner loss function relative to the other loss components and adjusting the influence of this term in the total loss function ($Total\ _{loss}$). Under the context of this research, the predetermined factor of α =0.05, defined by Chen et al.(2022), was used.

Input images are either 256x256 or 512x512; for our research, images of 256x256 were used, as the available computing resources limited the use of image size 512x512. Figure 9 shows the overall architecture of HEAT and its keys component.

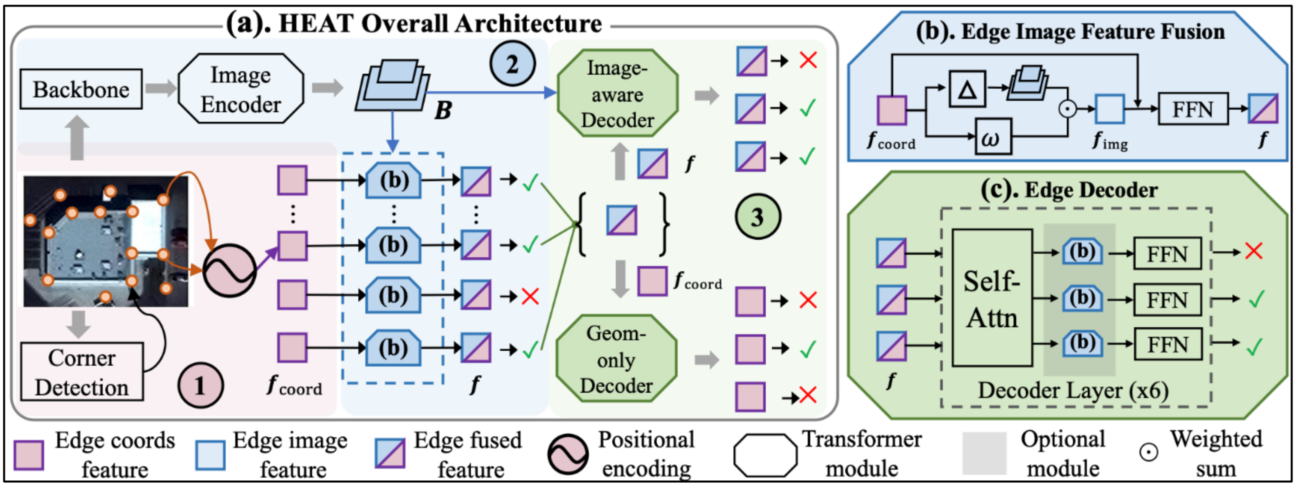


Figure 9. The overall architecture of the HEAT model (Chen et al., 2022).

3.4. Phase I: Data preparation

The input datasets for the further developed approach consist of RGB aerial images, building footprints with a buffer of around 2m to have spatial context information of every building, and building inner roof planes reference data. Considering that the reference data for the building footprint and inner roof planes were manually digitized over the aerial images (0.08 m resolution for Enschede and 0.10 m resolution for Sofia), some considerations were taken into account like some minor mismatches between the buildings samples reference data and the aerial image. Reference data were split into training/validation/testing following the same standard proposed by Chen et al. (2022) during experiments on the development of the HEAT model, as follows:

Stadsveld – 't Zwering, Enschede, The Netherlands (Figure 10), is comprised of a total of 2465 building samples. The dataset for this area is divided into 1972 (yellow buildings), 123 (blue buildings), and 370 (red buildings) samples randomly selected for training (80%), validation(5%), and testing (15%), respectively.

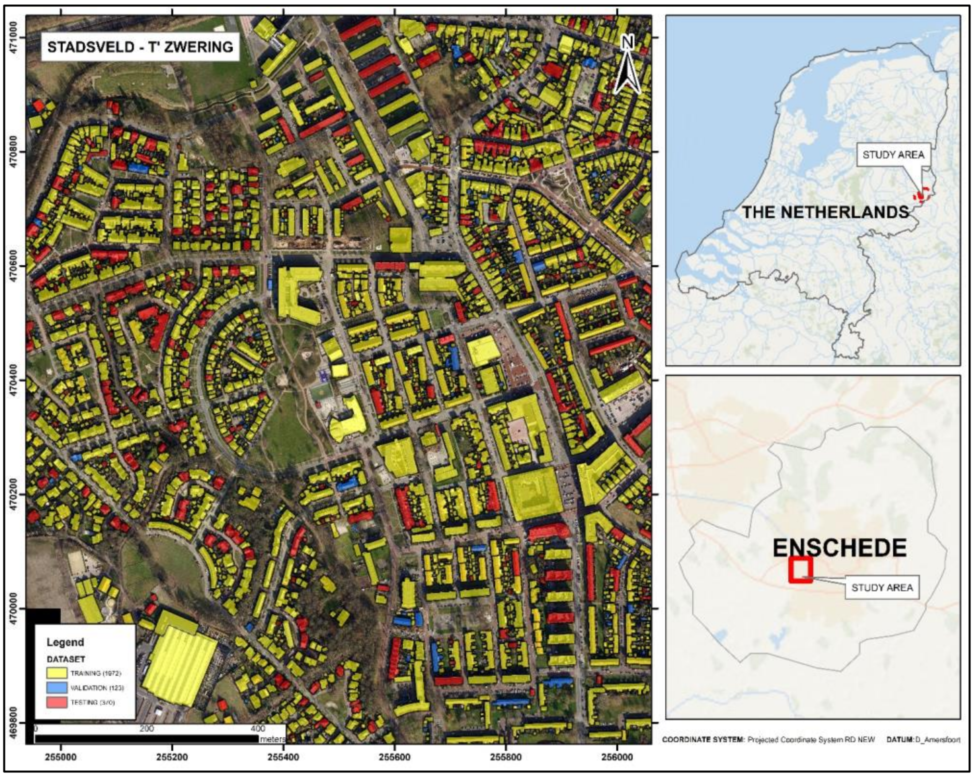


Figure 10. Location of the building's samples in Stadsveld – 't Zwering, Enschede, The Netherlands.

Lozenets, Sofia, Bulgaria (Figure 11), covers an area of around 812 hectares, with 1800 building samples. The buildings dataset for this area is divided into 1440 (yellow buildings), 90(blue buildings), and 270 (red buildings) samples randomly selected for training (80%), validation (5%), and testing (15%), respectively.

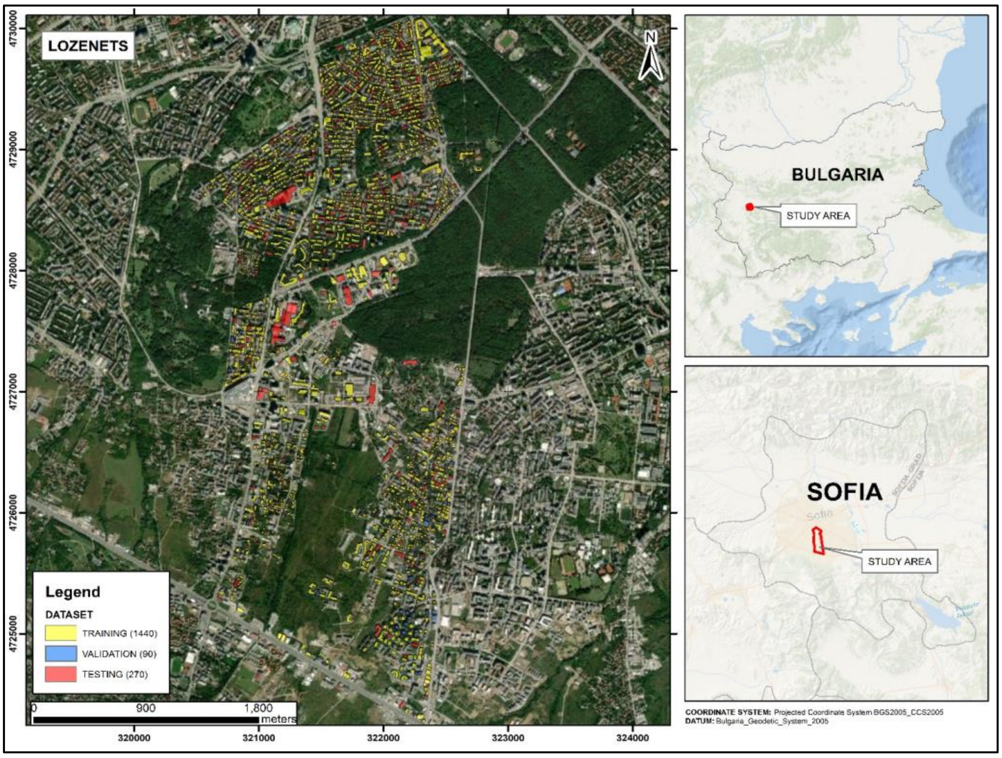
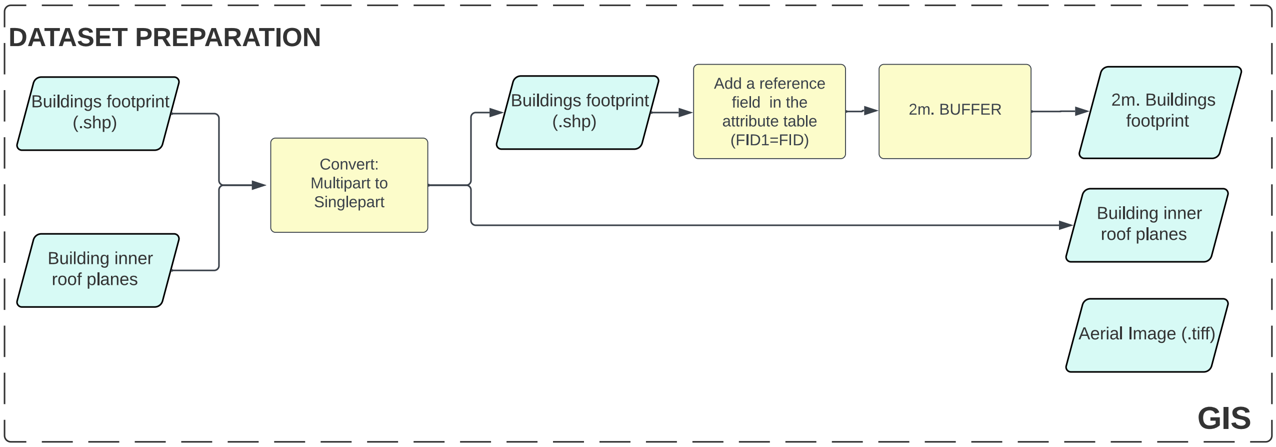


Figure 11. Location of the building's samples in Lozenets, Sofia, Bulgaria.

The building's inner roof plane samples and building footprints were digitalized and edited using GIS software. Different GIS tools were applied to prevent topological inconsistency between the digitalized planes. Subsequently, a buffer of 2m was created around the building footprint to capture the whole building and spatial context in each image sample of each building. Figure 12 shows the preprocessing process using GIS software.

Figure 12. Flowchart of data preprocessing process performed in GIS.


DATASET PREPARATION
Buildings footprint (.shp)
Building inner roof planes
Convert: Multipart to Singlepart
Buildings footprint (.shp)
Add a reference field in the attribute table (FID1=FID)
2m. BUFFER
2m. Buildings footprint
Building inner roof planes
Aerial Image (.tiff)
GIS

In the pre-processing stage implemented in our methodology, the building's external polygons, inner roof planes, and the image sections encompassing each building sample must be resized to a uniform image dimension size of 256x256. This resizing is a requirement for creating the required dataset compatible with the chosen deep learning method. To achieve this requirement, a resizing strategy defined by the bounding box of the external 2-meter polygon around each building was implemented. Figure 13 illustrates the mentioned resizing strategy, while equations 10 and 11 provide the mathematical representation to translate the features from real-world feature coordinates (lat, lon) to the 256x256 pixel image format coordinates.

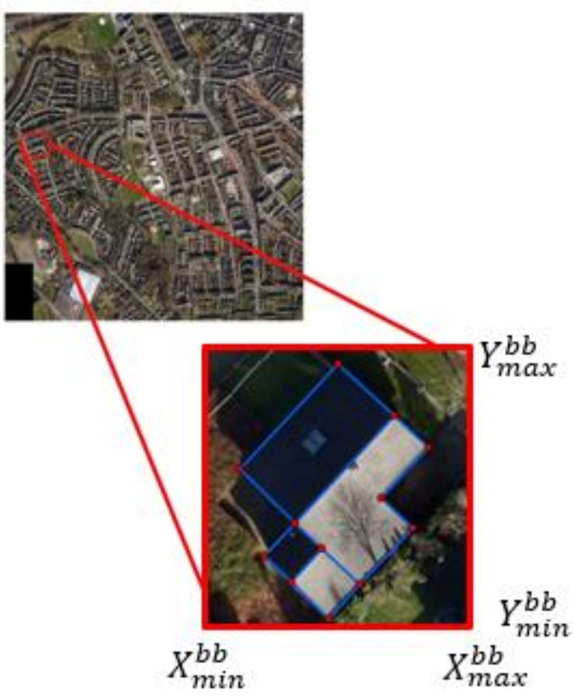


Figure 13. From real-world feature coordinates to the 256x256 pixel image format coordinates.

Resize equations:

$$X_{img} = \frac{255\,(X_{cord}^{real} - X_{min}^{bb})}{X_{max}^{bb} - X_{min}^{bb}} \qquad \textbf{(10)}$$

$$Y_{img} = 255 - \frac{255(Y_{cord}^{real} - Y_{min}^{bb})}{Y_{max}^{bb} - Y_{min}^{bb}} \qquad \textbf{(11)}$$

Where X_{img}, Y_{img} are coordinates that represent the new coordinates within the 256x256 pixel image size, X_{cord}^{real}, Y_{cord}^{real} are coordinates that represent real-world coordinates of the features (points/corners) within the bounding box, $X_{min}^{bb}, Y_{min}^{bb}$are coordinates that represent the minimum bounding box coordinates defined by the external 2-meter polygon around each building, X_{max}^{bb} , Y_{max}^{bb} are coordinates representing the maximum bounding box coordinates defined by the same external 2-meter polygon around each building. 255 is the maximum value within the 0 to 255-pixel image coordinates range.

To meet the dataset requirement of the selected deep learning approach, all the required input datasets are generated. Aerial images are clipped and resized into smaller patches of 256x256 pixels, in accordance with the bounding box defined by the external polygons, which surround each building with a 2-meter margin (RGB Images).

The information concerning the coordinate reference system and the bounding box coordinates of the 2m-external building polygons for each building sample are stored in individual text files per building sample (SPATIAL REFERENCE). This step is essential for later processes involving resizing from image pixel coordinates back to real-world coordinates.

The building's roof structure, composed of many inner roof planes, is encoded into a Python dictionary as a planar graph on image coordinates. These dictionary files contain each building roof sample's planar graph, corners, edges, and geometric interrelations. Figure 14 depicts a diagram illustrating the described data preparation process, performed using Python 3.8.

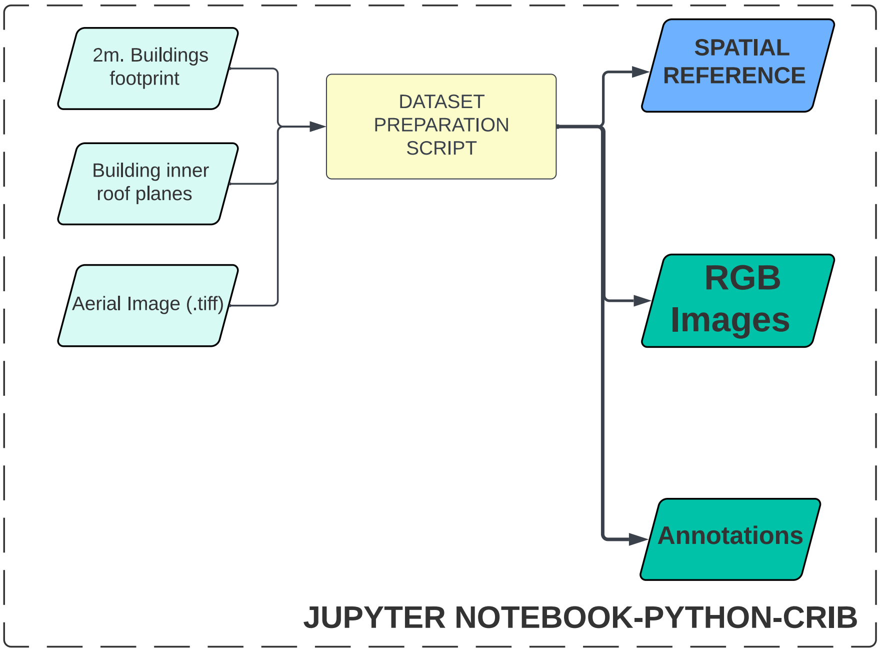


Figure 14. Flowchart of data preprocessing process performed – phase I.

3.5. Phase II: Training the deep learning model

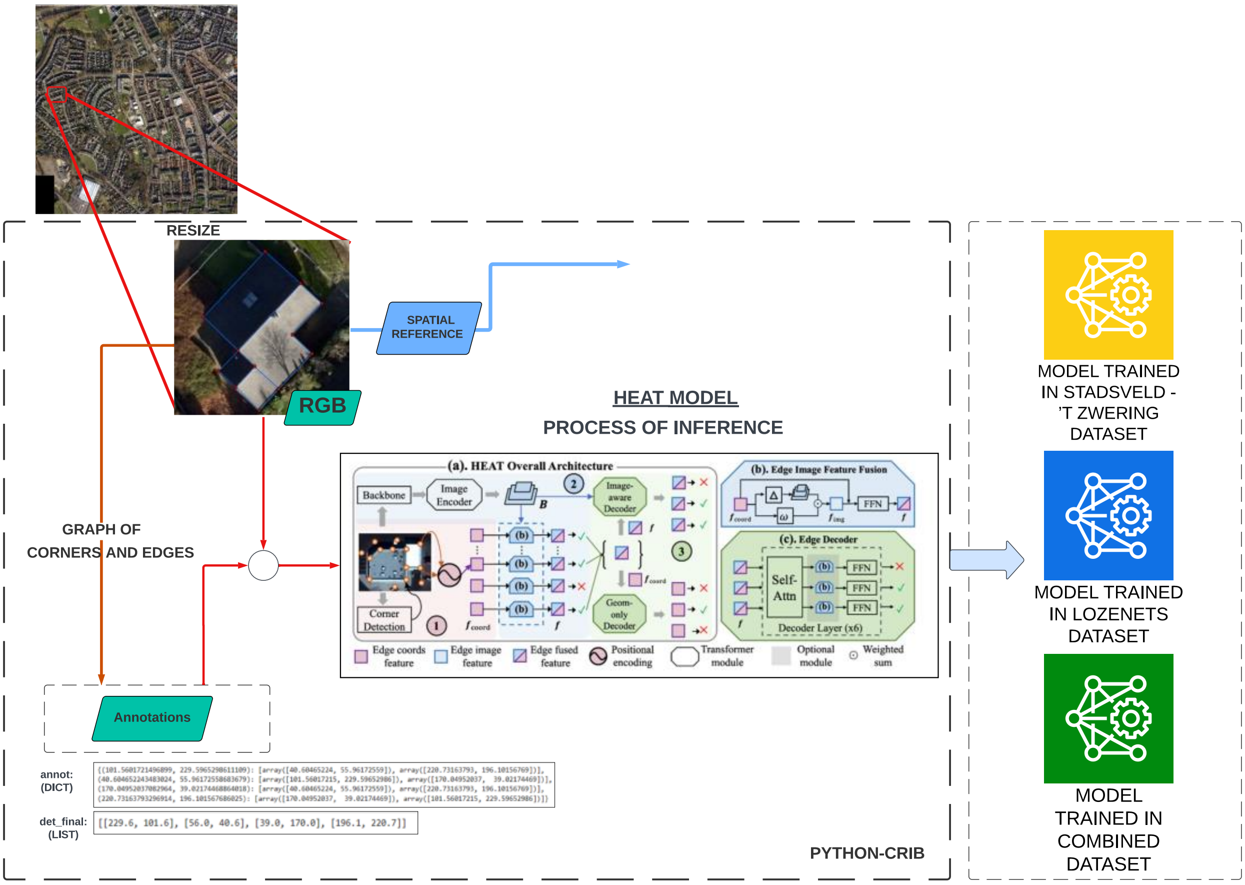
The training process starts using the HEAT pre-trained model. This model was configured and trained according to the following parameters: The original HEAT model was developed in Python3.7 and Pytorch1.5.1, the image encoder contains only one Transformer layer, while the edge decoders have six. The backbone of the model is ResNet-50.. The optimizer used for the model is the Adam optimizer, with an initial learning rate of $2\mathrm{e}-4$ and a weight decay factor of $1\mathrm{e}-4$. The learning rate decays by a factor of 10 for the last 25% of epochs. Regarding LETR, the model was trained for 800 epochs (0-799) based on the dataset size (1601 building image samples from Paris, Las Vegas, or Atlanta) without a hyper-parameter search.

The training strategy in the current research consisted of using the prior knowledge of the pre-trained model as a starting point. Based on previous experiments, an arbitrary number of 646 epochs was set. The training process was monitored using the validation accuracy value to find the best model with the highest accuracy within that training session to be selected for further application. The models were trained using Python 3.8 and Pytorch1.12.1, available in the cloud computing platform of The Center of Expertise in Big Geodata Science (CRIB). The training parameters for the different sets of datasets are presented as follows in Table 4.

Table 4. Dataset and parameters used for training.

Model	**Dataset size**			**Image size**	**Batch size**	**Max number of corners per image**
	Training	**Validation**	**Total**			
The model trained on Stadsveld – 't Zwering, Enschede, The Netherlands dataset	1972	123	2095	256	16	150
The model trained on Lozenets, Sofia, Bulgaria dataset	1440	90	1530			
The model trained on a combined dataset from Stadsveld – 't Zwering and Lozenets dataset	3412	213	3625			

Figure 15 shows an overview of Phase II, which shows the environment in which the training process was performed and what input datasets from Phase I are needed to perform the training process. As outputs from this Phase, three models are obtained according to the different training sets used.



```
{(101.5601721496899, 229.5965298611109): [array([40.60465224, 55.96172559]), array([220.73163793, 196.10156769])],
(40.604652243483024, 55.96172558683679): [array([101.56017215, 229.59652986]), array([170.04952037,  39.02174469])],
(170.04952037082964, 39.02174468864018): [array([40.60465224, 55.96172559]), array([220.73163793, 196.10156769])],
(220.73163793296914, 196.101567686025): [array([170.04952037,  39.02174469]), array([101.56017215, 229.59652986])]}
```

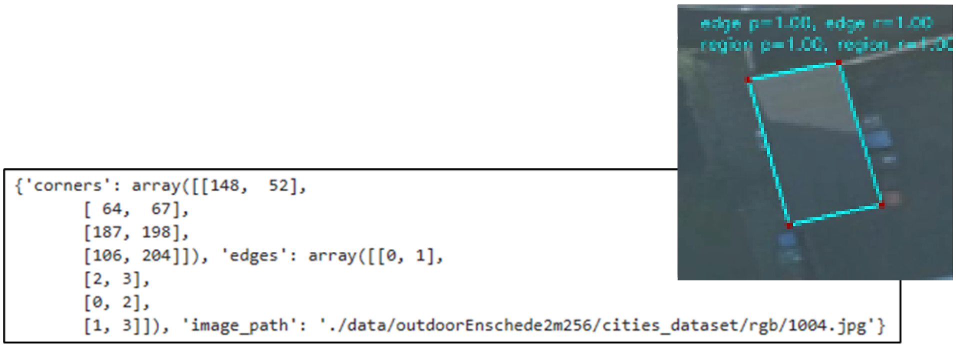
```
[[229.6, 101.6], [56.0, 40.6], [39.0, 170.0], [196.1, 220.7]]
```

Figure 15. Flowchart of training process performed in phase II.

3.6. Phase III: Building roof planes extraction

Upon completion of the training phase, the research progresses to the third phase, which centers around evaluating the performance of the trained models, which centers around evaluating the performance of the trained models in delineating the building's inner roof planes on the images samples into planar graphs and subsequently conducting the post-processing operation to convert the obtained planar graph of the building roof structure into a vector format dataset.

The model is applied to the predefined testing datasets to delineate the inner roof planes of the building samples. The outputs of this application are captured in a Python dictionary with three keys. Figure 16 illustrates the configuration of the reconstructed planar graph in a Python dictionary:



```
{'corners': array([[148,  52],
       [ 64,  67],
       [187, 198],
       [106, 204]]), 'edges': array([[0, 1],
       [2, 3],
       [0, 2],
       [1, 3]]), 'image_path': './data/outdoorEnschede2m256/cities_dataset/rgb/1004.jpg'}
```

Figure 16. The obtained planar graph after applying the model to a building image sample.

The three different keys are described as follows:

- **'corners'**: This key maps to a 2D array of integers. Each row in the array represents the x and y coordinates of a corner in an image.
- **'edges'**: This key maps to a 2D array of integers. Each row in the array represents a pair of corners (specified by their indices in the 'corners' array) that form an edge. So, for example, the edge [0, 1] means an edge between the first and second corners in the 'corners' array.
- **'image_path'**: This key maps to a string that specifies the path to an image file. This image corresponds to the inferred corners and edges on the input-image building sample.

The observed coordinates of the inferred corners on the images fall within the range of 0-255. This range corresponds to the pixel dimensions of the input image 256x256, indicating that the coordinates are expressed in image pixel units. The following strategy will focus on converting the obtained planar graph from pixel units to real-world coordinates.

This conversion process will leverage a method similar to the one used in Phase I for data preparation but in an inversed manner. As the delineated buildings' inner roofs plane graphs are named with their corresponding input image names, the image file name will be used to map the corresponding polygon bounding box used for clipping this image sample from the original aerial image. That information is captured in the spatial reference text file saved in Phase I.

The transformation equations that will be used to resize and georeference from image units to real-world coordinates are a variation of the resize equations 10 and 11. These equations are defined as follows in equations 12 and 13:

$$Y_{cord}^{real} = \frac{(255 - Y_{img})(Y_{max}^{bb} - Y_{min}^{bb})}{255} + Y_{min}^{bb} \qquad \textbf{(12)}$$

$$X_{cord}^{real} = \frac{(X_{max}^{bb} - X_{min}^{bb}) * X_{img}}{255} + X_{min}^{bb} \qquad \textbf{(13)}$$

Where X_{cord}^{real} , Y_{cord}^{real} are coordinates that represent the converted corners from the dictionary to real-world coordinates, $X_{min}^{bb}, Y_{min}^{bb}$ are coordinates that represent the minimum bounding box coordinates defined by the external 2-meter polygon around each building that is saved in the spatial reference text file from Phase I, X_{max}^{bb} , Y_{max}^{bb} are coordinates that represent the maximum bounding box coordinates defined by the same external 2-meter polygon around each building that is saved in the geom_txt file from Phase I, X_{img}, Y_{img} are coordinates representing the image coordinates within the 256x256 image size from the obtained graph stored in a Python dictionary. The number 255 is the maximum value within the range of from 0 to 255-pixel image coordinates.

Once every file is resized and georeferenced to its real-world location, the next step is to merge all the converted building inner roof planes and convert them into vector format, which is illustrated in Figure 17.

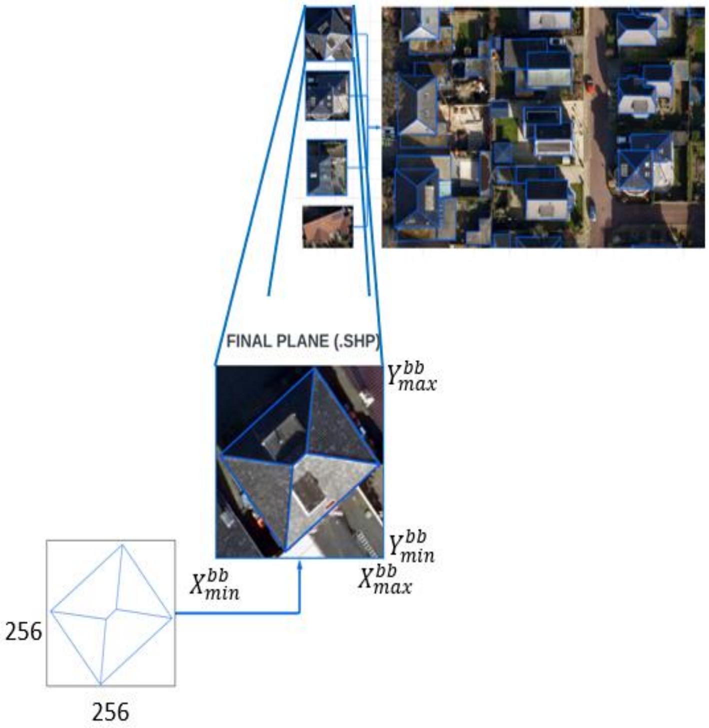


Figure 17. Building inner roof plane extracted from image coordinates to real-world coordinates.

The overall pipeline of phase III is shown in Figure 18. This process is performed using Python 3.8 in the cloud computing platform CRIB. The building's inner roof planes were obtained in polyline vector format and converted into polygon vector format using GIS software tools.

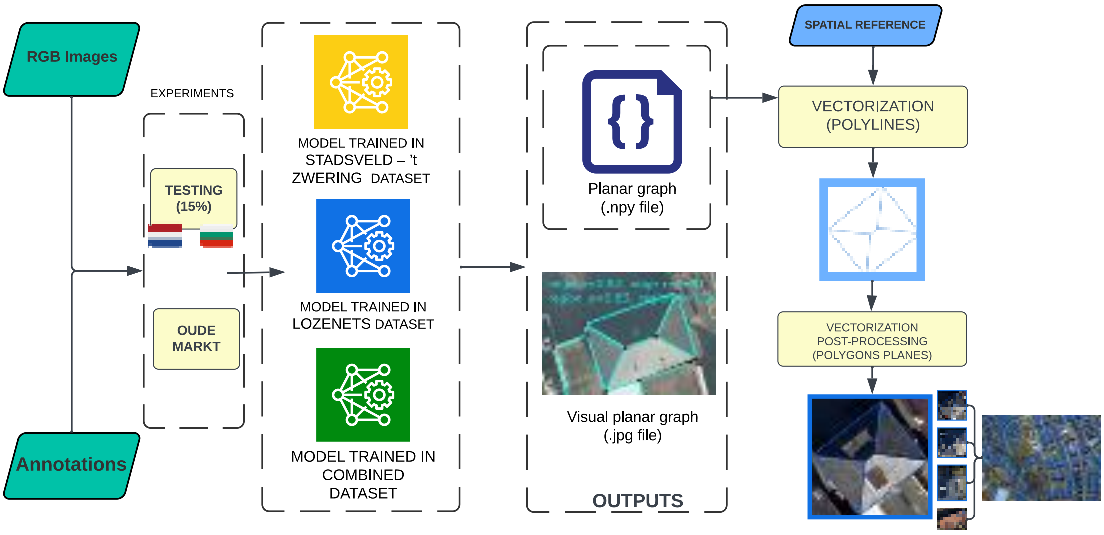

RGB Images
Annotations
EXPERIMENTS
TESTING (15%)
OUDE MARKT
MODEL TRAINED IN STADSVELD – 't ZWERING DATASET
MODEL TRAINED IN LOZENETS DATASET
MODEL TRAINED IN COMBINED DATASET
Planar graph (.npy file)
Visual planar graph (.jpg file)
OUTPUTS
SPATIAL REFERENCE
VECTORIZATION (POLYLINES)
VECTORIZATION POST-PROCESSING (POLYGONS PLANES)

Figure 18. The overall pipeline of Phase III.

3.7. Phase IV: 3D Modelling

The fourth phase focused on the applicability of the obtained building's inner roof plane structures for creating a LOD2 3D city model. This procedure adheres to the methodology for 3D city modelling outlined in the 3DBasemap extension of the commercial GIS software ArcGIS. The methodology proposed a multistep procedure combining different GIS tools to combine the building's inner roof planes in polygon vector format, DSM., DTM, and nDSM. This combination aids in inferring the roof form and subsequently facilitates the representation of LOD2 3D building objects.

Under the scope of this research, this phase was tested only for the Oude Markt dataset. Figure 19 shows the process in GIS software to create the LOD2 3D building models by integrating the different mentioned datasets.

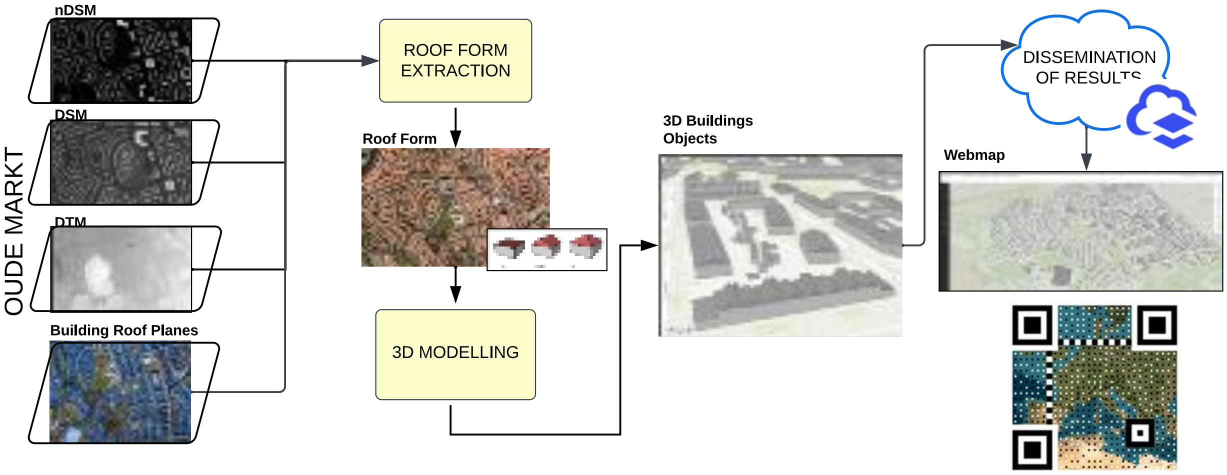


Figure 19. Flowchart of the Building 3D modelling process-phase IV.

3.8. Phase V: Method evaluation

To assess the performance of the designed workflow. The workflow was tested across stages: Buildings' inner roof planes delineation, vectorization, and 3D modelling. However, the present research did not define a metric to assess the performance of the whole workflow.

3.8.1. Buildings' inner roof planes delineation

This section assessed the model's performance in delineating the buildings' inner roof planes on the image samples following the same metrics used by Chen et al.(2022) by evaluating the model's performance in detecting corners, edges, and regions by using the standard formulas of precision (equation 14), recall (equation 15), and F1 score (equation 16) which are described as follows.

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP)+False\ Positive\ (FP)} \quad \textbf{(14)}$$

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP)+False\ Negative\ (FN)} \quad \textbf{(15)}$$

$$F1\ score = 2\ X\ \frac{Precision\ x\ Recall}{Precision+\ Recall} \quad \textbf{(16)}$$

Metrics for corners:

The method starts by extracting corner data from the ground truth (annotation file) and the model's predictions (planar graph file), and then a temporal list to track is created. This list tracks which ground truth corners have been matched with detected corners (predicted). A corner is successfully predicted if a ground-truth corner is located within a Euclidean distance of an 8-pixel radius. In instances where multiple corners are detected around a single ground-truth corner, only the nearest one is deemed correct, with the others classified as false positives. Finally, the method calculates recall, precision, and F1 score using the standard formulas. Figure 20 illustrates the process of corner detection.

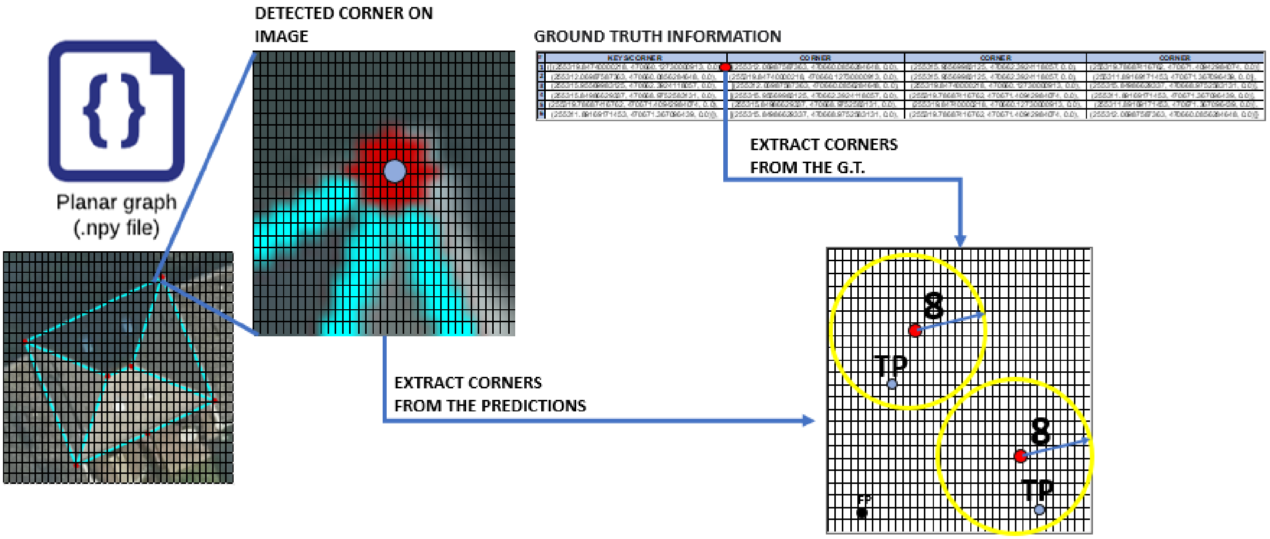
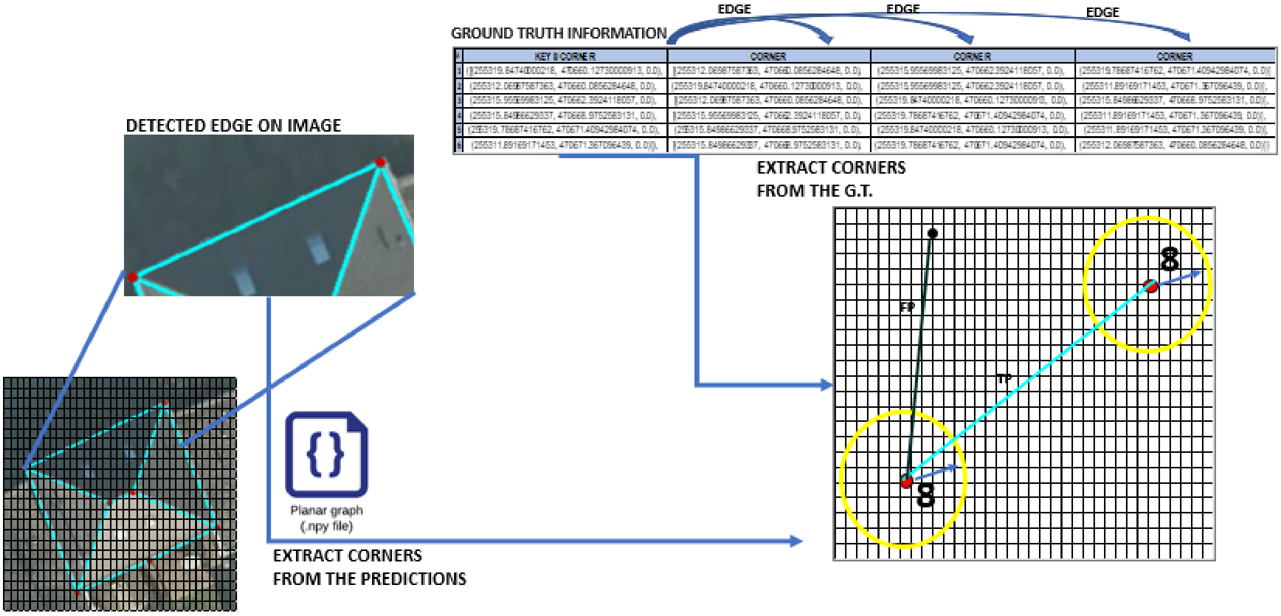


Figure 20. Corners detection.

Metrics for edges:

An edge is considered a true positive if its end corners are both detected and the pair of corners exist in the ground truth data (annotation file). The process starts by iterating through the detected edges. For each edge, it checks if the edge's corners were detected. If not, it counts it as a false positive and moves to the subsequent detection. If the edge's corners were detected, it checks for a match between the detected edge and the ground truth edges (whether the two corners of the detected edge are the same as the two corners of any ground truth edge, regardless of order). If a match is found, it counts as a true positive. If no match is found, it counts as a false positive. Finally, the method calculates recall, precision, and F1 score using the standard formulas. Figure 21 illustrates the process of edge detection.

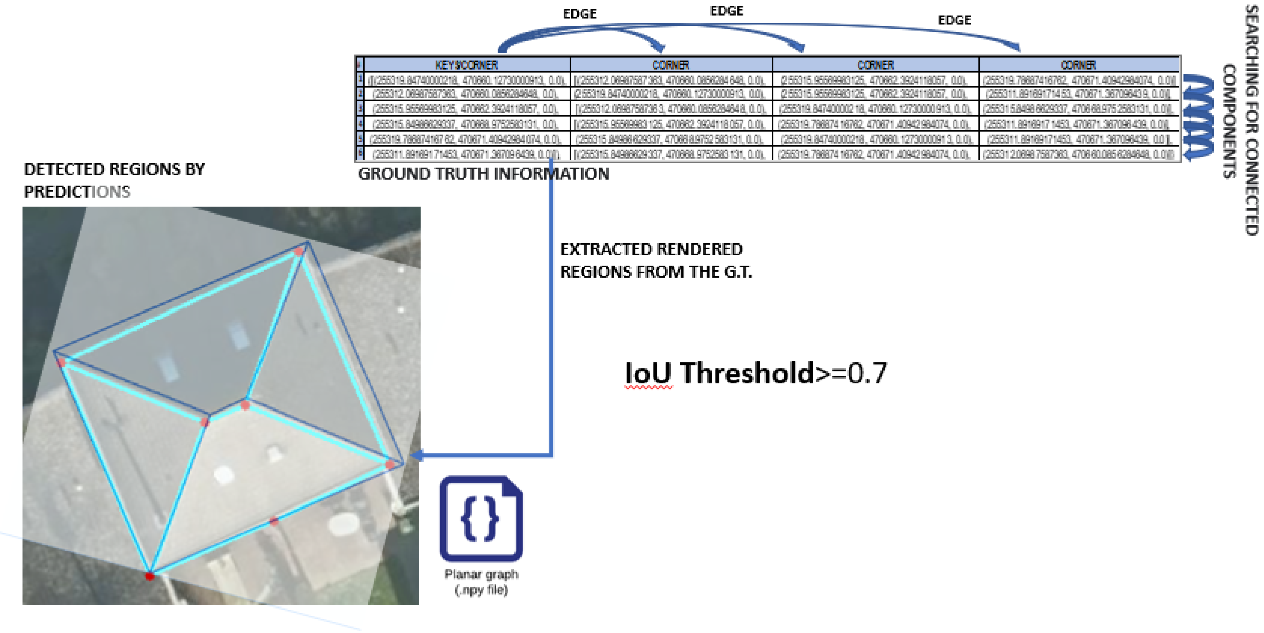


#	KEY II CORNER	CORNER	CORNER	CORNER
1	([(255319.84740000218, 470660.12730000913, 0.0),	[(255312.06987587363, 470660.0856284648, 0.0)	(255315.95569983125, 470662.3924118057, 0.0),	(255319.78687416762, 470671.40942984074, 0.0)]
2	(255312.06987587363, 470660.0856284648, 0.0),	(255319.84740000218, 470660.12730000913, 0.0),	(255315.95569983125, 470662.3924118057, 0.0),	(255311.89169171453, 470671.367096439, 0.0)],
3	(255315.95569983125, 470662.3924118057, 0.0),	[(255312.06987587363, 470660.0856284648, 0.0),	(255319.84740000218, 470660.12730000913, 0.0),	(255315.84986629337, 470668.9752583131, 0.0)],
4	(255315.84986629337, 470668.9752583131, 0.0),	[(255315.95569983125, 470662.3924118057, 0.0)	(255319.78687416762, 470671.40942984074, 0.0),	(255311.89169171453, 470671.367096439, 0.0)],
5	(255319.78687416762, 470671.40942984074, 0.0),	(255315.84986629337, 470668.9752583131, 0.0),	(255319.84740000218, 470660.12730000913, 0.0),	(255311.89169171453, 470671.367096439, 0.0)],
6	(255311.89169171453, 470671.367096439, 0.0)]),	[(255315.84986629337, 470668.9752583131, 0.0)	(255319.78687416762, 470671.40942984074, 0.0),	(255312.06987587363, 470660.0856284648, 0.0)])

Figure 21. Edge detection.

Metrics for regions:

Regions are detected by rendering the detected closed shapes formed by the detected corners and edges. These connected shapes are connected components. Each connected component corresponds to a region. The function then calculates IoU for each detected region and the ground truth regions. If the IoU is greater than or equal to 0.7, the ground truth region has not been matched before, and the region does not overlap with any other region; it is considered a true positive; otherwise, it is a false positive. Note that this metric does not consider the positioning and sharing of corners and edges. Finally, the method calculates recall, precision, and F1 score using the standard formulas. Figure 22 illustrates the process of region detection.


EDGE
EDGE
EDGE
SEARCHING FOR CONNECTED COMPONENTS
DETECTED REGIONS BY PREDICTIONS
GROUND TRUTH INFORMATION
EXTRACTED RENDERED REGIONS FROM THE G.T.
IoU Threshold>=0.7
Planar graph
(.npy file)

#	KEY I/CORNER	CORNER	CORNER	CORNER
1	([(255319.84740000218, 470660.12730000913, 0.0),	[(255312.06987587363, 470660.0856284648, 0.0),	(255315.95569983125, 470662.3924118057, 0.0),	(255319.78687416762, 470671.40942984074, 0.0)]
2	(255312.06987587363, 470660.0856284648, 0.0),	(255319.84740000218, 470660.12730000913, 0.0),	(255315.95569983125, 470662.3924118057, 0.0),	(255311.89169171453, 470671.367096439, 0.0)],
3	(255315.95569983125, 470662.3924118057, 0.0),	[(255312.06987587363, 470660.0856284648, 0.0),	(255319.84740000218, 470660.12730000913, 0.0),	(255315.84986629337, 470668.9752583131, 0.0)],
4	(255315.84986629337, 470668.9752583131, 0.0),	[(255315.95569983125, 470662.3924118057, 0.0),	(255319.78687416762, 470671.40942984074, 0.0),	(255311.89169171453, 470671.367096439, 0.0)],
5	(255319.78687416762, 470671.40942984074, 0.0),	(255315.84986629337, 470668.9752583131, 0.0),	(255319.84740000218, 470660.12730000913, 0.0),	(255311.89169171453, 470671.367096439, 0.0)],
6	(255311.89169171453, 470671.367096439, 0.0)]),	[(255315.84986629337, 470668.9752583131, 0.0),	(255319.78687416762, 470671.40942984074, 0.0),	(255312.06987587363, 470660.0856284648, 0.0)])

Figure 22. Region detection.

3.8.2. Vectorization

The Intersection over Union was used as a vector format evaluation metric for the predicted building inner roof planes. As given in equation 17, IoU is calculated by dividing the intersection area by the union area of predicted inner roof planes (A) and ground truth building inner roof planes (B). This metric evaluates the accuracy of final outputs obtained after extracting and vectorizing all the building's inner predicted planes.

$$IoU = \frac{Area\ (A \cap B)}{Area\ (A \cup B)} \quad \textbf{(17)}$$

3.8.3. 3D Modelling

The Root Mean Square Error (RMSE) was employed to quantify the disparity between the DSM and the generated LOD2 3D city model based on the building's inner roof planes. For this computation, each pixel within the inner roof planes is accounted for, and to ensure compatibility with the DSM resolution, the LOD2 3D CITY model is rasterized to a resolution of 0.2 meters.

The process begins with the calculation of all residuals on the inner roof planes considering all the contained pixels, and this is achieved by calculating the difference between the modeled building height ($Building_{pixel\ value}$) and the actual height which is given by the DSM ($DSM_{pixel\ value}$), after obtaining the residuals, the residuals are squared to neutralize any negative values and assign greater importance to more significant errors.

Moreover, the mean of these squared residuals, known as the Mean Squared Error (MSE), was then computed by dividing the sum of the squared residuals by the total number of pixels within the plane (N). Finally, the RMSE is calculated by taking the square root of the MSE, as indicated in Equation 18.

$$RMSE_{Building\ Inner\ Roof\ Plane} = \sqrt{\sum_{i=1}^{N} \frac{(DSM_{pixel\ value} - Building_{pixel\ value})^2}{N}} \quad \textbf{(18)}$$

This process was conducted by using GIS tools and following the methodology developed by Dimitriadou & Nikolakopoulos (2022). A lower RMSE value means a better alignment of the LOD2 3D city model with the DSM. Conversely, a higher RMSE value suggests a less accurate alignment, indicating that the 3D building model's predictions have more significant discrepancies with the DSM.

3.9. Implementation Details

The approach was implemented using Python 3.8 and Pytorch1.12.1, using the cloud computing server of CRIB for ITC, using images size 256x256.
To implement the HEAT model, it is necessary to install the required packages and compile the deformable-attention modules from (Zhu et al., 2020). The training details are described in Phase 2.

3.10. Summary

In this chapter, the study areas of the current research were introduced, and a detailed explanation of the HEAT model's architecture, the design framework around it for buildings' inner roof planes delineation, extraction, and LOD2 3D modelling. The methodology of the research is divided into five different phases. The metrics and evaluation strategies to assess performance across the phases are described.

4. RESULTS AND ANALYSIS

This research conducted a series of experiments to evaluate the different trained models' performance across multiple created test datasets from different areas. Stadsveld - 't Zwering, and Oude Markt, in Enschede, The Netherlands, and Lozenets in Sofia, Bulgaria. Our evaluation metrics are divided into four key sections: training, buildings' inner roofs planes delineation, vectorization, and 3D Modelling.

The training section shows the performance of the conducted training for the different trained models. The buildings' inner roofs planes delineation assessed the model's ability to infer and delineate the inner roof planes on the provided building image samples.

In the vectorization section, the focus shifts to the conversion process of the planar graph of each building sample into a vector polygon shapefile format. Here, the Intersection over Union (IoU) of all building inner roof planes was computed and compared against the ground truth building inner planes. The 3D Modelling section is dedicated to analyzing data from the Oude Markt area, as it is the designated test dataset for this phase in the methodology. Therefore, the presented results in this section are specific to this dataset.

4.1. Training

Each training session was monitored using the calculated loss during the training and the accuracy for the validation. For the model trained on Stadsveld – ’t Zwering, Enschede, The Netherlands dataset, the loss curve shows a decreasing trend around the whole training session. Meanwhile, the validation curve shows that the highest validation value of 0.76 was obtained after 474 epochs (epoch 1273 considering the whole HEAT framework) and then showed a stable trend during the rest of the training session (rest 172 epochs). Figure 23 shows this training session's loss and accuracy curves with a red mark on the highest accuracy value.

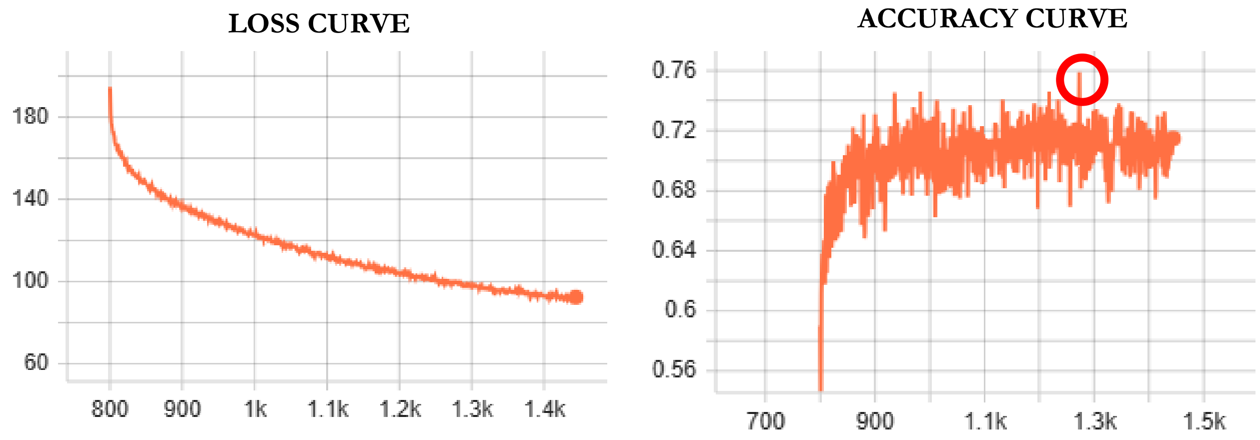


Figure 23. Loss and accuracy curves for the training session using Stadsveld – ’t Zwering, Enschede, The Netherlands dataset.

For the model trained on the Lozenets dataset, the validation curve shows that the highest validation value of 0.72 was obtained after 131 epochs (epoch 930 considering the whole HEAT framework) and then showed a decreasing trend during the rest of the training session (rest 515 epochs). Figure 24 shows this training session's loss and accuracy curves with a red mark on the highest accuracy value.

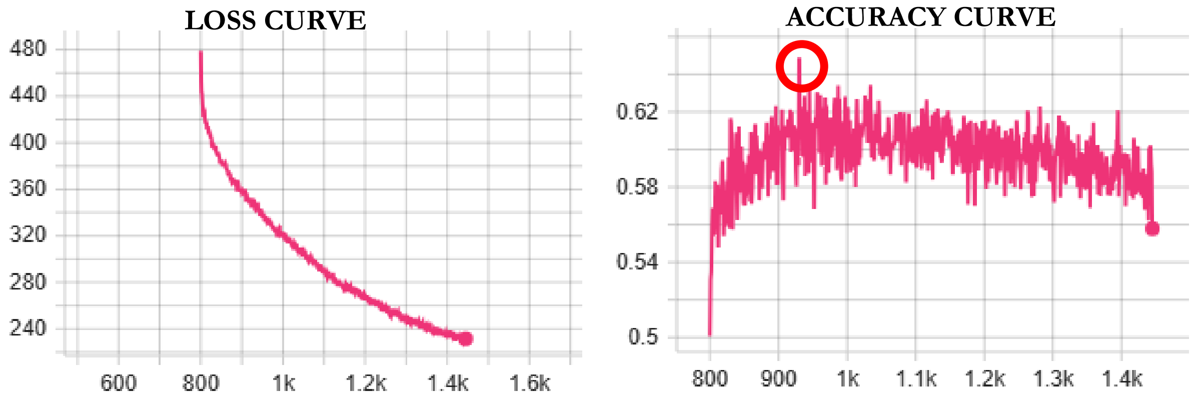


Figure 24. Loss and accuracy curves for the training session using Lozenets, Sofia, Bulgaria dataset.

For the model trained on the combined dataset of Stadsveld – ’t Zwering, and Lozenets, Sofia, the loss curve shows a decreasing trend around the training session. Meanwhile, the validation curve shows that the highest validation value of 0.65 was obtained after 406 epochs (epoch 1205 considering the whole HEAT framework) and then showed a decreasing trend during the rest of the training session (rest 244 epochs). Figure 25 shows this training session’s loss and accuracy curves with a red mark on the highest accuracy value.

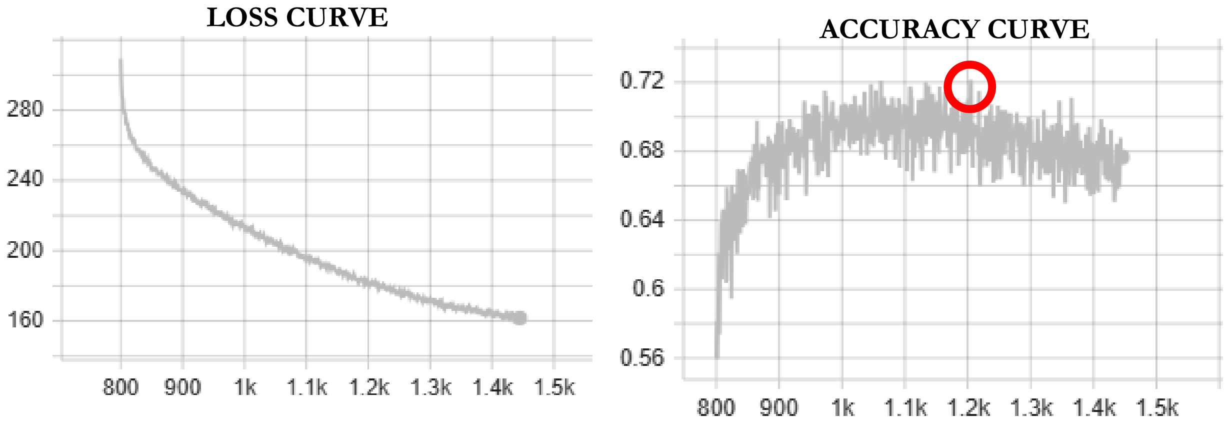


Figure 25. Loss and accuracy curves for the training session using Lozenets, Sofia, Bulgaria dataset.

During all three training sessions, the loss curves display a consistent downward trend throughout the training session, indicating that the model is effectively learning and improving its predictions over time. However, after reaching their peaks, curves plateau for the model trained on the Stadsveld – ’t Zwering validation dataset and show a decreasing trend for the other two cases. It is interpreted that the model might have reached an optimal state where further training does not significantly enhance its performance on the validation set and is a precursor of overfitting.

Once the two training datasets were combined, the combined dataset model showed performance similar to the Enschede dataset but not superior to this. In addition, the best model performance of the combined model was reached in fewer epochs than the model trained on the Stadsveld – ’t Zwering, Enschede dataset. Figure 26 compares the loss and accuracy curves of the different trained models. Nevertheless, each model's performance is not comparable because its validation datasets are different.

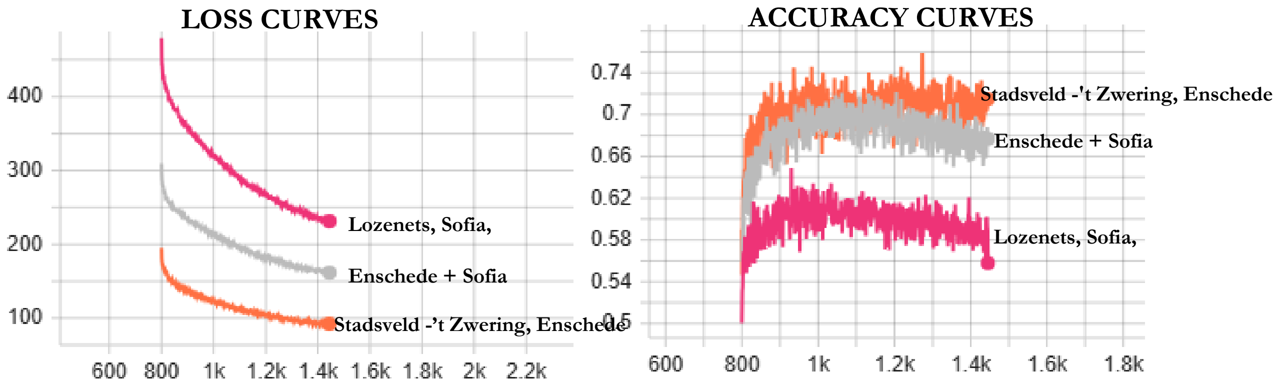


Figure 26. Loss and accuracy curves for the different trained models.

4.2. Buildings' inner roofs planes delineation

4.2.1. Quantitative analysis

Table 5 shows the evaluation metric for the buildings' inner roofs planes delineation stage. This section shows the obtained values for predicting corners, edges, and regions using the different models and test datasets. The numbers in bold indicate the highest value obtained for the model on the analyzed parameters among all the models. In the cases where more than one model showed similar performance, both values are shown in bold.

To simplify the names of the trained models, the model trained on the Stadsveld-'t Zwering dataset is named “MODEL TRAINED ON ENSCHEDE DATASET”, the model trained in Lozenets dataset is named “MODEL TRAINED ON SOFIA DATASET”, and the model with the combined dataset is named “MODEL TRAINED ON COMBINED DATASET (ENSCHEDE + SOFIA)”.

Table 5. Quantitative evaluations on building inner roof plane reconstruction on the different testing sets.

TESTING AREA	MODELS	CORNERS			EDGES			REGIONS		
		PRECISION	RECALL	F1_SCORE	PRECISION	RECALL	F1_SCORE	PRECISION	RECALL	F1_SCORE
Stadsveld -'t Zwering, Enschede, The Netherlands	MODEL TRAINED ON ENSCHEDE DATASET	**0.85**	0.68	**0.76**	**0.61**	0.50	0.55	0.72	**0.64**	**0.68**
	MODEL TRAINED ON SOFIA DATASET	0.52	**0.72**	0.60	0.34	0.48	0.40	0.41	0.56	0.47
	MODEL TRAINED ON COMBINED DATASET (ENSCHEDE + SOFIA)	**0.85**	0.68	**0.76**	**0.61**	**0.51**	**0.56**	**0.73**	**0.64**	**0.68**
Oude Markt, Enschede, The Netherlands	MODEL TRAINED ON ENSCHEDE DATASET	**0.69**	0.46	0.55	**0.38**	0.24	0.29	**0.49**	0.30	0.37
	MODEL TRAINED ON SOFIA DATASET	0.43	**0.64**	0.51	0.22	**0.34**	0.27	0.27	0.40	0.32
	MODEL TRAINED ON COMBINED DATASET (ENSCHEDE + SOFIA)	0.60	0.55	**0.57**	0.31	0.29	**0.30**	0.44	**0.43**	**0.43**
Lozenets, Sofia, Bulgaria	MODEL TRAINED ON ENSCHEDE DATASET	**0.84**	0.27	0.41	0.39	0.12	0.19	0.45	0.13	0.21
	MODEL TRAINED ON SOFIA DATASET	0.80	**0.53**	**0.63**	**0.44**	**0.31**	**0.37**	**0.47**	**0.37**	**0.41**

MODEL TRAINED ON COMBINED DATASET (ENSCHEDE + SOFIA)	0.81	0.50	0.62	**0.44**	0.30	0.36	**0.47**	0.35	**0.41**

As is shown in Table 4, for the testing dataset of Stadsveld-'t Zwering, the model trained on the training dataset of Stadsveld-'t Zwering shows similar performance to the model trained in the combined dataset of Stadsveld -'t Zwering and Lozenets In contrast with the model trained on the Lozenets training dataset which shown the poorest performance. These results suggest that the combined dataset did not improve the models' performance for the Stadsveld -'t Zwering testing area.

When evaluated on the testing dataset of Lozenets, both the model trained on the training dataset of Lozenets and the model trained in the combined dataset of Stadsveld -'t Zwering and Lozenets exhibited a comparable level of performance. In contrast, the model trained on the Stadsveld -'t Zwering training dataset shows the poorest performance. These findings suggest that using the model trained in Enschede does not have the necessary structural ability to infer structural roof structures as the buildings samples from Sofia.

However, a contrasting pattern emerged in the case of the testing dataset of Oude Markt. Here, the model trained on the combined training dataset of Stadsveld -'t Zwering and Lozenets outperformed the model trained solely on the training Stadsveld -'t Zwering dataset and the model trained on the training Lozenets dataset.

4.2.2. Qualitative analysis

Stadsveld -'t Zwering, Enschede, The Netherlands

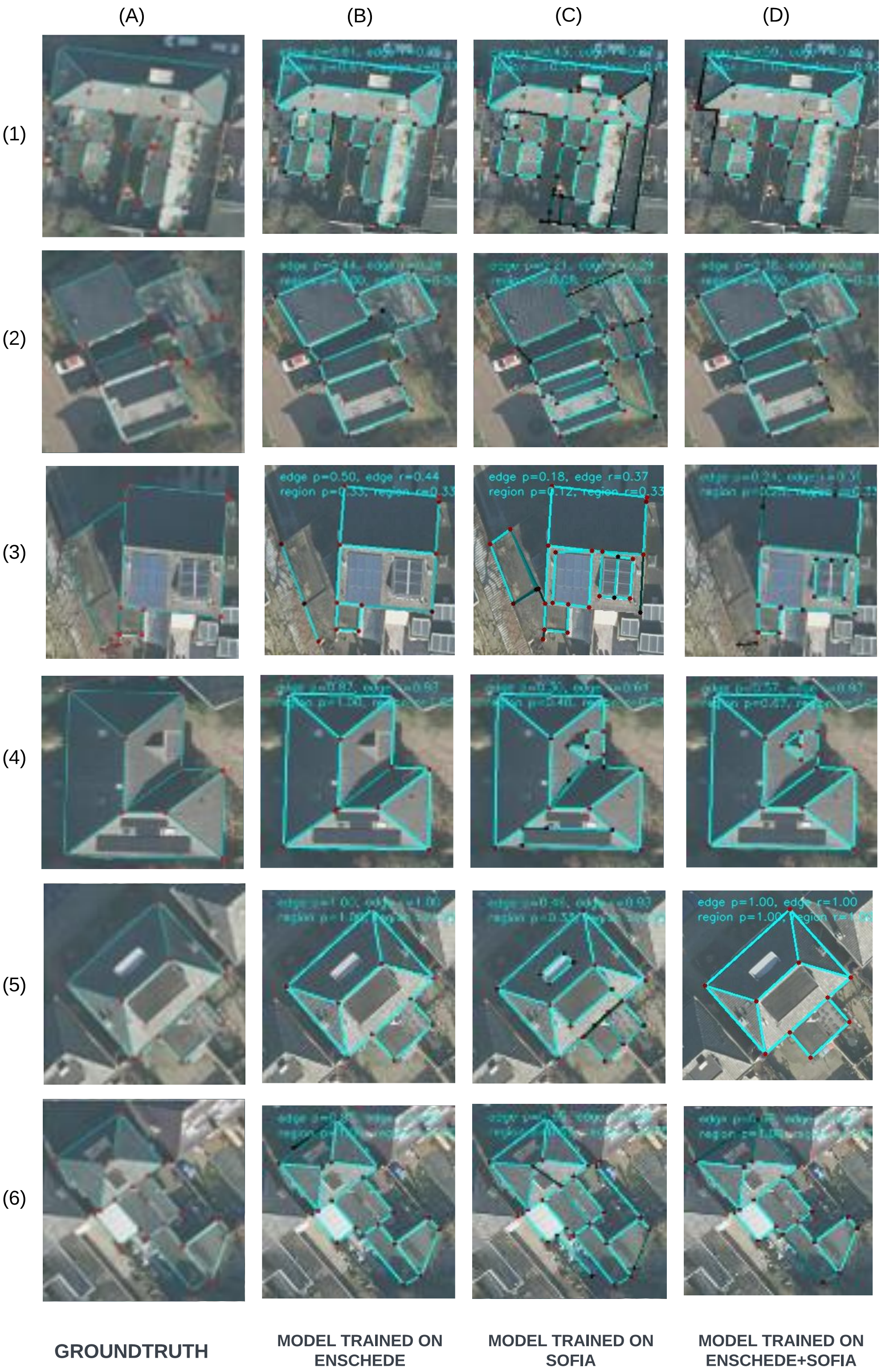

(A)
(B)
(C)
(D)
(1)
(2)
(3)
(4)
(5)
(6)
edge p=0.50, edge r=0.44
region p=0.33, region r=0.33
edge p=0.18, edge r=0.37
region p=0.12, region r=0.33
edge p=1.00, edge r=1.00
region p=1.00, region r=1.00
GROUNDTRUTH
MODEL TRAINED ON ENSCHEDE
MODEL TRAINED ON SOFIA
MODEL TRAINED ON ENSCHEDE+SOFIA

Figure 27. Qualitative evaluations on inner roof plane inference in the Enschede area, with image size 256x256.

Figure 27 provides a qualitative comparison of the delineation results obtained for the Stadsveld -'t Zwering testing area using three different models: the base model trained on the Enschede dataset, the model trained on the Sofia dataset, and the model trained on a combined dataset of Enschede and Sofia compared with the ground truth. The red points on the images represent the predicted corners. The edges are represented by: Cyan color, the most confident ones. Black color, the less confident edges.

Oude Markt, Enschede, The Netherlands

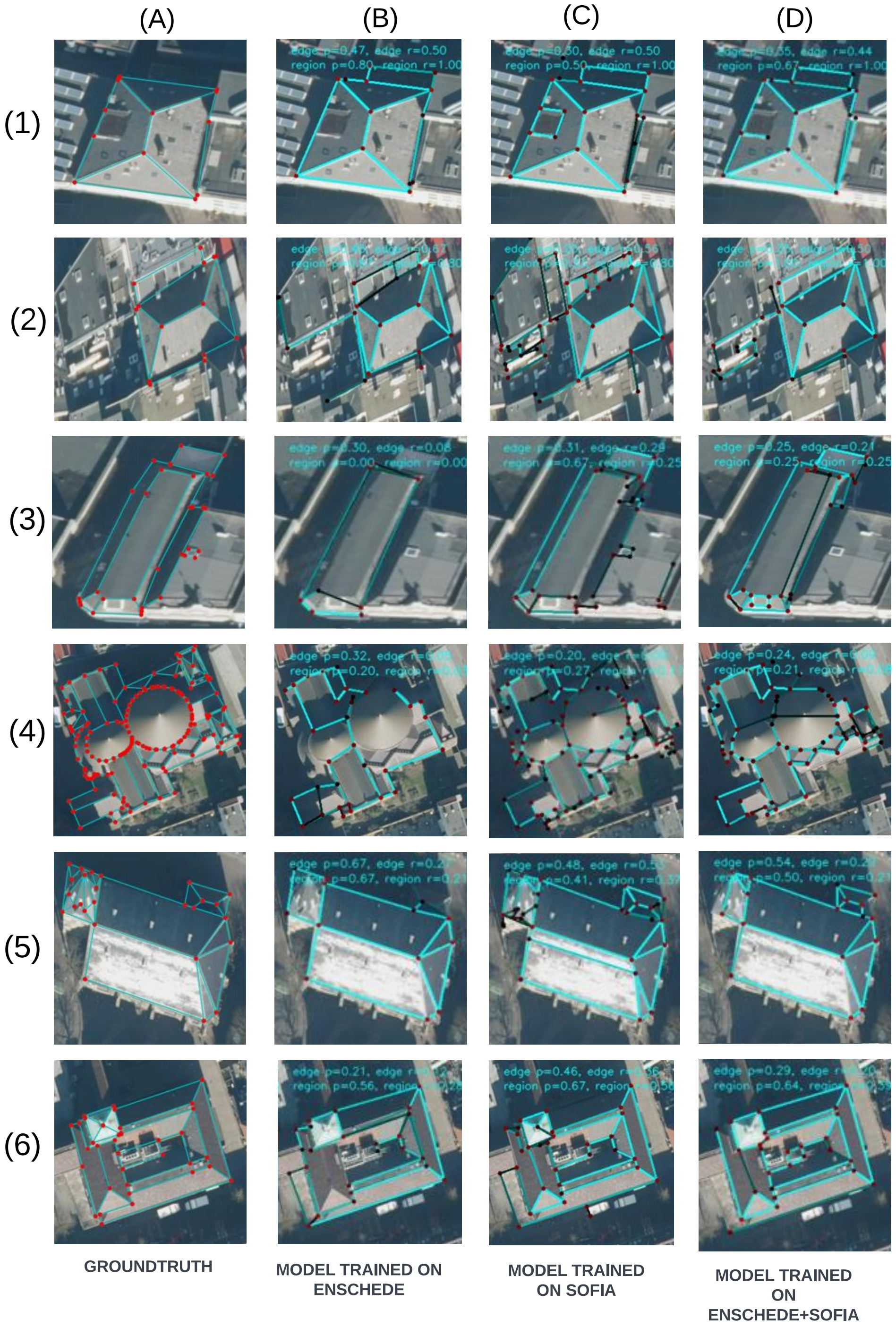

(A)
(B)
(C)
(D)
(1)
(2)
(3)
(4)
(5)
(6)
edge p=0.47, edge r=0.50
region p=0.80, region r=1.00
edge p=0.30, edge r=0.50
region p=0.50, region r=1.00
edge p=0.35, edge r=0.44
region p=0.67, region r=1.00
edge p=0.30, edge r=0.08
region p=0.00, region r=0.00
edge p=0.31, edge r=0.29
region p=0.67, region r=0.25
edge p=0.25, edge r=0.21
region p=0.25, region r=0.25
GROUNDTRUTH
MODEL TRAINED ON ENSCHEDE
MODEL TRAINED ON SOFIA
MODEL TRAINED ON ENSCHEDE+SOFIA

Figure 28. Qualitative evaluations on inner roof plane inference in the Oude Markt area, with image size 256x256.

Figure 28 provides a qualitative comparison of the inference results obtained from the Oude Markt, Enschede testing area using three different models the base model trained on the Enschede dataset, the model trained on the Sofia dataset, and the model trained on a combined dataset of Enschede and Sofia compared with the goundtruth.

Lozenets, Sofia, Bulgaria

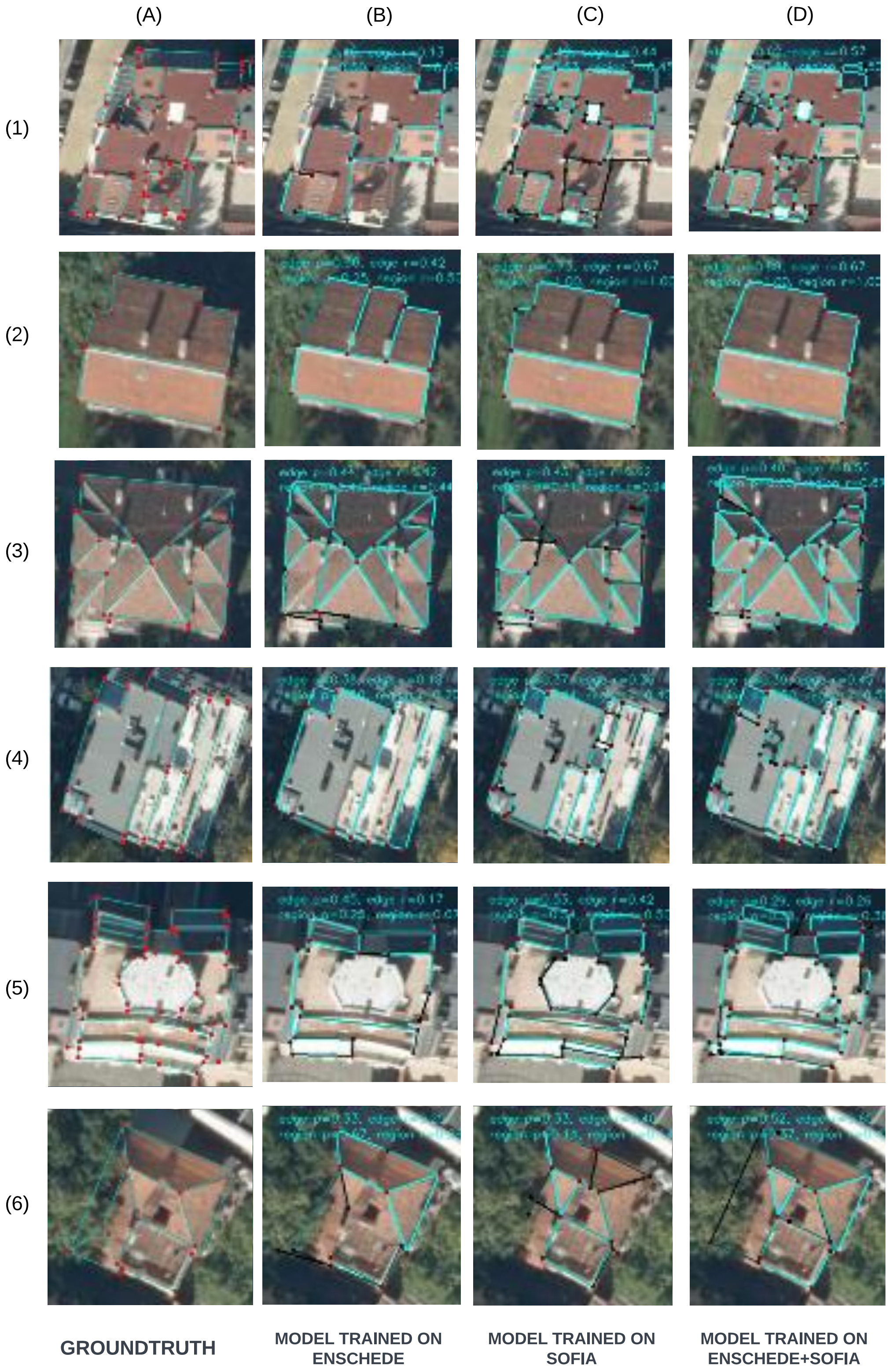


Figure 29. Qualitative evaluations on inner roof plane inference in the Sofia area, with image size 256x256.

Figure 29 provides a qualitative comparison of the inference results obtained from the Sofia testing area using three different models: the base model trained on the Enschede dataset against the groundtruth, the model trained on the Sofia dataset, and the model trained on a combined dataset of Enschede and Sofia.

This section shows that all three models could identify corners and edges on the buildings' image samples. For "simple roof structures" (Figure 28-row 4, Figure 29-row 1, Figure 30-row 2) residential buildings, the models do not face challenges in identifying corners and edges on the samples and delineating the complete roof structure. However, in samples with "complex roof structures" (Figure 28-row 1, Figure 29-row 4, Figure 30-row 6), like historic buildings, buildings with circular roof shapes, buildings occluded with trees, and buildings with high heights, the model is challenged, and some edges are identified with lower confidence (black edges).

Another point to consider is the number of regions that are delineated on the image samples. There are cases in which the model can delineate regions that were not present in the ground truth because the ground truth was delineated based on obtaining a rooftop with a LOD2. (Figure 28-row 4, Figure 29-row 1, Figure 30-row 1) This ability to delineate "extra regions" impacts the quantitative metrics as these regions are considered false positives and decrease the scores on the metrics.

Moreover, even the quantitative metrics showed that for the Sofia testing dataset, the model trained on Sofia showed a slightly superior performance than the model trained on the combined dataset. The sample in Figure 30-row 6 shows the capacity of the model trained on the combined dataset to retrieve roof elements under the vegetation, which was not possible by the other models.

4.3. Vectorization

4.3.1. Quantitative analysis

Table 6 shows the evaluation metric for the vectorization stage of the workflow. Quantitatively evaluates model performance for the various defined test dataset sets. This metric addressed the model's performance for delineating the building's inner roof plane and subsequently vectorization into a polygon vector format. Values in bold correspond to the model with the highest IoU between the compared models for the respective testing dataset.

Table 6. Quantitative evaluations on building inner roof plane reconstruction on the different testing sets.

TESTING AREA	MODEL	IoU
Stadsveld-'t zwering, Enschede, The netherlands	MODEL TRAINED ON ENSCHEDE DATASET	**0.82**
	MODEL TRAINED ON THE COMBINED DATASET (EN-SCHEDE + SOFIA)	0.80
Oude markt, enschede, The netherlands	MODEL TRAINED ON ENSCHEDE DATASET	0.66
	MODEL TRAINED ON THE COMBINED DATASET (EN-SCHEDE + SOFIA)	**0.82**
Lozenets, Sofia, Bulgaria	MODEL TRAINED ON SOFIA DATASET	**0.71**
	MODEL TRAINED ON COMBINED DATASET (EN-SCHEDE + SOFIA)	0.70

The conducted experiments suggest a similar behavior with the building's inner roof plane delineation phase in which the model trained with the combined training dataset did not improve the models' performance for the Stadsveld - 't Zwering, and Lozenets testing area. However, for the Oude Markt testing area, the combined model performs better than the model trained on Stadsveld - 't Zwering.

Notice that for the Oude Markt testing area, the performance of the model trained on Lozenets was not computed because the buildings' inner roof planes delineation metric showed the poorest performance in delineating the building's inner roof plane.

4.3.2. Qualitative analysis

The vectorization section was performed using the GIS software ArcGIS. The GIS tool "from feature to polygon". The tool creates polygons features from delineated closed boundaries. If there is a gap and the polygon is not closed, the tool will not draw the polygon. The vectorization process executed through ArcGIS took the boundaries of the inner roof planes obtained from the previous phase. As a result, the closed buildings' inner roof planes are obtained.

Stadsveld-'t Zwering, Enschede, The Netherlands

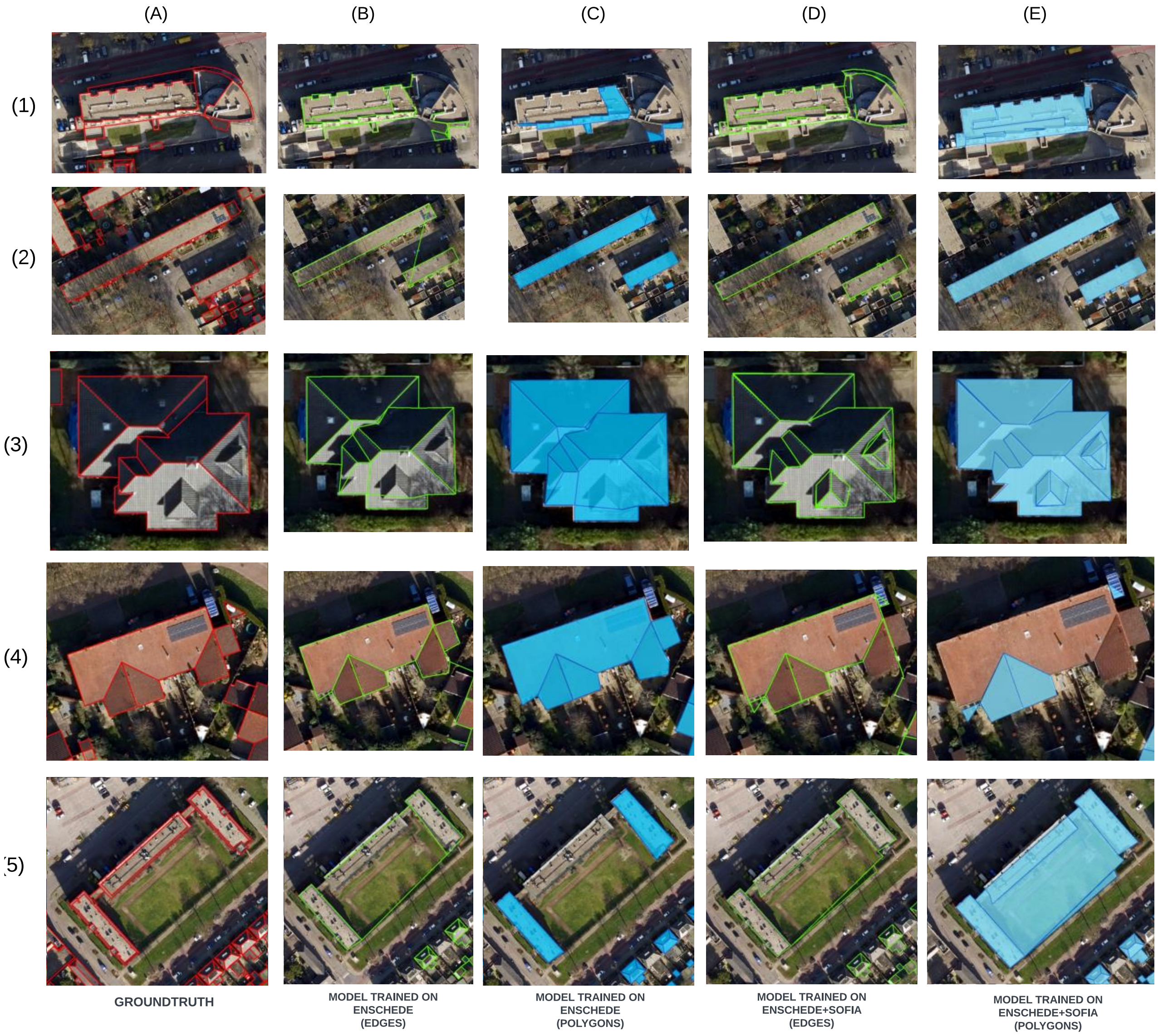

(A)
(B)
(C)
(D)
(E)
(1)
(2)
(3)
(4)
(5)
GROUNDTRUTH
MODEL TRAINED ON ENSCHEDE (EDGES)
MODEL TRAINED ON ENSCHEDE (POLYGONS)
MODEL TRAINED ON ENSCHEDE+SOFIA (EDGES)
MODEL TRAINED ON ENSCHEDE+SOFIA (POLYGONS)

Figure 30. Comparison of the obtained vector polygon planes after applying the different model- Stadsveld -'t Zwering.

Figure 30 provides a qualitative comparison of the vectorizations results obtained for the Stadsveld - 't Zwering testing area using the two best models from the building inner roof plane stage: the base model trained on the Enschede dataset and the model trained on a combined dataset of Enschede and Sofia, for large complex structures such as Figure31-row1 and Figure31-row5, the model trained on the Enschede dataset faces challenges in delineating all the edges of the building's inner roof planes, which is later impacted the vectorization results.

Oude Markt, Enschede, The Netherlands

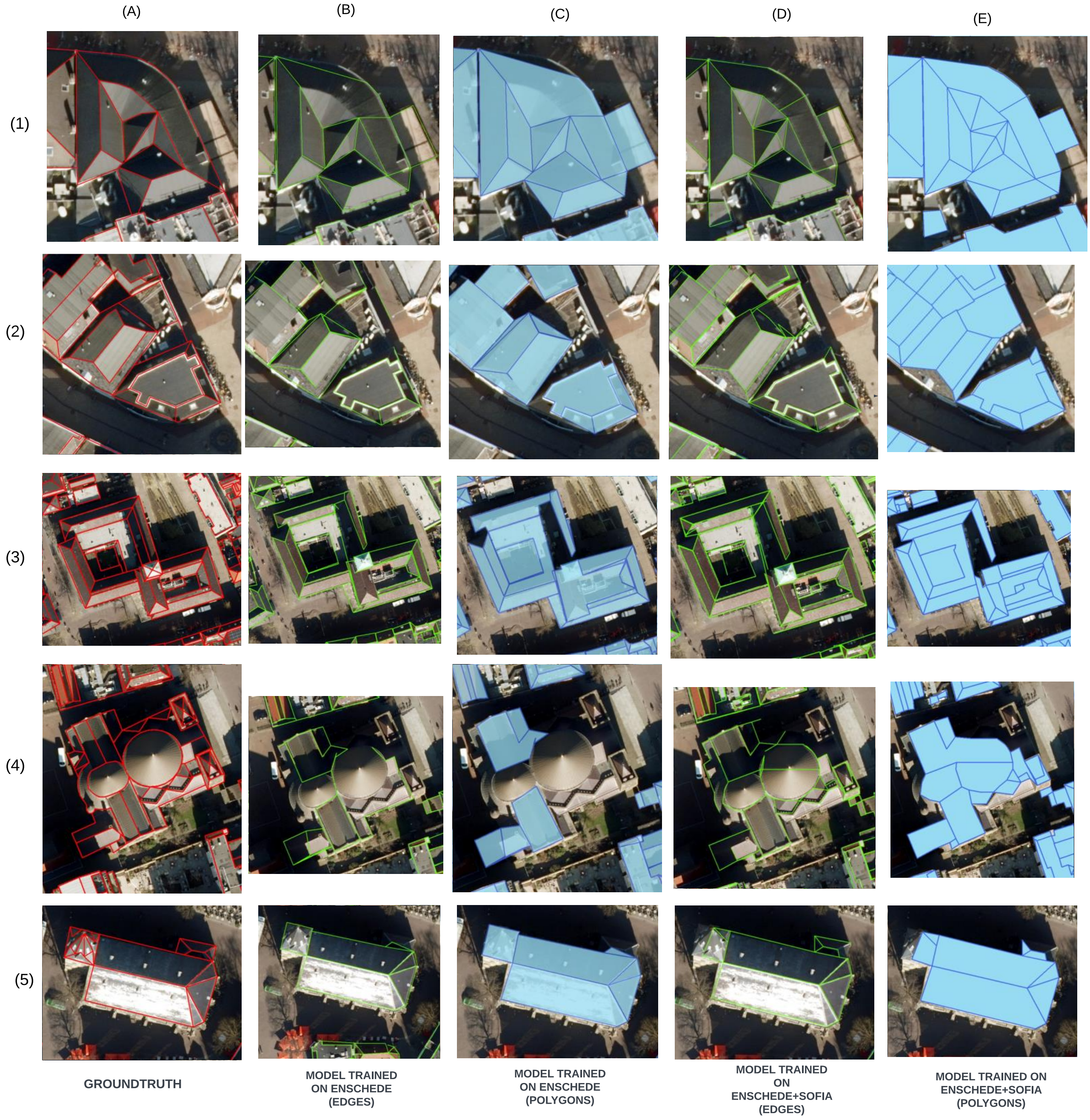

(A)
(B)
(C)
(D)
(E)
(1)
(2)
(3)
(4)
(5)
GROUNDTRUTH
MODEL TRAINED ON ENSCHEDE (EDGES)
MODEL TRAINED ON ENSCHEDE (POLYGONS)
MODEL TRAINED ON ENSCHEDE+SOFIA (EDGES)
MODEL TRAINED ON ENSCHEDE+SOFIA (POLYGONS)

Figure 31. Comparison of the obtained vector polygon planes after applying the different model- Oude Markt.

Figure 31 provides a qualitative comparison of the vectorizations results obtained for the Oude Markt testing area using the two best models from the building inner roof plane stage: the base model trained on the Enschede dataset and the model trained on a combined dataset of Enschede and Sofia. As is shown in all the samples, the model trained on the combined dataset can delineate the building's inner roof plane structures in more detail than the model trained in the Enschede dataset. They show that the model trained on the combined dataset performs better for delineating roof structures with great details for this testing dataset.

Lozenets, Sofia, Bulgaria

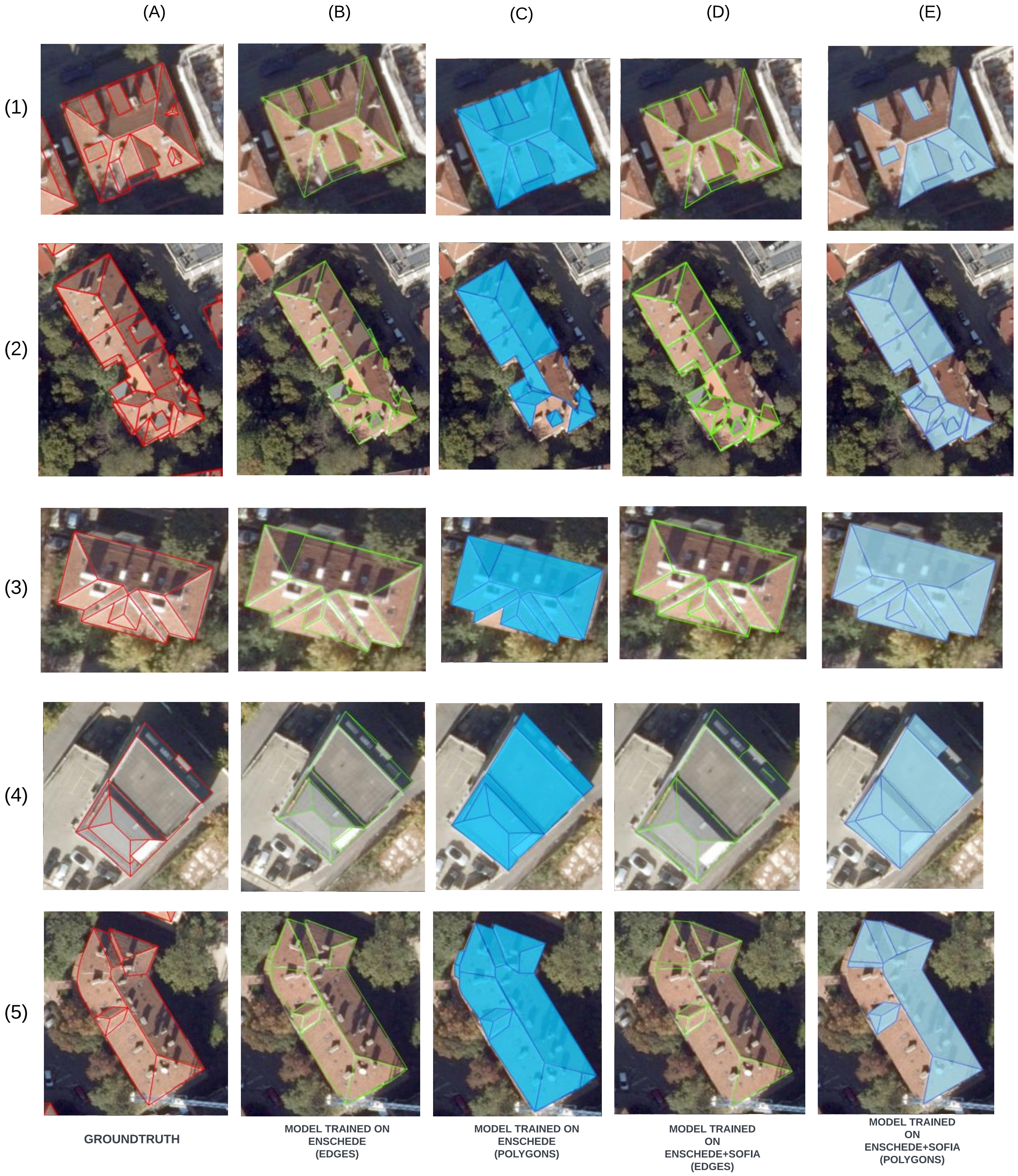

(A)
(B)
(C)
(D)
(E)
(1)
(2)
(3)
(4)
(5)
GROUNDTRUTH
MODEL TRAINED ON ENSCHEDE (EDGES)
MODEL TRAINED ON ENSCHEDE (POLYGONS)
MODEL TRAINED ON ENSCHEDE+SOFIA (EDGES)
MODEL TRAINED ON ENSCHEDE+SOFIA (POLYGONS)

Figure 32. Comparison of the obtained vector polygon planes after applying the different model-Lozenets.

Figure 32 provides a qualitative comparison of the vectorizations results obtained for the Lozenets testing area using the two best models from the building inner roof plane stage: the base model trained on the Sofia dataset and the model trained on a combined dataset of Enschede and Sofia.

As shown in the quantitative metrics, there were no significant differences between the best models in the IoU metrics for the cases of Stadsveld-'t Zwering and Lozenets. However, in Oude Markt's case, the quantitative metrics show a big difference, reflected in the qualitative evaluation between the numbers of predicted building inner roof planes.

4.4. 3D Modelling

4.4.1. Quantitative analysis

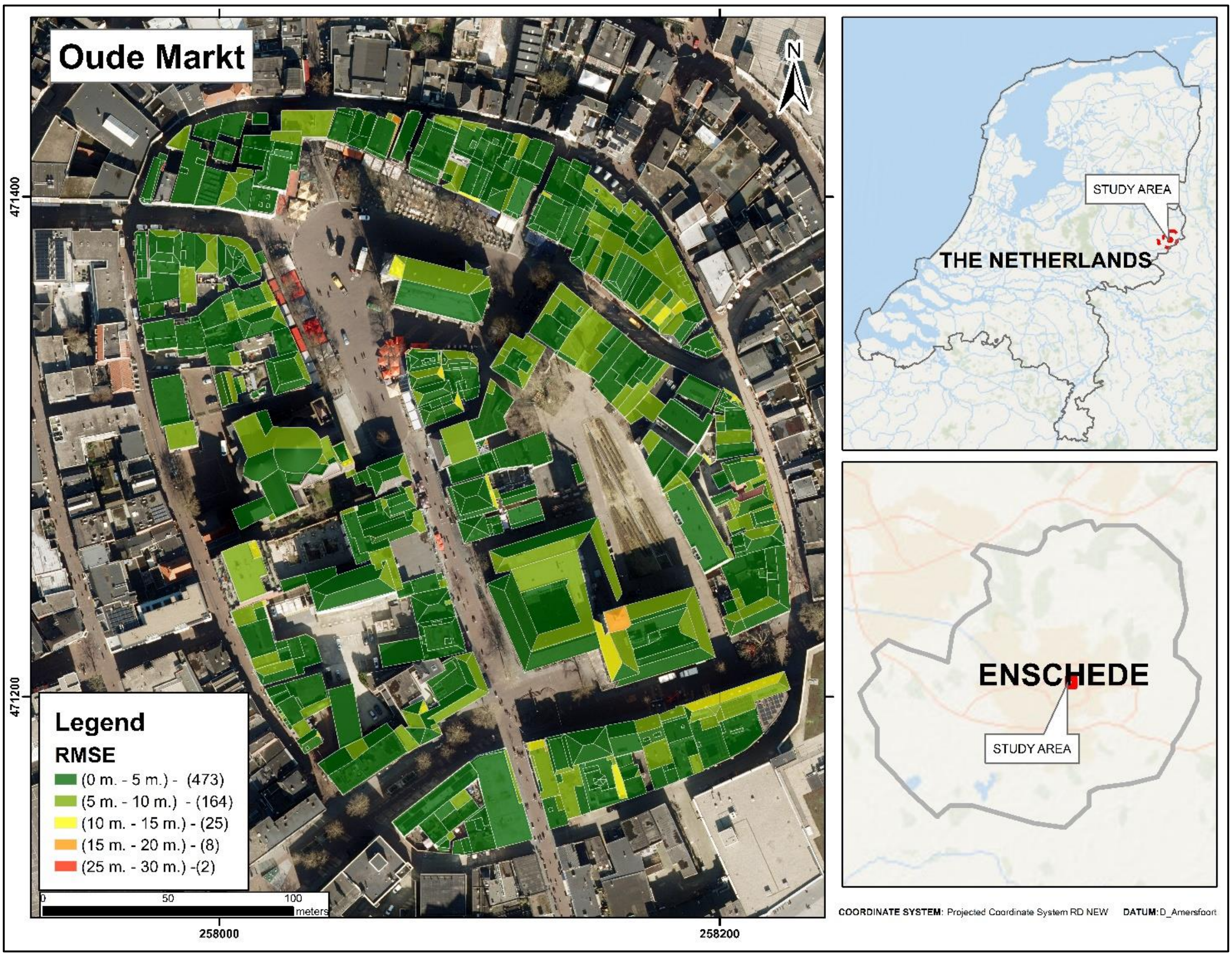


Figure 33. RMSE computed in the 3D model of the Oude Markt, Enschede, The Netherlands.

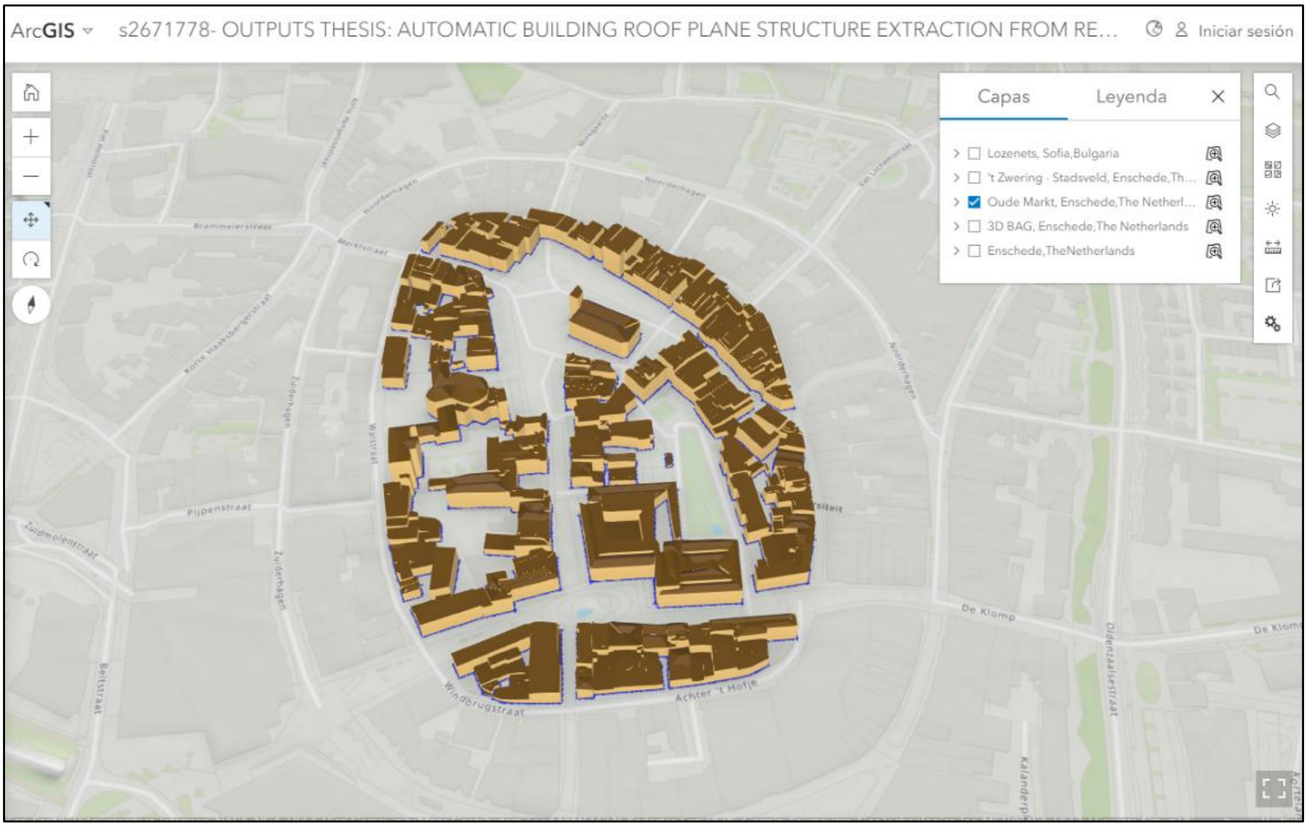
Figure 33 shows the calculation of the RMSE for the building's inner roof planes of the generated LOD2 3D city model for the Oude Markt area. The quantitative evaluations show that 70% of the total building's inner roof planes show discrepancies between 0m-5m, 24% of the total building's inner roof planes show discrepancies between 5m-10m, 4% of the total building's inner roof planes show discrepancies between 10m-15m, a slightly higher value than 1% of the total building's inner roof planes show discrepancies between 15m-20m, and less than

1% of the total building's inner roof planes show discrepancies between 25m-30m between the DMS and the generated LOD2 3D City model.

In terms of area, the quantitative evaluations show that 72% of the total building's inner roof planes show discrepancies between 0m-5m, 26% of the total building's inner roof planes show discrepancies between 5m-10m, 1.99% of the total building's inner roof planes show discrepancies between 10m-15m, and the rest 0.01% the building's inner roof planes that show discrepancies between 15m-30m between the DMS and the generated LOD2 3D City model.

4.4.2. Qualitative analysis

Using ArcGIS, the LOD2 3D city model for the Oude Markt area was generated using the obtained polygons from the model trained on the combined dataset. To provide a more user-friendly and interactive visual representation of the generated 3D city model, a webmap was developed. Interested readers can access the webmap via the following link: https://arcg.is/1raWvS0



ArcGIS s2671778- OUTPUTS THESIS: AUTOMATIC BUILDING ROOF PLANE STRUCTURE EXTRACTION FROM RE... Iniciar sesión

Capas Leyenda

- □ Lozenets, Sofia,Bulgaria
- □ 't Zwering - Stadsveld, Enschede,Th...
- ☑ Oude Markt, Enschede,The Netherl...
- □ 3D BAG, Enschede,The Netherlands
- □ Enschede,TheNetherlands

Figure 34. Webmap showing the generated LOD2 3D model.

Figure 34 shows an interface of the generated webmap to explore the results of the LOD2 3D city model. From the created LOD2 3D City model, 566 out of 672 building inner roof planes were modeled with a flat roof form, 101 building inner roof planes were modeled with a Gable roof form, and 5 building inner roof planes were modeled with a Hip roof form.

A visual analysis of the generated LOD2 3D model can be performed by comparing it visually against the 3D Building Attribute Geometry (3D BAG). The 3D BAG is a 3D dataset of all buildings in the Netherlands. Initiated by the Delft University of Technology, the 3D BAG integrates data from the Dutch national building registry (BAG) and the national height dataset (AHN) to create 3D representations of all the buildings at different LODs (Peters et al., 2022).

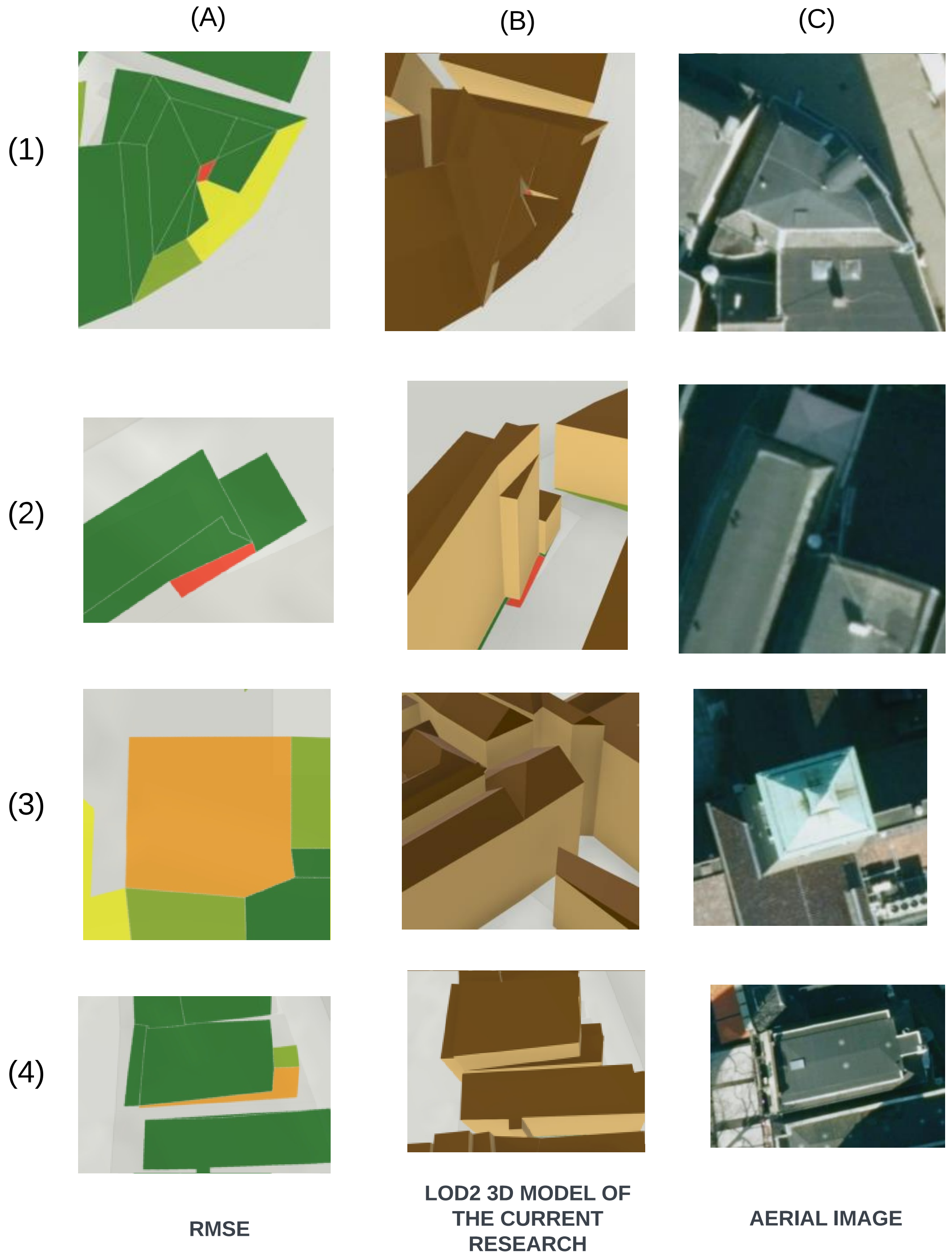


Figure 35. Comparative between the calculated RMSE, the generated LOD2 3D model, and the aerial image of the building sample.

Figure 35 illustrates a visual comparison of some samples comparing the calculated RMSE, the generated LOD2 3D model, and the aerial imagery from the corresponding sample. The samples with the highest RMSE are the LOD2 3D models with the highest height discrepancies between the accurate and modeled heights, as shown in all the building samples presented in Figure 35.

Analyzing the discrepancies between the created LOD2 3D model and DSM shows that some errors occurred when the buildings inner roof planes of the building used to generate the LOD2 model were not accurately aligned with the position on the DSM, which caused these buildings' inner roofs planes were not modeled correctly.

4.5. Summary

This chapter provided quantitative and qualitative comparisons across our inference, vectorization, and 3D modelling phases. Models trained on a combined dataset from Enschede and Sofia demonstrated superior performance in capturing the details of inner roof planes across all test areas. However, when testing our model in the Sofia area, a finding suggest that the Enschede dataset did not enhance the model's performance. This is likely due to the relative simplicity of roof structures in the Enschede dataset compared to the more complex structures found in Sofia's building samples. The generation of a 3D model is contingent upon the availability of a Digital Elevation Model, which is a limitation to consider in future work.

All the generated outputs from all the performed experiments can be revised on the created webmap, via the following link: https://arcg.is/1raWvS0

5. DISCUSSION

5.1. Reflection on the performance of the created framework

The current research has scaled up the application of the work developed by Chen et al. (2022) HEAT of planar graph reconstruction from 2D raster images, which primarily involves the detection and classification of primitives on images to yield corners and edges (Chen et al., 2019), ultimately resulting in a final planar graph (F. Zhang et al., 2021). For which a framework based on HEAT was developed.

In the previous results section, the performance of the three trained models has been compared quantitatively and qualitatively. The combined model showed similar performance and, in some aspects, slightly better performance than the models trained in their respective areas. However, after vectorization and testing the models in Oude Markt area, a different area from the trained area, the differences between models were notable, as the original training area in Enschede was based on residential roof structures, the Oude Markt area is comprised of different historical, public and commercial buildings with topology similarities as the buildings samples from the Sofia training area.

Unlike traditional segmentation models that cannot detect straight edges easily (Hossain & Chen, 2022), our proposed framework is trained to detect corners, classify and draw the geometry relationships between them, obtaining remarkable achievements such as the model trained on a combined dataset could detect building inner roof planes, even when vegetation obscures edges, and by detecting the end corners of the building inner roof plane (Figure 29-6).

.

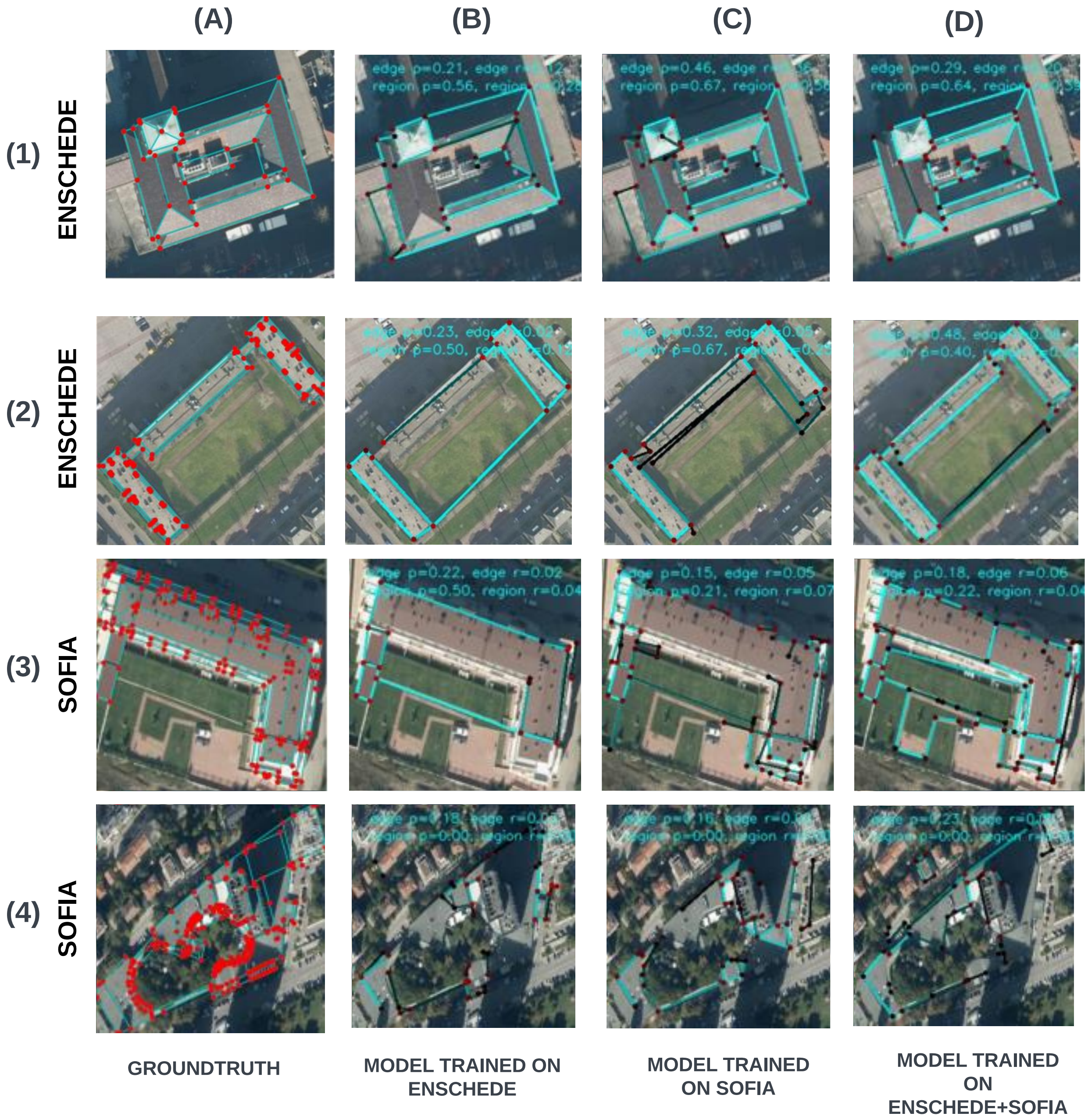


Figure 36. Limitations of the method in distinguishing between different types of flat surfaces.

Despite the robust performance of our approach in delineating inner roof plane structures, there are a few scenarios where it shows limitations. A noteworthy limitation arises when the image samples contain large areas of ground. When the trained model encounters these areas, it predicts the ground as a plane. This misinterpretation can be attributed to the model's difficulty distinguishing between different types of flat surfaces, such as the ground and the roofs of the buildings, especially when the ground occupies a significant portion of the image and the shape of the building (Figure 36). This issue could lead to inaccuracies in the final 3D model, as the ground is incorrectly represented as part of the building roof structure. In addition, this misinterpretation can lead to significant errors that will be accounted for to measure the model performance.

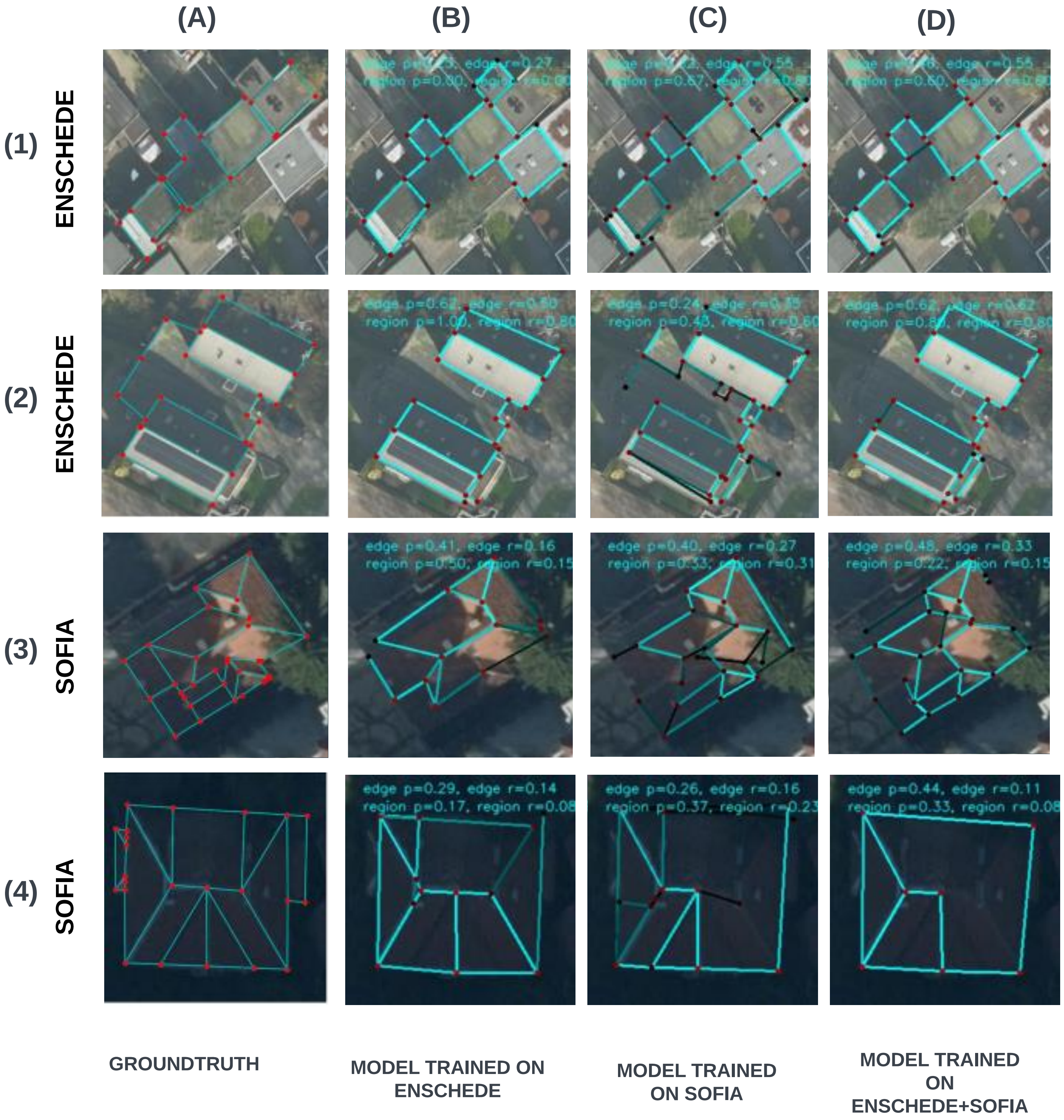

(A)
(B)
(C)
(D)
(1)
ENSCHEDE
(2)
ENSCHEDE
(3)
SOFIA
(4)
SOFIA
edge p=0.62, edge r=0.60
region p=1.00, region r=0.80
edge p=0.24, edge r=0.35
region p=0.43, region r=0.60
edge p=0.62, edge r=0.62
region p=0.80, region r=0.80
edge p=0.41, edge r=0.16
region p=0.50, region r=0.15
edge p=0.40, edge r=0.27
region p=0.33, region r=0.31
edge p=0.48, edge r=0.33
region p=0.22, region r=0.15
edge p=0.29, edge r=0.14
region p=0.17, region r=0.08
edge p=0.26, edge r=0.16
region p=0.37, region r=0.23
edge p=0.44, edge r=0.11
region p=0.33, region r=0.08
GROUNDTRUTH
MODEL TRAINED ON ENSCHEDE
MODEL TRAINED ON SOFIA
MODEL TRAINED ON ENSCHEDE+SOFIA

Figure 37. Limitations of the method when inferring inner roof planes in image samples with multiple buildings on the occlusion image.

Another drawback appears when inferring inner planes to image samples that contain multiple buildings or deal with occlusions. The model can struggle to accurately differentiate and isolate the planes associated with individual buildings, mainly when these structures are closely situated or their planes intersect in the image. Consequently, the model's capacity to accurately reconstruct each building's inner roof planes can be compromised. Furthermore, occlusions present another challenge. The model's performance can be hampered when parts of the building or its roof plane are obscured in the image, leading to incomplete or inaccurate reconstructions, as it showed in Figure 37.

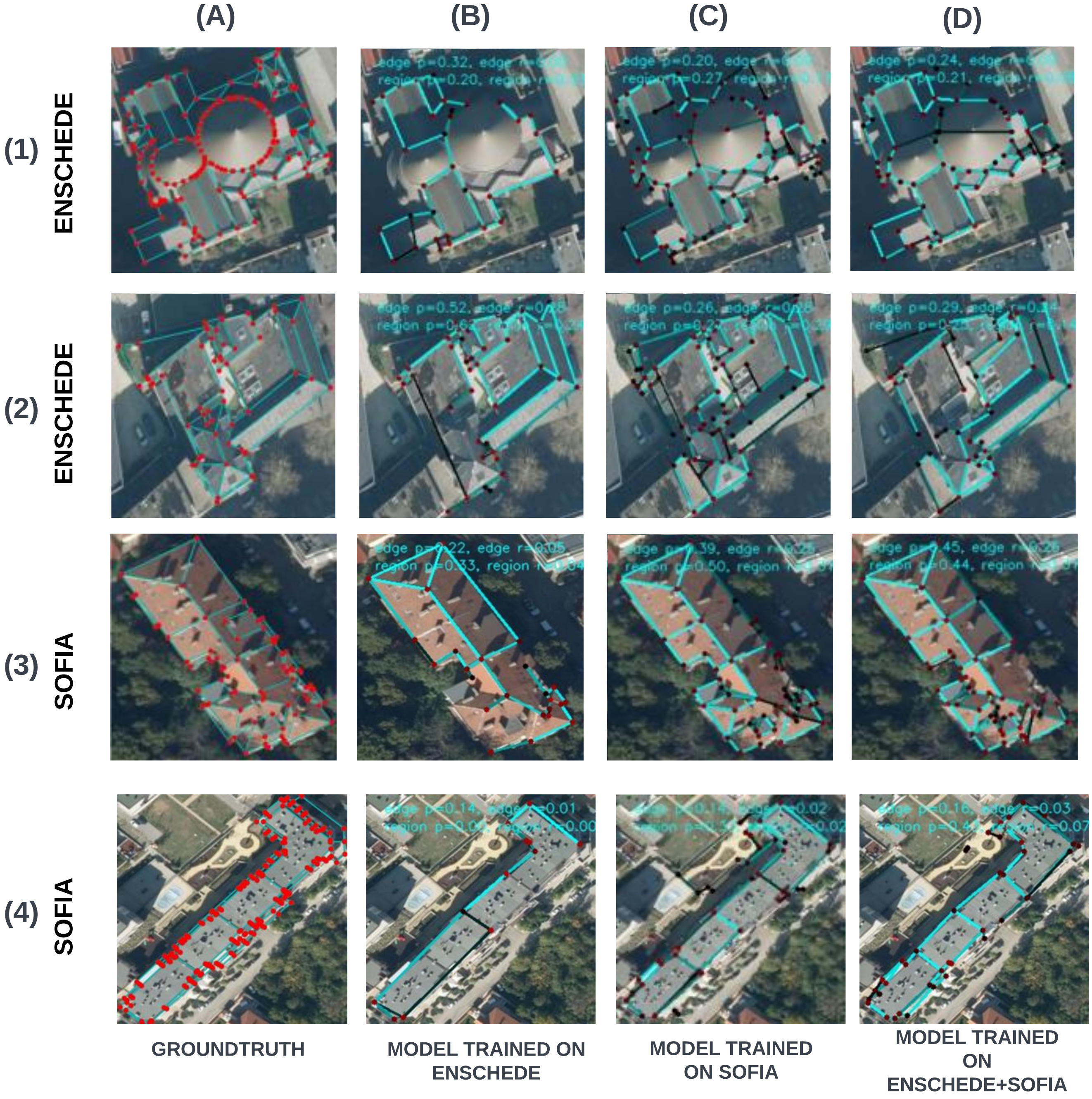


Figure 38. Limitations of the method when inferring inner roof planes with many corners on its structure.

Another complication appears when the model attempts to infer inner planes on image samples with complex roof graphs containing many corners to generate circular roof planes or complex roof structures. The model struggles to accurately interpret and reconstruct the delicate geometrical aspects of these complex roof structures. Specifically, it often fails to correctly identify and process planes with numerous corners in its structure, leading to an inaccurate generation of circular roof planes. This limitation can significantly affect the reconstructed roof plane's quality, particularly in roofs with circular or curved features. As the complexity and the number of corners in the roof graph increase, the model's performance tends to decrease, indicating a potential area of weakness in handling geometric complexity, as shown in Figure 38.

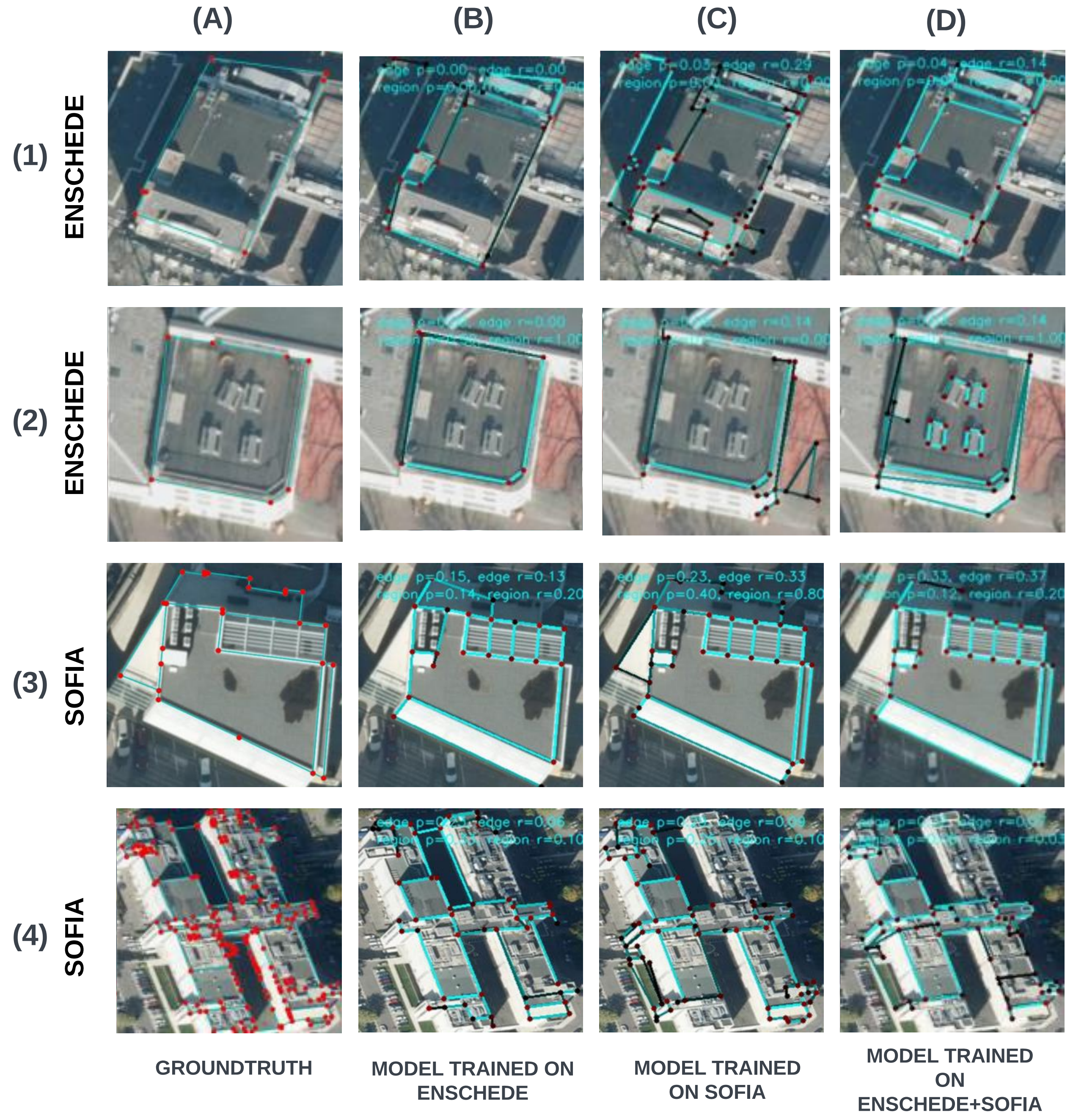


Figure 39. Limitations of the method when some facades are inferred as inner roof planes.

A significant limitation emerges when the model misinterprets the facades of big buildings as inner roof planes. This confusion can lead to inaccuracies in the delineation of roof planes, as the vertical surfaces of the facades are incorrectly incorporated into the roof plane structure. This misinterpretation can be especially problematic for buildings with complex or non-standard designs where the distinction between facades and roof planes may not be visible from the image data (Figure 39). This flaw could be attributed to the model's difficulty distinguishing between different structural elements in large, packed, complex building structures.

In terms of the LOD2 3D model. This research generated a LOD2 3D model using ArcGIS software tools. Despite this achievement, several potential areas for improvement were identified, such as refining the quality of input data, addressing topological errors in the buildings' inner roof planes, exploring open-source software alternatives, and incorporating supplementary data.

To qualitatively evaluate the generated LOD2 3D model, A visual comparison of the developed 3D model for the current research with the widely accessible Google Earth 3D model and the 3D BAG model. The 3D BAG model, initiated by Delft University of Technology, offers a 3D dataset of all buildings in the Netherlands. It merges data from the Dutch national building registry (BAG) and the national height dataset (AHN), presenting a robust standard for comparison (Peters et al., 2022), which could present an alternative path to evaluating the generated LOD 2 3D model against pre-established 3D models.

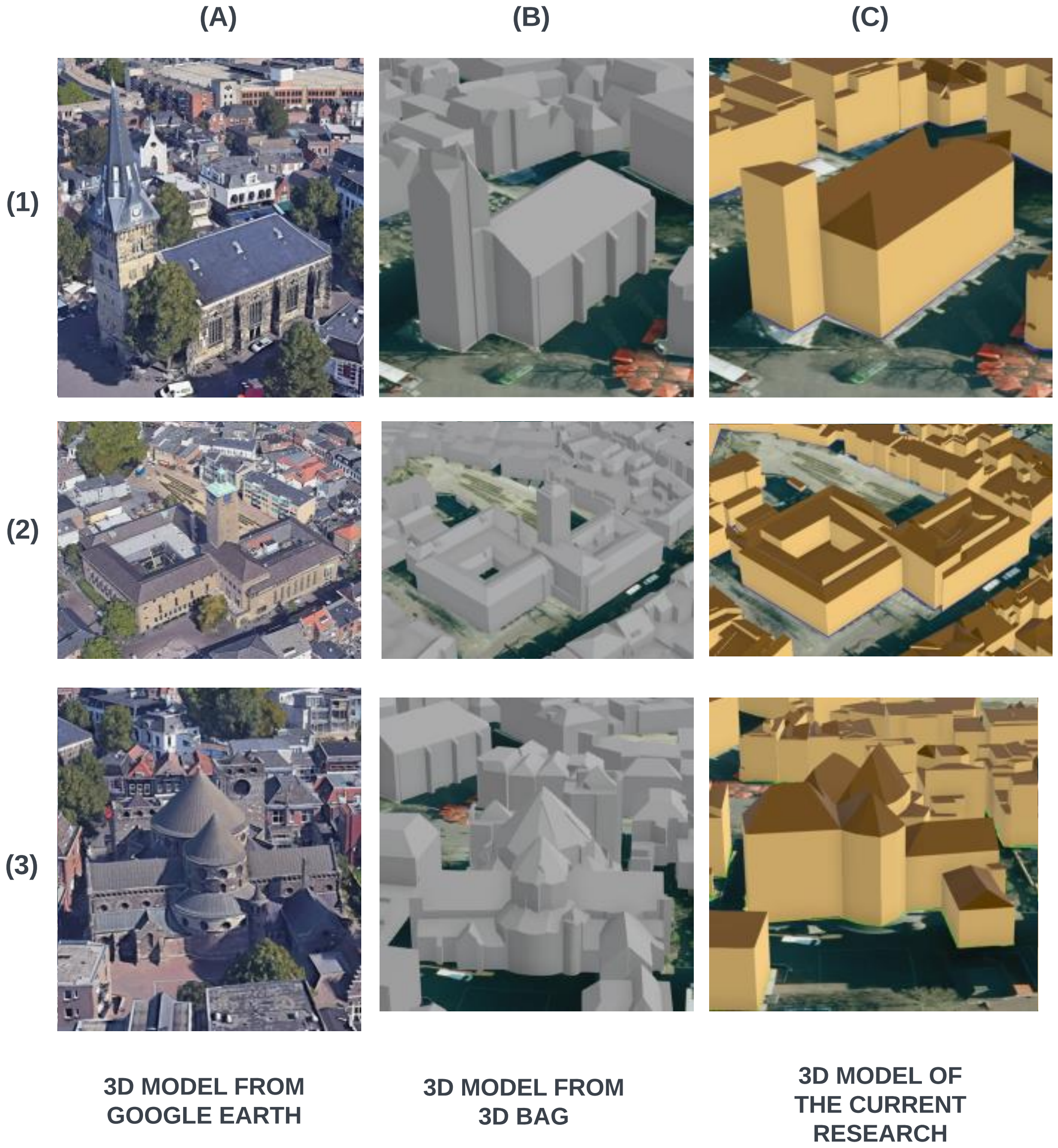


Figure 40. Limitations of the implemented 3D approach.

Figure 40 compares a highly detailed 3D model from Google Earth, the 3DBAG 3D model, and the 3D model of current research. Quantitative validation metrics of our 3D model against other established 3D models, such as the 3D BAG model, could enhance the model assessment and provide a robust validation platform. The goal would be to minimize the discrepancy between the reconstructed model and its real-world counterpart or similar 3D models, such as 3DBAG, which can be used as a reference for 3D models, thereby increasing the practical utility and accuracy of the 3D models generated by our approach.

5.2. Applicability of the created approach and further improvements

The applicability of the presented approach is quite broad in urban applications. Here are a few potential areas of application:

- In urban planning and development, the proposed approach facilitates the creation of detailed 3D models of urban environments (Lafarge et al., 2010).
- To enhance the current 3D models based on LIDAR, as one of the challenges of 3D model reconstruction based on point cloud segmentation is roof segmentation, the presented approach facilitates the task of the roof partitioning of the different roof planes in a roof structure (Peters et al., 2022).
- 3D cadastre, LOD2 3D models offer a much more accurate and practical representation of property rights than traditional 2D cadastre systems (Paulsson, 2007).
- In natural disasters, the proposed approach can be applied to detect building changes. Moreover, accurately generating LOD2 3D models of buildings can help emergency responders plan and execute rescue operations more effectively (Kolbe, 2009).
- Solar potential analysis, the detailed roof structure information can be used to determine the solar potential of buildings, which could support sustainable energy planning and promote solar potential analysis studies(Kausika et al., 2016).
- In telecommunication, generated LOD2 3D models offer a platform to evaluate suitable locations for placing antennas and predicting signal coverage (Becker et al., 2011).

While the approach has shown promising results, further refinement and validation may be necessary for specific applications at bigger scales.

Considering the previously discussed applicability. The approach can be improved using the following recommendations, which can be seen as suggestions for improvement for further studies:

1) Collecting multiple and varied building samples for training data can improve the model's performance (Shorten & Khoshgoftaar, 2019).
2) Use a sampling strategy to select buildings samples and not take buildings samples from an area not representative of a whole city (Du et al., 2015).
3) Adapting the model to accept images with more channels and not just RGB could potentially enhance the model's performance. This additional channel could contain supplementary information not present in the RGB images (Kenzhebay, 2022).

As a basis, a pre-trained model from buildings samples from Paris, Las Vegas, and Atlanta from the SpaceNet Challenge (Van Etten et al., 2018) was used. The generated models were trained and tested in a typical dutch residential neighborhood and Bulgarian complex areas with various roof types.

As discussed, the results may vary when the same approach is used in different geographical locations. Our trained models with the combined dataset of The Netherlands and Bulgaria can have good spatial transferability to other areas with a similar topology architecture design. However, the trained model may produce poor results if the new test area has non-similar buildings with complicated topology roof structures (e.g., big cities with skyscrapers or different roof materials). As the model is flexible, performing a transfer learning to the model on a subset of building roofs in the selected new areas is possible. On the other hand, if most of the buildings in the test area have basic flat roof structures, the model will perform better because there will be no need to detect the inner rooflines and reconstruct complex building roof structures.

Future studies might focus on improving the HEAT model's capacity to distinguish between various surface types in light of the difficulties found during this research. In addition, the complexities encountered in multiple buildings and occlusions scenarios underscore the need for additional refinement. As well as future iterations of the model should aim to improve its recognition and processing of complex roof structures with multiple corners, thereby enhancing the accuracy and fidelity of the reconstructed roof planes. These issues could be tackled by incorporating an additional channel in the input images or refining the building samples in the training process. However, adding extra channels will also increase the complexity of the model and the computational resources required to train and run it, and it will require that the input images be more complex and require extra information that could be easy to access (Q. Zhang et al., 2018). Therefore, it is crucial to ensure that the benefits of adding an extra channel outweigh these additional costs.

In addition, as the model can perform planar building reconstruction building per building, its application to a complete is conditioned to access to the building footprint of each building, so its application in other tasks, such as slum delineation, is not possible, as slum areas area composed of areas where the built-up environment is very dense. With some modifications, the method could be used for building footprint delineation in areas where the cadastral system is parcel-based and lacks building footprint information.

5.3. Ethical considerations

In the current research project, some ethical considerations were taken into account. The use of deep learning for the automatic reconstruction of building inner roof planes introduces ethical considerations. One significant concern is privacy; extracting building roof structures from satellite or aerial images might infringe on individual privacy rights, given that these images, despite their resolution, can reveal details about residential properties (Ha et al., 2020). Any data collected this way must be handled cautiously and adhere to individual privacy rights to avoid unwanted access (Bae et al., 2018).

Further, the potential for bias in deep learning models is well-known. Suppose a model is trained on data from specific neighborhoods or building structures. In that case, it may not perform as well when applied to different areas or types of buildings, leading to potentially biased outcomes (Mehrabi et al., 2021)). The issue of transparency and accountability also comes into play. Due to the complexity of the created deep learning model, it might not be easy for the public in general to understand it, thus challenging to be accountable for their performance (Mazijn et al., 2022).

Lastly, military considerations as reflected by Chen et al. (2022). In the original HEAT work, the development of outdoor reconstruction frameworks like those presented in the current research could inadvertently promote the use of satellite images for military purposes.

.

6. CONCLUSIONS

This study proposed a multi-phase framework that employs HEAT (Holistic Edge Attention Transformer), a deep learning model built on a transformer-based neural architecture to automatically extract building inner roof planes, which can be considered prerequisites for generating LOD2 3D city models. The proposed approach performs planar architecture delineation of building outline and inner line roofline extraction, which can be generalized to different roof topologies. Furthermore, the obtained inner roofs planes outputs have been used to generate a LOD2 3D city model. Notably, the proposed framework excels in extracting regularized building inner roofs planes structures in vector format without extra post-processing steps, addressing a challenge encountered in some image segmentation methods.

Based on the quantitative assessment, models tailored to specific study areas successfully delineated building inner roofs planes structures in their areas. However, a model trained on a combined dataset from both study areas demonstrated slightly superior performance. In the case of the study area of Enschede, the base HEAT model trained with a combined dataset composed of building roof samples from the study areas in Enschede and Sofia showcased comparable performance (Regions: F-Score: 0.68) to the model trained exclusively on building inner roof planes samples from Enschede study area. For the case of the study area of Sofia, the model trained only with Sofia building inner roofs planes samples demonstrated comparable results performance (Regions: F-Score: 0.41) to the model trained on the combined dataset from Enschede and Sofia.

Once the inner roof planes were converted into vector polygon shapefile format, the performed tests showed that the model trained only with building samples from Enschede itself performed slightly better (IoU=0.82) than the model trained with a combined dataset composed of building roof samples from Enschede and Sofia, (IoU=0.80). A similar situation was seen in the Sofia study area, in which the model trained only with building samples from Sofia itself performed slightly better (IoU=0.71) than the model trained with a combined dataset composed of building roof samples from Enschede and Sofia (IoU=0.70). However, it still has limitations, such as topological errors like overlap and gaps between polygons, that will require further GIS post-processing to correct it.

As outlined in the discussion section, deep learning models could be sensitive to bias. Changing the area could affect the model's performance, as demonstrated by a subsequent experiment conducted in another area in Enschede. In the Oude Markt, for inferring inner roof planes, the model trained in a combined dataset of building roof samples from the study areas of Enschede and Sofia showed a superiority (Regions: F-Score: 0.43) against the model trained only with the building roof samples of the Enschede study area (Regions: F-Score: 0.37). The difference is even more notorious when the planes are converted into vector polygon shapefile format. The model trained with a combined dataset of roof samples from Enschede and Sofia showed better results (IoU=0.82) than the model trained only with roof samples from Enschede. (IoU=0.66).

According to the qualitative assessment, both models performed similarly, generating straighter edges and fewer undetected corners. Nevertheless, the model trained on a combined dataset could detect building inner roof planes, even when vegetation obscures edges (Figure 29-6D).

The developed framework performed well in delineating and extracting buildings' inner roof planes. However, it has limitations, such as failing to predict all corners in a complex roof derived into roof structures with missing building inner roof plane tropology (Figure 30-5, Figure 31-4, Figure 32-3). The obtained output is a structure composed of only edges and how these edges are connected. Therefore, post-processing is still required to convert the planar graph structure into inner roof planes.

This study has demonstrated the feasibility of creating a 3D city model with a LOD2 is feasible by integrating the building's inner roof planes with DSM, DTM, nDSM. The generated LOD2 3D model offers an opportunity to be improved. The outputs generated by the proposed combinations of different remote sensing datasets with GIS and deep learning techniques confirm our strategy's viability and create new directions for future study and growth in urban mapping and 3D city modelling.

6.1. Answers to research questions.

SO 1: To acquire knowledge in planar graph reconstruction from 2D raster images

1. **What is the process for planar graph reconstruction?**

Planar graph reconstruction from a 2D raster image takes images with an artificial structure, producing CAD-level reconstructions with corners and edges, a typical family of this tasks outdoor architecture building reconstruction from satellite or high-resolution images (Nauata & Furukawa, 2020). The typical process consists of first a primitive detection on the images to obtain corners and edges, then classifying the obtained corners and edges based on their geometrical relationship to derive the final planar graph finally (Chen et al., 2019).

2. **How to apply planar graph reconstruction for building roof plane structures extraction?**

Building roof plane structure extraction corresponds to an outdoor architecture reconstruction process. In this context, the generated outputs of a planar graph reconstruction task applied to a building in a 2D raster image represent the roof structure, where corners and edges form each component form each roof component. (Chen et al., 2022). Selecting and developing a framework around a planar graph reconstruction method makes extracting the roof plane structure feasible.

SO 2: To prepare the dataset for the further deep learning-based approach

1. **What dataset and resources are needed to implement the selected approach?**

A 2D RGB image (0.08 cm for Enschede and 0.10 cm. for Sofia), the buildings footprints of the buildings, building inner roof planes annotations, and defined workflows combining GIS and the deep learning methodology.

2. **What further data processing is needed?**

Correct mismatches between the 2D RGB image, the building footprints, and the building's inner roof planes, and avoid errors in digitizing interior rooflines and outlines.

SO 3: To design a deep learning-based framework to extract building roof plane structures in vector format from aerial images

1. **How can the selected deep learning approach for planar graph reconstruction be adapted to extract building roof plane structures?**

Once the deep learning approach is selected, the next step is to build an end-to-end framework. That adapts the available input dataset to the input requirements of the selected deep learning approach and converts the obtained outputs from the deep learning approach to the desired output format.

HEAT (Chen et al., 2022) was selected as the deep learning approach for the current research. Our dataset, composed of 2D RGB images, the buildings footprints of the buildings, and building inner roof planes, was formatted to the datasets requirements of the HEAT model and used to create a dataset for training, validation, and testing.

2. How to apply the developed framework in two different selected study areas?

To apply the developed framework in two study areas is suggested to follow the developed methodology in this research:

- Data pre-processing: Gather the necessary data for each study area, 2D raster images representing the study areas' such as high-resolution aerial images, a dataset of building footprints, and its corresponding inner roof planes, to generate datasets for training, validation, and testing.
- Model Training: Train the deep learning HEAT model using the preprocessed data for each study area. The model would learn to infer and delineate the planar graphs based on the given roof structures of the training dataset.
- Inferring: Use the developed, trained model to delineate building inner roof planes on the testing dataset.
- Post-processing: Convert the Inferring stage's obtained outputs to the desired dataset format.
- Evaluation and Fine-tuning: Assess the performance of the training models. Take metrics based on criteria relevant to the research topics, such as precision, recall, F score, or other assessment metrics. Another option is to train the model on additional data to improve its performance.

SO 4: To develop a LOD2 3D city model from the obtained roof plane structures vector format dataset.

1. What is the level of detail of the obtained 3D models?

Using ArcGIS commercial software tools, it was possible to reach a LOD2 3D model by combining inner roof planes with DSM, DTM, and nDSM.

2. What are the further improvements for the obtained 3D model?

Several measurements can be considered for further improvements for the created 3D LOD2 model:

- Refine the input data quality, as the dataset's quality is linked to the generated 3D model's quality.
- The inner roof planes used as input have some topological errors (as there was a post-processing stage to correct the mentioned topological errors); addressing topological errors in the inner roof planes and discrepancies between these planes and the 3D datasets could significantly refine the model. This could be accomplished by manual post-processing or GIS tools to identify topological errors.
- Explore open-source software alternatives and novel methodologies. Additionally, incorporating supplementary data, such as roof type and geometry information, may enhance the overall model's depth and accuracy.
- To assess and validate the generated 3D LOD2 model against other 3D models, such as the 3D BAG model, in order to identify inconsistencies, iteratively improve the 3D model, and validate the produced model

SO 5: To assess the performance of the developed approach

1. What are the performance differences between the two study cases?

The experiments in this research showed that models tailored to specific study areas effectively detected corners, edges and delineated building inner roof plane structures. However, a model trained on a combined dataset from both study areas showed a superior performance once the models were tested in a different area, with a different roof structure than the training area. Visually, both models resulted in straighter edges and fewer missed detections, but limitations such as incomplete region roof plane prediction were observed. Nevertheless, the model trained on a combined dataset could detect building inner roof planes, even when vegetation obscures edges.

After delineation of the roof graph structure, post-processing was required to convert the planar graph structure into inner roof planes in vector polygon format. Upon conversion, The situation was similar to the delineation stage. The model trained on the combined dataset showed superior performance in the area, with a different roof structure than the training area.

2. **What are the strengths and limitations of the developed framework?**

The strengths of the developed approach are given as followings:

Strengths:

1. The inner building roof delineation stage can be performed entirely using open–source tools.
2. Using the based HEAT model as starting point for training, it was possible to harness previous knowledge of the roof structure of the models. This optimizes the training in requires less training time in further training sessions on different datasets.
3. The trained model from the selected deep learning demonstrated outperformance as the models could delineate planes beyond the considered on-the-ground truth.
4. The model trained on a combined dataset could detect building inner roof planes, even when vegetation obscures edges.
5. The model can perform roof structure extraction for multiple buildings in one image sample.
6. Compared with image segmentation, the developed approach can delineate and extract straight edges, which derives into more regularized planar graph structures without extra post-processing.
7. Transferability, the approach can be transferred to different areas, and the performance be improved by executing a transfer learning on the new areas.
8. The generated LOD2 3D model showed a high visual similarity with the 3DBAG LOD2 3D model which was built using LIDAR point cloud.

Nevertheless, the proposed method has several limitations:

Limitations:

1. The selected deep learning approach (HEAT) can be computationally intensive to train and run, requiring significant computational resources. Due to a lack of computing resources, only experiments with image size 256x256 were performed.
2. Requires large and varied labeled training data to learn different complex geometry structures.
3. The approach faces challenges in delineating roof-building structures with complex shapes, such as roofs with circular configurations or samples featuring densely packed roof structures.
4. Limited generalization, the performance of the deep learning approach may be limited to the specific domains it has been trained on. Without further adaptation or training, it may not generalize well to other domains or scenarios.
5. Lack of hyperparameter-tunning, different experiments varying different hyperparameters such as the learning rate, optimizer, or the amount of regularization were not performed. This means that these hyperparameters must be carefully considered to improve the model's performance, often through trial and error.
6. The 3D phase was performed using commercial software. This limited the application of the approach to specific commercial software, ArcGIS, which could limit its applicability.
7. Lack of a metric that assesses the whole proposed approach. Even though some metrics to assess the performance in different phases are presented (building inner roof delineation, vectorization, 3D modelling), there is a lack of more complete metrics for the vectorization phase and the 3D phase, and a metric that could assess the effectiveness of the whole presented pipeline.

In further studies, collecting more varied training data will improve performance. Besides, adding another dimension in the input architecture of the model is a suggestion that can be explored to improve the model's ability to differentiate roof structures and the ground in the context of the image samples, as this could allow the use of the nDSM as another dimension in the image sample. Moreover, the applicability of the presented approach could be explored for building footprint extraction tasks and on-point cloud segmentations for 3D modeling using LIDAR.

LIST OF REFERENCES

Alidoost, F., Arefi, H., & Tombari, F. (2019). 2D image-to-3D Model: Knowledge-Based 3D building reconstruction (3DBR) using single aerial images and convolutional neural networks (CNNs). *Remote Sensing*, *11*(19). https://doi.org/10.3390/rs11192219

Awrangjeb, M., Zhang, C., & Fraser, C. S. (2013). Automatic extraction of building roofs using LIDAR data and multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *83*, 1–18. https://doi.org/10.1016/j.isprsjprs.2013.05.006

Bae, H., Jang, J., Jung, D., Jang, H., Ha, H., Lee, H., & Yoon, S. (2018). *Security and Privacy Issues in Deep Learning*. *00*(0), 1–20. http://arxiv.org/abs/1807.11655

Bauda, M.-A., Chambon, S., Gurdjos, P., & Charvillat, V. (2015). Geometry-based Superpixel Segmentation Introduction of Planar Hypothesis for Superpixel Construction. *VISAPP 2015 - 10th International Conference on Computer Vision Theory and Applications; VISIGRAPP, Proceedings*, *1*, 227–232. https://doi.org/10.5220/0005354902270232

Becker, T., Nagel, C., & Kolbe, T. H. (2011). Integrated 3D Modeling of Multi-utility Networks and Their Interdependencies for Critical Infrastructure Analysis. *Lecture Notes in Geoinformation and Cartography*, *XXXVIII*, 1–20. https://doi.org/10.1007/978-3-642-12670-3_1

Biljecki, F. (2013). The concept of level of detail in 3D city models. In *PhD Research Proposal, Delft University of Technology: Vol. II* (Issue 62). http://repository.tudelft.nl/assets/uuid:cea5a207-e796-4691-9440-13362cf8654c/291180.pdf

Chen, J., Liu, C., Wu, J., & Furukawa, Y. (2019). Floor-SP: Inverse CAD for Floorplans by Sequential Room-wise Shortest Path. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 2661–2670. https://doi.org/10.1109/ICCV.2019.00275

Chen, J., Qian, Y., & Furukawa, Y. (2022). HEAT: Holistic Edge Attention Transformer for Structured Reconstruction. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2022-June*, 3856–3865. https://doi.org/10.1109/CVPR52688.2022.00384

Cui, S., Yan, Q., & Reinartz, P. (2012). Complex building description and extraction based on Hough transformation and cycle detection. *Remote Sensing Letters*, *3*(2), 151–159. https://doi.org/10.1080/01431161.2010.548410

Deng, T., Zhang, K., & Shen, Z. J. (Max). (2021). A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. *Journal of Management Science and Engineering*, *6*(2), 125–134. https://doi.org/10.1016/j.jmse.2021.03.003

Dimitriadou, S., & Nikolakopoulos, K. G. (2022). Development of the Statistical Errors Raster Toolbox with Six Automated Models for Raster Analysis in GIS Environments. *Remote Sensing*, *14*(21). https://doi.org/10.3390/rs14215446

Dimitrov, H., & Petrova-Antonova, D. (2021). 3D city model as a first step towards digital twin of Sofia City. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *43*(B4-2021), 23–30. https://doi.org/10.5194/isprs-archives-XLIII-B4-2021-23-2021

Du, S., Zhang, F., & Zhang, X. (2015). Semantic classification of urban buildings combining VHR image and GIS data: An improved random forest approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, *105*, 107–119. https://doi.org/10.1016/j.isprsjprs.2015.03.011

Elberink, S. O., & Vosselman, G. (2009). Building Reconstruction by Target Based Graph Matching on Incomplete Laser Data: Analysis and limitations. *Sensors*, *9*(8), 6101–6118. https://doi.org/10.3390/s90806101

Girard, N., Smirnov, D., Solomon, J., & Tarabalka, Y. (2020). Polygonal Building Extraction by Frame Field Learning Nicolas. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 1805–1808. https://doi.org/10.1109/IGARSS39084.2020.9324080

Golnia, M. (2021). *Building Outline Delineation and Roofline Extraction : a Deep Learning Approach* [University of Twente]. http://essay.utwente.nl/88990/

Gui, S., & Qin, R. (2021). Automated LoD-2 model reconstruction from very-high-resolution satellite-derived digital surface model and orthophoto. *ISPRS Journal of Photogrammetry and Remote Sensing*, *181*(August), 1–19. https://doi.org/10.1016/j.isprsjprs.2021.08.025

Ha, T., Dang, T. K., Le, H., & Truong, T. A. (2020). Security and Privacy Issues in Deep Learning: A Brief Review. *SN Computer Science*, *1*(5), 1–15. https://doi.org/10.1007/s42979-020-00254-4

Hajji, R., Yaagoubi, R., Meliana, I., Laafou, I., & Gholabzouri, A. El. (2021). Development of an integrated BIM-3D GIS approach for 3D cadastre in morocco. *ISPRS International Journal of Geo-Information*, *10*(5). https://doi.org/10.3390/ijgi10050351

Hang, L., & Cai, G. (2020). CNN BASED DETECTION of BUILDING ROOFS from HIGH RESOLUTION SATELLITE IMAGES. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(3/W10), 187–192. https://doi.org/10.5194/isprs-archives-XLII-3-W10-187-2020

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2020). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 386–397. https://doi.org/10.1109/TPAMI.2018.2844175

Hossain, M. D., & Chen, D. (2022). A hybrid image segmentation method for building extraction from high-resolution RGB images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *192*(August), 299–314. https://doi.org/10.1016/j.isprsjprs.2022.08.024

Huang, J., Stoter, J., Peters, R., & Nan, L. (2022). City3D: Large-Scale Building Reconstruction from Airborne LiDAR Point Clouds. *Remote Sensing*, *14*(9), 1–18. https://doi.org/10.3390/rs14092254

Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., & Ma, Y. (2018). Learning to Parse Wireframes in Images of Man-Made Environments. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 626–635. https://doi.org/10.1109/CVPR.2018.00072

Jiang, J., Shu, Y., Wang, J., & Long, M. (2022). *Transferability in Deep Learning: A Survey*. *1*, 1–48. http://arxiv.org/abs/2201.05867

Kausika, B., Moshrefzadeh, M., Kolbe, T. H., & Van Sark, W. (2016). 3D Solar Potential Modelling and Analysis: A case study for the city of Utrecht. *Eu Pvsec 2016*, *5*(2), 1–4. https://mediatum.ub.tum.de/doc/1303969/1303969.pdf

Kenzhebay, M. (2022). *Planar roof structure extraction from Very High-Resolution aerial images and Digital Surface Models using deep learning* (Issue June) [University of Twente]. http://essay.utwente.nl/91396/

Kolbe, T. H. (2009). Representing and exchanging 3D city models with CityGML. *Lecture Notes in Geoinformation and Cartography*, *September*, 15–31. https://doi.org/10.1007/978-3-540-87395-2_2

Kolbe, T. H., Gröger, G., & Plümer, L. (2005). CityGML: Interoperable access to 3D city models. *Proceedings of the Int. Symposium on Geo-Information for Disaster Management*, *March*, 883–899. https://doi.org/10.1007/3-540-27468-5_63

Lafarge, F., Descombes, X., Zerubia, J., & Pierrot-Deseilligny, M. (2010). Structural approach for building reconstruction from a single DSM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(1), 135–147. https://doi.org/10.1109/TPAMI.2008.281

Lee, J., Zlatanova, S., Gartner, G., Meng, L., & Peterson, M. P. (2009). 3D Geo-Information Sciences. In J. Lee & S. Zlatanova (Eds.), *Lecture Notes in Geoinformation and Cartography* (pp. 79–96). Springer-Verlag Berlin Heidelberg 2009.

Liu, K., Ma, H., Ma, H., Cai, Z., & Zhang, L. (2020). Building Extraction from Airborne LiDAR Data Based on Min-Cut and Improved Post-Processing. *Remote Sensing*, *12*(17), 2849. https://doi.org/10.3390/rs12172849

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). https://doi.org/10.3390/rs11070830

Lopac, N., Jurdana, I., Brnelić, A., & Krljan, T. (2022). Application of Laser Systems for Detection and Ranging in the Modern Road Transportation and Maritime Sector. *Sensors*, *22*(16), 1–27. https://doi.org/10.3390/s22165946

Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, *28*(5), 823–870. https://doi.org/10.1080/01431160600746456

Lundervold, A. S., & Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift Fur Medizinische Physik*, *29*(2), 102–127. https://doi.org/10.1016/j.zemedi.2018.11.002

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, *152*(March), 166–177. https://doi.org/10.1016/j.isprsjprs.2019.04.015

Marcos, D., Tuia, D., Kellenberger, B., Zhang, L., Bai, M., Liao, R., & Urtasun, R. (2018). Learning deep structured active contours end-to-end. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8877–8885. https://doi.org/arXiv:1803.06329v1

Mazijn, C., Prunkl, C., Algaba, A., Danckaert, J., & Ginis, V. (2022). *LUCID: Exposing Algorithmic Bias through Inverse Design*. 1–9. http://arxiv.org/abs/2208.12786

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, *54*(6), 1–34. https://doi.org/10.1145/3457607

Nauata, N., & Furukawa, Y. (2020). Vectorizing World Buildings: Planar Graph Reconstruction by Primitive Detection and Relationship Inference. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12353 LNCS*, 711–726. https://doi.org/10.1007/978-3-

030-58598-3_42

Ok, A. O. (2013). Automated detection of buildings from single VHR multispectral images using shadow information and graph cuts. *ISPRS Journal of Photogrammetry and Remote Sensing*, *86*, 21–40. https://doi.org/10.1016/j.isprsjprs.2013.09.004

Paulsson, J. (2007). 3D Property Rights: An Analysis of Key Factors Based on International Experience. In *Building* (Issue May).

Persello, C., Wegner, J. D., Hansch, R., Tuia, D., Ghamisi, P., Koeva, M., & Camps-Valls, G. (2022). Deep Learning and Earth Observation to Support the Sustainable Development Goals: Current approaches, open challenges, and future opportunities. *IEEE Geoscience and Remote Sensing Magazine*, *10*(2), 172–200. https://doi.org/10.1109/MGRS.2021.3136100

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., & Stoter, J. (2022). Automated 3D Reconstruction of LoD2 and LoD1 Models for All 10 Million Buildings of the Netherlands. *Photogrammetric Engineering and Remote Sensing*, *88*(3), 165–170. https://doi.org/10.14358/PERS.21-00032R2

Pintore, G., Ganovelli, F., Pintus, R., Scopigno, R., & Gobbetti, E. (2018). 3D floor plan recovery from overlapping spherical images. *Computational Visual Media*, *4*(4), 367–383. https://doi.org/10.1007/s41095-018-0125-9

Qin, Y., Wu, Y., Li, B., Gao, S., Liu, M., & Zhan, Y. (2019). Semantic segmentation of building roof in dense urban environment with deep convolutional neural network: A case study using GF2 VHR imagery in China. *Sensors (Switzerland)*, *19*(5), 1–12. https://doi.org/10.3390/s19051164

Rezaei, Z., Vahidnia, M. H., Aghamohammadi, H., Azizi, Z., & Behzadi, S. (2023). Digital twins and 3D information modeling in a smart city for traffic controlling : A review. *Journal of Geography and Cartography (2023)*, *6*(1), 1–27. https://doi.org/10.24294/jgc.v6i1.1865

Sandrini, C., & Ii, S. (2022). *Social transformations of inhabited spaces in Sofia , Bulgaria Clara Sandrini To cite this version : HAL Id : hal-03638406*.

Shojaei, D., Olfat, H., Rajabifard, A., & Briffa, M. (2018). Design and development of a 3D digital cadastre visualization prototype. *ISPRS International Journal of Geo-Information*, *7*(10). https://doi.org/10.3390/ijgi7100384

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/s40537-019-0197-0

Sinha, S. N., Mordohai, P., & Pollefeys, M. (2007). Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. *Proceedings of the IEEE International Conference on Computer Vision*, *January*, 1–8. https://doi.org/10.1109/ICCV.2007.4408997

Soilán, M., Truong-Hong, L., Riveiro, B., & Laefer, D. (2018). Automatic extraction of road features in urban environments using dense ALS data. *International Journal of Applied Earth Observation and Geoinformation*, *64*(July 2017), 226–236. https://doi.org/10.1016/j.jag.2017.09.010

Stekovic, S., Rad, M., Fraundorfer, F., & Lepetit, V. (2021). MonteFloor: Extending MCTS for Reconstructing Accurate Large-Scale Floor Plans. *Proceedings of the IEEE International Conference on Computer Vision*, *Iccv*, 16014–16023. https://doi.org/10.1109/ICCV48922.2021.01573

Sun, X. (2021). *Deep Learning-Based Building Extraction Using Aerial Images and Digital Surface Models* [University of Twente]. https://library.itc.utwente.nl/papers_2021/msc/gfm/sun.pdf

Van Etten, A., Lindenbaum, D., & Bacastow, T. M. (2018). *SpaceNet: A Remote Sensing Dataset and Challenge Series*. 1–21. http://arxiv.org/abs/1807.01232

Van Melik, R. (2009). Visualising the effect of private-sector involvement on redeveloped public spaces in the Netherlands. *Tijdschrift Voor Economische En Sociale Geografie*, *100*(1), 114–120. https://doi.org/10.1111/j.1467-9663.2009.00512.x

Van Oosterom, P. (2013). Research and development in 3D cadastres. In *Computers, Environment and Urban Systems* (Vol. 40, pp. 1–6). https://doi.org/10.1016/j.compenvurbsys.2013.01.002

Xu, Y., Xu, W., Cheung, D., & Tu, Z. (2021). Line Segment Detection Using Transformers without Edges. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4255–4264. https://doi.org/10.1109/CVPR46437.2021.00424

Xue, F., Lu, W., Chen, Z., & Webster, C. J. (2020). From LiDAR point cloud towards digital twin city: Clustering city objects based on Gestalt principles. *ISPRS Journal of Photogrammetry and Remote Sensing*, *167*(July), 418–431. https://doi.org/10.1016/j.isprsjprs.2020.07.020

Xue, N., Wu, T., Bai, S., Wang, F., Xia, G. S., Zhang, L., & Torr, P. H. S. (2020). Holistically-Attracted Wireframe Parsing. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2785–2794. https://doi.org/10.1109/CVPR42600.2020.00286

Ye, Z., Fu, Y., Gan, M., Deng, J., Comber, A., & Wang, K. (2019). Building extraction from very high resolution aerial imagery using joint attention deep neural network. *Remote Sensing*, *11*(24), 1–21.

https://doi.org/10.3390/rs11242970

Zhang, F., Nauata, N., & Furukawa, Y. (2020). Conv-MPN: Convolutional message passing neural network for structured outdoor architecture reconstruction. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2795–2804. https://doi.org/10.1109/CVPR42600.2020.00287

Zhang, F., Xu, X., Nauata, N., & Furukawa, Y. (2021). Structured Outdoor Architecture Reconstruction by Exploration and Classification. *Proceedings of the IEEE International Conference on Computer Vision*, 12407–12415. https://doi.org/10.1109/ICCV48922.2021.01220

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, *42*(October 2017), 146–157. https://doi.org/10.1016/j.inffus.2017.10.006

Zhang, Z., Li, Z., Bi, N., Zheng, J., Wang, J., Huang, K., Luo, W., Xu, Y., & Gao, S. (2019). PPGNET: Learning point-pair graph for line segment detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 7098–7107. https://doi.org/10.1109/CVPR.2019.00727

Zhao, W. (2022). *Extracting Geometric Features of Buildings from Remote Sensing Images* [University of Twente]. https://doi.org/10.3990/1.9789036554978

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*(March), 119–131. https://doi.org/10.1016/j.isprsjprs.2021.02.014

Zhao, W., Persello, C., & Stein, A. (2022). Extracting planar roof structures from very high resolution images using graph neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, *187*(February), 34–45. https://doi.org/10.1016/j.isprsjprs.2022.02.022

Zhou, Y., Qi, H., & Ma, Y. (2019). End-to-End Wireframe Parsing. In CVF (Ed.), *Proceedings of the IEEE International Conference on Computer Vision* (Vols. 2019-Octob, pp. 962–971). CVF. https://doi.org/10.1109/ICCV.2019.00105

Zhou, Y., Qi, H., Zhai, Y., Sun, Q., Chen, Z., Wei, L. Y., & Ma, Y. (2019). Learning to Reconstruct 3D Manhattan Wireframes from a Single Image. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 7697–7706. https://doi.org/10.1109/ICCV.2019.00779

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2020). Deformable DETR: Deformable Transformers for End-to-End Object Detection. *ICLR 2021*, 1–16. http://arxiv.org/abs/2010.04159