Leveraging Deep Learning for Enhanced Building Information Retrieval and Reconstruction from Remote Sensing Imagery

Dissertation

zur Erlangung
des Doktorgrades der Naturwissenschaften (Dr. rer. nat.)
des Fachbereichs Mathematik / Informatik
der Universität Osnabrück

Vorgelegt
von

Philipp Schuegraf

Prüfer der Dissertation
Prof. Dr. Peter Reinartz, Universität Osnabrück
Prof. Dr. Friedrich Fraundorfer, Technische Universität Graz, Österreich
Tag der mündlichen Prüfung: 21.11.2024

Abstract

In large cities around the world, city centres are often densely built-up. Due to the dense development, it is difficult to distinguish the buildings on the basis of satellite and aerial images and to reconstruct them in 3D. However, this is essential for applications such as urban planning, disaster management, solar energy potential assessment, flow simulation, and many others. Therefore, the goal of this dissertation is to provide methods for retrieving building information and reconstructing buildings in 3D. For this purpose, three methods are introduced.

The first method segments building sections on satellite and aerial images, as well as associated digital surface models (DSMs), by segmenting the separation lines between them using a fully convolutional neural network (FCN). In addition, the remaining pixels assigned to the building are segmented. The building and separation line segments are then used to obtain seamless building sections using the watershed transformation. In addition, morphology is used to close existing gaps in separation lines. To further improve the separation line and building segments, a loss function is used, which results in the FCN already producing fewer gaps before morphology is applied and building segments have straighter edges and sharper corners. The resulting building sections are then further processed into polygons and level of detail (LoD)-1 models. The method is robust and works on both aerial and satellite data. In addition, it can segment building sections in complex scenarios more accurately than the compared approach. The method manages to accurately segment even in highly complex scenarios with very small building sections and informal settlement after the FCN has been re-trained with data from different cities with dense buildings. This is particularly important for crisis management and humanitarian aid. Moreover, a method is introduced which vectorizes building footprints and regularizes their outlines in two steps. In the first step, a deep neural network predicts the primary orientation angle of the building polygon. In the second step, all vertices are adjusted, such that their inside angles are either 90° or 180° with respect to the primary orientation.

The second method, PLANES4LOD2, generates LoD-2 models based on aerial images and DSMs, as well as digital terrain models (DTMs). Similar to the first method, it starts with segmenting separation lines between building sections and but also extending them to roof planes. Each building is thus divided into sections and each section is divided into roof planes. This process is performed end-to-end by an FCN, which uses a novel depth attention module (DAM) to utilize DSM features effectively and efficiently. The roof surfaces are then converted to polygons. To obtain a 3D model, the height values from the DSM within each roof plane are passed to random sample consensus (RANSAC), which estimates robust plane parameters. The plane parameters are used to calculate the height values at the corners of the roof plane polygons. The experimental

evaluation shows that PLANES4LOD2 generates geometrically accurate, topologically consistent and semantically correct LoD-2 models. In comparison with SAT2LOD2, PLANES4LOD2 shows a significantly higher accuracy of 1.06 m in mean absolute error (MAE), especially in complex scenarios in densely built-up city centres. The baseline method only achieves 2.18 m in MAE.

The third method presented in this dissertation is called SAT2BUILDING. It is a method for LoD-2 reconstruction based on image data and DSMs, but focuses on satellite rather than aerial image data and does not require an external DTM. Alternatively, an FCN generates the building heights as an additional output. Instead of focusing on the segmentation of separation lines, like PLANES4LOD2, SAT2BUILDING clusters pixels to roof planes by calculating spatial embeddings related to the centre of the roof planes. This is more effective for satellite data with ground sampling distances (GSDs) ranging from 0.5 m to 0.7 m, as the separation lines are difficult to recognise. SAT2BUILDING is evaluated on a more rural scenario with simple roof shapes and distances between the buildings and an urban scenario with dense development and complex roof shapes. The metrics show the significantly higher accuracy of SAT2BUILDING compared to three baseline methods in both scenarios.

The three newly introduced methods as well as their detailed experimental evaluation and discussion represent a significant contribution to the retrieval of building information and 3D reconstruction of buildings, and enable their future use for a wide range of applications.

Zusammenfassung

In großen Städten weltweit sind Stadtzentren oftmals dicht bebaut. Aufgrund der dichten Bebauung ist es schwierig, die Gebäude auf Basis von Satelliten- und Luftbildern auseinander zu halten, sowie sie in 3D zu rekonstruieren. Dies ist jedoch für Anwendungen wie Stadtplanung, Disastermanagement, Solarenergiepotenzialerhebung, sowie Strömungssimulation unerlässlich. Daher widmet sich diese Dissertation der Segmentierung von Abschnitten und level of detail (LoD)-2 Rekonstruktion von komplexen und einfachen Gebäuden. Dazu werden drei Methoden eingeführt. Desweiteren, wird in einer Fallstudie in Medellín, Kolumbien die Wirksamkeit einer Methode zur Erkennung informeller Bebauung analysiert.

Die erste Methode segmentiert Gebäudeabschnitte auf Satelliten- und Luftbildern, sowie zugehörigen Digitalen Oberflächen Modellen (DOMs), indem sie die Trennlinien zwischen diesen mithilfe eines fully convolutional neural network (FCN) segmentiert. Zusätzlich werden die restlichen Pixel, welche dem Gebäude zugeordnet sind, segmentiert. Die Gebäude- und Trennliniensegmente werden dann verwendet, um mittels der Watershed-Transformation nahtlose Gebäudeabschnitte zu erhalten. Darüber hinaus wird Morphologie verwendet, um existierende Lücken in Trennlinien zu schließen. Um die Trennlinien- und Gebäudesegmente weiter zu verbesseren, wird eine Loss-Funktion eingesetzt, die dazu führt, dass das FCN bereits vor dem Einsatz von Morphologie weniger Lücken produziert und Gebäudeabschnitte geradere Kanten und spitzere Ecken haben. Die erhaltenen Gebäudeabschnitte werden dann noch zu Polygonen und LoD-1 Modellen weiterverarbeitet. Die Methode ist robust und funktioniert sowohl auf Luft- und Satellitendaten. Darüberhinaus kann sie Gebäudeabschnitte in komplexen Szenarien akurater segmentieren als der mit ihr verglichene Ansatz. Die Methode schafft es, nachdem das FCN mit Daten aus verschiedenen Städten mit dichter Bebauung neu traniert wurde, auch in hoch-komplexen Szenarien mit sehr kleinen Gebäudeabschnitten und informeller Bebauung diese akurat zu segmentieren. Dies ist Besonders für Krisenmanagement, sowie humanitäre Hilfe von großer Wichtigkeit. Desweiteren wird eine Methode vorgestellt, welche Gebäudesegmente vektorisiert und deren Umrisse in zwei Schritten regularisiert. Der erste Schritt ist die Vorhersage eines Hauptorientierungswinkels des Gebäude Polygons. Im zweiten Schritt werden dann alle Verticees im Polygon so angepasst, dass sie entweder einen Innenwinkel von 90° oder 180° im Verhältnis zum vorhergesagten Winkel haben.

Die zweite Methode, PLANES4LOD2, erzeugt LoD-2 Modelle auf Basis von Luftbildern und DOMs, sowie Digitalen Gelände Modellen (DGMs). Dabei baut sie, ähnlich wie die erste Methode, auf der Segmentierung von Trennlinien zwischen Gebäudeabschnitten und Dachflächen auf. Jedes Gebäude wird somit in Abschnitte und jeder Abschnitt in Dachflächen unterteilt. Dieser Prozess wird Ende-zu-Ende durch ein FCN

durchgeführt, welches ein neuartiges depth attention module (DAM) verwendet, um DOM features effektiv und effizient zu nutzen. Anschließend werden die Dachflächen zu Polygonen konvertiert. Um ein 3D Modell zu erhalten, werden die Höhenwerte aus dem DOM innerhalb jeder Dachfläche an random sample consensus (RANSAC) übergeben, womit robuste Ebenenparameter geschätzt werden. Die Ebenenparameter dienen der Berechnung der Höhenwerte an den Ecken der Dachflächenpolygonen. Die experimentelle Auswertung zeigt, dass PLANES4LOD2 geometrisch genaue, topologisch konsistente und semantisch korrekte LoD-2 Modelle erzeugt. Im Vergleich mit bestehender Software zeigt sich, das PLANES4LOD2 insbesondere in komplexen Szenarien in dicht-bebauten Stadtzentren eine deutlich höhere Genauigkeit von 1.06 m mean absolute error (MAE), im Vergleich zu 2.18 m MAE der Vergleichsmethode, erzielt.

Die dritte in dieser Dissertation vorgestellte Methode nennt sich SAT2BUILDING. SAT2BUILDING ist eine weitere Methode für die LoD-2 Rekonstruktion auf Basis von Bilddaten und DOMs, konzentriert sich jedoch auf Satelliten- anstatt Luftbilddaten und benötigt kein externes DGM. Stattdessen erzeugt ein FCN die Gebäudehöhen als zusätzlichen Ausgabewert. Anstatt sich, wie PLANES4LOD2, auf die Segmentierung von Trennlinien zu konzentrieren, clustert SAT2BUILDING Pixel zu Dachflächen, indem es räumliche Einbettungen berechnet, welche sich auf den Mittelpunkt der Dachflächen beziehen. Dies ist bei Satellitendaten mit ground sampling distances (GSDs) von 0.5 m bis 0.7 m effektiver, da dort die Trennlinien nur noch schwer erkennbar sind. Durch die Auswertung von SAT2BUILDING auf zwei Testgebieten, ein eher ländliches Szenario mit einfachen Dachformen und Abständen zwischen den Gebäuden und ein urbanes Szenario mit dicher Bebauung und komplexen Dachformen, zeigt sich die wesentlich höhere Genauigkeit von SAT2BUILDING im Vergleich zu drei Vergleichsmethoden in beiden Testgebieten.

Die drei neu eingeführten Methoden sowie deren ausführliche experimentelle Auswertung und Diskusion stellen einen erheblichen Fortschritt in der Extraktion von Gebäudeinformation sowie 3D Rekonstruktion von Gebäuden dar und ermöglichen deren zukünftigen Einsatz für vielfältige Anwendungen.

Acknowledgments

I would like to express my deepest gratitude to my Doktorvater, Prof. Dr. Peter Reinartz, for his unwavering support, invaluable guidance, and encouragement throughout my doctoral journey. His mentorship has been instrumental in shaping both my research and personal growth.

I am indebted to Prof. Dr. techn. Friedrich Fraundorfer for his insightful ideas and perspectives, which have enriched my research and broadened my horizons in ways I could not have imagined.

Special thanks are due to Dr. Ksenia Bittner, my main technical supervisor, whose dedication, expertise, and encouragement propelled me forward during the challenging phases of my dissertation. Her unwavering support and belief in my abilities were truly motivating.

I am profoundly grateful to my wife, Elisabeth, for her unwavering support, patience, and understanding throughout this journey. Her love and encouragement sustained me during the long hours of research and writing.

I extend my gratitude to my mother, Angelika, and my father, Werner, for their boundless love, encouragement, and unwavering belief in my capabilities. Their constant support has been a source of strength and inspiration.

My heartfelt appreciation goes to my siblings—Ben, Paula, and Karla—for their unwavering belief in my abilities, their encouragement, and their understanding. Their support has been a cornerstone of my journey, providing both motivation and solace during challenging times.

Lastly, I am grateful to all colleagues and mentors who have provided support, encouragement, and valuable insights along the way. Your contributions have played a significant role in shaping my academic journey.

Philipp Schuegraf
Oberpfaffenhofen, November 2024

Contents

1 Introduction

We live in the era of big data. There exist many sources of data, be it from industrial sensors or social networks. One large source of data is coming from earth observation. Every day, the Earth is orbited by many sensors mounted on satellites. Data acquisition campagnes from aerial vehicles like helicopters, aeroplanes and unmanned aerial vehicles (UAVs) are carried out. These sensors include synthetic apperture radar (SAR), light detection and ranging (LiDAR), thermal and optical sensors.

Optical sensors can produce stereo imagery, allowing the large-scale 3D reconstruction of the surface of the earth. Furthermore, they are, similar to SAR, relatively cheap. Yet, optical sensors provide easily interpretable and highly detailed images, whereas SAR delivers highly noisy data. On one hand, aerial imagery has high spatial resolution and is less prone to atmospheric occlusion than satellite imagery. On the other hand, obtaining satellite imagery is more energy efficient than obtaining aerial imagery from flight campagnes. Moreover, satellite imagery cover larger regions by each single acquisition. Overall, airborne and spaceborne imagery complement each other.

One of the most prominent features on aerial and satellite imagery in urban scenes are buildings. Next to roads, they form the built environment of cities. Detecting their outlines in imagery is a highly relevant task in urban planing and development, disaster management, environmental monitoring, crisis response and humanitarian aid. Moreover, the 3D reconstruction of buildings is crucial for applications like flow simulation of 5G waves, flood, or wind and solar panel recommendation.

Manual delineation of buildings on satellite or aerial imagery is a difficult and time-consuming task. Building boundaries can be occluded by high vegetation or shadows from taller objects and highly-trained human experts are necessary to annotate buildings in complex scenarios like densely built neighborhoods. Acquiring 3D models of buildings manually is even more expensive, since terrestrial laser-scanning requires a person that can handle the necessary devices and drives from house to house. This becomes infeasible for updating the 3D models of whole cities and even impossible for historic data, where only the overhead image exists and the built environment has changed over time.

Lately, advances in deep artificial neural networks allow to automatically learn features from data, given a large training dataset and sufficient computational power. Thanks to the large quantities of existing imagery as well as open cadastre data and 3D city models, as well as improved computer hardware, deep learning has become a valuable tool to automate the process of building information extraction and reconstruction. Yet, learning-free methods are still reliable approaches beyond feature extraction, making them ideal to be combined with deep learning methods.

1.1 Scope of the Dissertation

This thesis tackles two problems in the field of building information retrieval and reconstruction:

- **Objective 1: Building Section Instance Segmentation**

  In typical cities, houses are built next to each other, with no space in between them. This makes the detection of individual houses a challenging problem. Furthermore, instance segmentation methods like Mask-RCNN [1] struggle to correctly segment building sections without gaps or overlaps [2]. Yet, remote sensing imagery shows the separation line between building sections. Segmenting it together as a separate class with the remaining building section enables the usage of the watershed transformation [3], leading to seamlessly connected building sections. Another challenge in building segmentation is the confusion of buildings with road, which can have similar shape features (rectangular) and color (grey). Digital surface models (DSMs) enhances the distinction of buildings from road, leading to improved segmentation. The resulting segments are memory and computation intensive from the perspective of applications. And simple vectorization does not remove unnecessary vertices and can lead to loss of seamless connection. Hence, we vectorize building sections by tracing their boundaries and simplify the separation lines and the remainders of the polygons independently. For simple buildings, we propose a regularization method that makes each angle of the polygons rectangular. The resulting building polygons are limited to 2D applications. We tackle this by using the DSM together with the building sections to derive level of detail (LoD)-1 models (according to the city geography markup language (CityGML) standard [4]). Informal settlements are often sparsely mapped, which is due to low accessibility and high construction dynamics. We also show that our building section segmentation approach can support mapping in informal settlements.

- **Objective 2: Level of Detail-2 Reconstruction**

  Reconstructing buildings in LoD-2 requires more detailed elements than building sections. Yet, conventional methods can not detect roof planes in a robust way, because they rely on hand-crafted features [5, 6]. We employ a deep learning approach, PLANES4LOD2, that separates roof planes and building sections in the same style, by segmenting separation lines for each using a deep neural network. After we vectorized the roof planes, we want to project the polygons to the 3D space to obtain LoD-2 models using a DSM. Simply inserting the height values from the DSM as the z-coordinate of each vertex in the polygon will lead to highly irregular 3D polygons, since the roof plane polygon will usually not completely coincide with the building edge in the DSM. Hence, we utilize random sample consensus (RANSAC), an approach that is more robust to outliers, to obtain planar roof planes. Due to atmospheric effects and high ground sampling distance (GSD), separation lines become less visible in satellite imagery. Hence, we introduce another approach, SAT2BUILDING, for LoD-2 reconstruction that uses spatial em-

beddings, which focus on the center of roof planes instances to segment them. Furthermore, applications require 3D building models with height relative to the ground. But normalizing heights from DSM using a digital terrain model (DTM) is a challenging task, where conventional methods fail [7]. Hence, SAT2BUILDING predicts the normalized digital surface model (nDSM) as an additional output.

1.2 Guildeline for Reading

This is a *cumulative* dissertation, which is organized as follows. Chapter 2 introduces fundamental knowledge related to this thesis. Chapter 3 introduces the related work, Chapter 4 presents a new method for building section instance segmentation, based on the publication

[8]: ***P. Schuegraf***, *S. Zorzi, F. Fraundorfer, and K. Bittner, "Deep learning for the automatic division of building constructions into sections on remote sensing images," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 16, pp. 7186–7200, 2023*,

and shows its applicability to formal and informal building segmentation in endangered areas based on the publication

[9]: ***P. Schuegraf***, *D. Stiller, J. Tian, T. Stark, M. Wurm, H. Taubenböck, K. Bittner, "Ai-based building instance segmentation in formal and informal settlements," IEEE International Geoscience and Remote Sensing Symposium, 2024.*

Additionally, a method to regularize building footprints effectively and time-efficient is presented, which is based on the publication

[10]: ***P. Schuegraf***, *Z. Li, J. Tian, J. Shan, and K. Bittner, "Rectilinear building footprint regularization using deep learning," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2024.*

Chapter 5 introduces one method for LoD-2 reconstruction from aerial imagery based on the publication

[11]: ***P. Schuegraf***, *S. Shan, and K. Bittner, "Planes4lod2: Reconstruction of LoD-2 Building Models using a Depth Attention-Based Fully Convolutional Neural Network," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 211, pp. 425-437, 2024*

and one for LoD-2 reconstruction from satellite imagery based on the publication

[12]: ***P. Schuegraf***, *S. Gui, R. Qin, F. Fraundorfer, and K. Bittner, "Sat2building: Lod-2 Building Reconstruction from Satellite Imagery using Spatial Embeddings," Submitted to ISPRS Journal of Photogrammetry and Remote Sensing, 2024.*

Chapter 6 concludes this thesis.

1.3 Papers Related to the Dissertation

This section briefly describes contributions that where done while working on the dissertation and they are relevant for the content of the dissertation. These papers where either written by the author of the dissertation as the primary author or as a co-author.

The content in Chapter 4 is based on, but not limited to the work done in Schuegraf *et al.* [2]. There, the concept of building section segmentation based on touching borders was originally introduced and tested against Mask-RCNN. Chapter 4 extends Schuegraf *et al.* [2] by showing that the UNet-3+ [13] is more suitable than the classical UNet [14] for building section segmentation. Furthermore, the topology loss [15] is leveraged and the evaluation is extended in terms of generalization, multiple data sources and advanced baselines.

In Schuegraf *et al.* [16], a dataset called Roof3D for building section and roof plane segmentation is introduced together with a baseline method. This method is tested in various configurations in terms of handling multi-modal inputs, namely RGB data and photogrammetric DSM. Roof3D and the baseline method are used as the basis for the experimental part in Chapter 5.

Gui *et al.* [17] builds on SAT2LOD2 [18], the LoD-2 reconstruction pipeline from ortho imagery and nDSM. SAT2LOD2 reconstructs whole buildings, without separating them into sections. Contrastingly, Gui *et al.* [17] rely on the building sections predicted by the method presented in Chapter 4, to obtain refined LoD-2 models. This method is then used as a baseline in the experimental evaluation in Chapter 5.

2 Background

This chapter introduces basic concepts, that form the foundation for the methods analysed and utilized in this thesis.

2.1 Very High-Resolution Remote Sensing Imagery

Very high resolution (VHR) remote sensing imagery is acquired by sensors mounted on satellites or aerial vehicles. It includes ground sampling distance (GSD) of lower than 1 m. The GSD is the spatial extent of a single pixel of a remote sensing image. Low GSDs allow to detect objects like buildings, their components, roads, and cars. Special challenges of VHR remote sensing imagery are its high data volume due to the low GSD, and its high price in comparison to low-resolution imagery.

2.1.1 Light Detection and Ranging

Light detection and ranging (LiDAR) sensors are active remote sensing sensors (see Figure 2.3 (a)). Many of them belong to the VHR realm. They emit pulses and measure the time that the pulse takes to be reflected off the terrain or surface objects to the sensor. This allows to obtain a detailed height profile. A LiDAR sensor consists of a laser scanner that emits the pulse, a receiver to measure the time the pulse takes to be reflected back to the sensor and global positioning system (GPS). It is used exclusively from air. Among the advantages of LiDAR are the high geometrical accuracy of the obtained height-profile, the insensitivity to noise, and the fact that it can penetrate through clouds and vegetation, allowing operation in every weather scenario. Its downsides are the high cost of data acquisition, the limited penetration in certain surface materials, and that special software and expertise are required to derive meaningful information from LiDAR data.

2.1.2 Optical Stereo Imagery

Optical sensors are relatively cheap and allow easy interpretation of the data with less pre-processing than LiDAR. Furthermore, they include spectral information in the visible range of the electromagnetic spectrum, which is illustrated in Figure 2.1. The light that they receive was originally emitted by the sun and reflected by the surface of the earth, which makes them a passive sensor (see Figure 2.3 (b)). Night-time imagery also includes light from terrestrial sources. Optical sensors combined with GPS and the camera parameters allow the acquisition of geo-referenced imagery. If multiple images of the same scene are acquired it is called stereo imagery. Multiple images are usually acquired in short time intervals.

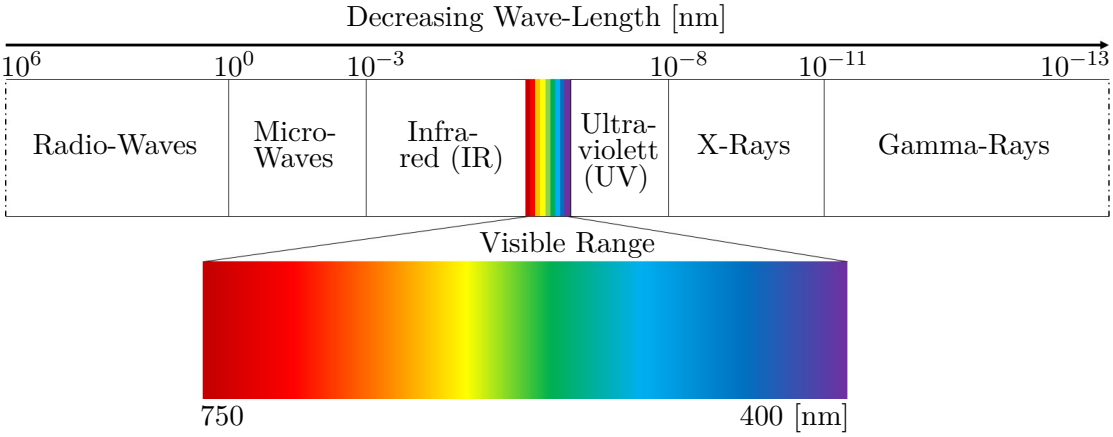


Figure 2.1: Illustration of the electromagnetic spectrum and the visible range.

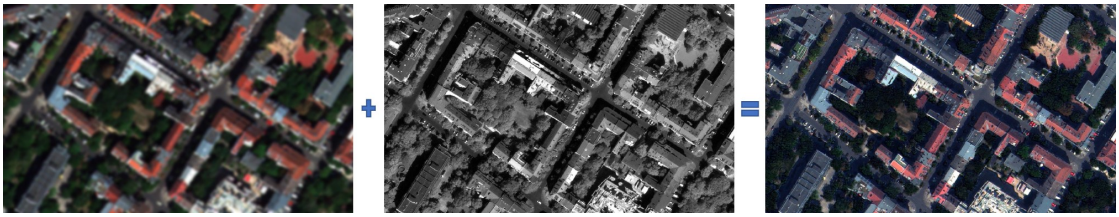


Figure 2.2: Illustration of pan-sharpening. PAN is short for panchromatic image and RGB is short for RGB image.

2.1.2.1 Pan-Sharpening

Commonly, cameras have multiple bands that have different advantages. The pancromatic band captures light at a broad sub-range of the visible range, which means it has low spectral resolution. On the other hand, the panchromatic band has a high spatial resolution (see middle in Figure 2.2), for example 0.31 m GSD in the WorldView-4 satellite. In the same satellite, there exist four spectral bands (infrared, red, green, and blue), which means a high spectral resolution. Other satellite have even more spectral bands. Yet, the spectral bands in WorldView-4 only have a spatial resolution of 1.24 m GSD (see left in Figure 2.2). This trade-off stems from the dedication of limited resources like data storage and transmission. Pan-sharpening is commonly used to combine the high spatial resolution of the panchromatic band with the high spectral resolution of the multi-spectral bands, as it is illustrated in Figure 2.2. In this thesis, color or multi-spectral bands are always pan-sharpened before further processing them.

2.1.2.2 Ortho-Rectification

On remote sensing imagery there is a shift of pixels that do not lie directly below the camera position. Furthermore, uneven terrain and camera tilt lead to more geometric inaccuracies. Such distortions lead to inconsistencies between imagery and other

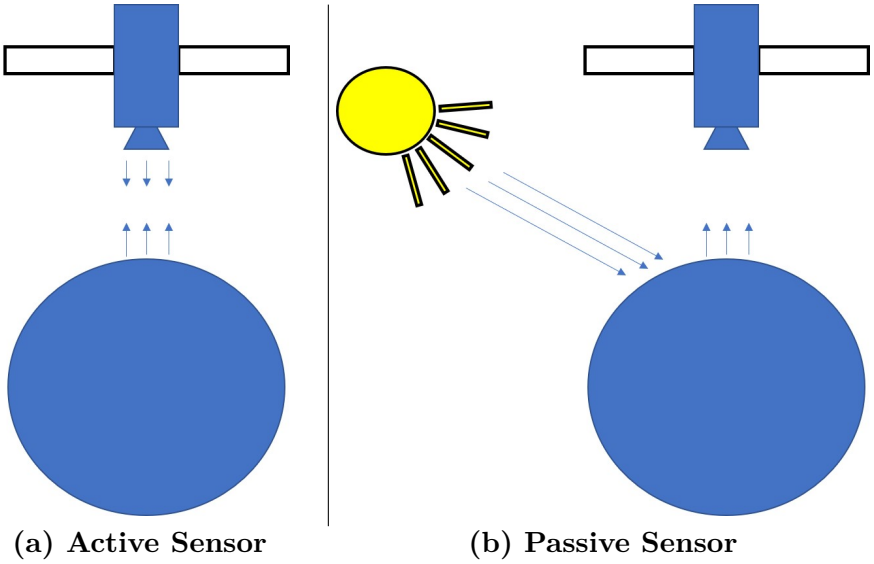


Figure 2.3: Illustration of active and passive sensors in remote sensing.

Figure 2.4: Illustration artifacts due to ortho-rectification. The building roof is mirrored at the areas of steep gradients in the digital surface model (DSM).

geo-information like building positions. Hence, ortho-rectification is used to obtain geographically accurate images. This process can lead to artifacts, especially on objects with sharp gradients like building facades (see Figure 2.4) and mountains. This makes the task of building information retrieval and reconstruction more challenging.

2.1.2.3 Aerial Imagery

Imagery that is acquired by aircrafts like helicopters, aeroplanes or unmanned aerial vehicles (UAVs) is called aerial imagery. Aircrafts usually operate inside the atmosphere of the earth at a height of up to 18 km. Aerial flight campaigns are limited by weather conditions and the cost of the pilot, the operator of the camera system, fuel and vehicle rent are relatively high. Yet, aerial imagery is an important source of information. It is uniquely valuable for tasks that require GSDs of smaller than 0.3 m like pedestrian de-

tection, car detection, and road condition assessment. In 3D reconstruction of buildings, small GSDs like 0.1 m allow a highly detailed insight into the roof geometry.

2.1.2.4 Satellite Imagery

Satellites are devices that operate above the Kármán line at 100 km above ground. Modern commercial satellites have a GSD of as small as 0.3 m (e.g. Pleiades Neo, WorldView-3, WorldView-4) and often times 0.5 m (e.g. Pleiades, WorldView-1, WorldView-2) in the panchromatic band. Satellites acquire imagery at lower cost than aerial imagery, since they need neither fuel nor pilots and operators. Their main costs are satellite design and construction, the sensor and the data transmission and processing on the ground. Satellite imagery is used in applications such as building detection, building damage assessment, 3D reconstruction, road detection, and land-cover classification.

2.1.2.5 Photogrammetric Digital Surface Models

The enhanced maneuverability of contemporary VHR satellites, such as WorldView-1, 2, 3, and 4, enables the capture of multiple images of a specific area from multiple different viewing angles within a single orbit. By merging DSMs derived from multiple image pairs using semi-global matching (SGM) algorithms [19–21], which circumvent the necessity for matching windows, it is possible to achieve a superior-quality DSM characterized by fewer outliers and more distinct object delineations. The DSM generation process involves two primary stages: 1) computation of stereo imagery orientation considering provided rational polynomial coefficientss (RPCs), and 2) dense image matching. Obtaining precise stereo matching is of utmost importance, as the density and accuracy of the resultant matching points directly impact the quality of DSMs (see visualization of example DSM in Figure 2.5).

Multi-View Image Orientation: Obtaining elevation models from multi-view VHR satellite imagery necessitates precise calculation of RPC camera parameters for each image. While the initial RPCs, derived from orbit and altitude information, serve as a foundation, they often fail to adequately align multiple images. Thus, refining the RPCs becomes imperative, as inaccuracies in satellite position and orientation can lead to systematic errors during dense image matching for DSM generation.

To rectify errors in sensor position and orientation, a bundle adjustment process is employed. This process ensures that observations of a single ground point across multiple images are consistent within a unified geodetic framework [22]. Various studies [21, 23, 24] have already utilized RPC bundle adjustment refinement procedures to establish satisfactory relative orientation between images. Correcting RPCs begins with identifying accurate tie points between stereo images. Initially located with one-pixel accuracy by scanning the search region, these tie points are then refined to sub-pixel precision using local least squares matching (LSM) [25].

For absolute orientation correction, optimal ground control points (GCPs) at sub-pixel accuracy are indispensable. GCPs represent known 3D points acquired from GPS observations or existing DSMs and high-resolution ortho-rectified images. However, acquiring

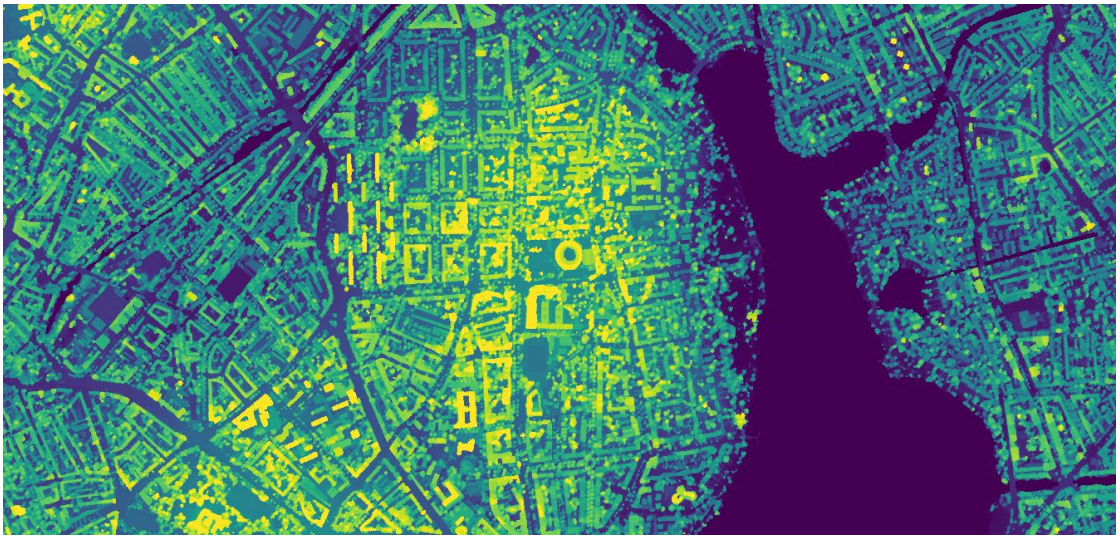


Figure 2.5: Illustration of a DSM showing Hamburg, Germany.

the necessary GCPs can be arduous or even infeasible, particularly in scenarios requiring rapid response, such as large-scale processing or crises. In such cases, where GCP information may not be readily available, contemporary research relies on automatic image-based techniques to extract and correct GCPs [26].

Dense Image Matching: To produce a detailed DSM from multi-view stereo images, the approach relies on employing the dense stereo matching technique SGM [19, 20]. SGM facilitates pixel-wise matching of mutual information and approximates global 2D smoothness constraints by extending 1D constraints in multiple directions across the image [20]. The core principles of SGM revolve around matching and generating a disparity map.

Following the establishment of robust relative orientation, matching occurs for all potentially corresponding pixels in the stereo pair using epipolar geometry. Unlike traditional window-based matching techniques, SGM does not employ window matching [19, 20], thereby ensuring reliable reconstruction of object edges. To circumvent strong local assumptions about surface shape, the matching process is formulated as an energy minimization problem, wherein aggregation costs are optimized from 16 directions to derive a disparity image D with minimal energy [20]

$$\varepsilon(D) = \sum_{p} \Big\{ C(p, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1] \Big\}. \quad (2.1)$$

The energy function in Equation (2.1) comprises three terms: a data term C which defines the pixel-wise matching cost between image pixels p and corresponding pixels in the disparity map D_p; and two regularization terms that promote similar disparities for neighboring pixels N_p while permitting large jumps in high-contrast areas. These regularization terms introduce constant penalties P_1 and P_2 for small and large disparity changes, respectively, ensuring the stability of the matching process. The detailed cost

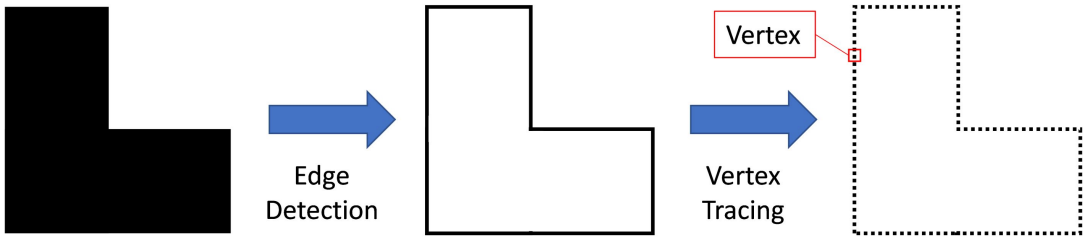


Figure 2.6: Illustration of vectorizing a building blob.

aggregation methodology is elucidated by Hirschmüller [20].

In the matching process image pairs are matched bidirectionally, allowing for the retention of consistent disparities while avoiding the fusion of highly mismatched regions. Small isolated regions, indicative of irregularities, are discarded. The resulting separate disparity maps from each image pair are then reprojected to the desired projection and merged to generate a unified DSM, typically employing a median filter [21].

2.2 Image Processing in Remote Sensing

Image processing is a key to remote sensing. Without suitable methods to process image data, satellite and aerial imagery could not be leveraged for applications.

2.2.1 Polygonization of Segments

For a computer to process visual data it is often advantageous to have it in vector format. Vector formats consist of geometric elements such as points, lines, polygons and collections of these primitives. Fully convolutional neural networks (FCNs) usually produce raster output which is computationally heavy to process and does not allow easy editing. In remote sensing, this means that pixel-blobs of buildings, roads, vegetation, and other objects have to be converted to vector format which is called vectorization. The way this is achieved depends on the object type. As an example, building blobs are vectorized by a two step procedure. The first step is to apply an edge filter like Sobel or Canny and to use morphology to obtain a single pixel-wide line which is called skeleton. The second step consists of first establishing a graph from the skeleton that contains pixel locations as nodes and edges between direct 8-neighbors. Finally, a random node is selected and used as initial vertex. While moving along the graph, visited nodes are regarded as vertices and added to the final polygon. The skeleton is completed when the initial node is visited for the second time.

2.2.2 Image Re-Sampling

In some cases, operations require or prefer a certain GSD. But the available data might be in higher or lower GSD, where a low GSD corresponds to a high resolution image and a high GSD corresponds to a low resolution image. Sometimes, the GSDs across

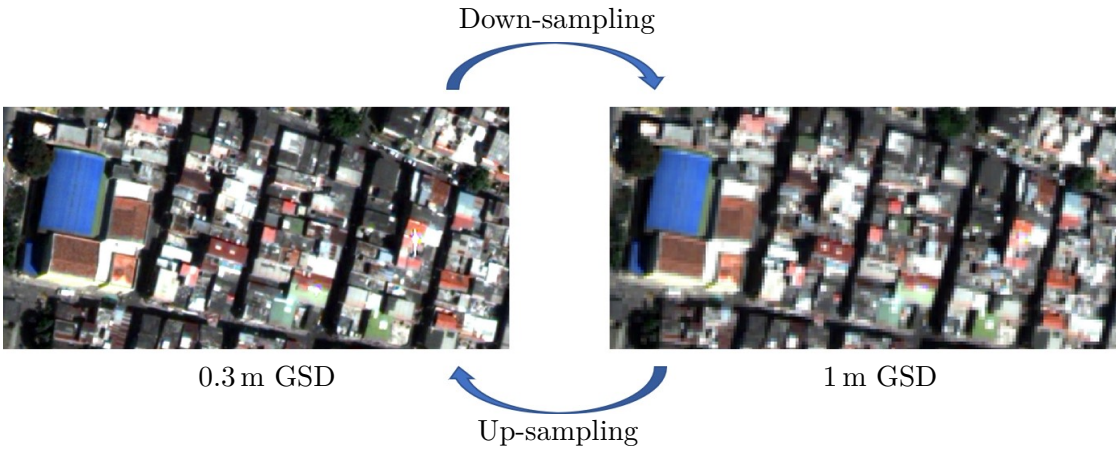


Figure 2.7: Illustration of resampling an image in an urban scenario in Medellín, Columbia.

a large dataset with minor resolution difference, needs to be close. Even though a large difference in available GSD and required GSD causes an included information gap that can not be closed, especially if higher resolution imagery is required, re-sampling is a technique that comes to help. There are mainly two types of image re-sampling, which are up-sampling (increasing the resolution) and down-sampling (decreasing the resolution). Both principles are illustrated for satellite color image in Figure 2.7.

Down-sampling an image is achieved by merging several pixels into one by a certain pattern. Up-sampling an image is accomplished by first adding pixels equally spread around existing pixels. Secondly, the newly inserted pixels are filled according to some schemes. One popular scheme or pattern for image re-sampling is nearest neighbor, where the value of the new pixel is the same as the pixel with the least distance. This kind of re-sampling is suitable if there is only a small number of allowed pixel values, like in semantic maps. Nearest-neighbor re-sampling causes block-patterns, which are unwanted in many image types like multi-spectral imagery or DSMs. In those cases, for each new pixel, multiple neighboring pixels of the original image are taken into account. Bilinear re-sampling uses a linear mixture of neighboring pixel values, depending on the distance to each of the neighboring pixels. Bicubic re-sampling uses a third order polynomial and not only direct neighbors, resulting in a more accurate but less efficient re-sampling.

2.3 Deep Learning

Deep learning is the discipline of training deep neural networks. Deep neural networks are functions $f_\theta(x) = f^{(n-1)}(f^{(n-2)}...(f^{(0)}(x))...))$ that parameterize a high-dimensional, non-linear mapping with θ. They can be used for classification and regression tasks. Deep neural networks consist of multiple stacked layers $f^{(i)}(x) = \sigma^{(i)}(l^{(i)}(x))$, which usually have non-linear activation functions $\sigma^{(i)}$ at layer i. The non-linearity allows modelling real-world input-output-relations. Layer $l^{(i)}(x) = n^{(i)}(c^{(i)}(x))$ commonly use normalization layers $n^{(i)}$, which makes information passed to the next layer $f^{(i+1)}$ more

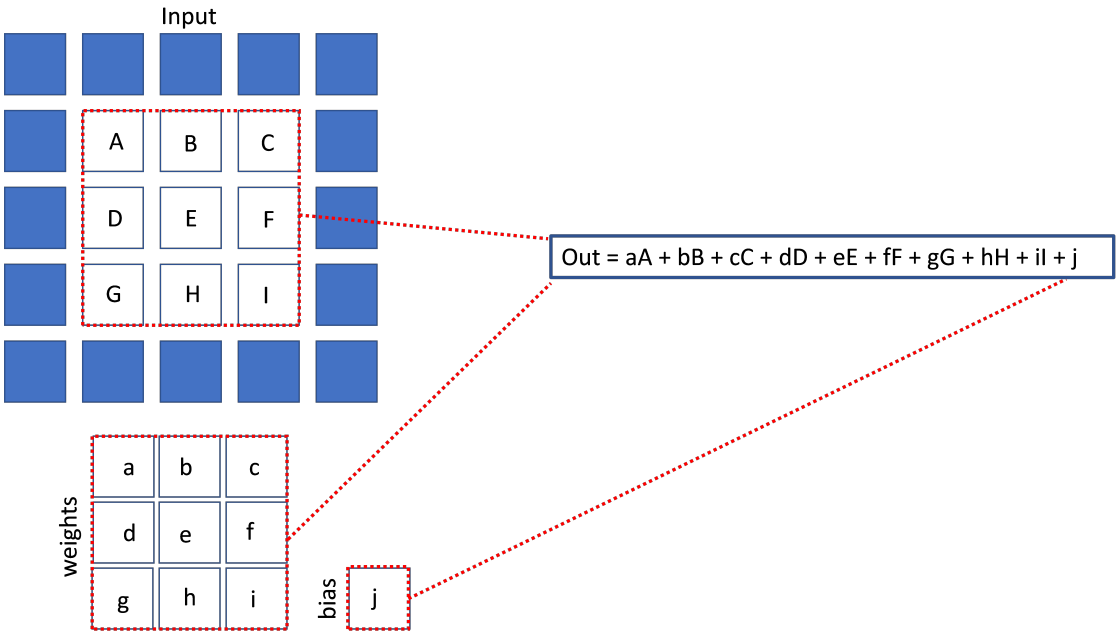


Figure 2.8: Illustration of the convolution operation using a 3x3 filter. The scenario shows the computation of a single cell in the output grid, based on a spatial window in the input grid. To compute a full output grid, the spatial window is shifted over the input, but the weights and bias remain the same everywhere.

predictable and hence simplifies learning in deep neural network architectures. The main component of neural networks are the neural layers $c^{(i)}$. They are linear transformations of their input, where the connectivity between input and output depends on the layer type. The parameters $\theta = (w, b)$ of $c^{(i)}$ are called weights and biases because they scale components of their input vectors and add on them.

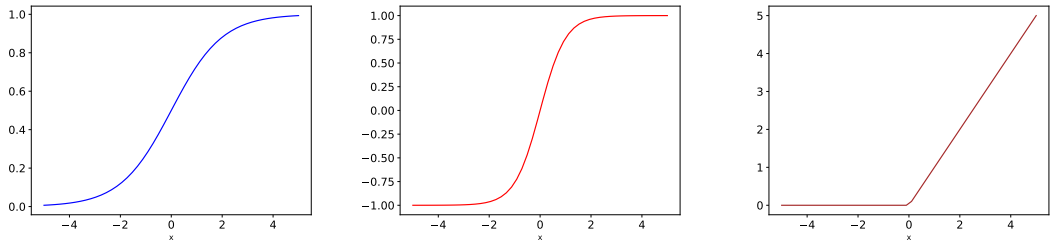
2.3.1 Convolutional Neural Network

In convolutional neural networks (CNNs) weights are shared across different locations in the input and each output component only interacts with input elements in a window which is usually squared. The main modules of typical CNNs are convolutional layers for 2D input

$$c^{(i)}(x)_p = conv_\theta^{(i)}(x)_p = b_{p_0,p_1}^{(i)} + \sum_{s=-W_1}^{W_1} \sum_{t=-W_2}^{W_2} w_{s,t}^{(i)} x_{p_0-s,p_1-t}, \tag{2.2}$$

where $p = (p_0, p_1)$ is the location of the output component, $s \in \{-W_1, ..., W_1\}$ and $t \in \{-W_2, ..., W_2\}$ are discrete weight indices. The convolution operation is visualized in Figure 2.8.

Batch normalization, layer normalization and instance normalization are common as normalization layers. Data is usually passed to neural networks in batches, meaning that



(a) sigmoid **(b) tanh** **(c) ReLU**

Figure 2.9: Illustration of different activation functions.

multiple inputs are processed in parallel. Furthermore, the number of channels varies in CNN layers. Batch normalization normalizes across the spatial and batch dimension of the feature maps, layer normalization across the spatial and channel dimension, and instance normalization only across the spatial dimension. Normalization in these layers means that the mean is subtracted and the result is divided by the standard deviation. To allow flexibility of these layers, new mean and standard deviation are used, which are treated as learnable parameters of the network.

Typical choices for non-linearities are tanh, sigmoid, and rectified linear unit (ReLU), as visualized in Figure 2.9. Tanh

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.3}$$

and sigmoid

$$\text{sigmoid}(x) = \frac{1}{e^{-x} + 1} \tag{2.4}$$

are smooth functions in their entire domain and are converging towards their maximum value for increasing x and their minimum value for decreasing x. These properties can lead to very small gradients, which is a problem in training CNNs. On the other hand, ReLU

$$\text{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \tag{2.5}$$

is a piece-wise linear function with zero gradient below zero and fixed gradient above zero.

In common CNN architectures, the number of channels steadily increases layer per layer. Since this would lead to an infeasible memory consumption, pooling layers are utilized. Pooling layers aggregate information across the spatial dimension. One type of pooling, which is usually employed in intermediate layers to steadily decrease the spatial resolution, is strided, windowed max-pooling. It takes the maximum value inside a window of the input and moves the window by a stride that is usually equal to its side-length. Another type of pooling, which is used as output layer by some CNNs, is global average pooling. It computes the average over the complete spatial dimension,

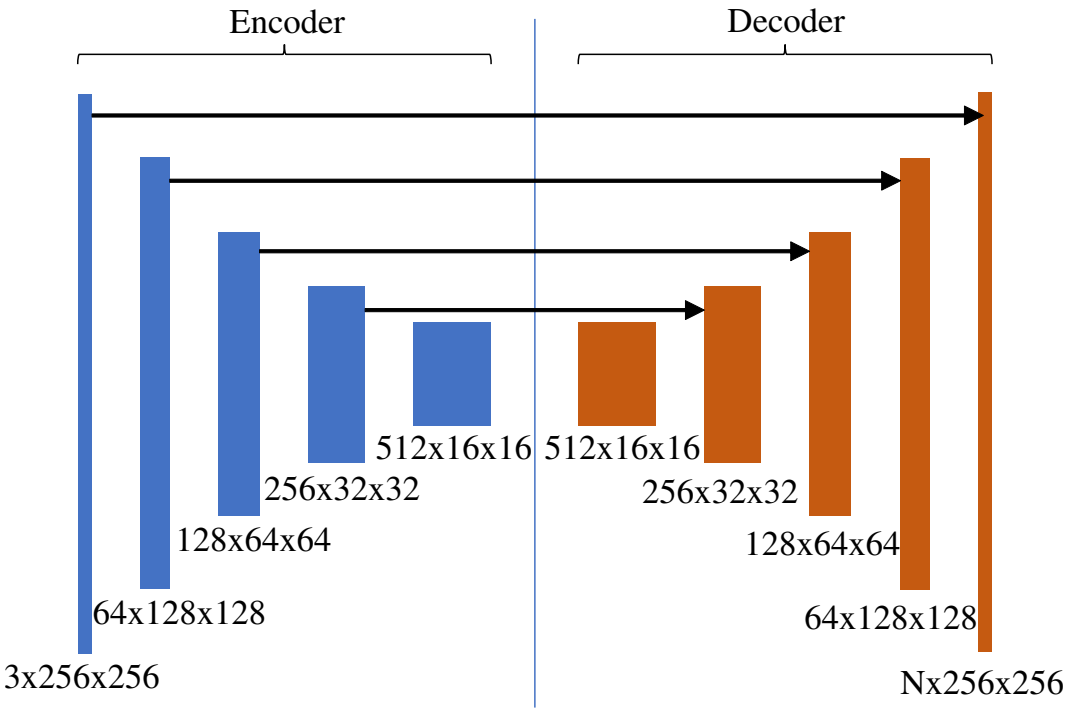


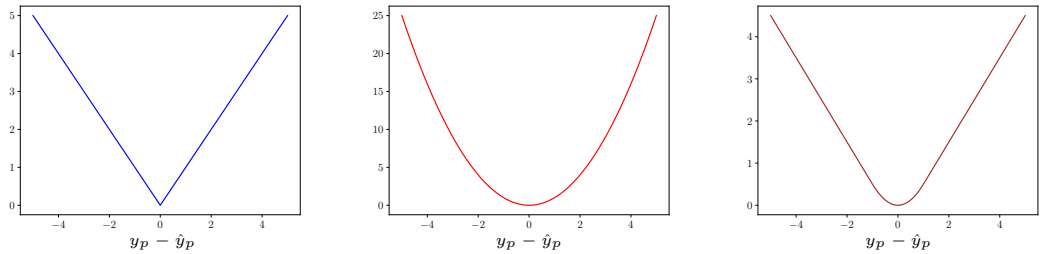
Figure 2.10: Illustration of a Unet architecture with N outputs at the same resolution as the input. The black arrows represent skip-connections. The blue blocks consist of convolution, normalization, activation, and pooling layers, whereas the brown blocks consist of upsampling and convolution or transposed convolution, normalization, and activation layers.

leaving the result with only a single value per channel. This value is then interpreted as the class-score or regression value.

But many times, the class-score or regression value is obtained by flattening the spatial and channel dimensions. Afterwards these vectors are passed to fully connected layers, where each output element is a linear transformation of all input elements. Fully connected layers serve to generate a vector of class-scores or regression values. In the case of classification, it is a usual choice to use the class with the highest score as the predicted class. For regression, a suitable activation function can be used to bring the values to a certain range, for example $[-1, 1]$ using tanh.

2.3.2 Fully Convolutional Neural Network

CNNs produce outputs independent of spatial location. But tasks like building segmentation in satellite imagery or normalized digital surface model (nDSM) regression require outputs of high spatial resolution. FCNs [27] can produce outputs of high spatial resolution. They have no fully connected layers, to preserve spatial location of features throughout the entire network. Due to memory restrictions, it is advantageous to keep the intermediate pooling layers from the CNNs. But this leads to low resolution outputs.



(a) $\mathcal{L}_{L1}$ **(b)** $\mathcal{L}_{L2}$ **(c)** $\mathcal{L}_{Huber}$

Figure 2.11: Illustration of different loss functions for regression tasks.

Hence, up-sampling and transposed convolution layers are utilized. Transposed convolutional layers expand single pixels in the input by multiplying them with various learnable parameters in an expanding window and adding bias. Alternatively, up-sampling can be used to increase the spatial resolution and be followed by convolutional layers that allow a learnable increase of resolution.

The shrinking part of the network is called the encoder, which is followed by the decoder, the expanding part of the network (see Figure 2.10). When the original FCN was introduced by Long *et al.* [27], the encoder and decoder had only a small number of connections. These connections were close to the transition between encoder and decoder, called the bottleneck. But such a low-connectivity scheme has two main downsides. The first is that few high resolution information from the encoder is used to recover spatial detail. The second is that long paths between a layer and the input can lead to vanishing gradients. To avoid these pitfalls, skip-connections from each encoder level to each decoder level were introduced by Ronneberger *et al.* [14].

2.3.3 Training Neural Networks

We have defined typical types of neural networks and their layers. Training them requires a task-dependent loss function. Regression tasks use loss functions (illustrated in Figure 2.11) such as L1

$$\mathcal{L}_{L1}(\hat{y}, y) = \frac{1}{|P|} \sum_{p \in P} |y_p - \hat{y}_p| \tag{2.6}$$

and L2

$$\mathcal{L}_{L2}(\hat{y}, y) = \frac{1}{|P|} \sum_{p \in P} \|y_p - \hat{y}_p\|_2^2, \tag{2.7}$$

where y is the ground truth, $\hat{y}$ is the prediction, and $p \in P$ is a pixel position. Since L2 causes steep gradients if the difference between y and $\hat{y}$ is large and L1 is not smooth near zero, the Huber-Loss is often used:

$$\mathcal{L}_{Huber}(\hat{y}, y) = \frac{1}{|P|} \sum_{p \in P} \begin{cases} |y - \hat{y}| - \frac{1}{2} & \text{if } |y - \hat{y}| > 1 \\ \frac{1}{2} \|y - \hat{y}\|_2^2 & \text{else.} \end{cases} \tag{2.8}$$

For segmentation tasks, the loss function is often the multi-class cross-entropy

$$\mathcal{L}_{CE}(\hat{y}, y) = \frac{1}{|C||P|} \sum_{p \in P} \sum_{c \in C} -y_p^c \times log(\text{softmax}(\hat{y}_p)_c), \tag{2.9}$$

where $c \in C$ is the semantic class label, $y_p^c \in \{0, 1\}$ is a binary ground truth label that determines whether pixel p belongs to class c, $\hat{y}_p$ is the predicted class score vector at pixel p, and

$$\text{softmax}(x)_c = \frac{e^{x_c}}{\sum_{i \in C} e^{x_i}} \tag{2.10}$$

converts arbitrary real-valued class-score vector x into a probability score for class c.

Training neural networks requires optimization techniques, such as evolutionary algorithms [28], simmulated annealing [29] and stochastic gradient descent [30]. Since stochastic gradient descent is flexible and efficient, it is the most common optimization algorithm for training neural networks. Stochastic gradient descent works by sampling a mini-batch (x, y) from the training data, where x is the input to the neural network $\hat{f}_\theta$, which has parameters $\theta_i \in \theta$, and y is the ground truth. After computing the forward pass $\hat{y} = \hat{f}_\theta(x)$, the network parameters get updated by

$$\theta_i' = \theta_i - \alpha(\nabla_\theta \mathcal{L}(\hat{y}, y))_i, \tag{2.11}$$

where θ_i' is the updated parameter and α is the learning rate, controlling the step-length of the optimization step. Usually, the data x gets sampled without replacement. After each sample has been processed once, which is often called an epoch or iteration, the procedure is repeated. The gradient computation is done by backpropagation [31], which is based on the chain-rule of calculus

$$\frac{\delta f(g(x))}{\delta x} = \frac{\delta f(g(x))}{\delta g(x)} \frac{\delta g(x)}{\delta x}. \tag{2.12}$$

A common variant of stochastic gradient descent is Adam [32]. Adam includes information of past gradients in the parameter update, which leads to a more direct way to a minimum in the loss function with respect to the parameters. Let $g_i^{(t)} = (\nabla_\theta \mathcal{L}(\hat{y}, y))^{(t)}$ be the gradient of the loss function with respect to the parameter vector θ at optimization step t. Then,

$$m^{(t)} = \beta_1 m^{(t-1)} + (1 - \beta_1) g^{(t)} \tag{2.13}$$

and

$$v^{(t)} = \beta_2 v^{(t-1)} + (1 - \beta_2)(g^{(t)})^2, \tag{2.14}$$

are the first and second order momenta and β_1, $\beta_2 \in [0, 1[$ are hyperparameters controlling the balance between current and past gradients in both momenta. The momenta $m^{(t)}$ and $v^{(t)}$ are usually initialized with 0, which makes them tend to be small, espes-

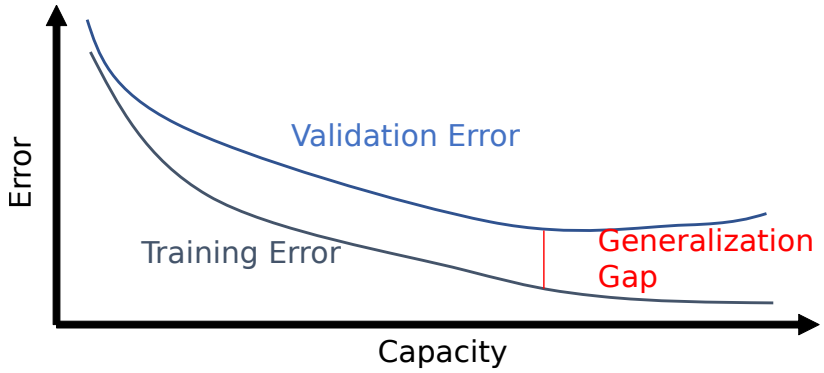


Figure 2.12: Illustration of training and validation error.

cially for β_1 and β_2 close to 1. This bias-corrected momenta are

$$\hat{m}^{(t)} = \frac{m^{(t)}}{1 - \beta_1^t} \tag{2.15}$$

and

$$\hat{v}^{(t)} = \frac{v^{(t)}}{1 - \beta_2^t}, \tag{2.16}$$

for $t \geq 1$. The parameter update is then computed by

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\alpha}{\hat{v}^{(t)} + \epsilon} \hat{m}^{(t)}, \tag{2.17}$$

where ϵ is a small number that avoids division by 0.

2.3.4 Data Splitting

The data that is used to compute parameter updates during training is called training data. Its primary purpose is to expose the neural network to samples of a function it shall approximate. Yet, the training data alone is not a good indicator to judge the success of the training process. How much a model fits the training data is called the capacity of the model. The capacity of a deep neural network grows with the number of layers and the number of parallel convolution operators at each layer. Furthermore, the capacity also increases with training time. Even though it is the goal of the training to increase the model capacity, this is only true to a certain point. If the model fits the training data too much, it has also learned the noise and irrelevant details instead of generalizable patterns. Hence, the generalization performance is usually measured by a second dataset, called the validation dataset. This dataset is entirely distinct from the training data and is not used in the parameter updates. In this way, it allows to observe the generalization performance of the model. In Figure 2.12, a typical relation between capacity, training error, and validation error is illustrated. Moreover, the generalization

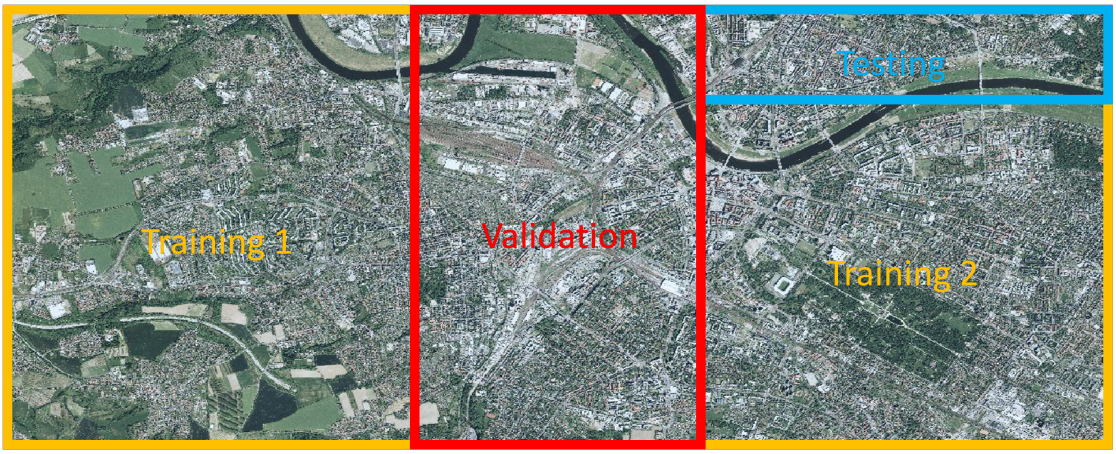


Figure 2.13: Example of splitting training-, validation-, and testdata in remote sensing using an RGB image in Dresden, Germany.

Figure 2.14: Example of cropping a large image into patches, which are overlapping by 50 % in vertical and horizontal directions.

gap indicates how much worse the model fits the validation data than the training data. Since the validation data is utilized to measure the generalization performance and also restrict the capacity, an additional dataset is needed to quantify the final generalization error of the model. This dataset is separate from training and validation data and is called test data.

When splitting the dataset, it is important that all three parts are diverse. The training dataset usually is the largest of the three, since it directly influences the performance of the model. The validation dataset is commonly a smaller dataset than the training dataset, as is the test dataset. When training a deep neural network with VHR imagery, there exist usually several large images. In Figure 2.13, it is visualized how each of these images can contribute to the three datasets. Due to memory restrictions, the large areas are not a direct input to the network, but are split in patches. This is illustrated in Figure 2.14, where each of the dotted boxes of different colors indicate one patch.

2.3.5 Model Regularization

Regarding the validation error of the trained model, it should be as small as possible. This can be achieved by restricting the capacity of the model. The model capacity is influenced, for example, by the number of epochs, the model size, and the range of the parameters. The more epochs the model is trained, the better it fits the training data. Yet, at some point, the validation error starts to increase. To find the ideal number of epochs, the model is trained for a large number of epochs, and the model with the lowest validation error is usually selected as the final model. This is called early stopping, and was introduced by Morgan *et al.* [33]. The model size can be restricted in multiple ways. One possibility is to train and validate on models of several sizes, like the EfficientNet [34] or ResNet [35] model families, and then select the model with the lowest validation error. These model families include models of varying size, making it easy to scale it to the optimal capacity. Another effective way to limit the effective size of the model is Dropout [36]. Dropout is a layer that can be arbitrarily stacked on top of convolutional or fully connected layers. It makes the subsequent layer use only a subset of the features computed by the convolutional layer. Hence, the model is forced to learn robust features, that are less dependent on each other. Since not all of the features are used, the capacity of the model is limited, which functions as a regularization. Another way of controlling the capacity of the model is to restrict the range of the parameters. This can be achieved by adding a norm of the parameters, for example the L_2-Norm to the loss

$$\mathcal{L}_{wd}(\hat{y}, y) = \mathcal{L}(\hat{y}, y) + \eta \frac{1}{|\theta|} \sum_{\theta_i \in \theta} \theta_i^2, \tag{2.18}$$

where η controls how strong weight decay influences the loss function. By using $\mathcal{L}_{wd}$ instead of $\mathcal{L}$, the parameters are forced to be small, which restricts the capacity of the model.

The above-mentioned regularizers or a subset of them are often used in deep learning applications. Whether or not to use them depends on the quantitative evaluation. If adding a regularization strategy leads to a lower validation error, it is worth including it.

2.3.6 Multi-Modal Networks

VHR remote sensing imagery often includes uncertainty about objects of interest. In such cases, using multiple modalities that complement each other comes to help. One example for this is road detection on orthorectified imagery. A bridge can make the road below it invisible, which introduces uncertainty about the road, lanemarks and cars on it. In Henry *et al.* [37], aerial imagery is utilized together with a spatially corresponding patch of open street map (OSM) data. The network utilized in their work is called a Skip-Fuse architecture, having two encoders and a single decoder. One of the encoders is dedicated to aerial imagery, the other one to OSM data. Their feature maps are fused at the skip-connections. This architecture makes the layers dedicated to each of the modalities independent. A sketch of this architecture is presented in Figure 2.15.

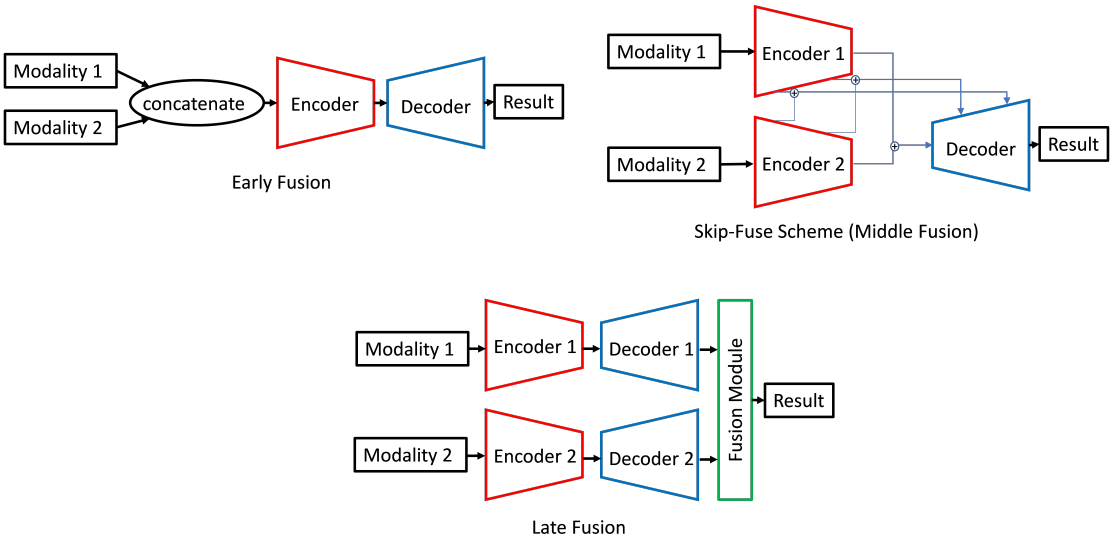


Figure 2.15: Illustration of multiple different multi-modal fusion architectures.

Bittner *et al.* [38] and Schuegraf *et al.* [39] have tackled the task of automatic building footprint generation from VHR satellite imagery and photogrammetric DSM. Both works found, that it is beneficial to have two separate decoders and a fusion module of the features of both decoders. Specifically, this kind of multi-modal network was compared against early fusion, which means that the RGB patch and the DSM patch are fused before passing them to the network. Find the early, middle, and late fusion visualized in Figure 2.15.

2.4 Summary

Modern advancements in technology offer extensive possibilities for analyzing and monitoring Earth's transformations. One effective method for swiftly acquiring comprehensive Earth-related data is through the exploration of global satellite imagery. This data provides detailed insights into the structures and landscapes of numerous cities, essential for various remote sensing applications. Utilizing the current high spatial and spectral resolutions, alongside height information derived from multi-view stereo satellite images, enables the extraction of building outlines and their 3D representations. However, these tasks pose challenges due to the diverse shapes and complexities of buildings. Recent advancements in methodologies, capable of automatically extracting task-specific features, hold promise for achieving precise results. Consequently, this dissertation delves into two issues concerning the automatic extraction and reconstruction of building information, employing deep learning-based approaches.

3 State of the Art

In this chapter, the area of research that the thesis is targetting is thoroughly examined in the subsequent sections. The disciplines included here are neural network architecture, building footprint extraction, building instance segmentation, level of detail (LoD)-2 reconstruction, and attention in building segmentation. The contents in this chapter are mostly taken from, but not limited to the papers Schuegraf *et al.* [8], Schuegraf *et al.* [11], Schuegraf *et al.* [12], and Schuegraf *et al.* [10].

3.1 Neural Network Architecture

The architecture of most common state-of-the-art (SOTA) semantic segmentation neural networks is fully convolutional, meaning that only convolutional layers carry automatically adaptable parameters. Such networks are called fully convolutional neural networks (FCNs). Furthermore, Ronneberger *et al.* [14] introduced the U-form to FCNs. The left half of the U-shaped network is called the encoder, deriving features of increasing depth and decreasing resolution. The right half of the U-shaped network is named decoder, which regains spatial resolution step-wise, taking into account feature maps from the encoder on multiple levels of resolution.

Even though the many skip-connections make the original U-Net a popular architecture, it is limited in the structure of its convolutional layers. In the original U-Net, the convolutional layers are purely sequential, meaning that each convolutional layer has as an input the output of the previous, potentially down-sampled convolutional layer. There are several modifications to this style of stacking convolutions. For instance, He *et al.* [35] have introduced ResNet, which applies shortcuts to the convolutions. In such blocks, called residual blocks, the input follows two paths. One path forwards the input and the other one applies convolutions to the input. Both paths are then merged by summation. This makes it easy for the network to learn the identity function in a residual block, which reduces overfitting. Different to ResNet, DenseNet [40] has a far higher re-use of feature maps and therefore requires less parameters. This is accomplished by stacked dense blocks, which propagate each feature map to all subsequent layers inside a block. The strategy presented by Henry *et al.* [37] not only takes advantage of DenseNet121 (DenseNet with 121 layers), as the backbone, but also uses its blocks in the decoder, mirroring the number of feature maps, feature maps' height and width as well as the number of layers on each level of resolution. This architecture is designed for road center line segmentation from aerial RGB imagery. Hence, it does not facilitate the injection of auxiliary information. This problem was solved by Henry *et al.* [37], where a fusion technique called SkipFuse is introduced to enhance the DenseNet121-U-Net by injecting open street map (OSM) information. In the architecture, the fusion of OSM

and RGB is achieved by summation of the feature maps at each level of resolution in the encoder. The sum is not propagated to any other layer in the encoder, but serves as the input to the skip-connections. Thus, both the RGB and OSM each have a separate backbone network that specializes solely on one of the two modalities. Since building segmentation profits from knowledge about height [2, 39, 41], this architecture is also suitable for building information extraction.

3.2 Building Segmentation

3.2.1 Building Footprint Extraction

Most of the work dedicated to dense pixel labeling of buildings in remote sensing imagery is in the realm of binary building footprint extraction using FCNs.

3.2.1.1 Footprint Extraction using Classical Methods

Traditional methods for building footprint extraction rely on geometrical models and analysis of building properties by experts. For instance, Huertas *et al.* [42] base their approach on the assumption that buildings are characterized by rectangular shapes and the presence of shadows, which help differentiate building outlines from non-building ones. This technique often produces building polygons with jagged lines. To create more precise footprint boundaries, Guercke *et al.* [43] employ the Hough Transformation to detect lines within an initial building footprint. They then use lines corresponding to peaks in the Hough space to construct a refined polygon. To enhance accuracy and robustness against variations in building appearance, some methods utilize datasets containing both depth and spectral images. Rottensteiner *et al.* [44] leverage Dempster-Shafer theory to fuse multiple features derived from light detection and ranging (LiDAR) digital surface model (DSM) and multispectral aerial imagery for building detection. These features include the normalized difference vegetation index (NDVI), the normalized digital surface model (nDSM), and measures of object roughness. The nDSM helps identify objects that rise from the ground, distinguishing them from buildings, while the roughness measure and NDVI primarily differentiate trees from buildings. Ekhtari *et al.* [45] also use a LiDAR nDSM to generate an initial building mask, which is refined using WorldView imagery. Initially, a rough building mask is created from the nDSM, which is then reduced to its rough edges. Subsequently, edges are detected in the spectral image, but these edges are often discontinuous and not limited to buildings. To filter out non-building edges, edges from the spectral image are masked by those from the depth-based building edges. Finally, polygons are fitted to the masked edges to eliminate discontinuities. Turlapaty *et al.* [46] compute depth information by fusing space-borne multi-angular imagery and utilize both multispectral and PAN images, which are combined through pansharpening. From the pansharpened image, the NDVI is calculated, and statistical properties of these data sources are input into a support vector machine, which classifies each pixel as either building or non-building.

Although these methods can be effective for certain areas and building types, a significant limitation is that models that rely on hand-crafted features often fail to handle complex building structures. Moreover, they depend heavily on the manual identification of relevant features by human experts.

3.2.1.2 Footprint Extraction using Deep Learning

Lately, deep learning has become the SOTA in building footprint extraction. Li *et al.* [47] leveraged an attraction field map (AFM) to make a U-Net [14] recognize building footprints more accurately. The authors cascade the U-Net with another FCN. The U-Net learns an alternative representation for building boundaries that is based on a limited set of pixels per building boundary pixel and the distance of a pixel to the boundary. Although, this method improves the shapes of the building boundaries, it does not prove that it works in dense, urban scenarios and the results showed visual irregularities like rounded corners and blob-like building shapes. One work that tackles these problems is presented by Zhang *et al.* [48], where authors adapt the topology loss [15] to be aware of building boundaries. Instead of using an ImageNet pre-trained VGG-network as in [15], the authors opt for a cyclic training scheme of the loss network. The resulted building mask is passed to the cyclic loss network to compute features of the intermediate layers. The authors do the same with the ground truth building footprint and then compute a regression loss between the features of the prediction and the features of the ground truth.

Liu *et al.* [49] introduce spatial residual inception, a module that successively fuses multi-level features to aggregate multi-scale information. Their proposed SRI-Net is especially accurate in detecting large buildings. Another method that targets multi-scale information is that of Zhu *et al.* [50], called MAP-Net, addresses the challenges of accurately extracting multiscale building footprints and precise boundaries from remote sensed imagery. Unlike traditional multi-scale feature extraction strategies, MAP-Net maintains spatial localization by using a multi-parallel path architecture. Each stage of this architecture is designed to gradually generate high-level semantic features while preserving fixed resolution. An attention module is then employed to adaptively squeeze the channel-wise features extracted from each path, optimizing multiscale feature fusion. Additionally, a pyramid spatial pooling module captures global dependencies, refining discontinuous building footprints. This approach ensures accurate extraction of building footprints across different scales, with improved precision in the boundaries.

Abdollahi *et al.* [51] perform building footprint segmentation based on aerial imagery. Their approach contains two main components. The first is a SegNet [52] with bi-directional long short term memory (LSTM). This kind of network allows to use both features with high spatial and high semantic resolution in two directions, which improves the performance in existence of noisy backgrounds. The second is a generative adversarial network (GAN) setting which helps in creating a long-range context between objects, also reducing the influence of background noise on the segmentation result.

The feature representation plays an important role in all building segmentation methods. Hence, several works have been dedicated to the investigation of utilizing height

information as an input. Bittner *et al.* [38] use the fact that spectral and depth features are complementing each other by merging a pansharpened RGB, a nDSM and a PAN in a convolutional network after passing each of the modalities through an FCN with skip-connections. Schuegraf *et al.* [39] even use an 8-channel multi-spectral image in its original resolution, pass it to a deconvolutional layer and merge it with the PAN in a residual block. The residual block's output is passed to a U-shaped network and in parallel, a DSM patch is also passed through a similar network. The outputs of the two streams are concatenated and passed through more convolutional layers to produce the building footprints as the final output.

3.2.2 Deep Learning-based Building Instance Segmentation

More recently, the research community is pushing the bar higher by moving from semantic segmentation of buildings to instance level segmentation of whole buildings and even building sections. In general, every method that extracts building footprints can be combined with the extraction of connected building-blobs to obtain instances. But this conventional approach does not take into account the complexity of some buildings, consisting of multiple simple sections. Hence, we consider the deep learning-based methods, that can separate sections, in this subsection.

3.2.2.1 Mask-RCNN-based

Most of works on building instance segmentation were carried out by utilizing Mask-RCNN [1]. Mask-RCNN uses a set of anchor bounding boxes, which it classifies as instance or not instance and refines their position. Finally, it segments the object inside the predicted bounding box on a pixel level. For example, Amo-Boateng *et al.* [53] have trained Mask-RCNN to segment wide-spread, low-density buildings in aerial imagery of rural Africa. The challenges in rural areas in Africa are the visibility of houses and the various roof-structures. But besides that, it is not hard to distinguish between buildings, since they are largely detached, which is much different in dense urban settlements. Furthermore, the shapes of the resulting masks of Mask-RCNN are irregular and therefore not suitable for many engineering applications. In work Zhao *et al.* [54], the authors propose a post-processing refinement of the resulting building masks to obtain regularized building polygons in vector format. They use Mask-RCNN to get initial building polygons, from which they produce a set of polygons by local manipulation of corners using neighboring vertices. To choose from the resulting polygon solutions, the authors use minimum description length (MDL) optimization. The resulting polygons are indeed regularized, but the experimental evaluation does not include building junctions and the resulting building polygons can not include inner-yards, since multi-polygons are not allowed. Wen *et al.* [55] extend classical Mask-RCNN in two ways. They use rotated bounding boxes instead of horizontal bounding boxes to better obtain a tightly defined region around each building. Furthermore, they integrate atrous convolutions [56] and an inception block [57] into the segmentation branch of Mask-RCNN, which improves segmentation performance. Chen *et al.* [58] perform a multi-step procedure to obtain

improved building instances. In the pre-processing step, they perform super-resolution. Using the up-sampled image data, they apply Mask-RCNN based on a vision transformer backbone [59].

3.2.2.2 Not Mask-RCNN-based

Much work has been done to segment buildings by representing them as polygons. One example is presented by Zhao *et al.* [60], where a graph neural network (GNN) called RSGNN is utilized to vectorize building roof-lines. This approach produces perfectly straight lines and sharp corners by design but also shows many failure cases. RSGNN could not capture curved roof-lines and relied on instance-level input. Another work that uses keypoints to detect building instances from remote sensing imagery is that of Li *et al.* [61]. There, a convolutional neural network (CNN) extracts features from an aerial image before a region proposal network (RPN) provides bounding boxes that likely correspond to locations of building instances. The features of the CNN in the proposed bounding boxes are extracted by regions of interest (RoI) align and an FCN generates a density map for keypoints. In a learning-free post-processing step, this keypoint density map is converted to a building instance polygon. The method produces regular building polygons and achieves high metric values on a publicly available dataset, but the study does not contain enough information to see how the method performs in densely built urban city centers.

The PolyMapper approach [62] directly forecasts buildings and road networks in vector form, but its efficacy on the CrowdAI dataset [63] is not satisfactory. Conversely, approximating shapes in images with polygons (ASIP) [64] surpasses the performance of PolyMapper. ASIP initiates polygons by segmenting the image into convex cells, followed by polygon refinement through an energy function. This function minimizes disparities between the fidelity of each polygon to the input image and its complexity.

A further approach to polygonize buildings presented by Girard *et al.* [65], where a frame field is learned as one output and a segmentation map of building interior and building boundary as another one. The frame field is a field of two vectors per pixel, containing the direction of the tangent and normal vectors of building boundaries. The frame field is used **(a)** to regularize the segmentation result and **(b)** in a polygonization post-processing step. This procedure is able to separate buildings with common borders, but relies on the right choice of thresholds to not either over- or undersegment building sections. While the method of Girard *et al.* [65] has only the RGB image as the input, the study of Sun *et al.* [66] concatenates the nDSM with the RGB before feeding it to the FCN. The integration of depth information leads to higher metric values and more regular polygons.

Zorzi *et al.* [67] present PolyWorld, which is a method for extracting regular building polygons that are suitable to produce vectorized buildings for many geoinformation system (GIS) applications that have a building layer. It has a fully convolutional feature extraction network, a self-attention GNN and an optimal connection network. Despite the good performance of PolyWorld, [68] presents the improved Re:PolyWorld method, which outperforms PolyWorld. Even though PolyWord produces regular building poly-

gons with superior metric values on CrowdAI [63], they do not separate neighboring building sections elegantly, because the permutation matrix, that encodes edge information, can not deal with buildings with shared vertices. Moreover, Re:PolyWorld does not utilize DSM information.

Furthermore, there are multiple works on building section instance segmentation which first segment the building sections and some representation of their borders and then instantiate building objects in a second step. For example, the methodology of Bai *et al.* [69] predicts the energy landscape of an image by a deep neural network with a discrete number of energy levels. Building section instances are obtained by cutting lower energy levels and setting them to background values and then extracting the connected components. Afterwards, the instances are expanded to close the resulting gaps between them. Wagner *et al.* [70] chose a slightly different approach. They use a tailored network architecture of three connected U-Nets to segment the building border, footprint and inner segment. The prediction of borders, which are exaggerated building borders, helps to separate the building well enough. In the post-processing step, the inner segments, which do not overlap, are labeled as instances and are buffered by a number of pixels to represent their original size. In contrast, Iglovikov *et al.* [71] predict the background, building and separation line between nearby buildings in a single network. This leaves the task of separating adjacent building sections open ended.

3.2.3 Building Outline Regularization

Building footprints and sections often have irregular outlines. To solve this problem, there has been significant research on building instance regularization. For instance, Li *et al.* [72] propose using the primary orientation angle along with a straightforward yet effective rectilinearization algorithm. However, their method for computing the primary orientation angle is not robust, as it depends on the minimal point density along the x and y axes after rotating the initial polygon by a candidate angle. The primary orientation direction might have an arbitrary number of vertices, depending on the roughness of the initial polygon.

Other notable work in this area includes Zebedin *et al.* [73], who regularize building outlines by filtering initial lines through a histogram of orientations and removing outliers. The remaining line directions are then used to reconstruct buildings with a regular appearance. This approach is flexible, as it is not limited to right angles. Similarly, Cui *et al.* [74] employ the Hough transform to group an initial set of line segments into two perpendicular sets of parallel lines representing the building boundary. They construct an initial graph with these lines, remove edges in low-contrast regions, and determine the final building boundary by searching for cycles in the graph. This method, however, relies on the completeness of the initial line detection and is restricted to rectangular buildings. Tian *et al.* [75] also use the Hough transform and line segment intersections to form building boundaries, allowing for two arbitrary main orientation directions.

More recent advancements involve end-to-end deep learning approaches for building outline regularization. Marcos *et al.* [76] propose learning the parameterizations of active contour models to refine initial building blobs into regular polygons. Gur *et al.*

[77] develop an end-to-end trainable pipeline that iteratively updates an initial set of points similar to active contour models, although the predicted polygon is not necessarily regularized. Similar to Marcos *et al.* [76] and Gur *et al.* [77], Hatamizadeh *et al.* [78] propose an end-to-end trainable active contour model-based building boundary extraction method that extends capabilities to various buildings within a patch, as initial contours are predicted by a CNN. Additionally, Zhao *et al.* [79] enhance PolyMapper, which uses an recurrent neural network (RNN) to recursively predict vertices.

3.3 3D Building Reconstruction

The 3D reconstruction of buildings has gained increasing attention in the last years. One main criterion to distinguish work in 3D reconstruction of building is the LoD according to the CityGML standard. Since the dissertation focuses on LoD-1 and -2, this section includes both of them.

3.3.1 LoD-1 Reconstruction

Reconstructing buildings in LoD-1 is a well-studied field. Dukai *et al.* [80] argue that LoD-1 models are sufficient or even advantageous for some applications. They provide a flexible service for LoD-1 model generation, where different approaches for base- and extrusion-height are provided for all approximately 10 million buildings of the Netherlands, which has good data coverage. Peters *et al.* [81] also provide nation-wide LoD-1 models in the Netherlands. Their method relies on 2D polygons and LiDAR point cloud and is highly sensitive to the inputs. Hence, they provide a software to review the quality of the input at different processing steps. Bagheri *et al.* [82] fuse OSM data with height information from either dense stereo matching or synthetic apperture radar (SAR). None of the above methods rely on deep learning. On the opposite, Yu *et al.* [83] propose a three stage approach that relies heavily on deep learning. First, the DSM is reconstructed using a CNN. Next, building edges are robustly extracted by another CNN. The building edges are vectorized and regularized. Finally, the building height is extracted from the generated DSM.

3.3.2 LoD-2 Reconstruction

LoD-1 reconstruction is not suffcicient for applications that require detailed information about roof structure. Hence, LoD-2 reconstruction is a highly pertinent task. The derivation of roof planes from imagery and/or height information is often tightly connected to the reconstruction of buildings in LoD-2. Hence, roof plane segmentation is studied in terms of a secondary task for LoD-2 reconstruction. Therefore, we do not distinguish the works dedicated to LoD-2 reconstruction from those dedicated to roof plane segmentation. The LoD-2 reconstruction has received relatively limited attention in remote sensing research. Nex *et al.* [5] presented a study that doesn't involve machine learning but depends on manually designed features to recreate 3D building rooftops.

This approach relies on utilizing the near-infrared channel, which may not be universally available. Additionally, the method struggles to accurately handle highly complex building structures. Arefi *et al.* [6] also employ a learning-free technique to create LoD-2 building reconstructions by utilizing both the DSM and the orthorectified image. Despite generating improved regular reconstructed buildings, this learning-free approach depends on manually designed features and consequently lacks robustness when encountering significant variations in the input data. Peters *et al.* [81] proposed a method for the reconstruction of buildings in LoD-2 with building sections and LiDAR point clouds as input. They use a region growing algorithm to partition the footprints into roof planes and detect their intersection lines. Another work that relies on normalized point clouds for LoD-2 reconstruction is that of Li *et al.* [84], where building primitives from a list of rooftypes are optimized given the point cloud at hand.

In the study conducted by Alidoost *et al.* [85], a single aerial image is employed to create LoD-2 building models. Their methodology involves initially training two distinct neural networks: one for estimating building heights and the other for extracting roof features such as eaves, ridges, and hips. Subsequently, a model-based technique is utilized to generate 3D building models. Recently, LoD-2 reconstruction was performed using deep learning methods by Lussange *et al.* [86], that use two consecutive Mask-RCNNs, called keypoint inference by segmentation (KIBS). The first Mask-RCNN performs roof plane segmentation, while the second detects roof plane corners and their respective heights in a categorical manner. Although the LoD-2 reconstruction results look promising, the resulting 3D geometries are not necessarily connected to individual buildings. Furthermore, KIBS is dependent on the oblique view image and hence doesn't generalize to dissimilar viewing angles than those in the training set. Even though learning-based approaches can achieve consistent city models, it is worth noting that the accuracy of the predicted heights solely from an image remains to be a potential limitation. Therefore, there is a need to use heights directly from or estimated based on a DSM. In Gui *et al.* [18], buildings are segmented by a semantic segmentation neural network. LoD-2 models are derived using learning-free methods and allowing the integration of OSM data. Their method is based on an ortho image and an nDSM as input.

3.4 Attention in Building Segmentation

In recent years, different flavors of attention have been implemented for building segmentation. One such work is Chen *et al.* [87], where the authors use self-attention for the semantic segmentation of buildings in optical remote sensing imagery. In Dai *et al.* [88], the authors use a location channel attention module to improve the segmentation of building edges in building and water segmentation. Another work that uses a combination of spatial and channel attention is Pan *et al.* [89]. Besides these cases of using attention in CNNs modules, Sun *et al.* [90] use a multi-resolution transformer that heavily depends on the attention mechanism for building and road segmentation. In Wang *et al.* [91], a hybrid model combines hierarchical feature extraction of CNNs with global

context modeling of transformers for urban scene semantic segmentation. Yet, all works introduced here use only spectral features for attention computations.

3.5 Summary

Building information extraction and reconstruction is widely studied. The methods used to segment buildings in image focus mainly on footprint segmentation. In this dissertation, the aim is to segment buildings on a section instance-level in a way that is tailored to building. For example, in this thesis, the building sections are not segmented by using bounding boxes, but separation lines, which allows seamless connection of neighboring building sections. The named methods for 3D reconstruction of buildings mostly rely on hand-crafted features, which are not robust to large variations in the input, whereas the dissertation proposes deep learning-based approaches, which are more robust. Furthermore, in the above-mentioned domains, the fusion of spectral imagery and photogrammetric DSM are yet to be improved. On the other hand, the dissertation proposed multiple ways to introduce height information into the pipeline of building segmentation and 3D reconstruction.

4 Building Section Instance Segmentation

In this chapter, a novel methodology for segmenting building sections based on aerial and satellite imagery, and photogrammetric digital surface model (DSM) is presented. It is mainly based on the peer-reviewed journal paper:

[8]: ***P. Schuegraf**, S. Zorzi, F. Fraundorfer, and K. Bittner, "Deep learning for the automatic division of building constructions into sections on remote sensing images," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 16, pp. 7186–7200, 2023.*

Furthermore, an application study is included in Section 4.3. There, the aforementioned methodology [8] is combined with additional training data and evaluated on highly dense and complex informal and formal built areas. Section 4.3 is based on the peer-reviewed conference paper:

[9]: ***P. Schuegraf**, D. Stiller, J. Tian, T. Stark, M. Wurm, H. Taubenböck, K. Bittner, "Ai-based building instance segmentation in formal and informal settlements," IEEE International Geoscience and Remote Sensing Symposium, 2024.*

Moreover, a method to regularize footprint, using a deep neural network for orientation prediction is proposed in Section 4.4. Section 4.4 is based on the peer-reviewed conference paper:

[10]: ***P. Schuegraf**, Z. Li, J. Tian, J. Shan, and K. Bittner, "Rectilinear building footprint regularization using deep learning," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2024.*

4.1 Problem Statement

Modern urban settlements are densely built, driven by urbanization. Mixed with different architectural styles and building requirements, densely built up cities show a variety of geometric and spectral building appearances. Most often, complex buildings are assembled of smaller, more manageable constructions, as can be seen in Figure 4.1. It turns out, that for endeavours to systematize the building inventories of cities, for example, to reconstruct building models in a level of detail (LoD) of higher than zero, it is rudimentary to be able to locate the outlines of the building primitives. We define these building primitives as parts of larger building constructions as provided by federal cadastre data, where the individual building is defined by a house number. The assignment of house numbers to buildings depends on societal factors, which are not always visible. Hence, the separation between buildings by house numbers is not always visible in the top-down view, which is a source of ambiguity in the judgement of the success of our method. On the other hand, if the touching border between two differently numbered

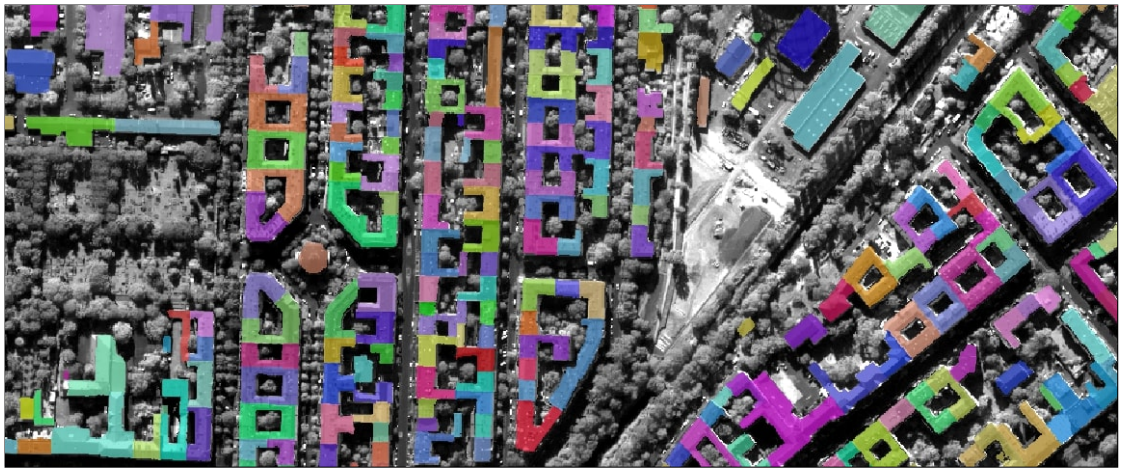


Figure 4.1: Overlay of our method's building section instance segmentation predictions over a panchromatic image in our test area showing Berlin, Germany.

houses is not visible, the two parts can safely be regarded as a single building for applications that require building outlines as a primary feature. Those applications include building 3D reconstruction, flow simulation (e.g. water, wind or 5G-waves), solar panel recommendation and natural crisis intervention. For those applications, a correct and detailed information of each building section is essential. For example, roof forms of the individual sections need to be distinguished to recommend the solar panel assembly side.

In applications such as building reconstruction, disaster monitoring, city planning and environment modelling for autonomous driving, building footprints are crucial. Most works on building footprint extraction produce raster outputs, whereas applications require them in vector format. A robust approach to obtain buildings in vector format is to first predict raster buildings using a neural network and then applying postprocessing that outputs polygons. The results achieved by conventional methods are either limited in terms of generalization capacity [73–75] or are not restricted sufficiently to prior knowledge of regularity [68, 76–79].

A manual delineation of the sections' outlines is very time consuming and expensive, owing to the fact that cities experience rapid growth and infrastructure advances. Accordingly, an automatic extraction of building sections is a highly pertinent task. But currently, segmenting multiple gapless buildings as one large building is set as a standard [6, 38, 39]. But remotely sensed images show much more beneficial features for building segmentation, since they are equipped with cameras with increasingly high spatial resolution with a ground sampling distance (GSD) as small as 0.3 m. We already showed in our previous work [2], that these features are appropriate to discern building sections visually. Even though building section instance segmentation based on separation lines outperforms the Mask-RCNN, it lacks regularity of the predicted sections. Zhang *et al.* [48] use the perceptual loss term, originally introduced in [15] for thin structure segmentation, to regularize building footprints. To complement optical data, pixel-wise height features like DSM can be used in building segmentation. Many optical satellites deliver

multi-view data, which is used to compute DSMs. Furthermore, spectral and height information have been leveraged jointly to detect buildings [6, 38, 39]. In our previous method [2], we leveraged the DSM to detect junctions of neighboring buildings.

4.2 Deep Learning for the Automatic Division of Building Constructions into Sections

4.2.1 Contributions

In this section, we extend our findings from Schuegraf *et al.* [2] by the following contributions:

- Our main contribution is the in-depth description of a method which uses the separation line between building instances instead of the whole borders [65] to obtain building sections. Furthermore, our method can handle an arbitrary number of instances as opposed to bounding-box based methods [1].
- We introduce a framework consisting of a deep learning network and a post-processing that extracts building footprints and building section separation lines from very high resolution (VHR) satellite images and DSMs. The framework is designed to produce sharp building edges, as well as complete and straight separation lines. Our framework utilizes the watershed transform together with multiple post-processing steps. The whole workflow is depicted in Figure 5.1.
- We fuse spectral and depth features by fusing them by summation at the skip-connections and the full scale aggregated skip-connections (FSA) from the U-Net-3+ in our proposed SkipFuse-U-Net-3+ architecture.
- We leverage an additional, feature-extraction-based loss-term that not only improves the regularity of separation lines at building junctions, but also regularizes building section shapes.
- We show the generalization capability of our method by evaluating our model, that was previously trained on data from Berlin, Germany, on an additional test area in Lyon, France, which includes different architectural styles and topographies and was acquired by a different sensor than the training data. The corresponding ground truth is provided by the municipality of Lyon.
- We evaluate our method on an open-source dataset consisting of aerial images, aerial-based DSMs and dense pixel-wise building section instance masks in the area of the German federal state of North-Rhine-Westphalia. Our benchmark dataset focuses on areas with highly complex urban building structures.
- Our results are suitable for vectorization and rendering them into LoD 1 models.

Regarding our previous publication [2], we highlight the following differences to this section:

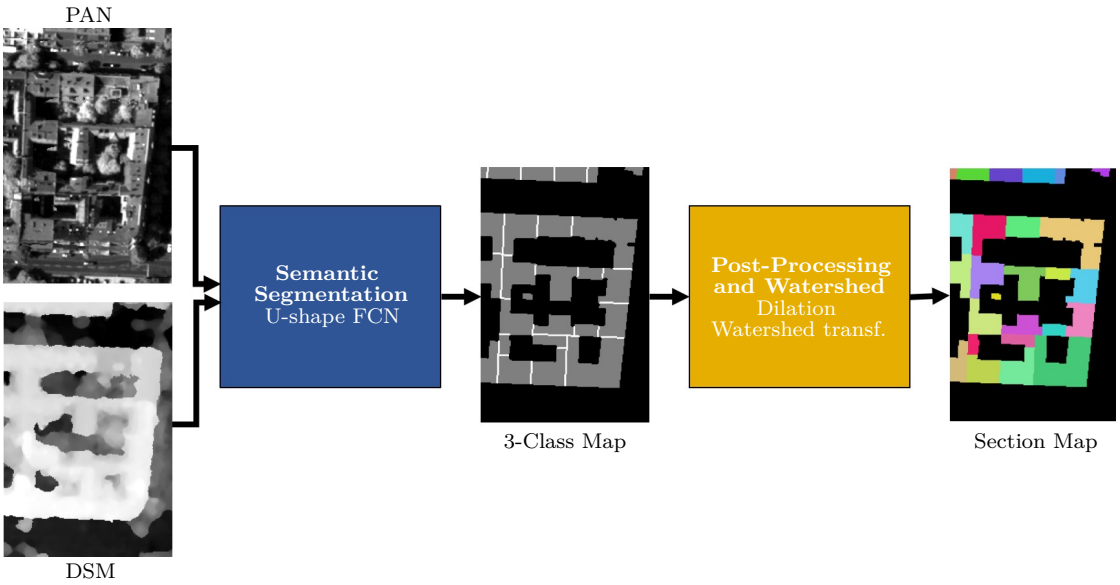


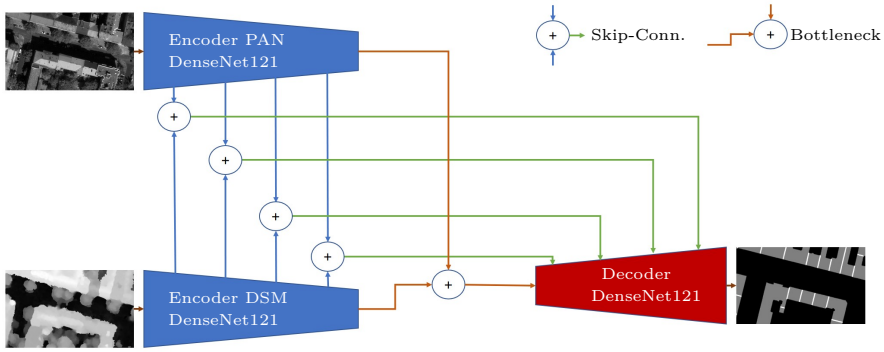
Figure 4.2: The overall workflow of our proposed methodology. First, a fully convolutional neural network (FCN) extracts a map of the three classes background, building and separation line. In the second step, the separation line is dilated and the watershed transform is used to obtain building section instances.

- We add a regularization term to our loss function to enhance building section shapes and improve the separation line.
- We evaluate our method on two additional datasets, where one originates from a satellite and the other one is aerial.
- We evaluate our trained model on satellite imagery of the city of Lyon, France to show the models generalization capability.
- After comparing our model to the Mask-RCNN in the previous work, now we compare with the more advanced framefield learning model.
- We provide more detailed descriptions of the presented method.

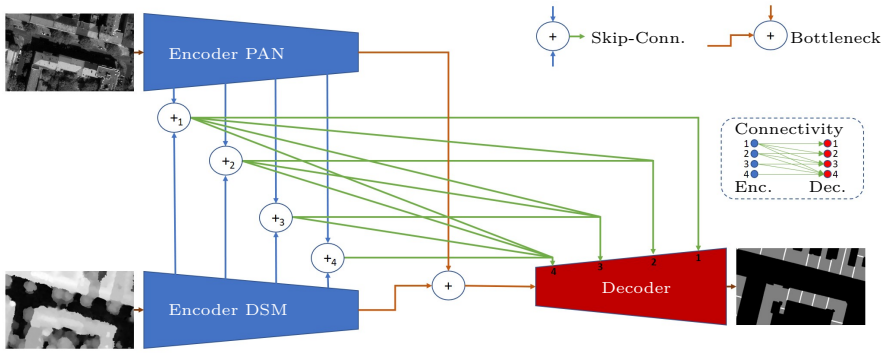
4.2.2 Methodology

4.2.2.1 Network Architecture

The network architecture is one of the most important parts of a deep learning based method. The SkipFuse-DenseNet121-U-Net was successfully used for building section instance segmentation in [2], but its skip-connections have access to only a single scale of the encoder. In work [13], FSAs are introduced to overcome this limitation by merging feature maps from all higher scale feature maps at each skip-connection. We propose to use the U-Net-3+ architecture from [13] for building section instance segmentation, with



(a) SkipFuse-DenseNet121-U-Net



(b) SkipFuse-U-Net-3+

Figure 4.3: Visualization of the SkipFuse-DenseNet121-U-Net and the SkipFuse-U-Net-3+. Both take as the input a patch of a spectral, i.e. panchromatic or RGB image and a patch of the corresponding DSM. The two modalities are summed at the skip-connections in both architectures. At the bottleneck, the features from both modalities are also summed for both architectures, but the SkipFuse-U-Net-3+ uses full scale aggregation.

a second encoder for DSM information integration. One encoder receives the panchromatic patch and the other receives DSM patch as in the SkipFuse-DenseNet121-U-Net. The feature maps at four different scales are summed from the two encoders and used as the input to the FSAs. We call the resulting architecture SkipFuse-U-Net-3+ (see Figure 4.18 **(b)**).

4.2.2.2 Loss Function

We train our proposed model on a combination of segmentation and regularization terms

$$\begin{aligned}\mathcal{L}_{TOTAL} = \mathcal{L}_{CE} + \mathcal{L}_{GD} + \lambda_{BM} \cdot (\mathcal{L}_{TOP})_{BM} \\ + \lambda_{TB} \cdot (\mathcal{L}_{TOP})_{TB},\end{aligned} \tag{4.1}$$

where λ regulates the contribution of the topological term towards the overall training process. In the following paragraphs, we provide details about the other components of $\mathcal{L}_{TOTAL}$. Note that for a compact notation, we do not note the parameter list for each

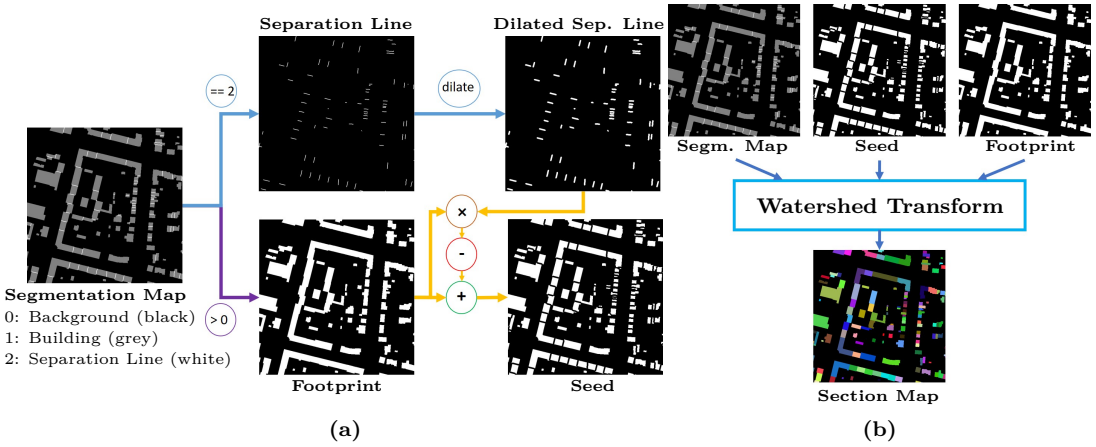


Figure 4.4: A visualization of the workflow of building section creation from 3-class maps by using the watershed transform and morphological processing. First, a seed is generated from the dilated separation line and the building footprints (see Figure 4.4a). Then, the 3-class maps together with the seed and the footprint as a mask is sent to the watershed transform to produce building section instances.

of the losses, which is (x, y, p) for $(\mathcal{L}_{TOP})_C$ and $\mathcal{L}_{GD}$ and (x, y, p, w) for $\mathcal{L}_{CE}$.

A common loss function for semantic tasks is the weighted cross-entropy loss

$$\mathcal{L}_{CE}(x, y, p, w) = -\sum_i y_i w_i \cdot log(p(x_i)), \tag{4.2}$$

where x is the input tensor, y is the ground truth and $p(.)$ denotes the softmax-output of the neural network model, i denotes the respective class and w is a vector of manually chosen loss balancing factors, which we set to $[1, 1, 4]$. The cross-entropy loss is used to maximize the predicted probability of the desired class at a certain pixel which often leads to rounded or blurry object boundaries. In contrary, the generalized dice loss [92]

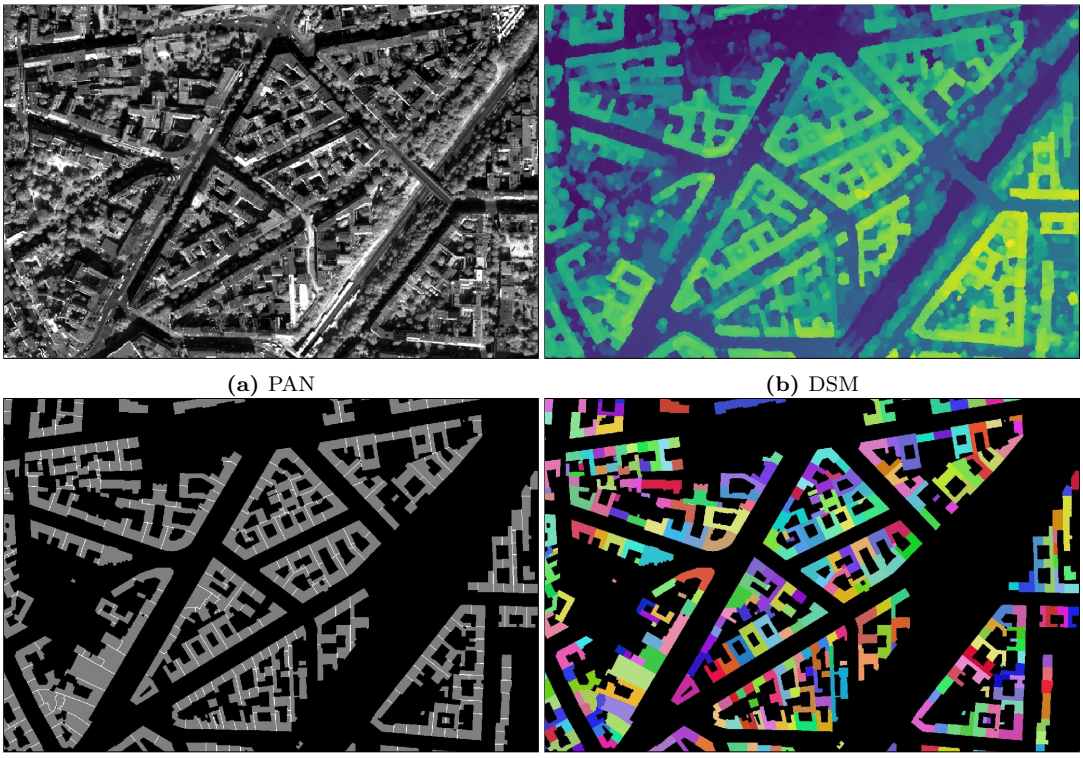
$$\mathcal{L}_{GD}(x, y, p) = 1 - 2 \cdot \frac{\sum_i v_i \sum_n y_{in} \cdot p(x_i)_n}{\sum_i v_i \sum_n y_{in} + p(x_i)_n}, \tag{4.3}$$

where v_i is the inverse frequency of the class i, is designed for crisp boundary detection. Even though $\mathcal{L}_{CE}$ and $\mathcal{L}_{GD}$ guide the training process towards confident and comprehensive predictions, they do not include spatial context beyond pixel level to the optimization target. But in remote sensing, the final result is demanded to be of spatial coherence. Hence, we aim to use an additional loss term which enforces local regularity with regards to the structure in the ground truth.

This loss-term is called topological loss [15]

$$(\mathcal{L}_{TOP})_C(x, y, p) = \sum_{n=1}^{N} \sum_{m=1}^{M_n} ||l_n^m(y_C) - l_n^m(p(x)_C)||_2^2, \tag{4.4}$$

where l_n^m denotes feature map m of layer n of a pre-trained convolutional neural network



(a) PAN **(b)** DSM

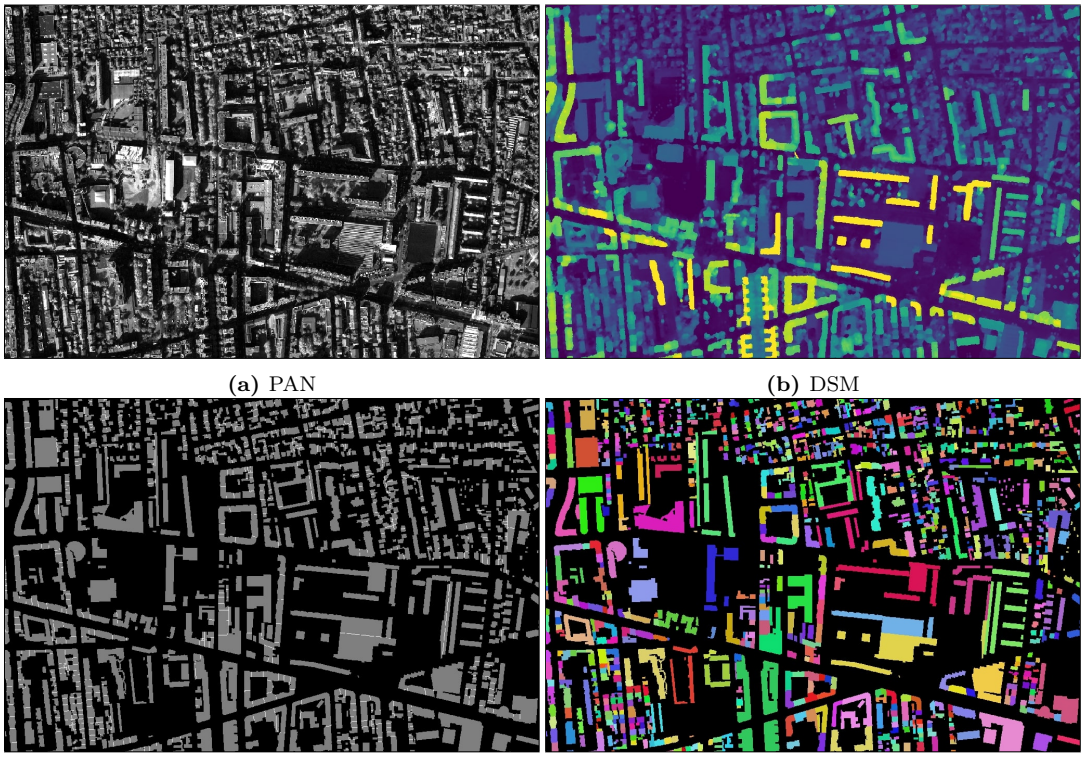
(c) GT Footprint **(d)** GT section instances

Figure 4.5: Excerpt from our WorldView-1 building section instance segmentation dataset. We use this dataset for experimental evaluation of our method and ablation.

(CNN) and C is the class on which to apply the term. For the sake of simplicity, we omit C if we mention the version of the topology loss for binary segmentation tasks as in work [15]. The topological loss penalizes structural irregularities like over-rounded corners and bumpy edges by comparing geometrical features of the segmentation patch with those of the ground truth patch. Hence, it serves as regularization with regard to typical building shapes and their separation lines. In the work of Mosinska *et al.* [15], the networks output is concatenated three times with itself to obtain three input channels, since the feature extracting CNN is pre-trained on RGB images and therefore, a single input is not enough. We follow that strategy too. But different to Mosinska *et al.* [15], we not only use $\mathcal{L}_{TOP}$ to improve the thin line, in our case the separation line, but also the building section mask. Hence, we evaluate $\mathcal{L}_{TOP}$ once on the building section output channel and once on the separation line output channel of our network. We set l_n^m to all channels of the feature maps after each of the five max pooling operations of the VGG19 feature extraction network, which was pretrained on the imagenet dataset.

4.2.2.3 Post-Processing

Our FCN architecture produces a 2-dimensional map of three classes, which can not directly be interpreted as a map of building sections instances. Instead, we use the



(a) PAN **(b)** DSM

(c) GT Footprint **(d)** GT section instances

Figure 4.6: Excerpt from our test set in Lyon, France. We use this dataset to test the generalization capability of our model.

watershed transform [3] to use the network's output for building section instance segmentation. The watershed transform takes a single channel intensity image together with a seed image and a mask as its input and interprets the intensities as a topological surface. Then, a flooding of the surface is simulated, where water starts flooding from the seeds and is caught in basins which are separated by watershed lines that correspond to high image intensities. The mask restricts the virtual water to flow in specific regions. The areas enclosed by watershed lines are then interpreted as the objects. As depicted in Figure 4.4, for building section instance segmentation, the intensity image is replaced by the 3-class map of background, building and separation line. We obtain the seed image by subtraction of the dilated binary separation line mask from the binary building mask in the foreground regions of the binary building mask.

The dilation, which is carried out using a disk-shaped structuring element helps to fill holes in the separation lines, leading to better separation between sections but also badly influences the sections by deforming the seed. With the watershed transform, two predicted, neighboring building sections will have no gap between them. This facilitates regularity of the building sections' separation.

City	Split	Region of Interest (EPSG:32632)	Area km^2
Bonn, Germany	Training	363675.0:372281.0,5614866.5:5629411.5	125
Cologne, Germany	Training 1	352000.25:354777.25,5638999.75:5649999.75	31
Cologne, Germany	Training 2	355277.25:361000.25,5638999.75:5649999.75	34
Cologne, Germany	Training 3	354777.25:355277.25,5638999.75:5646999.75	4
Cologne, Germany	Validation	354777.25:355277.25,5646999.75:5648999.75	1
Cologne, Germany	Test	354777.25:355277.25,5648999.75:5649999.75	0.5

Table 4.1: Geographic coordinates of the **Aer50-NRW** dataset. The provided format is $x_{min} : x_{max}, y_{min} : y_{max}$.

4.2.2.4 Vectorized LoD-1 Generation

After producing a map of building sections in raster format, we aim to generate building polygons and LoD-1 models to demonstrate the possible usage in remote sensing and geoinformation system (GIS) applications. We achieve this by extracting the coordinates of the boundary polygons of each building section by a simple tree search and the removal of redundant corners if they do not lead to a change of direction in the boundary polygon. To integrate height information, we mask the input DSM with the binary map for the corresponding building section. Then, we remove the lowest two percents of height values to reduce the influence of pixels which are mistakenly segmented as building pixels, but belong to the background. The remaining height values are averaged to obtain the mean building height. For our visualization, we model the ground by removing above terrain objects with morphological filters as in the work of Qin *et al.* [93].

4.2.3 Experiments

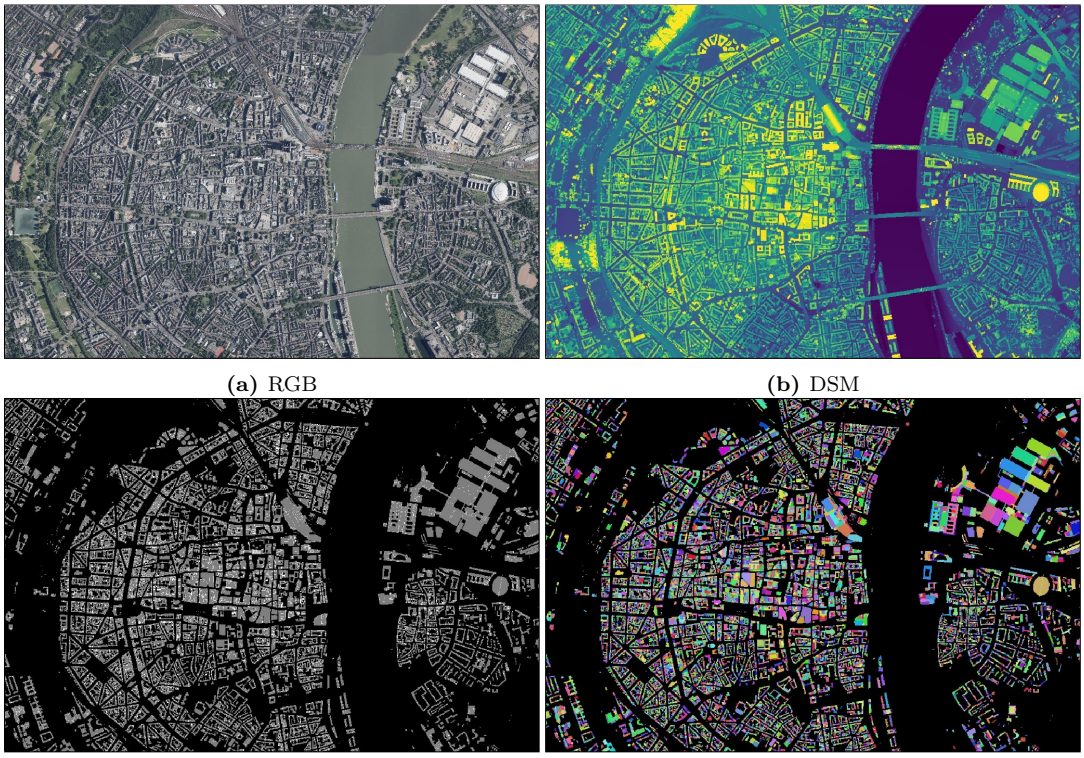
4.2.3.1 Data

We use three different experimental datasets. **Sat50-Berlin** consists of a panchromatic and a DSM images as inputs, paired with a building mask image as a ground truth (GT) with values in $\{0, 1, 2\}$. By stereo matching of multiple different views over the same scene [19], the DSM was computed. A building instance is defined by the coordinates of houses with individual addresses, provided by the German Federal Agency for Cartography and Geodesy [1].

The ground truth raster is converted to the same resolution and size as the input. The raster values are 0 for background and integer values for each individual building, thus every pixel is linked to an individual building instance. Then, the touching borders between buildings are initially calculated by searching for neighbors for each instance. In a refinement step, morphology is applied to remove gaps between building sections. Afterwards, these borders are merged with a binary image of the building mask, so the result is a raster image with three values: 0 for background, 1 for building mask and 2 for touching borders. See an excerpt of **Sat50-Berlin** in Figure 4.5.

Panchromatic images and DSMs of our satellite dataset **Sat50-Berlin** originate from

[1] http://www.businesslocationcenter.de/downloadportal



(a) RGB **(b)** DSM

(c) GT Footprint **(d)** GT section instances

Figure 4.7: Excerpt from our aerial building section instance segmentation dataset **Aer50-NRW**. We use this dataset to provide a public benchmark for the evaluation of building section instance segmentation methods that rely not only on spectral information, but also on depth information like from a DSM.

high resolution WorldView-1 data with a GSD of 0.5 m and shows the city of Berlin, Germany. The selected training area covers an area of 387 km^2. Two non-overlapping tiles, that are separate from the training data, are cropped from the WorldView-1 image for each of validation and test. The validation and test tile cover an area of 7 km^2 and 2 km^2 respectively. Additionally, to test the generalization capability of our method, we use a WorldView-2 panchromatic image and DSM of the city of Lyon, France. See an overview of the selected area in Figure 4.6.

As the second dataset, we use **Sat30-Berlin** with GSD of 30 cm as in our previous work [2] to serve as a basis for comparison. In work [2], the ground truth was shifted more with respect to the image in the training, validation and test set, which we corrected and hence in this work, and, as a result, got slightly different metric values. Apart from the shifting, the ground truth was generated in the same ways as for **Sat50-Berlin**. The training, validation and test set cover an area of 81 km^2, 13 km^2 and 2 km^2, respectively. Furthermore, the test set of **Sat30-Berlin** matches geographically with the test set of **Sat50-Berlin**.

Our last dataset, **Aer50-NRW** (see Figure 4.7) is an open source dataset consisting of aerial RGB imagery and an aerial DSM showing the cities of Cologne and Bonn,

Germany, with a ground sampling distance of 0.5 m. The original GSD of the RGB images is 0.1 m. Hence, we apply cubic interpolation in the downsampling step to acquire high quality RGB images in 0.5 m GSD. The ground truth is arranged in the same way as in **Sat50-Berlin**. The utilized RGB, DSM and ground truth data are freely available, supplied by the government of the federal state North-Rhine-Westphalia, Germany [2]. To allow comparison of different methods that utilize depth and spectral information for building instance segmentation, we provide the geographic coordinates in Table 4.1. Furthermore, we manually corrected some parts of the test area to obtain more meaningful metric values [3].

The final step of preparing the three datasets is to normalize all input data before training. Since generalization is an important factor of the quality of the final model's prediction, we leverage a normalization scheme that is robust to variations in input images. We compute the 2nd and 98th percentile of the whole panchromatic or RGB image and set all pixels with values below the 2nd percentile to the 2nd percentile and all pixels with values above the 98th percentile to the 98th percentile. Then, we normalize each patch to the range $[-1, 1]$, where -1 represents the 2nd percentile and 1 represents the 98th percentile. Note that, for the RGB data, we do percentile cutting and normalization for each band individually. Since we do not rely on absolute height information, we normalize the DSM to range $[-1, 1]$, where -1 represents the minimum height of the patch and 1 represents the maximum height of the patch.

4.2.3.2 Training Process

In this subsection, we point out the elements of the training that are similar for all experiments if not stated otherwise. For optimization, we used the Adam [32] optimizer with $beta_1 = 0.9$, $beta_2 = 0.99$ and an initial learning rate of 0.001. To increase the speed of the training, an exponential learning rate decay of 0.9 per epoch is used if not stated otherwise. The training is done in a mini-batch procedure for 50 epochs with validation after each epoch. For some experiments, the number of epochs deviates from 50, if either a different model or a different data augmentation strategy is chosen. To increase the variety of training samples, we do augmentation by random shifting of the patches, that are cut out from the remote sensing image. The shift is uniformly randomly distributed in between 0 and half of the patch height or with for both the vertical or horizontal direction.

4.2.3.3 Metrics

To evaluate the performance of experiments, several metrics are calculated for each class separately. For each class, a binary mask is computed for the ground truth and class predictions and the corresponding number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) are calculated.

[2] https://www.opengeodata.nrw.de/produkte/geobasis
[3] https://github.com/dlrPHS/Aer50-NRW

The precision

$$Pre_c = \frac{TP}{TP + FP} \tag{4.5}$$

is a measure for how good the segmentation method does not predict the negatives as positives for a particular class, the recall

$$Rec_c = \frac{TP}{TP + FN} \tag{4.6}$$

gives insight into how complete the pixels of a certain class are segmented, the $F1$

$$F1_c = 2 \times \frac{Pre_c * Rec_c}{Pre_c + Rec_c} \tag{4.7}$$

is a metric which combines precision and recall, such that the $F1$ is drawn stronger towards the lower of the two and the intersection over union (IoU)

$$IoU_c = \frac{TP}{TP + FP + FN} \tag{4.8}$$

is the ratio of overlapping pixels of the prediction and ground truth over their union.

Next to the background, there are the classes building mask (BM) and touching borders (TB). All the previous metrics are listed for the class $c \in \{BM, TB\}$. Since in this work we focus on instance segmentation, the common instance metrics mean average precision (mAP) and mean average recall (mAR) are used to evaluate the models on an instance level. Both these metrics rely on computing the IoU of predicted instance masks and ground truth masks, where the mAP tends to punish predicted masks, which cannot reach a certain threshold and the mAR has a reciprocal relation with the number of ground truth instances that are not matched by any of the predicted instances in terms of a threshold. To get a balanced metric, the $F1_{IS}$ is computed similar as in Equation 4.8, but with the mAP and mAR instead of precision and recall.

4.2.3.4 Inference Process

After training the models, we test them on unseen data to evaluate their performance. We pre-selected a single tile as the test area for each of the data set in the same city as the training data, except one experiment, where the cross-city generalization capacity is evaluated by predicting building section instances in a different city. The tile is split in patches of size 512×512 pixels. The patches are overlapping by 256 pixels in horizontal and vertical dimension each. The resulting masks are averaged at the overlapping regions. After the computation of the prediction scores for the three different classes background, building model and touching borders, the *argmax* classifier is applied to obtain a class label at each pixel. Next, dilation and watershed are applied to complete separation lines and obtain instances from the semantic output. To obtain quantitative evaluation, we use the same metrics as in work [2]. Note that the subscript IS denotes

instance segmentation and is evaluated per instance, whereas the corresponding semantic segmentation metric is evaluated per pixel. In addition to the quantitative view, we aim for a more comprehensive presentation and discussion of the experiment's results by visualizing them.

4.2.3.5 Experiment Descriptions

The Mask-RCNN [1] is a common instance segmentation network. In our previous work [2], we already showed that our usage of a separation line is superior to the bounding-box based approach of the Mask-RCNN. On the **Sat30-Berlin** dataset, it achieves an mAP of 0.27, leveraging only the RGB as the input image and a U-Net-ResNet50 as the backbone network, as compared to 0.33 mAP of an RGB-based U-Net-ResNet50 utilized in the touching borders setting with watershed post-processing. Hence, we omit the Mask-RCNN experiment here to focus on the following experiments.

First, we describe the experiments that where carried out on our WorldView-1 based satellite dataset **Sat50-Berlin**. We use this dataset, since a larger area is covered and hence, more diversity is included by this dataset than by our **Sat30-Berlin** dataset.

We evaluate the SkipFuse-DenseNet121-U-Net trained with several different loss functions, enhance its performance by stronger data augmentation and then change the architecture to demonstrate the effectiveness of our proposed SkipFuse-U-Net-3+ architecture for building section instance segmentation.

In the first experiment, we train the SkipFuse-DenseNet121-U-Net architecture using $\mathcal{L}_{CE}$ (Equation (4.2)) and $\mathcal{L}_{GD}$ (Equation (4.3)) as the optimization target. We regard this experiment as our baseline and to test that our method not only works with RGB and DSM modalities as in our previous work [2] but also with PAN and DSM modalities.

To show that the $(\mathcal{L}_{TOP})_{TB}$ (see Equation (4.4)) regularizes separation line segmentation at building junctions, we add it to the loss function with which we train the SkipFuse-DenseNet121-U-Net. We set $\lambda_{TB} = 0.2$ to bring the regularizing term to a similar scale as the pixel wise term.

The topology loss was originally developed to improve the segmentation of thin structures. But since it uses not only the thin features of its pretrained feature extractor, we argue that it is also useful for the regularization of building section segmentation. Therefore, we add the topology loss terms $(\mathcal{L}_{TOP})_{TB}$ and $(\mathcal{L}_{TOP})_{BM}$ (Equation (4.4)) to $\mathcal{L}_{CE}$ (Equation (4.2)) and $\mathcal{L}_{GD}$ (Equation (4.3)) for the building class with weight $\lambda_{BM} = 0.1$ and the separation line class with weight $\lambda_{TB} = 0.2$.

To further enhance the performance of our SkipFuse-DenseNet121-U-Net, we add data augmentation. Additionally, to the already enabled random patching, we rotate all input and ground truth patches by a random angle between 0 and 360 degrees, to increase the variety of training samples. Due to more variety in the training dataset, we increase the learning rate decay rate of the exponential scheduler from 0.9 to 0.99, which helps exposing the model to a greater diversity of training samples in a tight range of learning rate values. In Table 4.2 and Table 4.3, the additional augmentation along with the adapted training schedule is noted with "+Aug". We select the model after 212 epochs of training for testing, since it has high evaluation metrics on the validation set.

As the next experiment, we change the architecture of our segmentation model. The inputs and outputs are structured similar as in the SkipFuse-DenseNet121, but the SkipFuse-U-Net-3+ uses FSA to aggregate more scale information at the skip-connections. To leverage FSA for building section instance segmentation, we train the SkipFuse-U-Net-3+ with the same schedule and augmentation strategy as in the previous paragraph.

Automatic building section instance segmentation methods need to generalize not only to previously unseen inputs with similar roof-structures, but also to total different roof styles. Therefore, we evaluate the trained SkipFuse-U-Net-3+ as described in the previous paragraph on panchromatic WorldView-2 imagery and DSM from the city of Lyon, France. While carrying out this experiment, we noticed that the model gives weaker responses in the touching borders channel of its output. Therefore, in addition to the argmax classifier, we add all pixels to the touching borders class that achieve a softmax score of higher than 0.25. Hence, during inference, we compute the output of four versions of each patch. In detail, the three applied transformations are horizontal flipping, vertical flipping and both horizontal and vertical flipping. The network's output scores are averaged over the four resulting outputs.

Since we want to demonstrate that our proposed architecture and training scheme is also applicable to pansharpened satellite imagery, we carry out two experiments.

To have a baseline, we retrain our model from [2] on the **Sat30-Berlin** dataset as described in [2].

Next, we demonstrate how our model, trained similar as on **Sat50-Berlin** but on the **Sat30-Berlin** dataset, is capable to predict high quality building section instances from pansharpened RGB imagery together with the DSM.

Model	Dataset	Loss	$F1_{BM}$	IoU_{BM}	$F1_{TB}$	IoU_{TB}	$meanIoU$
SkipFuse-DenseNet121-U-Net	Sat50-Berlin	$\mathcal{L}_{CE} + \mathcal{L}_{GD}$	0.9061	0.8284	0.5239	0.3549	0.7067
SkipFuse-DenseNet121-U-Net	Sat50-Berlin	$\mathcal{L}_{CE} + \mathcal{L}_{GD} + (\mathcal{L}_{TOP})_{BLD}$	0.9059	0.828	0.5247	0.3557	0.7067
SkipFuse-DenseNet121-U-Net	Sat50-Berlin	$\mathcal{L}_{CE}+\mathcal{L}_{GD}+(\mathcal{L}_{TOP})_{TB}+(\mathcal{L}_{TOP})_{BM}$	0.9107	0.836	0.4914	0.3257	0.7007
SkipFuse-DenseNet121-U-Net	Sat50-Berlin + Aug	$\mathcal{L}_{CE}+\mathcal{L}_{GD}+(\mathcal{L}_{TOP})_{TB}+(\mathcal{L}_{TOP})_{BM}$	0.9136	0.8409	0.5444	0.3740	0.7186
SkipFuse-U-Net-3+	Sat50-Berlin + Aug	$\mathcal{L}_{CE}+\mathcal{L}_{GD}+(\mathcal{L}_{TOP})_{TB}+(\mathcal{L}_{TOP})_{BM}$	**0.9142**	**0.8420**	**0.5618**	**0.3906**	**0.7246**

Table 4.2: Semantic segmentation metric results on **Sat50-Berlin**.

Model	Dataset	R	Loss	$F1_{IS}$	mAP	mAR
SkipFuse-DenseNet121-U-Net	Sat50-Berlin	6	$\mathcal{L}_{CE} + \mathcal{L}_{GD}$	0.3916	0.3453	0.4523
SkipFuse-DenseNet121-U-Net	Sat50-Berlin	6	$\mathcal{L}_{CE} + \mathcal{L}_{GD} + (\mathcal{L}_{TOP})_{BLD}$	0.4089	0.3664	0.4626
SkipFuse-DenseNet121-U-Net	Sat50-Berlin	5	$\mathcal{L}_{CE}+\mathcal{L}_{GD}+(\mathcal{L}_{TOP})_{TB}+(\mathcal{L}_{TOP})_{BM}$	0.4144	0.3672	0.4756
SkipFuse-DenseNet121-U-Net	Sat50-Berlin + Aug	5	$\mathcal{L}_{CE}+\mathcal{L}_{GD}+(\mathcal{L}_{TOP})_{TB}+(\mathcal{L}_{TOP})_{BM}$	0.4551	0.4066	**0.5167**
SkipFuse-U-Net-3+	Sat50-Berlin + Aug	5	$\mathcal{L}_{CE}+\mathcal{L}_{GD}+(\mathcal{L}_{TOP})_{TB}+(\mathcal{L}_{TOP})_{BM}$	**0.4568**	**0.4124**	0.5120

Table 4.3: Instance segmentation metric results on **Sat50-Berlin**. R denotes the size of the structuring element of the dilation of the separation line in the post-processing.

As a means to compare our proposed approach with current state-of-the-art methodology, we perform an additional experiment on the open source aerial dataset **Aer50-NRW**. **Aer50-Nrw** includes building instance maps, which are, unlike in existing public datasets like CrowdAI [63], separated at section borders.

Model	Dataset	Loss	$F1_{BM}$	IoU_{BM}	$F1_{TB}$	IoU_{TB}	$meanIoU$
SkipFuse-DenseNet121-U-Net	Sat30-Berlin	$\mathcal{L}_{CE}$	0.918	0.8484	**0.4377**	**0.2802**	**0.6904**
SkipFuse-U-Net-3+	Sat30-Berlin + Aug	$\mathcal{L}_{CE}$+$\mathcal{L}_{GD}$+$(\mathcal{L}_{TOP})_{TB}$+$(\mathcal{L}_{TOP})_{BM}$	**0.9214**	**0.8542**	0.3956	0.2466	0.6817

Table 4.4: Semantic segmentation metric results on **Sat30-Berlin**.

Model	Dataset	R	Loss	$meanIoU$	$F1_{IS}$	mAP	mAR
SkipFuse-DenseNet121-U-Net	Sat30-Berlin	4	$\mathcal{L}_{CE}$	**0.6904**	0.4546	0.4104	0.5109
SkipFuse-U-Net-3+	Sat30-Berlin + Aug	9	$\mathcal{L}_{CE}$+$\mathcal{L}_{GD}$+$(\mathcal{L}_{TOP})_{TB}$+$(\mathcal{L}_{TOP})_{BM}$	0.6817	**0.4874**	**0.4497**	**0.5320**

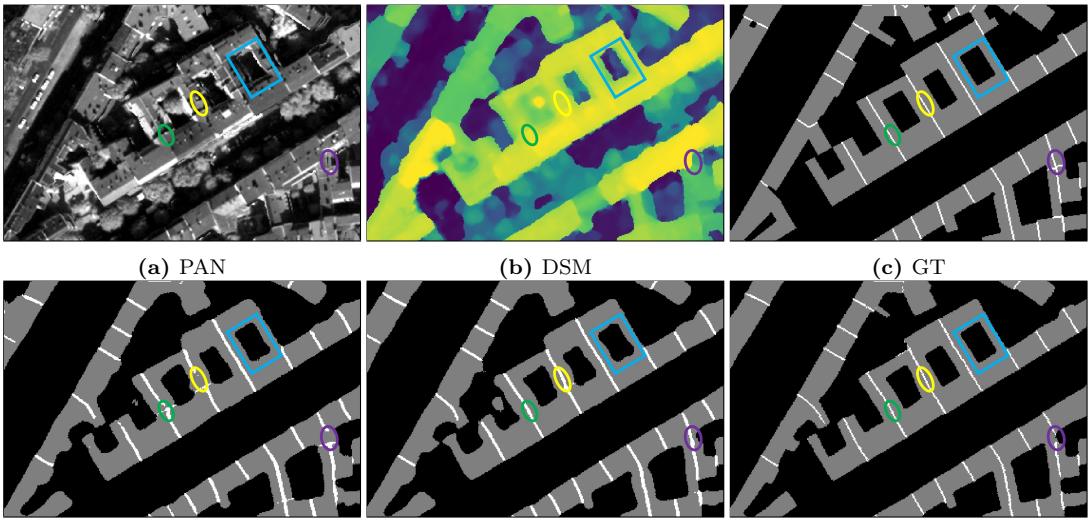
Table 4.5: Instance segmentation metric results on **Sat30-Berlin**. R denotes the size of the structuring element of the dilation of the separation line in the post-processing.

To compare the method proposed in this chapter with the method introduced by Girard *et al.* [65], we train a U-Net-ResNet50 with the frame field learning method on our aerial dataset **Aer50-NRW**. For this training, we use the public code from [**ff_code**]. We set the number of epochs to 500 and select the model after 86 epochs, which has the lowest validation loss. The learning rate is set to 0.001 and the batch size is chosen to be 8. As the polygonization procedure, we select the active skeleton model. The active skeleton model consists of **(a)** skeleton initialization and **(b)** skeleton refinement. In difference to [65], which uses only a single threshold to obtain the initial building interior and exterior segmentations, we use two different thresholds. First, the building interior segmentation is thresholded by 0.9. From the resulting footprints, the initial building polygon skeleton is computed. The initial interior edge, which allows the splitting of adjoining buildings, is obtained by thresholding the edge segmentation at 0.45. Both the interior and the exterior edges are joined to build the initial skeleton that is passed to the refinement step. After producing the polygons, we rasterize them, giving each polygon and its interior an ID, to allow us to compare the method to our ground truth as in all other experiments. Since the frame field learning method does not explicitly produce the touching borders class, we do not provide metric values for the semantic segmentation, but only for the instance segmentation results. We use the same backbone network as in [65]. Note, that the frame field learning method does not leverage a DSM, which is one of the key-features of our method.

To demonstrate how our SkipFuse-U-Net-3+ performs on a dataset that is fully public available, we re-train it on the **Aer50-NRW** dataset.

4.2.3.6 Vectorization and LoD-1 DSM

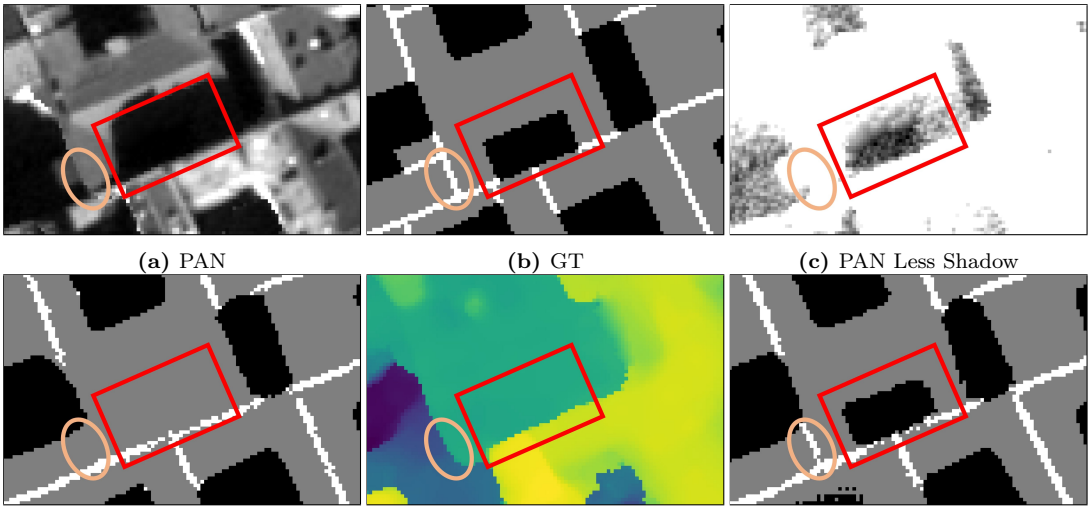
To show how our building section instances can be used for further applications, we have rendered an LoD-1 DSM as described in Section 4.2.2.4. The resulting LoD-1 DSM is visualized. We do not perform metric computation for the LoD-1 DSM, since we do not have suitable ground truth.



(a) PAN **(b)** DSM **(c)** GT

(d) Without $\mathcal{L}_{TOP}$ **(e)** $(\mathcal{L}_{TOP})_{TB}$ **(f)** $(\mathcal{L}_{TOP})_{TB} + (\mathcal{L}_{TOP})_{BM}$

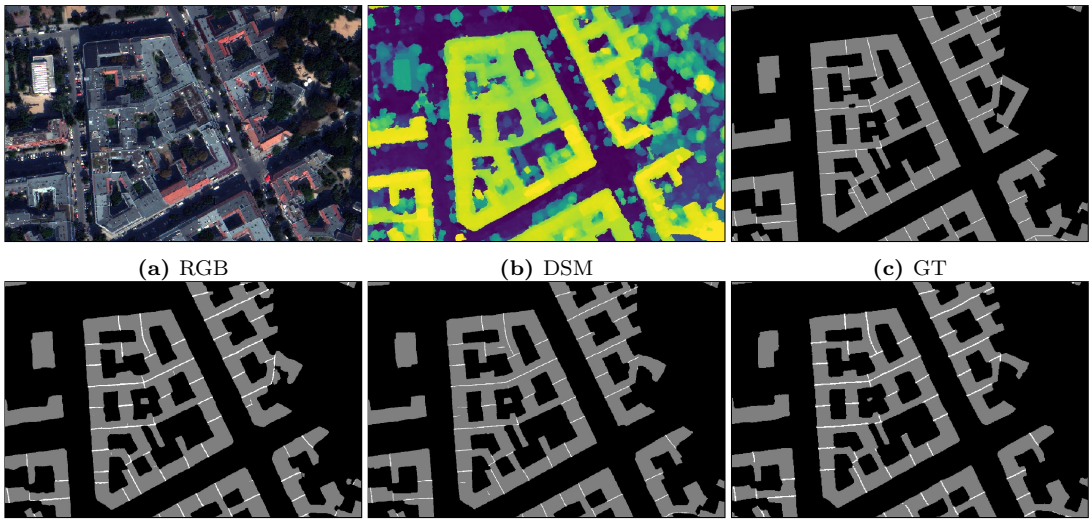
Figure 4.8: A visual comparison of the panchromatic image in **(a)**, the DSM in **(b)**, the three class map in **(c)** and the results of the SkipFuse-DenseNet121-U-Net trained with different losses. Adding $(\mathcal{L}_{TOP})_{TB}$ removes holes of touching borders (green, yellow and purple ovals). Also adding $(\mathcal{L}_{TOP})_{BM}$ leads to sharper edges of the building sections (blue box) and thinner touching borders (green, yellow and purple ovals).



(a) PAN **(b)** GT **(c)** PAN Less Shadow

(d) SkipFuse-DenseNet121-U-Net **(e)** DSM **(f)** SkipFuse-DenseNet121-U-Net Extra Augmentation

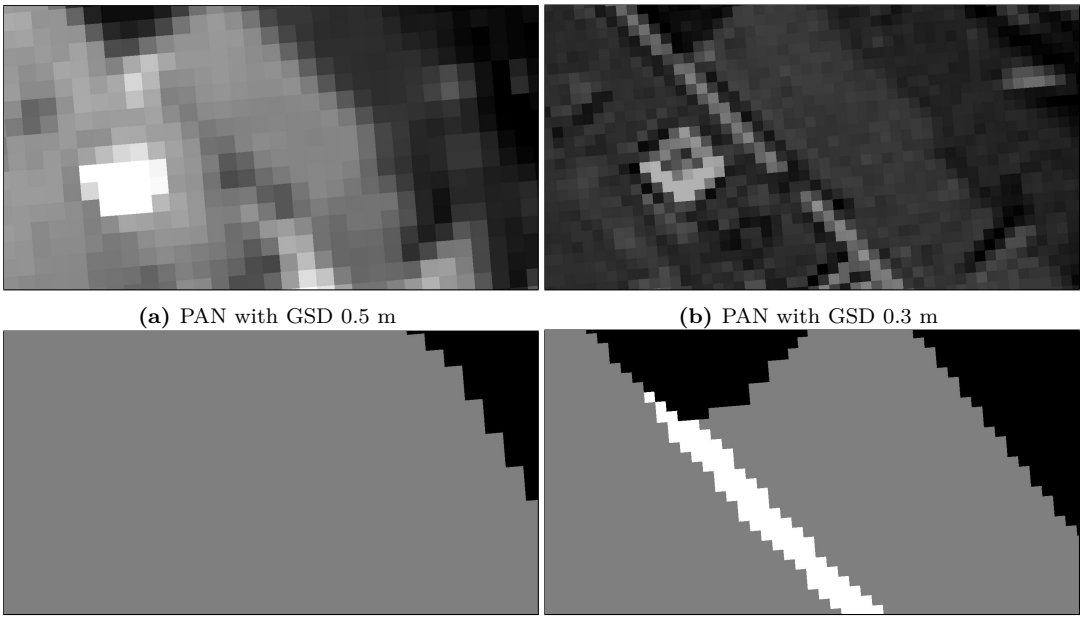
Figure 4.9: In **(a)** the PAN, in **(c)** the PAN cut off at the 0- and 20−percentiles and in **(e)** the DSM are visualized. In **(b)**, **(d)** and **(f)**, the ground truth, prediction from SkipFuse-DenseNet121-U-Net and prediction from SkipFuse-DenseNet121-U-Net with additional data augmentation are given.



(a) RGB **(b)** DSM **(c)** GT

(d) SkipFuse-DenseNet121-U-Net **(e)** SkipFuse-U-Net-3+, WorldView-4 **(f)** SkipFuse-U-Net-3+, WorldView-1

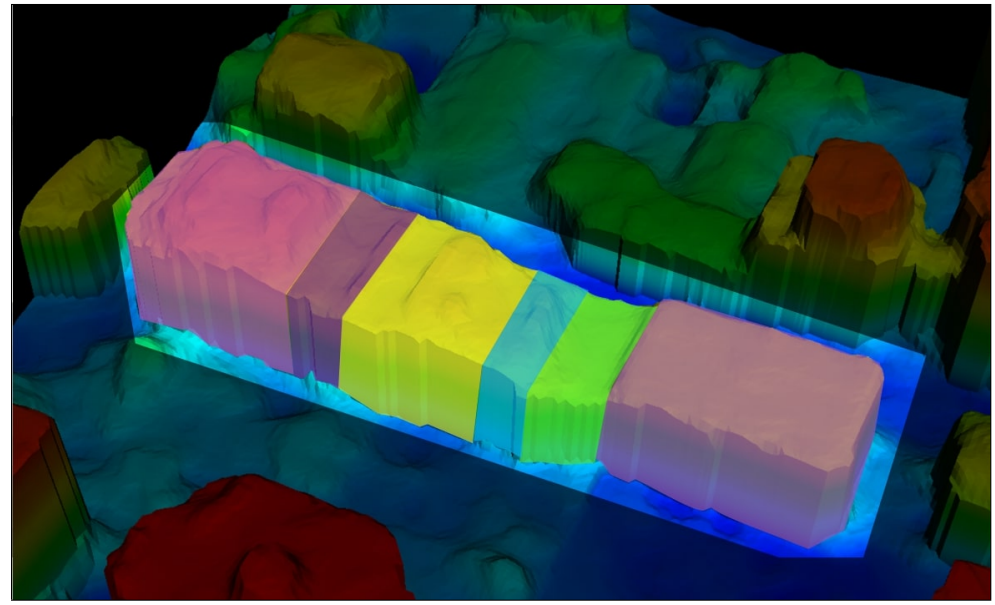
Figure 4.10: Visualization of the results of two different models on two different datasets.



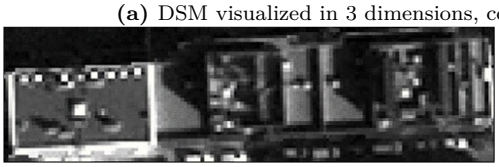
(a) PAN with GSD 0.5 m **(b)** PAN with GSD 0.3 m

(c) SkipFuse-U-Net-3+ on WorldView-1 data **(d)** SkipFuse-U-Net-3+ on WorldView-4 data

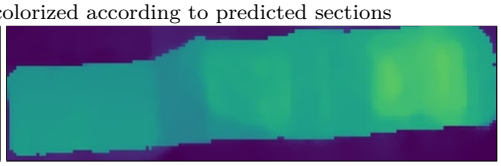
Figure 4.11: A visualization of the results of the SkipFuse-U-Net-3+ on WorldView-1 and WorldView-4 data. In **(a)** and **(b)**, a junction of two buildings as seen from WorldView-1 and WorldView-4 is shown. In **(c)** and **(d)** the respective results of the SkipFuse-U-Net-3+ is visualized.



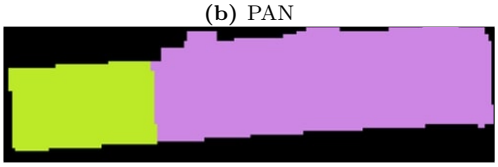
(a) DSM visualized in 3 dimensions, colorized according to predicted sections



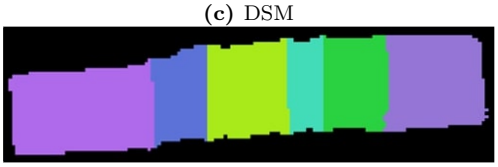
(b) PAN



(c) DSM



(d) GT section instances



(e) SkipFuse-U-Net-3+

Figure 4.12: Results from our generalization experiment. The SkipFuse-U-Net-3+ can disentangle neighboring roofs in this example. The predicted instances contains more detailed sections than the ground truth.

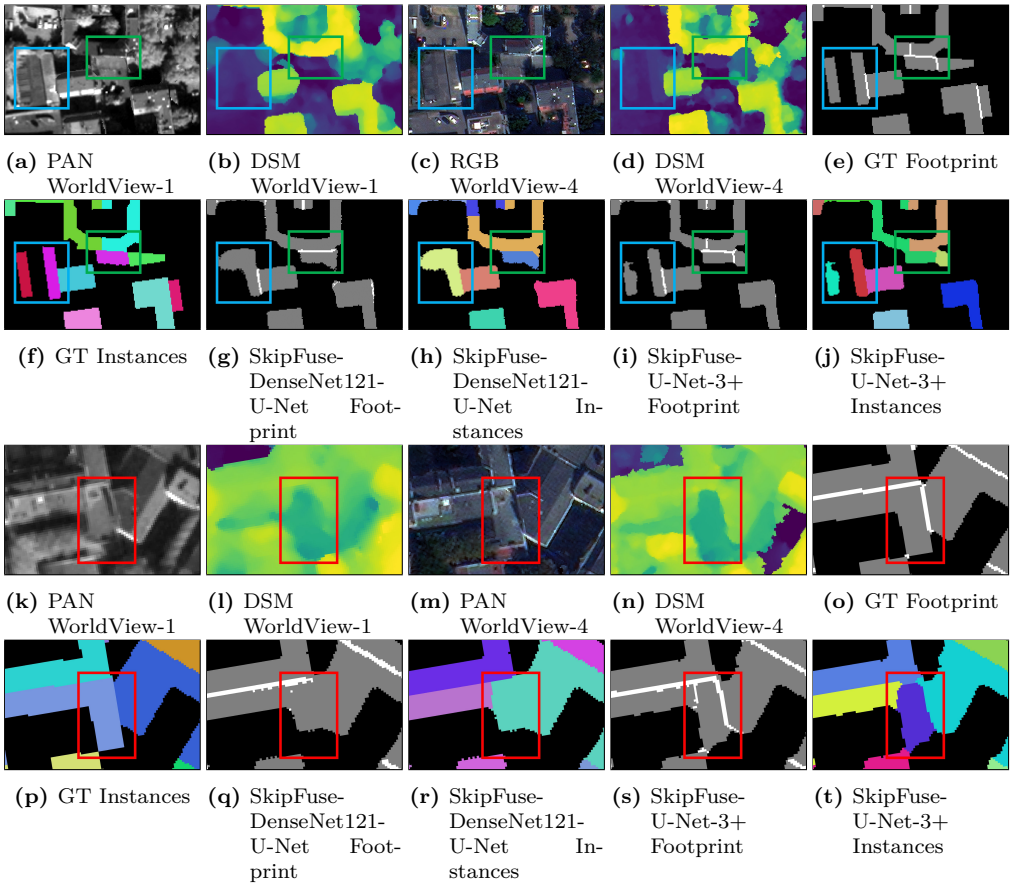
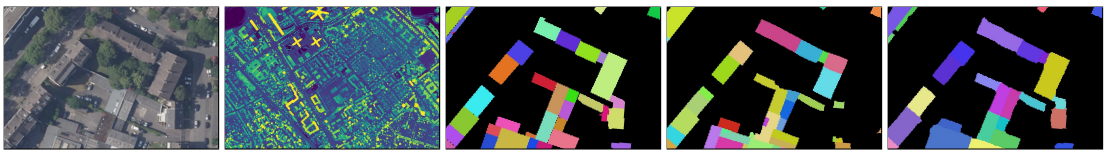


Figure 4.13: Two examples (**(a)** - **(j)** and **(k)** - **(t)**). The result of the SkipFuse-U-Net-3+ is more accurate in two ambiguous cases than the SkipFuse-DenseNet121-U-Net. In the colored boxes in **(a)**, **(b)**, **(k)** and **(l)**, it is hard to distinguish between building instances. Higher resolution World-View-4 imagery and DSM in **(c)**, **(d)**, **(m)** and **(o)** shows that the prediction of the SkipFuse-U-Net-3+ has split the two buildings correctly, whereas the SkipFuse-DenseNet121-U-Net fails to capture the fine contrast. Note, that none of the two networks used the World-View-4 data for prediction.

4.2.4 Results & Discussion

First we report our findings of the experiments on **Sat50-Berlin**. Find the quantitative results for the semantic segmentation task and the instance segmentation task in Table 4.2 and Table 4.3, respectively.

As the baseline model, we trained the SkipFuse-DenseNet121-U-Net with the segmentation losses $\mathcal{L}_{CE}$ and $\mathcal{L}_{GD}$. The model scores an mAP of 0.3453, which is much smaller than 0.4104 of the model trained on the dataset of our previous paper [2]. It is notable, that in [2], we used RGB imagery, which is not available for some satellites like WorldView-1. Also, both metric values can not directly be compared, since the imagery was acquired at different GSDs. The effect of different GSDs on separation line visibility can be recognized in Figure 4.11.



(a) RGB **(b)** DSM **(c)** GT Instances **(d)** Frame Field Learning **(e)** Ours

Figure 4.14: Comparison of the results of our SkipFuse-U-Net-3+, trained with the perceptual loss for both the touching borders class and the building class, against a ResNet101-U-Net, trained with the frame field learning method [65].

Since the quality of the touching borders is crucial for the separation of buildings into its sections, we want to regularize it. Typical pixel-wise losses as $\mathcal{L}_{CE}$ can not capture patch-level information. Here we compare our initial loss $\mathcal{L}_{CE} + \mathcal{L}_{GD}$ with $\mathcal{L}_{CE} + \mathcal{L}_{GD} + (\mathcal{L}_{TOP})_{TB}$. In Table 4.3, we can see the quantitative improvement that is achieved by adding $(\mathcal{L}_{TOP})_{TB}$. Both losses lead to similar scores on $F1_{BM}$, IoU_{BM}, $F1_{TB}$ and IoU_{TB}, but on the $F1_{IS}$, mAP and mAR, the model trained with the topological loss in the touching borders class scores 0.4089 over 0.3916, 0.3664 over 0.3453 and 0.4626 over 0.4523. The explanation for the difference in instance segmentation metrics and semantic segmentation metrics can be explained by qualitative inspection. In Figure 4.8e and Figure 4.8d, it is visualized that the addition of $(\mathcal{L}_{TOP})_{TB}$ reduces holes in touching border objects, which was similarly shown for road segments in work [15]. Hence, building sections are separated more accurate and hence, the instance segmentation metrics improve, even though the $F1_{TB}$ and IoU_{TB} are almost equal for both experiments.

Not only touching borders, but also the building model class should have regular appearance. Therefore, we trained our FCN on $\mathcal{L}_{TOTAL}$ and compared it with the model trained on $\mathcal{L}_{CE} + \mathcal{L}_{GD} + (\mathcal{L}_{TOP})_{TB}$ and the model trained on $\mathcal{L}_{CE} + \mathcal{L}_{GD}$. In Table 4.3, the model trained on $\mathcal{L}_{TOTAL}$ achieves the better performance than the previous experiments. It scores 0.4144 $F1_{IS}$, 0.3672 mAP and 0.4756 mAR, even though its performance drops on $F1_{TB}$ from 0.5247 to 0.4914 and on IoU_{TB} from 0.3557 to 0.3257. Compared to the model trained without both $\mathcal{L}_{TOP}$ terms, it can be seen how the topology loss improves the shape of the footprints in Figure 4.8. The footprints of the segmentation of the model trained with $\mathcal{L}_{TOP}$ have straighter edges in many places and even detect an inner yard, that the unregularized model does not detect. Furthermore, the model trained on $\mathcal{L}_{TOTAL}$ segments touching borders as thin lines as opposed to the model trained with $(\mathcal{L}_{TOP})_{TB}$ but without $(\mathcal{L}_{TOP})_{BM}$. This helps the post-processing to produce more accurate sections, since the building sections are less deformed by separation lines which are to thick.

Since it is crucial that the model is trained in a way that avoids overfitting, we applied further data augmentation techniques. In Table 4.3 it can be observed that the applied data augmentation leads to an improvement in mAP of 0.0133. In $F1_{BM}$ and IoU_{BM} the strongly augmented model does not improve over the not augmented model. In contrast, the strongly augmented model exceeds the performance of the SkipFuse-DenseNet121-U-Net without the extra augmentations in $F1_{TB}$ and IoU_{TB}. The touching borders class

Model	Dataset	Loss	$F1_{BM}$	IoU_{BM}	$F1_{TB}$	IoU_{TB}	$meanIoU$
SkipFuse-U-Net-3+	Aer50-NRW	$\mathcal{L}_{TOTAL}$	0.9148	0.8429	0.5499	0.3792	0.7218
FrameField-Resnet101-U-Net	Aer50-NRW	See [65]	-	-	-	-	-

Table 4.6: Semantic segmentation metric results on **Aer50-NRW**. Note, that in [65], the full border is predicted instead of the only the touching borders. Hence, the metric results are not comparable and we ommit them here.

Model	Dataset	R	Loss	$F1_{IS}$	mAP	mAR
SkipFuse-U-Net-3+	Aer50-NRW	1	$\mathcal{L}_{TOTAL}$	**0.2905**	**0.2454**	**0.3559**
FrameField-Resnet101-U-Net	Aer50-NRW	-	See [65]	0.2202	0.1776	0.2899

Table 4.7: Instance segmentation metric results on **Aer50-NRW**. R denotes the size of the structuring element of the dilation of the separation line in the post-processing.

is much more underrepresented than the building model class. As previously stated, we weight the touching borders class with a factor of 4 in $\mathcal{L}_{CE}$ and the inverse class frequency in $\mathcal{L}_{GD}$. Therefore, a fewer amount of pixels has a relatively large effect on the gradient of the model. But with the touching border dropout as we applied it here, the variety of input-output-relation in the touching border class increases. In Figure 4.9, a case of an improved separation line and occlusion robustness of the model trained with strong augmentations is visualized.

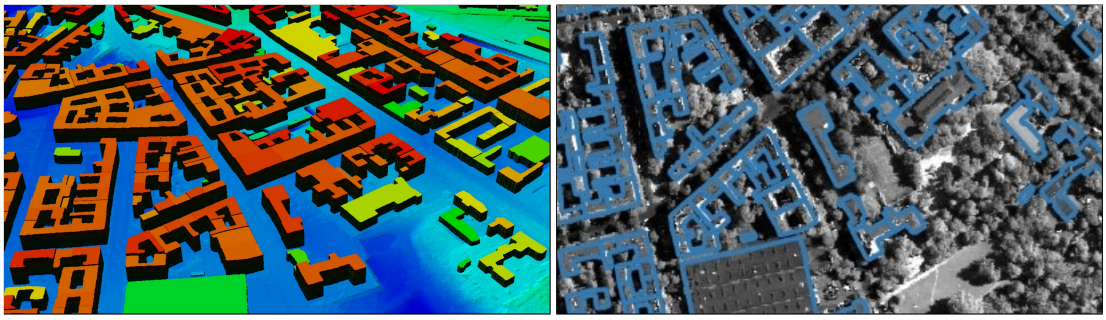
We evaluated the SkipFuse-U-Net-3+ on the **Sat50-Berlin** dataset. As can be seen in Table 4.2 and 4.3, the SkipFuse-Unet-3+ outperforms all the other models on **Sat50-Berlin**. It achieves the $meanIoU$ of 0.7246 and $F1_{IS}$ of 0.4568, which shows that the SkipFuse-U-Net-3+ performs better not only in the realm of semantic segmentations, but its predictions are more suitable for building section instance segmentation than those of the SkipFuse-DenseNet121-U-Net. In the visual inspection in Figure 4.13, we observe that the SkipFuse-U-Net-3+ is more accurate than the SkipFuse-DenseNet121-U-Net in places with high visual ambiguity. FSAs allow the SkipFuse-U-Net-3+ to take even small spectral and height differences into account when making the prediction.

After seeing only roofs of Berlin during training, we leveraged the SkipFuse-U-Net-3+ to make building section instance prediction in the city of Lyon, France. Quantitatively, the SkipFuse-U-Net-3+ achieves $meanIoU$ 0.582 and mAP score 0.1165 using a dilation radius of 3. But visually, the model's behavior is confirmed as more accurate than the ground truth (see Figure 4.12), which gives us an indicator for why the metrics are much lower than on our test area in Berlin.

As a means to demonstrate how our findings from the current chapter complement our previous work, we present our results on **Sat30-Berlin**. See the quantitative results on the semantic and instance segmentation tasks in Table 4.4 and Table 4.5.

We re-trained the model from [2] on a slightly modified dataset. In comparison to our previous paper, the performance dropped by about 0.01 from 0.42 mAP to 0.41 mAP. But since we shifted the ground truth to match the satellite imagery better, the results are now more useful and can not directly be compared in terms of metric values.

Since we want to show that the improvements we achieved in the ablation study using the WorldView-1 data, we also trained our proposed model, the SkipFuse-U-Net-3+,



(a) LoD-1 DSM **(b)** Vectorized Format

Figure 4.15: Visualization of our predicted building sections as **(a)** an LoD-1 DSM and **(b)** vectorized polygons. These two variants of our results are useful for further applications.

using our proposed regularization loss scheme. From Table 4.4 and Table 4.5, it can be seen that our proposed method has higher values in all metrics but the $F1_{TB}$ and the IoU_{TB}. In Figure 4.10, the separation lines produced by the SkipFuse-U-Net-3+ are considerably thinner than those infered by the SkipFuse-DenseNet121-U-Net, which is caused by the regularization loss term $(\mathcal{L}_{TOP})_{TB} + (\mathcal{L}_{TOP})_{BM}$. Even though thinner lines are more regular, they lead to dropping semantic segmentation metrics, because small deviations of the position of the predicted line may cause them to be completely off the corresponding ground truth. On the other hand, the SkipFuse-U-Net-3+ achieves an $F1_{IS}$ 0.4874, mAP 0.4497 and mAR of 0.532, which is an improvment of 0.0328, 0.0393 and 0.0211 respectively over the SkipFuse-DenseNet121-U-Net trained without the perceptual loss terms. Hence, the small drop in performance in the touching borders class does not effect the resulting instances severely. Furthermore, the SkipFuse-U-Net-3+ that was trained and evaluated on WorldView-4 data produces more regular building shapes than the same model trained and evaluated on WorldView-1 data. We attribute more of the gain to the change in GSD than to the higher spectral resolution. In Figure 4.11, it can be seen that even in the panchromatic image, the separation line, which is hard to detect even for the human eye, can be detected more easily on the WorldView-4 data and is detected by the corresponding model on the WorldView-4 data, but not on the WorldView-1 data.

Next, we present the results of the experiments carried out on the aerial open-source dataset **Aer50-NRW**. Find the quantitative results on the semantic and instance segmentation tasks in Table 4.6 and Table 4.7, respectively.

As the baseline model on our benchmark dataset **Aer50-NRW**, we evaluated the framefield learning method to segment building sections. Quantitatively, the framefield learning model achieves an mAP of 0.1776. It is shown in Figure 4.14, that the framefield learning model produces instances of regular appearance with straight lines and sharp corners. But in some places, the framefield learning model produces more instances than are actually present in the image and the ground truth.

On the other hand, our SkipFuse-U-Net-3+, which does not have access to the tangent direction and hence, can not use the active skeleton model to refine the building bound-

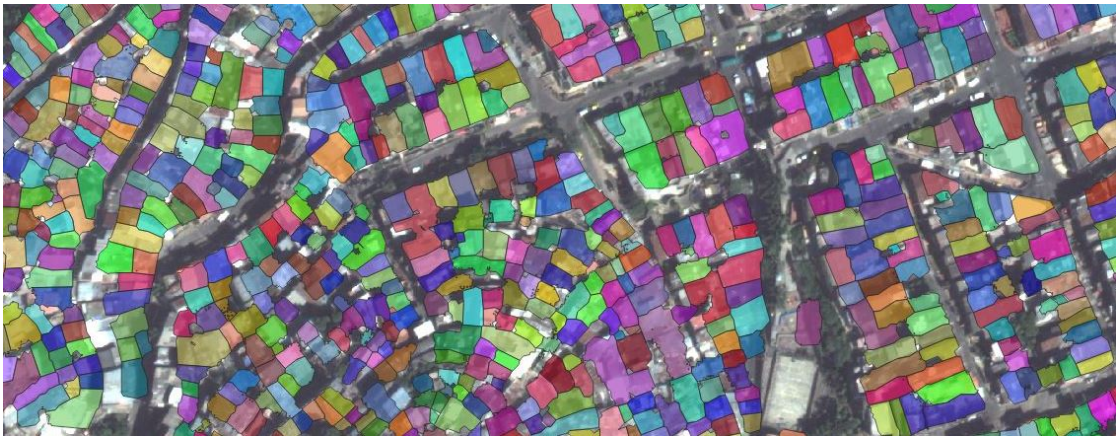


Figure 4.16: Example of selected test area in Medellin and the obtained instance segmentation results from the proposed methodology.

ary, produces less regular building sections. But quantitatively, our method improves over the framefield learning model by 0.0678 mAP. Since our method utilizes the DSM, it is more robust to variations in the RGB imagery.

As can be observed in Figure 4.15a, our building sections, combined with the corresponding DSM and digital terrain model (DTM) are a suitable basis to produce a regularized LoD-1 DSM. Furthermore, in Figure 4.15b, the corresponding polygons are visualized on top of the input panchromatic image. The polygons are visually accurate representations of the underlying building sections.

4.3 Informal Building Instance Segmentation

4.3.1 Application Description

In the undulating slopes of Medellín, Colombia, where challenging topography intertwines with the vulnerability of communities, the threat of landslides emphasizes the pursuit of precise insights into this complex urban landscape. We urgently require additional information to assess the potential risks and determine the necessity of employing artificial intelligence for the instance segmentation of buildings. Medellín, faces unique challenges as it is nestled in a valley surrounded by steep slopes. Historical urbanization, driven by rapid industrial and economic growth in the mid-20th century, led to an influx of migrants settling informally on the city's outskirts [94]. The escalation of informal housing, exacerbated by conflicts between paramilitary forces and guerrilla groups, particularly in later years, has extended the city into precarious, hard-to-reach areas on these steep slopes.

The informal settlements, characterized by low-quality building fabric, are highly vulnerable to landslide hazards due to frequent heavy rainfall and the presence of weak, erosive rocks in the bedrock [95]. Effectively countering this risk requires precise knowledge of the at-risk areas and an understanding of the potential impact on the population.

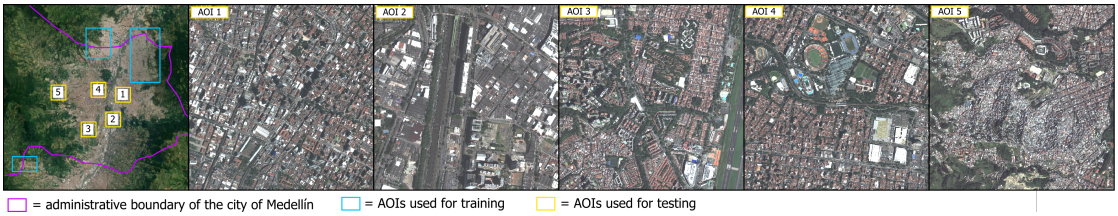


Figure 4.17: Locations of the AOIs within the administrative area of Medellín and detailed maps for the AOIs used for testing.

However, official population data for Medellín accurately geolocates formal residents but significantly underrepresents the more recent informal settlements on steep slopes, leading to a substantial underestimation of the exposed population [96].

Detecting informal settlements poses significant challenges [97, 98] due to limited data availability and the uncertainty associated with reference data [99, 100]. This difficulty is further aggravated by the intra-and-inter urban variability observed in informal areas [101]. While the utilization of very high-resolution imagery proves advantageous for informal settlement classification, as demonstrated in [102], the detection of individual buildings within informal settlements remains an extremely challenging task, often requiring manual intervention [103].

To address this critical knowledge gap, we leverage high-resolution remote sensing imagery in combination with a DSM, capable of detecting single buildings in the challenging urban environment of Medellín. This approach will be tested in formal settlements, but even more pertinent and demanding is its application in the city's informal settlements (see Figure 4.16). The morphological characteristics of these areas, marked by small-scaled, intricate, and densely packed structures, are indicative of informal settlements [104]. In this study, we employ a SkipFuse-UResNet34 in order to generate building instances, thus leading to a more comprehensive building mask compared to official data sources. This enhanced building mask serves as a crucial tool to estimate the population at risk of landslides, allowing for a comparison with official data and filling the current spatial knowledge void.

Overall, this section has the following contributions:

- building instance segmentation in formal settlement,
- building instance segmentation in informal settlement,
- empowering disaster aid by providing a building map layer.

4.3.2 Study area and data

The study area is the municipality of Medellín (Figure 4.17). Three regions of interest (RoI) were used for training, each with different sizes, i.e. from east to west, 3x5.5km, 2.5x3km, and 2.5x1.5km. The testing was undertaken in five distinct RoIs within Medellín, each covering an expanse of 1.5x1.5km. The selection of the test RoIs was guided

by two primary considerations. Firstly, they were chosen for their high building density, rendering them well-suited for our proposed application. Secondly, these RoIs showcase a diverse array of morphological building types, effectively capturing the inherent structural complexity of buildings. RoI 1 is situated in the central business district, RoI 2 is characterized by a substantial industrial area, RoI 3 primarily consists of residential building structures with varying heights, RoI 4 incorporates large communal and industrial buildings alongside low- and mid-rise structures, and RoI 5 features a densely-built informal area located at the steep slopes on the outskirts of Medellín.

For the study area of Medellín, we use very high resolution RGB satellite imagery from WorldView-3 with a geometric resolution of 0.3 m. For training the model, additional imagery data depicting Berlin city from WorldView-4, Bonn city from open source aerial data, and Hamburg city from WorldView-2 were used. All images were brought to a geometric resolution of 0.3 m.

In contrast to prior research, our model utilized a pansharpened RGB image in addition to DSM generated by semi-global matching (SGM) technique [19], rather than relying on a single panchromatic channel and DSM. The RGB images enables better visualization and interpretation of the details present in the scene. For a challenging task like detecting small buildings in informal areas, spectral information plays a crucial role. It provides the network with additional clues about various structures, particularly in cases of dissimilar textures.

We rely on official building cadaster data for training, validation, and testing of our method. Cadaster data from Germany for the cities of Berlin, Bonn, and Hamburg, were used for training and validation. In addition to these, cadaster data from Medellín was also used for the purpose of training, validation, and testing. This dataset encompasses the footprints for a significant portion of building structures in Medellín. Nevertheless, it is important to highlight that the official building cadaster does not include numerous informal or recently constructed buildings.

4.3.3 Methodology

With this paper, we are testing the applicability of our previously developed methodology Section 4.2 for building sections instance segmentation on a new challenging area of Medellín. The proposed method comprises two consecutive steps: Initially, a SkipFuse-UResNet34 architecture (see Figure 4.18) is used to segment buildings and separation lines between building sections as a 3-class problem. The process utilizes RGB and DSM images fed into two distinct encoders. To preserve fine-grained spatial information, feature maps, obtained at four distinct scales from the two encoders, are aggregated by summation and serve as the input for the full-scale skip-connections. During the network training, a combination of three losses—Weighted Cross Entropy Loss, Dice Loss, and Topology Loss—was employed. For more details, please, refer to Section 4.2.

Next, a map of building section instances is generated using the watershed transform [3] as a post-processing step. Mainly, the watershed transform interprets the obtained 3-class map of background, building, and separation line together with a seed image and a mask as a topographical surface. The seed map and mask are extracted

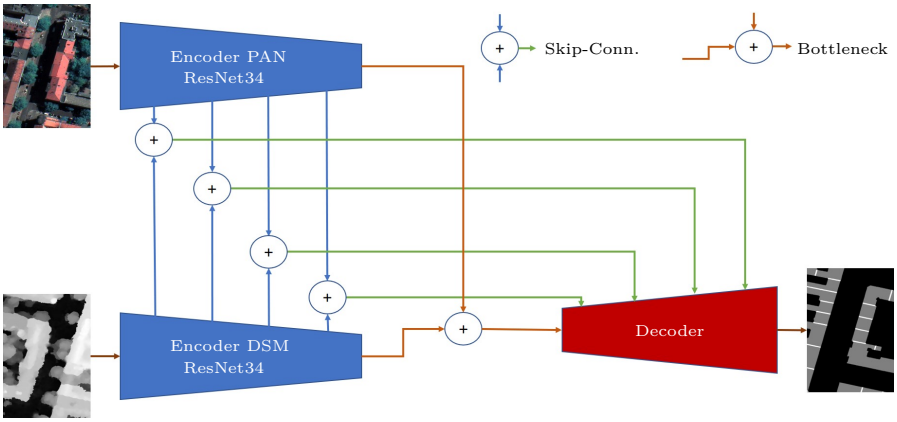
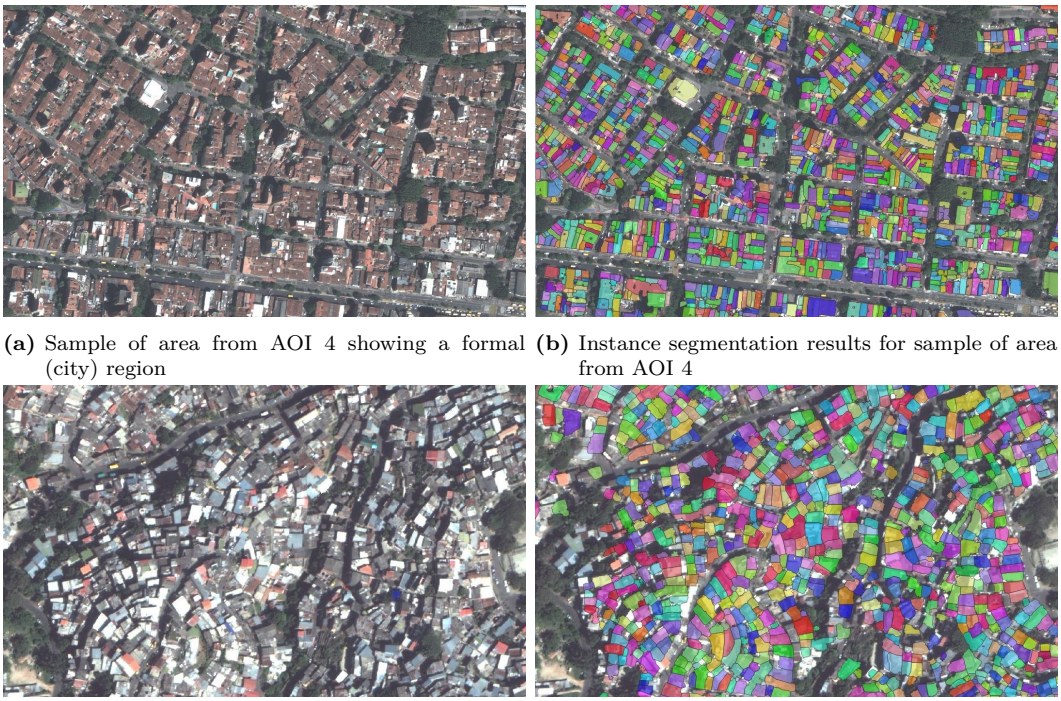


Figure 4.18: Visualization of the utilized network architecture.



(a) Sample of area from AOI 4 showing a formal (city) region

(b) Instance segmentation results for sample of area from AOI 4

(c) Sample of area from AOI 5 showing an informal settlement

(d) Instance segmentation results for sample of area from AOI 5

Figure 4.19: Detailed visual analysis of building instance segmentation results on two challenging areas from AOI 4 and 5.

from the predicted information about buildings and separation lines. Subsequently, watershed transform simulates a flooding scenario where water initiates flooding from the seeds and settles into basins. These basins are marked by watershed lines, aligning with high image intensities. The mask confines the virtual water flow to specific regions. The enclosed regions delineated by watershed lines are then recognized as objects.

Table 4.8: Quantitative results for IOU, FPR, FNR metrics of building class and overall accuracy evaluated on five selected AOIs for testing.

AOI	IoU_{BLD}	FPR_{BLD}	FNR_{BLD}	OA
1	0.694	0.125	0.234	0.816
2	0.761	0.039	0.192	0.902
3	0.669	0.066	0.223	0.889
4	0.708	0.078	0.198	0.878
5	0.620	0.086	0.254	0.866

We conduct evaluations on five selected RoIs that combine both formal urban environments and informal settlements. Those complex areas are selected for the purpose to better demonstrate the strength of the proposed methodology.

4.3.4 Results and Discussion

Figure 4.19 illustrates two samples of complex and very dissimilar regions of Medellín together with the obtained instance segmentation results from our proposed methodology.

In a city region (see Figure 4.19a), buildings can have very complex shapes with many sections within one building construction. To distinguish those sections or separation lines between those sections is a challenging task even for a human eye. Nevertheless, one can recognize a city layout within this region, which cannot be done by observing the informal settlement area (Figure 4.19c). Although buildings within informal settlements often exhibit simple rectangular shapes, their dense arrangement makes it challenging to distinguish each individual construction. Analysing the obtained instance segmentation results for both regions in Figure 4.19b and Figure 4.19d, we can say that our developed methodology enables the identification of even the smallest building segments, regardless of the building complexity or the region.

To quantify the quality of instance segmentation results, we evaluate the metrics

$$\text{IoU}_{BLD} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{4.9}$$

$$\text{FPR}_{BLD} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{4.10}$$

$$\text{FNR}_{BLD} = \frac{\text{FN}}{\text{FN} + \text{TP}} \tag{4.11}$$

and

$$\text{OA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{4.12}$$

where TP, TN, FP, FN are true positive, true negative, false positive and false negative of the building class.

Our approach achieves an IoU around or above 0.7 on the AOIs 1-4. In the informal AOI 5, it still scores 0.62 IoU, whereas the overall accuracy (OA) is high at above 0.8 for all AOIs. The false positive rate (FPR) is very low for all AOIs, which indicates that the method produces few false positives. On the other hand, the resulting false negative rate (FNR) is above 0.19 for all AOIs, pointing at the issue that many small buildings are very hard to detect.

4.4 Building Footprint Regularization

4.4.1 Contributions

Our goal is to achieve regularized building polygons, assuming that all vertices of buildings should have a rectangular angle. Whereas this assumption not always holds, it is sufficient for most buildings, especially residential houses and industrial buildings. A 90° angle always needs a reference axis, which is the primary orientation of a polygon. Even though for perfectly regularized polygons, in most cases the primary orientation is that of the longest sidelength, we are dealing with irregular polygons. Deep learning allows us to automatically learn features from a large training dataset. First, we use deep learning to extract building footprints from ortho imagery and photogrammetric digital surface model (DSM). Next, with our regularization framework called primary orientation learning (POL), we train a 1D convolutional neural network (CNN), from whose output we compute the primary orientation angle in continuous space. Subsequently, we use a learning free and iterative approach to insert vertices that make the initial polygon rectilinear, i.e. having 90° angles at every vertex. Figure 4.20 shows an example of a regularized building outline obtained by our method. Overall, the contributions of this sections are

- continuous prediction of primary orientation angle,
- highly efficient primary orientation angle inference,
- and guaranteed rectilinear resulting building polygons.

4.4.2 Methodology

To obtain regularized building footprints, two steps are required. In the first step, we used the trained network from Section 4.3.1. During the inference of that network, we merged neighboring building sections into larger building footprints, since our rectilinearization method is not designed for seamless neighboring sections. In the second step, we extracted building border pixels to form a polygon, followed by predicting the corresponding primary orientation, and then applied our rectilinearization algorithm to obtain regular polygons.

Figure 4.20: A building boundary regularization example in our test region, Braunschweig, Germany. The left figure represents the initial vectorization of the building footprint and the right one is the final regularized building boundary.

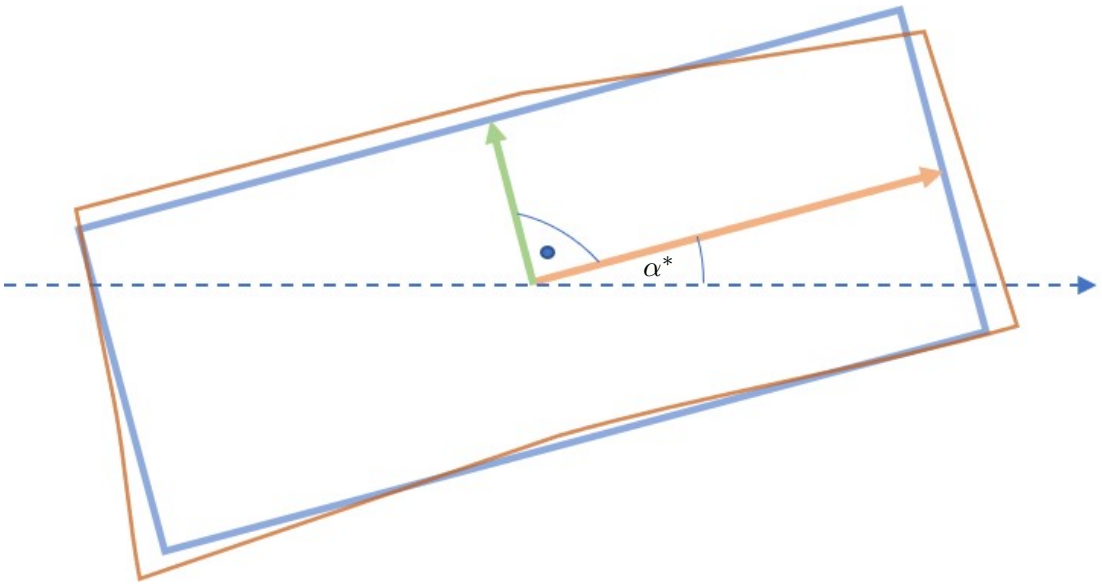


Figure 4.21: Visualization of a regular polygon (blue), an irregular polygon (brown), the primary orientation axis (orange) and the secondary orientation axis (green).

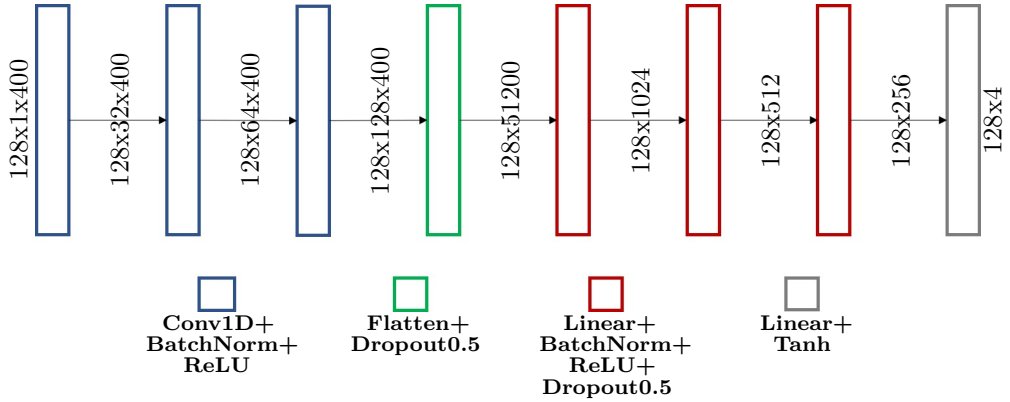


Figure 4.22: Visualization of the architecture of our proposed POL network. Convolutional layers extract local features for every vertex and linear layers compute global features and the regression output.

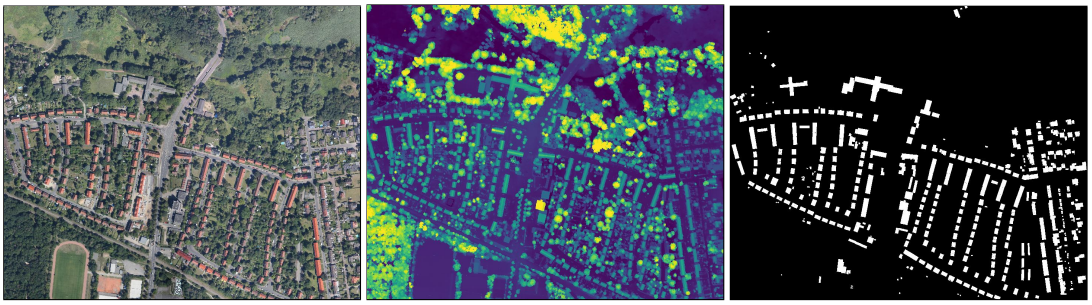


Figure 4.23: Our test area in Braunschweig, Germany. Three layers included within the data are shown, RGB image (left), DSM (middle), and the ground truth building footprints (right).

4.4.2.1 Primary Orientation Learning: Vectorization and Regularization

The previously described method to obtain building footprints delivers them in raster format. To obtain instances, we first generate initial polygons and then refine them based on their primary orientation.

4.4.2.1.1 Initial Polygon Generation We applied a tree search to obtain an ordered set of boundary pixels, forming a polygon. This polygon has many redundant vertices and has irregular appearance, because of limited ground sampling distance (GSD) and imperfect building footprints. To remove many redundant vertices and simplify the polygon, we applied Douglas-Peucker [105] with tolerance $\epsilon = 1.2$ m.

4.4.2.1.2 Direction Prediction & Rectilinearization The next step is based on the assumption, that the boundaries of a building are aligned with only two directions. We define the direction, along which the regular polygon stretches most, the primary orienta-

tion α^*. The 90° rotated primary axis is called secondary orientation $\beta^* = \alpha^* + 90°$. This can be seen in Figure 4.21, where the orange arrow represents the primary orientation axis, being rotated with respect to the blue, dashed arrow by angle α^*. For regular polygons, α^* can be obtained by computing the angle of the linesegment with the longest sidelength with respect to the positive x-axis (blue, dashed line). But for irregular polygons, the longest sidelength has no meaning. We represent a polygon $\mathcal{P} = [v_0, v_1, ..., v_{n-1}]$, which is a clockwise ordered set of n vertices $v_i \in \mathcal{V}$ with $\mathcal{V} = \{v_0, v_1, ..., v_{n-1}\}$ as a vector $\mathbf{p} = [x_0, y_0, x_1, y_1, ..., x_{n-1}, y_{n-1},$
$0, ..., 0]^T$ with trailing zeros to bring each vector to the fixed length 400, which facilitates the length of all polygons in our dataset. We passed a minibatch of such vectors to a network consisting of 1D convolutional, rectified linear unit (ReLU), batch normalization, dropout and linear layers (see Figure 5.11). This network predicts the primary and secondary orientation angles $\hat{\alpha}$ and $\hat{\beta}$ by the parameters c_0 and c_2 of the complex polynomial

$$f(z) = z^4 + c_2 z^2 + c_0, \tag{4.13}$$

where

$$c_0 = u^2 \tag{4.14}$$

$$c_2 = -(u^2 + v^2) \tag{4.15}$$

$$\Leftrightarrow \begin{cases} u &= \sqrt{-\frac{1}{2}\left(c_2 + \sqrt{c_2^2 - 4c_0}\right)} \\ v &= \sqrt{-\frac{1}{2}\left(c_2 - \sqrt{c_2^2 - 4c_0}\right)}. \end{cases} \tag{4.16}$$

The ambiguity of the sign and order when regressing an angle directly is resolved in this representation of the orientation. We borrow this idea from [65], where the coefficients are predicted at each pixel of an image along the boundary of buildings. However, we only predict a single complex value for each c_0 and c_2 for each polygon. Hence, we obtained 4 scalars for each input $\mathbf{p}$ from the network, from which we calculate the two complex numbers u and v using Equation (4.16). Then, we converted each of u and v into an angle with respect to the positive x-axis using trigonometry.

The network is trained using two loss functions, the first loss function

$$\mathcal{L}_{align} = |f(e^{i\theta^*}; \hat{c_0}, \hat{c_2})|^2, \tag{4.17}$$

where $\hat{c_0}$ and $\hat{c_2}$ are the predicted complex polynomial coefficients, enforces alignment of the prediction with the ground truth primary orientation angle θ^*. The second loss function

$$\mathcal{L}_{align90} = |f(e^{i\theta^{*T}}; \hat{c_0}, \hat{c_2})|^2, \tag{4.18}$$

enforces that the predicted secondary angle is aligned with $\theta^{*T} = \theta^* - \pi$. The total loss is

$$\mathcal{L} = \mathcal{L}_{align} + 0.2 \times \mathcal{L}_{align90} \tag{4.19}$$

Next, we applied the following rectilinearization algorithm for each of $\hat{\alpha}$ and $\hat{\beta}$, closely following [72]:

1. Rotate the irregular polygon by $\hat{\theta} \in \{-\hat{\alpha}, -\hat{\beta}\}$
2. Given a clockwise ordered set of vertices $\mathcal{V} = \{v_0, v_1, ..., v_{n-1}\}$, where vertex v_i has coordinates (x_i, y_i), generate a line list $L = \{l_0, l_1, ..., l_{n-1}\}$;
3. Select the oblique line segments in L. Then for each oblique line segment $l_i \in L$,
   a. Calculate two candidate points to be inserted based on the two subsequent vertices v_i and v_{i+1}:

$$v1_c = (x_i, y_{i+1})$$
$$v2_c = (x_{i+1}, y_i)$$

   b. The relative position of each candidate point relative to l_i is determined using

$$d1 = \begin{vmatrix} x_i & x_{i+1} & x_i \\ y_i & y_{i+1} & y_{i+1} \\ 1 & 1 & 1 \end{vmatrix},$$
$$d2 = \begin{vmatrix} x_i & x_{i+1} & x_{i+1} \\ y_i & y_{i+1} & y_i \\ 1 & 1 & 1 \end{vmatrix},$$

   where $d1$ is the relative position of $v1_c$ and $d2$ that of $v2_c$.

   c. Since we are dealing with clockwise-oriented polygons, a negative $d1$ or $d2$ means that either $v1_c$ or $v2_c$ is outside the polygon and hence is inserted into the polygon between v_i and v_{i+1}.

Since we applied the above algorithm twice for two different angles, we selected the rectilinear polygon that has the higher intersection over union (IoU) with the irregular polygon.

4.4.3 Experiments

We carried out two experiments. Both experiments are based on the footprints from our raster footprint extraction method, trained according to [8]. The first experiment is the baseline evaluating the method on our Braunschweig, Germany test region. The second experiment is our neural network based regularization on the same test region. See a visualization of the test area in Figure 4.23.

4.4.3.1 Baseline

The baseline method is that of [72]. The main difference to our approach is that the baseline uses a learning free procedure to obtain the primary orientation angle.

Figure 4.24: Our results in vector format on some part of the test area. Red polygons represent predicted building outlines, green polygons are ground truth polygons. The resulting polygons have regular shapes, i.e. right angles at every vertex with a low number of vertices. Even non-rectangular buildings are successfully regularized.

4.4.3.2 Primary Orientation Learning

We trained the proposed POL network on a dataset consisting of 92600 regular building polygons from public sources of the cities of Berlin, Cologne and Hamburg, Germany, as well as Medellin, Columbia and validated after every epoch on 958 polygons of Cologne, Germany. Since the trained model should work on irregular polygons, we slightly shifted each vertex of the regular polygons by a 2D normal distribution centered at the original vertex position with standard deviation 0.5 m. Additionally, we randomly rotated every polygon and adjusted the corresponding ground truth angle accordingly to increase the variety of training samples. We used the Adam optimizer with learning rate 0.001, batch size 128 and multiplied the learning rate by 0.9 after every ten epochs. We let the training run for 500 epochs and selected the model that performed best on the validation dataset.

Table 4.9: Quantitative evaluation of our method (POL) and the baseline method. We jointly evaluated quality and efficiency metrics. The baseline method for building regularization does not rely on machine learning, hence it has no training time.

Exp.	IoU	F1	Prec	Rec	Inf. Time	Train. Time	ε
Basline	0.7946	0.8855	0.8861	0.8850	78.261 ms	-	2.7355°
POL	0.7940	0.8852	0.9056	0.8657	2.879 ms	01:38:57 (hh:mm:ss)	4.2447°

We extracted the initial polygons from the predicted raster footprints by tracing the pixels along the boundary of each connected component. Then, we applied Douglas-Peucker with tolerance 1.2 m to simplify the initial polygon. We applied our trained POL network to the simplified polygon to obtain two orientation angles. Then, we applied the rectilinearization algorithm for each of the predicted angles and selected the polygon that has the larger IoU with the simplified polygon.

4.4.3.3 Evaluation

To judge the capability of our proposed method, we evaluated it on an RGB and DSM showing an area in Braunschweig, Germany. The data was captured by an aerial 3K camera at 0.1 m GSD and downsized to 0.3 m GSD.

We use the Common metrics to evaluate building footprint quality are

$$IoU = \frac{TP}{TP + FP + FN}, \tag{4.20}$$

$$Prec = \frac{TP}{TP + FP}, \tag{4.21}$$

$$Rec = \frac{TP}{TP + FN}, \tag{4.22}$$

and

$$F1 = 2 \times \frac{Prec \times Rec}{Prec + Rec}, \tag{4.23}$$

where TP, FP, FN are the true positive, false positive and false negative of the building class. Additionally, we provide the inference time, training time and the angle prediction error

$$\varepsilon = |\hat{\theta} - \theta^*|. \tag{4.24}$$

For POL, we processed the polygons of the whole test area at once and divided the inference time by the number of polygons. The experiments were carried out on a server with an NVIDIA GeForce RTX 2080 Ti GPU with 11019 MB for the neural network inference and a Intel® Xeon® Gold 6230 CPU @ 2.10GHz for the baseline inference. The server has 504 GB working memory. To gain more insight into the results, we visualized

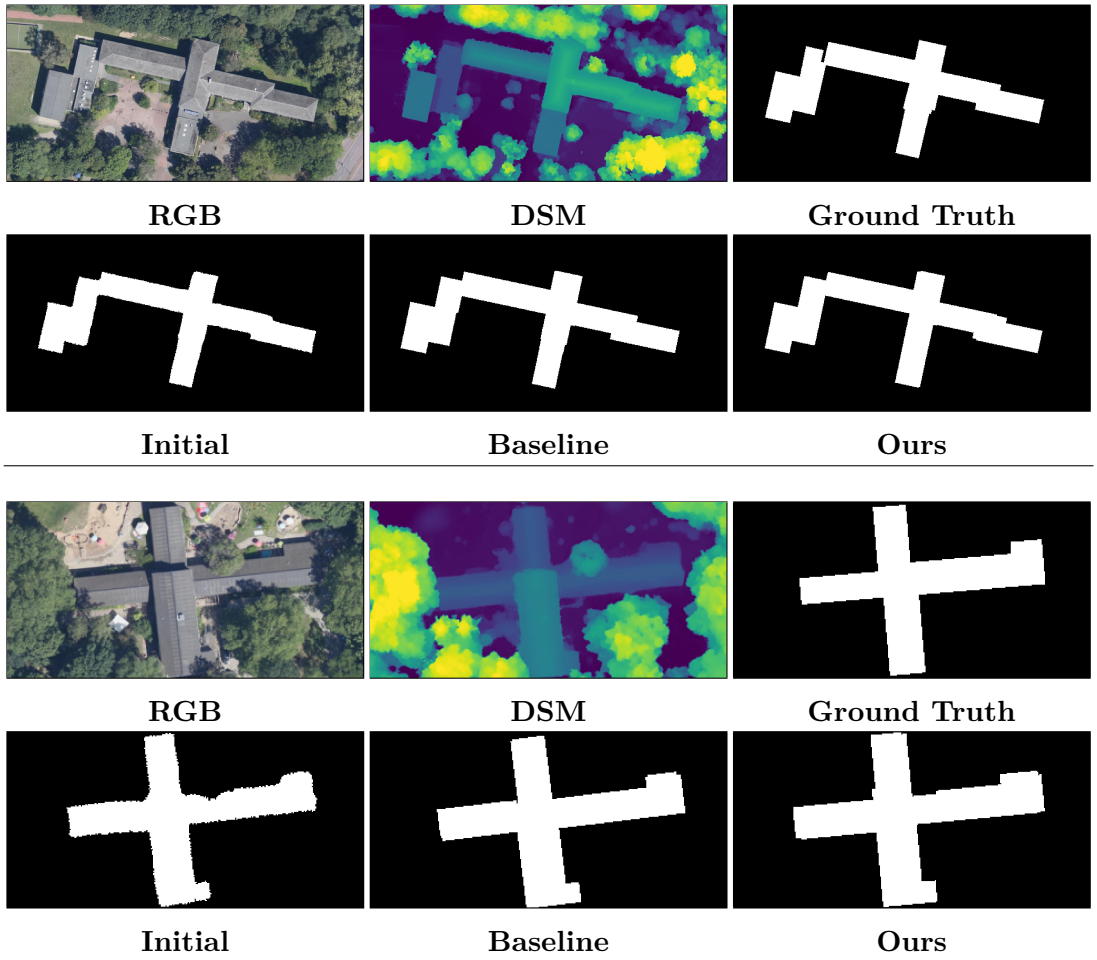


Figure 4.25: Visual result of two buildings in our test region. The baseline and our result are rectangular at every vertex, whereas the initial segmentation has irregular appearance. The horizontal line splits two different cases.

both results next to the ground truth.

4.4.4 Results

We listed the metrics of resulting building footprint quality and training/inference time in Table 4.9. It shows that both the baseline method and our proposed POL achieve very similar or almost identical results in IoU and F1, whereas POL has a higher precision and the baseline has a higher recall. The similarity in IoU and F1 are explained by the fact that we used the identical initial footprints and regularization does not have a large effect on these metrics. On the other hand, the baseline method tends to add the new vertices more on the outside of the ground truth polygon. These results show that our approach for footprint regularization is not worse than the baseline in terms of quality. This can be verified visually in Figure 4.25, where both the baseline and our

method have perfectly regular appearance. In Figure 4.24, the high quality of most of the resulting building footprints is visualized. Although the satisfying overall result, we encountered some missing detections. Those are due to tiny building size, low contrast or lack of visibility in the RGB image, which makes it hard for the footprint predictor to recognize them. Furthermore, the baseline achieves an angular error ε of about 1.5° lower than our POL. POL predicts angles continuously which removes ambiguity from the angle prediction and avoids a method intrinsic error of up to 1.0°, which the baseline method includes. On the other hand, our learning based method was trained only on slightly alternations of the regular ground truth polygons, which leads to a domain gap between training and test polygons. Furthermore, we used the orientation of the longest side as the ground truth annotation, which is inaccurate in many cases but easy to obtain for large quantities of ground truth polygons. However, the error of 4.2447° is still very low, but the baseline needs to test 181 possible angles to achieve this results, which results in the inference time of 78.261 ms, whereas POL only needs 2.879 ms to infer a single primary orientation angle. This computational advantage can be explained by two reasons. The first is the aforementioned necessity of the baseline to compute the axis density for 181 possible angles. The second one is the capability of batch processing in POL. POL can process the about 500 predicted initial polygons in the test area in a single forward pass in parallel.

4.5 Summary

In this chapter, we presented a framework for the separation of building constructions into sections. We extended the U-Net-3+ architecture to effectively make use of DSM features by applying the SkipFuse multi-modality fusion-scheme. Our proposed method consists of **(a)** prediction of pixel-wise buildings and separation lines by the SkipFuse-U-Net-3+, **(b)** transformation of the semantic output into instance-wise output by the watershed transform and morphological operations. The SkipFuse-U-Net-3+ was trained with both a pixel-wise and a topology-aware loss on space-borne images and DSMs. The experiments show the generalization capability of this two-step-procedure to extract building sections on an area which is different from our training area in terms of source satellite and roof-top appearance. We re-trained the SkipFuse-U-Net-3+ on aerial data and outperform the current state-of-the-art methodology. In both cases of aerial and satellite data, the procedure generates building sections with sharp edges and straight seams to neighboring touching instances. Our networks achieve high quality results in terms of metrics and visual inspection. Additionally, we publish the details and evaluation ground truth of our **Aer50-NRW** dataset, consisting of aerial RGB and DSM data showing the cities of Bonn and Cologne, Germany, together with the ground truth, which was manually adapted in the test area to facilitate meaningful metric evaluation. We presented simple method that convert the resultant instances from raster to vector format and produce a level of detail (LoD)-1 model, utilizing the input DSM. In the future, we will investigate both the fusion of RGB and panchromatic images integrated in the network and multi-task learning to extract additional building information.

Moreover, our proposed building instance segmentation approach is able to identify single building instances in both settlement types, formal and informal. This framework can obtain detailed instance segmentation masks, especially for informal regions, facilitates the accurate counting of built houses and estimation of the population residing in these areas. This information becomes critical during disasters and for coordinating humanitarian aid efforts.

Additionally, we presented POL, a framework to predict real-valued, primary orientations of initial, irregular polygons in an end-to-end trainable manner. We leveraged those angles for accurate and efficient building polygon regularization, using a simple yet effective rectilinearization algorithm. Furthermore, we demonstrated the generalization capability of POL on polygons that are very different to those in the training dataset. Our analysis showed that our method achieves similar results as those of the reference method but overcomes the limitation of discrete valued angles.

5 Level of Detail-2 Reconstruction

In this chapter, two approaches that use instance segmentation for level of detail (LoD)-2 reconstruction from aerial imagery and photogrammetric digital surface model (DSM) (Section 5.2), and satellite imagery and photogrammetric DSM (Section 5.3) are presented. The content of this chapter is based on the peer-reviewed journal papers:

[11]: ***P. Schuegraf***, *S. Shan, and K. Bittner, "Planes4lod2: Reconstruction of LoD-2 Building Models using a Depth Attention-Based Fully Convolutional Neural Network," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 211, pp. 425-437 ff., 2024*

and

[12]: ***P. Schuegraf***, *S. Gui, R. Qin, F. Fraundorfer, and K. Bittner, "Sat2building: Lod-2 Building Reconstruction from Satellite Imagery using Spatial Embeddings," Submitted to Journal of Applied Earth Observation and Geo Information, 2024.*

5.1 Problem Statement

Urbanization is one of the mega-trends that pose massive challenges for humanity. Many of these challenges are linked to buildings, the main structural elements of cities. From disaster management, flood simulation, landslide simulation to solar panel recommendation, all need precise knowledge of building locations, dimensions and appearances. A 3D building model at LoD-2, according to the CityGML standard [4], is required in those applications. One possible way to obtain LoD-2 city models is to scan these structures with terrestrial laser scanning. Yet, this is a very time and energy consuming approach, and can't quickly take into account changes in the housing stock of a city or historic buildings in large quantities. Laser-scanning from the air involves a light detection and ranging (LiDAR) sensor, which provides robust geometrical information but lacks spectral information and is much more expensive and less efficient than an optical camera. Photos from multiple angles of a scene allow the derivation of a photogrammetric DSM. Although it is more noisy than a LiDAR DSM, it is less cost-intensive and accompanied by spectral information. To make use of these data, a key step is to extract features from them. Conventional methods rely on hand-crafting such features to detect buildings and their components [5, 6, 81], but these features are often not robust to strong variations in the data. On the other hand, deep learning allows to automatically learn features from high-dimensional data, making it ideal for image recognition in remote sensing.

Although several studies have carried out LoD-2 reconstruction from airborne sensor data [5, 6, 18, 81, 85, 86] only few of them use deep learning [18, 85, 86] and none of them predicts the main planar components (i.e. roof planes) of each roof, directly based on an

image and photogrammetric DSM. As such, there is a need to uniquely identify building sections even if they have common borders. Note that we regard building sections as parts of a building with distinguishable roof-structure according to a single roof type.

The rest of this chapter is organized as follows. Section 5.2 describes a method for LoD-2 reconstruction from aerial data, noted as PLANES4LOD2, in detail. In Section 5.3, SAT2BUILDING, a method for LoD-2 reconstruction from satellite data, is introduced. Section 5.4 concludes this chapter.

5.2 PLANES4LOD2: Reconstruction of LoD-2 Building Models using a Depth Attention-based Fully Convolutional Neural Network

5.2.1 Contributions

The work in this sections extends our previous work reported in Schuegraf *et al.* [16]. In that paper, we introduced a dataset for instance segmentation of buildings and their respective roof planes, named Roof3D. Along with the data, we presented a method that jointly segments building sections and roof planes using a U-Net with a ResNet-34 backbone. Although this method showed promising results when operating solely on the image data, integrating the DSM led to a drop in performance. In Section 5.2, we introduce a depth attention module (DAM) to improve the prediction performance for both building sections and roof planes. Here, attention refers to a mechanism that models the interactions in a feature map by learning weights for computing a weighted sum of the input. In our case, the weights are calculated from DSM features. We show that the Efficient-NetB3 is a more suitable backbone for the task at hand. In the meantime, we are able to reduce the number of primitive classes in the preliminary semantic segmentation task from 5 to 4 by removing the outer boundary of building. Additionally, we make use of building sections and planes to derive an LoD-2 reconstruction of our test region in Cologne, Germany. To demonstrate the generalization capability of our method to dissimilar architectural styles and geographical locations, we perform an inference on a separate test region in Braunschweig, Germany.

In this section, we introduce a new approach, PLANES4LOD2, which has the following contributions:

- It predicts building sections and roof planes jointly, such that each roof plane is uniquely connected to a building section.
- It utilizes the predicted building sections and roof planes to achieve a complete LoD-2 reconstruction, which is represented both as a 3D shapefile and an LoD-2 DSM.
- The introduced attention module, DAM, is able to effectively and efficiently utilize the geometric features of a photogrammetric DSM in a U-Net architecture with an EfficientNetB3 backbone.

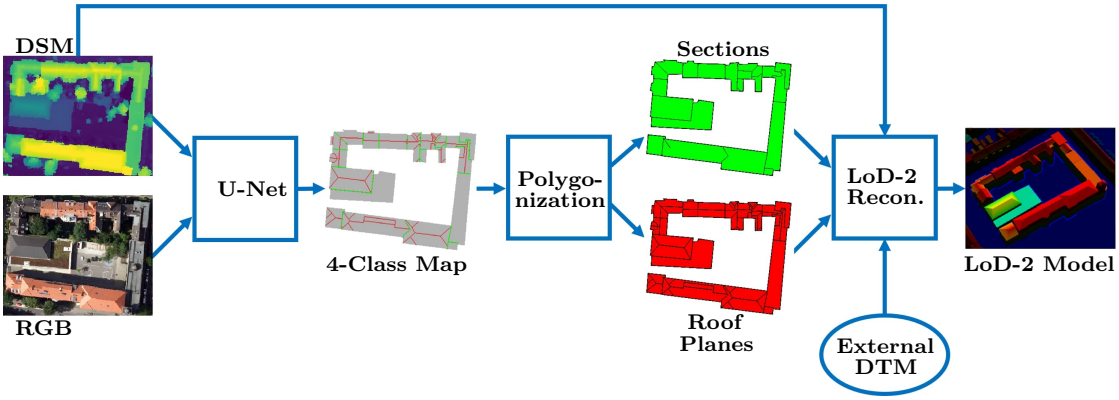


Figure 5.1: The overall workflow of PLANES4LOD2. The RGB imagery and DSM patches are passed to U-Net to produce a 4-class map. Polygonization yields building sections and roof planes. Using an external digital terrain model (DTM), LoD-2 reconstruction generates a vectorized 3D building model.

- By using two independent datasets, we show the superiority of the combination of spatial and spectral attention. Furthermore, we demonstrate the generalization capability of our approach to a test region that is dissimilar in architectural style and geographical location from the primary test region.

5.2.2 Methodology

We will first give an overview of our workflow, the PLANES4LOD2 method, and then describe its three major steps, including instance segmentation, polygonization and LoD-2 reconstruction.

5.2.2.1 Overview

The LoD-2 reconstruction of buildings can be achieved using three main inputs: **1)** building sections, **2)** building planes and **3)** a normalized digital surface model (nDSM).

The definition of building section is often ambiguous. It often refers to a building that has a primitive roof structure, but it can also be interpreted as the building belonging to a building address. In the end, the definition is tightly connected to the ground truth. The data from a public source that we use for training is based on the address definition. On the other hand, addresses are not always visibly discernible. Hence, for the hand-labeled data in the inference, we use the roof primitive definition. In the rest of chapter, we also refer to building section as a roof.

The nDSM is obtained by subtracting a DTM, acquired from a public source, from the photogrammetric DSM. We derive the polygons of building sections and building planes by a two-step procedure. The first step consists of passing an RGB image together with a photogrammetric DSM to fully convolutional neural network (FCN), which then produces a 4-class segmentation map. The four classes are background, separation lines between building sections, separation lines between roof planes that do not lie at the

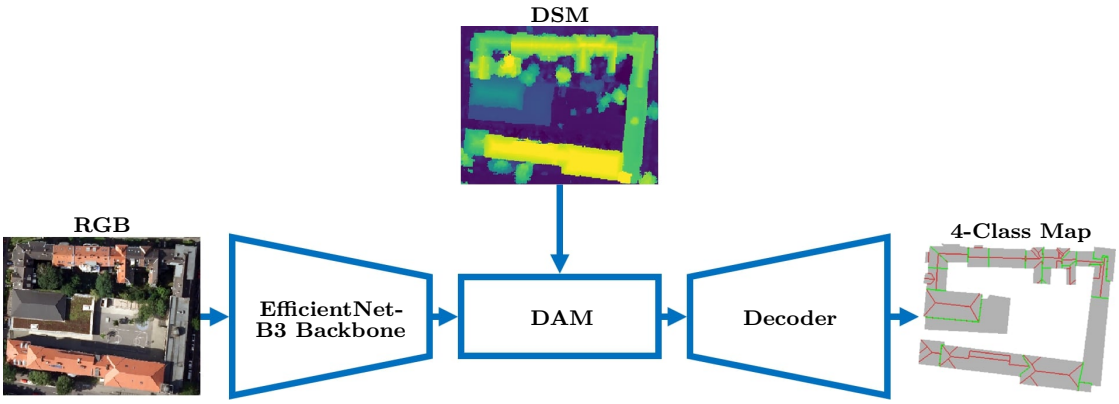


Figure 5.2: Our proposed DepthAtt-EfficientUnetB3 architecture. The EfficientNet-B3 backbone extracts features from the RGB data, which are then enriched in the DAM module by DSM information. The decoder reconstructs geometrical details to produce a 4-class map.

junctions between sections, and building segments. In the second step, holes in the line classes are filled using morphological dilation. Then, raster instances are obtained using the watershed transform. Afterwards, the resulting raster instances are polygonized and simplified. As the last step, the polygons and the nDSM are used to generate the LoD-2 model. In that step, random sample consensus (RANSAC) is used to fit 3D roof planes, while ridge lines are generated by intersecting roof planes. Figure 5.1 shows the overall workflow of PLANES4LOD2 as described above.

5.2.2.2 Network Architecture

For the task of building section segmentation and roof plane segmentation, UResNet34 has been leveraged in Schuegraf *et al.* [16]. Yet, this architecture has multiple drawbacks. First of all, the ResNet architecture has been outdated by the success of the EfficientNet architecture. Second, UResNet34 doesn't gain from the inclusion of height information, since neighboring buildings may not vary in height, but only in spectral appearance. Thereby, the network is confronted with confusing information. This observation also holds when including the SkipFuse-scheme to the UResNet34 [16]. Consequently, we propose the DepthAtt-EfficientUnetB3 architecture. In Figure 5.11, the individual parts of our architecture are outlined. The first part of the name DepthAtt refers to a depth attention mechanism that we call DAM, which is in the center of Figure 5.11. DAM is applied at the last layer of the encoder of a U-Net architecture. It receives a photogrammetric DSM patch as the input. DAM uses two different attention mechanisms based only on the DSM, leveraging height features at the deep part of the network. We apply a sequence of strided convolutional layers, ReLU activations and maximum pooling layers to the DSM, as is visualized in the upper part of Figure 5.3. The convolution operations allow automatic learning on features from the raw height information. ReLU introduces non-linearity to the network. The stride in the convolutions and the maximum pooling layers bring the height features to the same resolution as the feature maps of the bottleneck of the image network. These height features are then used in two

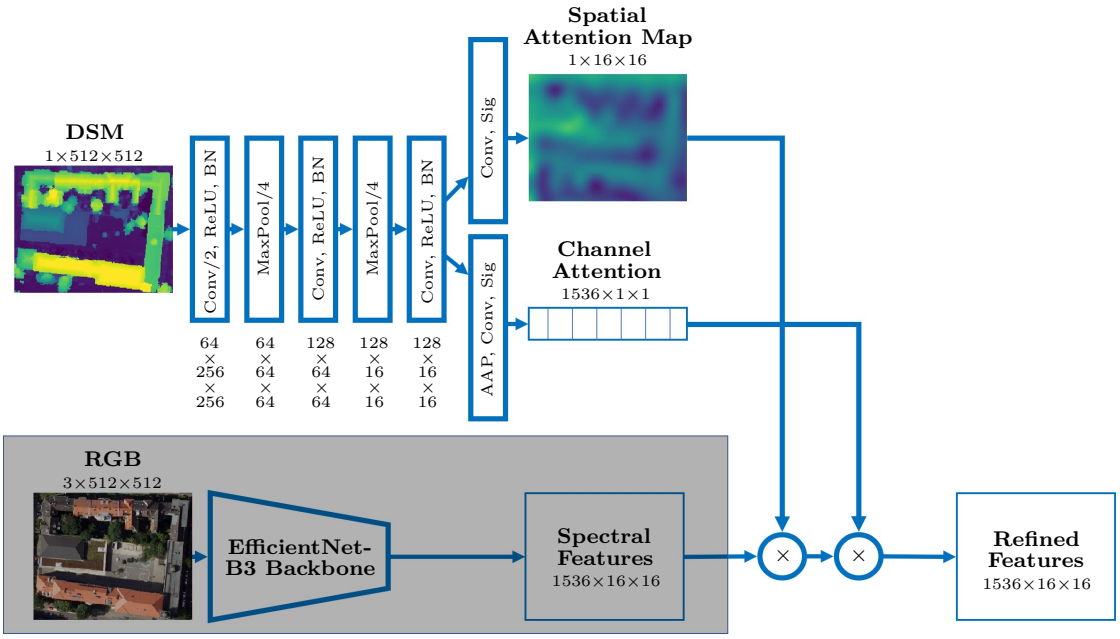


Figure 5.3: The structure of DAM. The shadowed area is not part of DAM but is visualized to show the origin of the spectral features. Conv and Conv/2 refer to convolutional layers with stride 1 and 2. ReLU means rectified linear unit, BN represents batch normalization, and MaxPool/4 refers to a maximum pooling layer with stride 4. AAP refers to adaptive average pooling and Sig stands for the sigmoid function, which maps its inputs to the range $[0, 1]$. The spatial attention map is visualized with 32-times the original resolution using bi-cubic interpolation.

types of attention layers. Both attention layers consist of a convolutional layer followed by a sigmoid activation. Yet, one of the attention layers uses a channel-wise convolution to place attention on features other than regions, whereas the other attention layer places the attention on pixels, to enhance features in certain spatial locations, which are derived from the DSM. DAM enables fusion of the RGB data and DSM at a coarse spatial resolution (see lower part of Figure 5.3), namely at the bottleneck. Hence, small spatial shifts between the two inputs affect little to the extracted feature maps. Moreover, since the DSM is only used in the attention mechanism, the network focuses on the features from the RGB image, but can use height features to suppress noise and guide the training process. The EfficientUnetB3 receives only an RGB image patch and extracts features sequentially in the encoder, leveraging the EfficientNetB3 architecture [34], which is shown on the left side of Figure 5.11. As for the decoder (right part of Figure 5.11), what we used is similar to U-Net in Baheti *et al.* [106]. This includes skip-connections to allow for better information flow from the encoder to the decoder. Our implementation of the EfficientUnetB3 is mostly based on an implementation that is publicly available on github[1]. During the training, we use a *softmax* activation, since it is required by our loss functions. When doing inference, we use the argmax of the

[1] https://github.com/zhoudaxia233/EfficientUnet-PyTorch

network outputs to produce class predictions for each pixel.

5.2.2.3 Polygonization

Although the raster results are valuable for some applications, most further applications, e.g. LoD-2 reconstruction, require vector data as the input. Hence, we convert our 4-class maps to two different vector layers. Note that when we refer to simplification algorithms in the following paragraphs, we always simplify common borders of polygons and the rest of the polygons separately to avoid irregular gaps between neighboring instances. Section polygons with an area smaller than $4\,\mathrm{m}^2$ are dropped, since they most likely correspond to false positive noise.

We achieve the separation of buildings into sections by using the same learning-free post-processing scheme as in Schuegraf *et al.* [8]. We treat the plane separation line as part of a section. Hence, this leaves us with three classes: background, building segment and section separation line. Then we use the watershed transform to infer instances. As the seed for the watershed transform, we dilate the section separation line, using a disk with radius $R_{\mathrm{sec}} = 6$ as the structuring element, and remove it from the building segment. The mask element for the watershed transform is the inverse of the background class raster. The surface map will be the segmentation raster with value 0 for background, value 1 for building segment, and value 2 for building section separation. To obtain boundary pixels, we use tree search and simplify the resulting polygon by utilizing the Douglas Peucker algorithm [105] with tolerance $\epsilon_{\mathrm{sec}} = 0.5\,\mathrm{m}$.

For the generation of a roof plane vector layer, we follow the same procedure as for the building section layer. The only difference is that we reconstruct the plane separation line by using both the building section separation and plane separation as the separation line. We again apply dilation to improve separation between sections, but with a disk of radius $R_{\mathrm{plane}} = 6$. To simplify the roof plane polygons with the Douglas Peucker algorithm, we use the tolerance $\epsilon_{\mathrm{plane}} = 0.5\,\mathrm{m}$.

5.2.2.4 LoD-2 Model Generation

The next task is to generate the LoD-2 model based on our predicted roof plane geometries. As the first step, we count the number of predicted planes of each building section. For a single plane, we estimate the roof plane parameters (a_i, b_i, c_i, d_i) using RANSAC [107]. The parameters define a plane with the equation $a_i x + b_i y + c_i z + d = 0$ for roof plane i. We then check whether the plane is nearly horizontal or parallel to the xy-plane ($a_i \sim 0$, $b_i \sim 0$, $z_i \sim 1$). In that case, we improve regularization by assuming complete flatness of the roof plane and average the height value of all vertices inside the roof polygon to obtain a single height value at all vertices.

If there are two planes for a roof, we assume that it is a gable roof. Even though not all roofs with two planes are of roof type gable, this assumption holds for most buildings in our datasets. We estimate a plane for each of the roof planes using RANSAC. For plane estimation, we sample all height values from the nDSM that lie in the area surrounded by the roof plane polygon. Next, we use the two sets of plane parameters

(a_1, b_1, c_1, d_1) and (a_2, b_2, c_2, d_2) to compute their intersection line in the point-slope expression $\overrightarrow{l}(t) = \overrightarrow{p_0} + t \times \overrightarrow{s}$, where $p_0 = [x_0, y_0, z_0]^\intercal$, $\overrightarrow{s} = [\delta_x, \delta_y, \delta_z]^\intercal$ with

$$\delta_x = b_1 c_2 - b_2 c_1, \tag{5.1}$$
$$\delta_y = a_2 c_1 - a_1 c_2, \tag{5.2}$$
$$\delta_z = a_1 b_1 - a_2 c_1, \tag{5.3}$$

$$x_0 = \begin{cases} 0, & \text{if } \delta_x \neq 0 \\ (d_1 c_2 - d_2 c_1) \div \delta_y, & \text{if } \delta_x = 0 \wedge \delta_y \neq 0 \\ (d_2 b_1 - d_1 b_2) \div \delta_z, & \text{if } \delta_x = 0 \wedge \delta_y = 0 \wedge \delta_z \neq 0, \end{cases} \tag{5.4}$$

$$y_0 = \begin{cases} (c_1 d_2 - c_2 d_1) \div \delta_x, & \text{if } \delta_x \neq 0 \\ 0, & \text{if } \delta_x = 0 \wedge \delta_y \neq 0 \\ (d_1 a_2 - d_2 a_1) \div \delta_z, & \text{if } \delta_x = 0 \wedge \delta_y = 0 \wedge \delta_z \neq 0, \end{cases} \tag{5.5}$$

$$z_0 = \begin{cases} (b_2 d_1 - b_1 d_2) \div \delta_x, & \text{if } \delta_x \neq 0 \\ (d_2 a_1 - d_1 a_2) \div \delta_y, & \text{if } \delta_x = 0 \wedge \delta_y \neq 0 \\ 0, & \text{if } \delta_x = 0 \wedge \delta_y = 0 \wedge \delta_z \neq 0 \end{cases} \tag{5.6}$$

and $t \in \mathbb{R}$. We intersect this line with the union polygon of the two roof planes using a line search. There, we iteratively evaluate the point-slope expression for different pairs (t_0, t_1), check whether the line that passes through the two resulting points $\overrightarrow{l}(t_0)$ and $\overrightarrow{l}(t_1)$ intersects the union polygon, until we find a pair (t_0, t_1). At the two intersection points, we use their average height according to the point-slope expression. The heights of the remaining vertices of the union polygon are complemented using the initial planes parameters. Next, we split the union polygon through the intersection line defined by the two intersection points. As a result, we yield two roof plane 3D-polygons with consistent height at the ridge line, i.e. avoiding vertical jumps of elevation.

For buildings with more than two planes, we use RANSAC to determine the plane parameters similar as for two planes. If the normal of a roof plane indicates a non-inclined plane, we model it as a flat roof with the average elevation at all vertices. We model the non-flat roof planes by using their estimated plane parameters to complement the height values at the vertices.

To complete the building models, we further include ground and wall polygons.

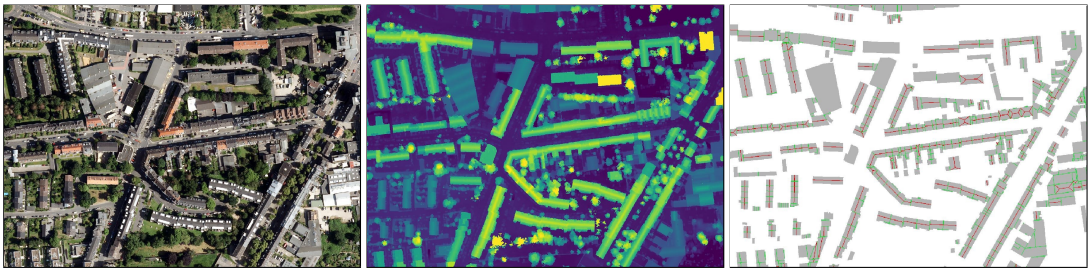


Figure 5.4: Excerpt from the training data of Roof3D. The RGB imagery was captured with a ground sampling distance (GSD) of 0.1 m, whereas the DSM was computed with 0.5 m GSD. Before being passed to the network, both of them are resampled at 0.3 m GSD using bicubic interpolation, since the ground truth is generated at 0.3 m GSD.

5.2.3 Experiments

5.2.3.1 Data

We use two different datasets for the experiments in this chapter. The first dataset is Roof3D [16], with data from the cities Cologne and Berlin for training and Cologne for evaluation. The RGB imagery and photogrammetric DSMs in Roof3D are comprised of real and synthetic pairs. The addition of the synthetic data increases the size of the training dataset and comes along with perfectly matching ground truth. Next to the perfect annotations of the synthetic data, Roof3D includes two more sources of ground truth. One source includes building outlines from the German building cadastre and coarse roof plane annotation from a semi-automatic method based on laser-scanning. The other source is manual annotation of real image and DSM pairs. The first testing region is that of Roof3D in Cologne, Germany, which we use for ablation, is annotated manually and has exclusively non-synthetic inputs. See Figure 5.4 for a visualization of an area in the training data. Furthermore, the testing region does not geographically overlap with the training data. Refer to Schuegraf *et al.* [16] for further details about Roof3D. For the construction of a reference LoD-2 DSM, we use public data[2]. This reference data stems from a semi-automatic method that uses cadastre data and laser scanning, which often leads to erroneous annotations. The testing set of Roof3D originates from the same flight campaign as some of the images used for training.

As for our second dataset, showing a part of the city Braunschweig, Germany, is solely used for testing and stems from a different flight campaign, with different lighting conditions, architectural styles and viewing angles, leading to dissimilar artefacts in the orthorectified imagery. We use the same tiling scheme for all our tests as in Roof3D. Both the RGB data and DSM in the two datasets have 0.3 m GSD after resampling.

5.2.3.2 Training Details

It is important to train an FCN according to the requirements of the task at hand. One important aspect is the choice of the loss function, which defines the learning objective

[2] https://www.opengeodata.nrw.de/produkte/geobasis

together with the ground truth. We use the weighted cross-entropy loss, which is a standard choice for semantic segmentation tasks, with weight 1 for the background class, 6 for the roof plane separation, 6.2 for the building section separation and 1.5 for the building segment class. We obtained these values by using the median frequency weighting heuristic

$$\mathrm{w}_{cl} = \frac{\mathrm{freq}_m}{\mathrm{freq}_{cl}}, \tag{5.7}$$

where freq_m is the median of the frequencies of pixels of each class and freq_{cl} is the pixel frequency of class cl. However, the cross-entropy loss is known to generate models producing blurry objects. To obtain sharper object boundaries, we combine the cross-entropy loss with the generalized dice loss [92], which has inverse frequency class weights. Where noted ("Topo") we also use the topological loss [15] to regularize the semantic raster output of the respective network. Topological loss was previously applied to regularize building footprints [8, 48]. We apply it to the building segment class (weighted with 0.05 in the loss function) and the union of building section separation and roof plane separation lines (0.1), as two separate terms in the loss function. As the optimization algorithm, we leverage AdamW [108] with weight decay of 0.0001, as it is a common choice for training FCNs.

5.2.3.3 Evaluation Metrics

For evaluation, we use two kinds of metrics. The first ones are for the evaluation in 2D, and the second in 3D.

To quantitatively evaluate the two instance segmentation tasks, building section and roof plane segmentation, we use average precision (AP) and average recall (AR). The harmonic mean of these two is

$$F1_{INST} = 2 \times \frac{\mathrm{AP} \times \mathrm{AR}}{\mathrm{AP} + \mathrm{AR}}. \tag{5.8}$$

AP and AR are two commonly used metrics for instance segmentation. The two metrics highly depend on the overlap between the predicted instances and ground truth instances and are thus highly discriminative. Furthermore, ambiguous ground truth can lead to low values of these metrics. AP focuses on the quality of the predicted results by considering both precision and recall, while AR focuses solely on the proportion of relevant items that are successfully retrieved. Hence, AR responds better to over-segmenting methods, whereas AP has a higher score on under-segmenting methods. Both metrics are based on the polygonized results and polygonized ground truth. Since these metrics only give insight to quantitative aspects of the results, we also carry out a visual inspection for qualitative evaluation in some of the experiments.

For the quantitative evaluation of the reconstructed LoD-2 DSM with our and reference

methods, we use the root-mean-squared error (RMSE)

$$\text{RMSE} = \sqrt{\frac{\sum_p |\hat{h}_p - h_p|^2}{N}}, \tag{5.9}$$

where p is the respective pixel, N is the total number of pixels, $\hat{h}_p$ is the predicted elevation at pixel p and h_p is the reference elevation at pixel p. Yet, the RMSE is sensitive to the scale of the values and to outliers. Hence, we use a more robust regression metric, mean absolute error (MAE)

$$\text{MAE} = \frac{\sum_p |\hat{h}_p - h_p|}{N}. \tag{5.10}$$

Another metric, which originates from stereo matching and optical flow, is the T_t-error

$$\text{T}_t = \frac{1}{N} \sum_p \begin{cases} 1 & \text{if } |\hat{h}_p - h_p| \geq t \\ 0 & \text{otherwise} \end{cases}, \tag{5.11}$$

which gives the percentage of pixels, where the predicted height has an absolute deviation of more than t from the ground truth, where t is expressed in meters. We use the strict T_1-error and the T_3-error to gain a better overall understanding of the quality of the predictions of our method.

5.2.3.4 Experiment Descriptions

For the analysis of our method, we perform multiple sets of experiments.

The public Roof3D dataset is suitable for the evaluation of algorithms on the tasks of segmenting building sections and roof plane extraction. Hence, we use it to carry out an ablation study to find the best setting of our architecture.

As a baseline model, we train UResNet34 for the 4-class semantic segmentation task using only RGB imagery. Post-processing techniques, as outlined in Section 3, are applied to obtain building sections and roof planes. In Schuegraf *et al.* [16], it was shown that Fuse-UResNet34 does not improve UResNet34, even though it has auxiliary height information as input. To address such drawbacks, we experiment with DepthAtt-UResNet34 with channel and spatial attention, leveraging DAM. To discern the impact of attention mechanisms, we evaluate DepthAtt-UResNet34 with only spatial attention, only channel attention, and both channel and spatial attention. Spectral attention uses only the features from the RGB image to derive attention maps. In an effort to determine the efficacy of depth and spectral attention, we introduce SpecAtt-UResNet34 with both channel and spatial attention. Additionally, we test the combination of spectral and depth attention in SpecDepthAtt-UResNet34. Given the success of EfficientNet in various image recognition tasks, we explore the performance of the DepthAtt-EfficientUnetB3 with channel and spatial attention architecture. We evaluate the EfficientNetB3 backbone and compare them to DepthAtt-UResNet34. To enhance regularization in the segmentation outputs, we introduce the topology loss to DepthAtt-EfficientUnetB3-Topo channel &

Table 5.1: Results of various models for the building section segmentation task on the Roof3D dataset. ↑ indicates that the higher values of the metrics correspond to better quality.

Architecture	MODALITY	$AP \uparrow$	$AR \uparrow$	$F1_{INST} \uparrow$
UResNet34	RGB	0.183	0.371	0.245
Fuse-UResNet34	RGB+DSM	0.176	0.365	0.237
DepthAtt-UResNet34 channel & spatial	RGB+DSM	0.201	0.390	0.265
DepthAtt-UResNet34 spatial	RGB+DSM	0.179	0.365	0.240
DepthAtt-UResNet34 channel	RGB+DSM	0.170	0.359	0.231
SpecAtt-UResNet34 channel & spatial	RGB	0.194	0.379	0.257
SpecDepthAtt-UResNet34 channel & spatial	RGB+DSM	0.183	0.359	0.242
DepthAtt-EfficientUnetB3 channel & spatial	RGB+DSM	**0.207**	**0.398**	**0.272**
DepthAtt-EfficientUnet-B3-Topo channel & spatial	RGB+DSM	0.197	0.361	0.255

spatial architecture.

We leverage DepthAtt-EfficientUnetB3-Topo channel & spatial to derive building sections and roof planes from pairs of RGB imagery and photogrammetric DSMs. The input photogrammetric DSM is normalized using a DTM from a public source to extract heights above ground. Then, we apply our LoD-2 reconstruction method from Section 5.2.2.4. We use SAT2LOD2 [18] for comparison to our method with the software described in Gui *et al.* [109]. We feed only the ortho image and photogrammetric DSM to SAT2LOD2, omitting the open street map (OSM) data.

One of the great promising properties of deep learning-based algorithms is their generalization capability. To test this, we apply our DepthAtt-EfficientUnetB3-Topo channel & spatial to a dataset that does not geographically overlap with the Roof3D dataset. Since this dataset stems from an entirely different campaign, this implies not only different architectural styles, but also different viewing angles and lighting conditions leading to a different appearance of buildings in the ortho image than those in the Roof3D dataset, as well as different architectural styles.

5.2.4 Results

5.2.4.1 Roof3D

In this subsection, we compare the quantitative results as in Tables 5.1 and 5.2 and the qualitative results from a visual inspection of the models trained and evaluated on the Roof3D dataset. The regression metrics for the 3D reconstruction task are provided in Table 5.3.

UResNet-34 successfully segments building sections and roof planes.

Comparing Fuse-UResNet34 and DepthAtt-UResNet34 channel & spatial, we observe that the latter is scoring higher metric values. Hence, the noise suppression and feature refinement of DAM lead to improved metrics. On the contrary, directly incorporating DSM makes it harder for the network to focus on RGB data, which contains the most important spectral information. Inspecting metric scores of the three models with different depth attention settings, the combination of channel and spatial attention outperforms

Table 5.2: Results of various models in the roof plane segmentation task on the Roof3D dataset. The second-last and third-last row correspond to identical metric values. ↑ indicates that the higher values of the metrics correspond to better quality.

Architecture	MODALITY	$AP \uparrow$	$AR \uparrow$	$F1_{INST} \uparrow$
UResNet34	RGB	0.115	0.279	0.163
Fuse-UResNet34	RGB+DSM	0.119	0.289	0.169
DepthAtt-UResNet34 channel & spatial	RGB+DSM	0.127	0.295	0.178
DepthAtt-UResNet34 spatial	RGB+DSM	0.100	0.267	0.146
DepthAtt-UResNet34 channel	RGB+DSM	0.117	0.283	0.166
SpecAtt-UResNet34 channel & spatial	RGB	0.123	0.282	0.171
SpecDepthAtt-UResNet34 channel & spatial	RGB+DSM	0.109	0.265	0.154
DepthAtt-EfficientUnetB3 channel & spatial	RGB+DSM	0.138	0.303	0.190
DepthAtt-EfficientUnetB3-Topo channel & spatial	RGB+DSM	**0.149**	**0.312**	**0.202**

Table 5.3: Comparison of the LoD-2 reconstruction results on the two test regions. ↓ indicates that the lower values of the metrics correspond to better quality

Method	Dataset	RMSE ↓	MAE ↓	$T_1 \downarrow$	$T_3 \downarrow$
SAT2LOD2	Roof3d	5.28 m	2.18 m	0.26	0.15
PLANES4LOD2	Roof3d	**3.34 m**	**1.06 m**	**0.18**	**0.07**
SAT2LOD2	Braunschweig	2.52 m	0.71 m	0.10	0.08
PLANES4LOD2	Braunschweig	**1.39 m**	**0.24 m**	**0.04**	**0.02**

spatial attention and channel attention. Using only one of the two attention mechanisms is not sufficient [110], but the combination of both leads to a more effective use of the features provided by the RGB image encoder.

We observe in Table 5.1 that the depth attention improves the performance of spectral attention in SpecAtt-UResNet34 channel & spatial. The combination of spectral and depth attention in SpecDepthAtt-UResNet34 is ranking even behind SpecAtt-UResNet34 channel & spatial. The attention provided by the input RGB data is helpful, but does not provide as much additional information as the depth attention does. Averaging the attention maps from the spectral and depth information does not seem to be the best way to make use of both mechanisms. We compared eight different settings of EfficientNet as the backbone and EfficientNetB3 outperforms all other versions on both tasks. The replacement of the ResNet34 backbone by the EfficientNetB3 backbone in DepthAtt-EfficientUnetB3 channel & spatial consistently outperforms all other backbones on all metrics. The most likely reason for the superiority of EfficientUnet over ResNet34 is its fine-grained scalability as compared to UResNet34. This allows us to choose a properly dimensioned feature extractor.

Training DepthAtt-EfficientUnetB3 spatial & channel with the topology loss leads to a drop in performance on building segmentation, but to a rise in performance on roof plane segmentation. The strength of introducing the topology loss is that it makes the predictions visually more similar to the ground truth. Because of the complex junctions of separation lines between roof planes, the model profits strongly if it is pushed to

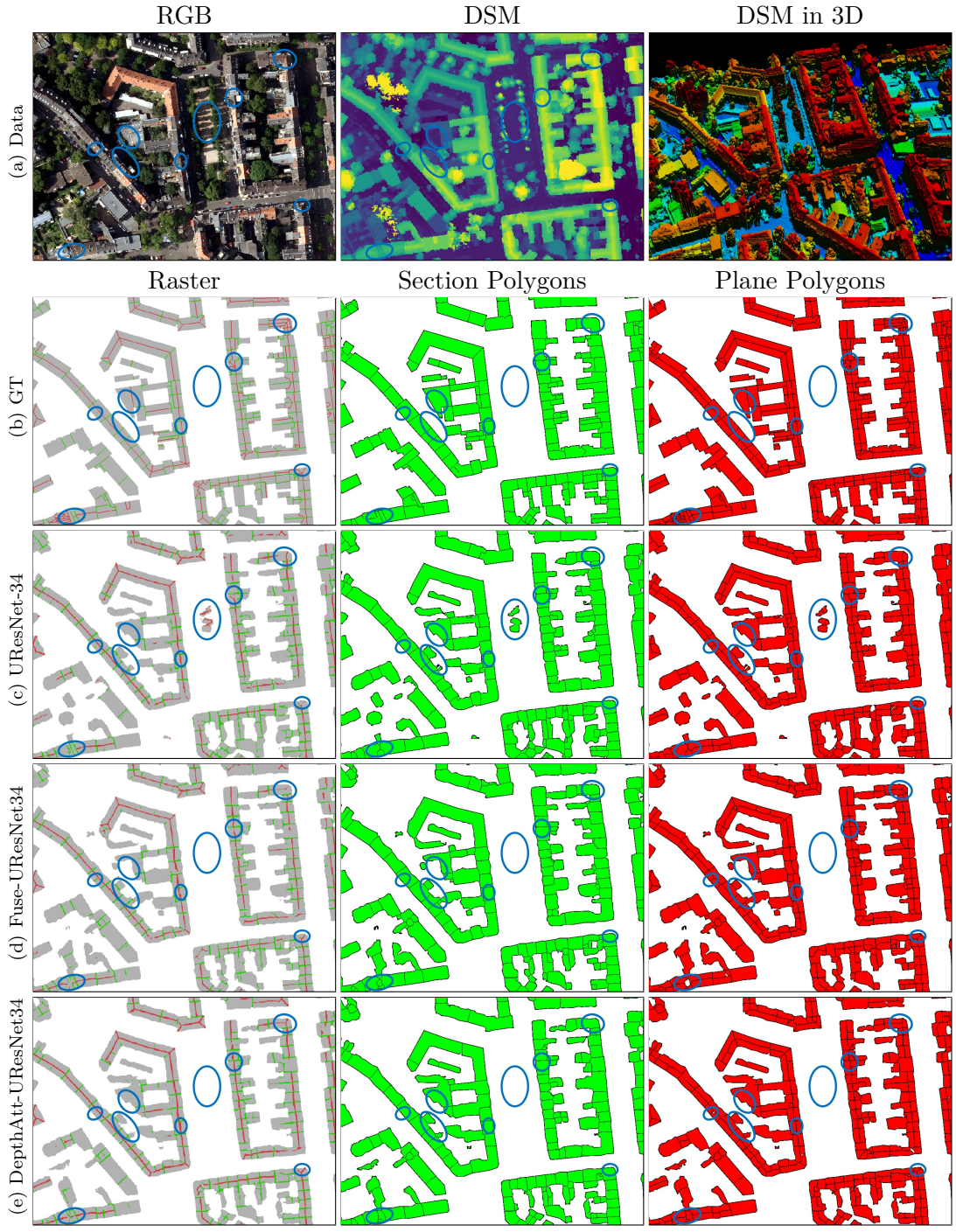


Figure 5.5: Visualization of the 2D results on a crop of the Roof3D test region. Row (a) shows the input data. Row (b) shows the reference ground truth and (c) the prediction of the UResNet-34. Row (d) presents the results derived from the Fuse-UResNet-34 and (e) those of the DepthAtt-UResNet34 channel & spatial. Blue oval highlight the differences.

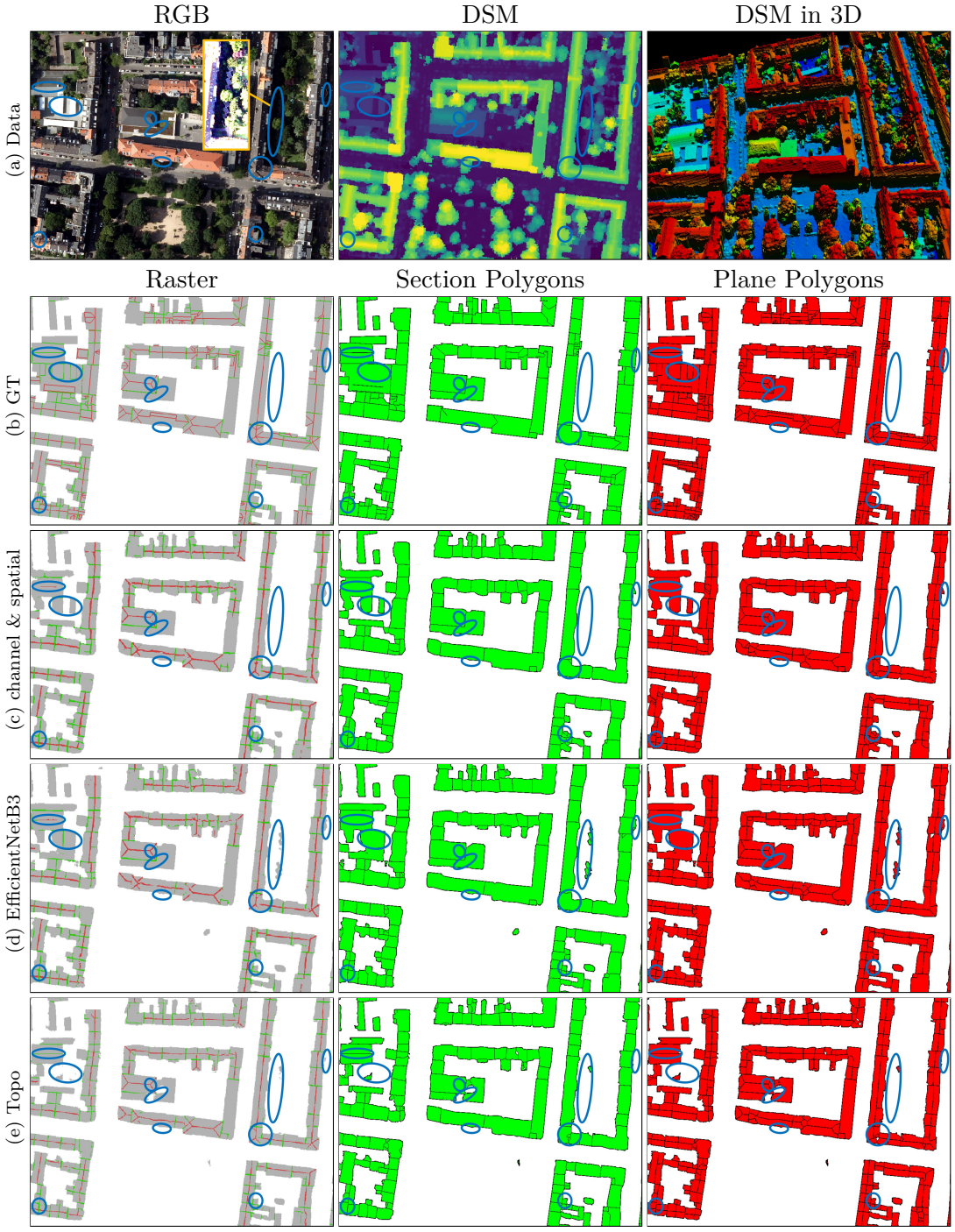


Figure 5.6: Visualization of the 2D results on another crop of the Roof3D test region. Row (a) shows the input data. Row (b) presents the reference ground truth and (c) the prediction of the DepthAtt-UResNet34 channel & spatial. Row (d) shows the results of the DepthAtt-EfficientUnetB3 channel & spatial and (e) those of the DepthAtt-EfficientUnetB3-Topo channel & spatial. Blue ovals highlight the differences.

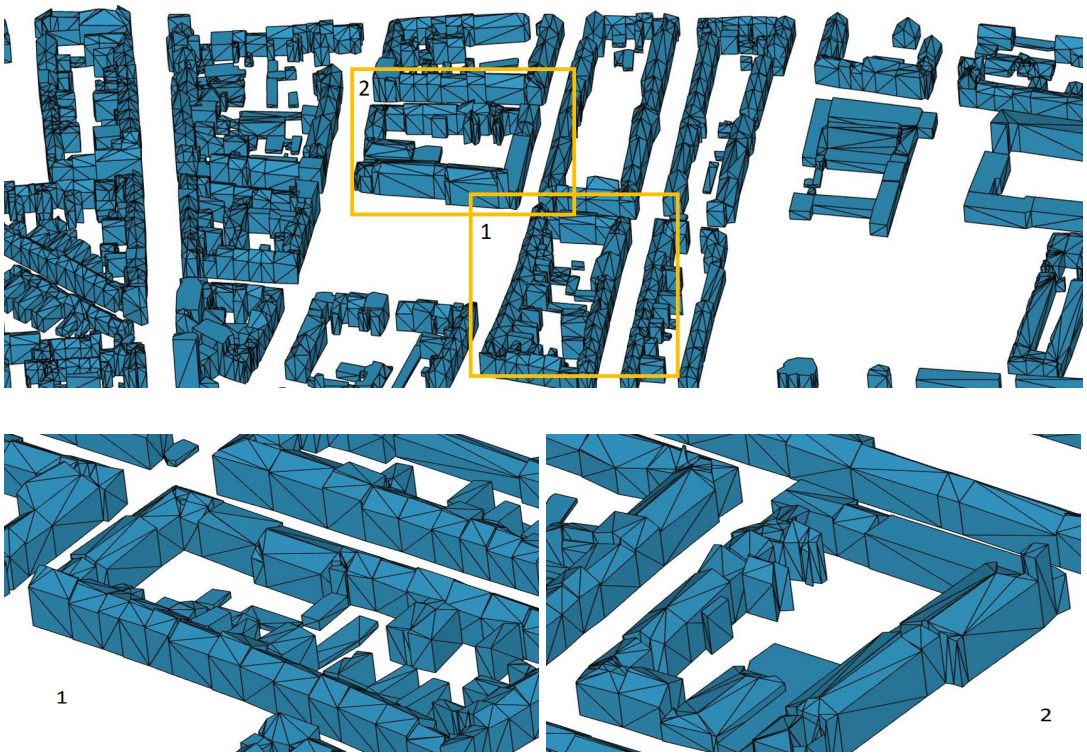
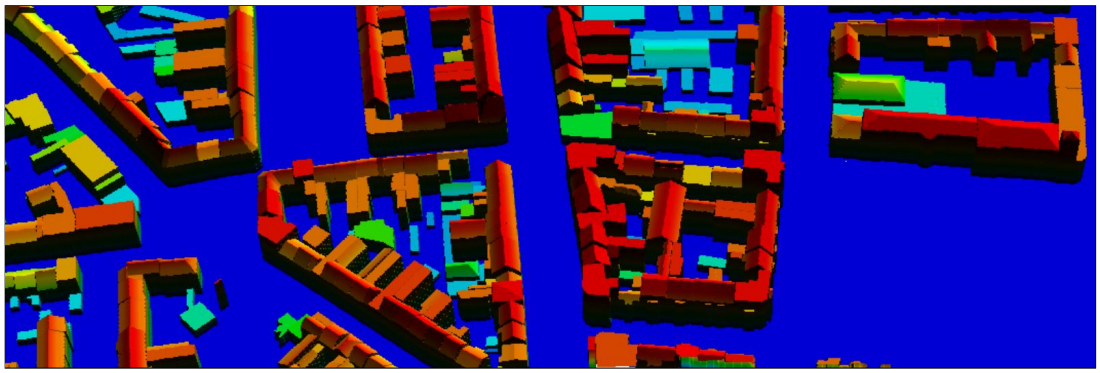


Figure 5.7: The resulting 3D LoD-2 model in vector format of a scene in the Roof3D test region. The image in the top row shows an overview, whereas the bottom row gives two detailed views.

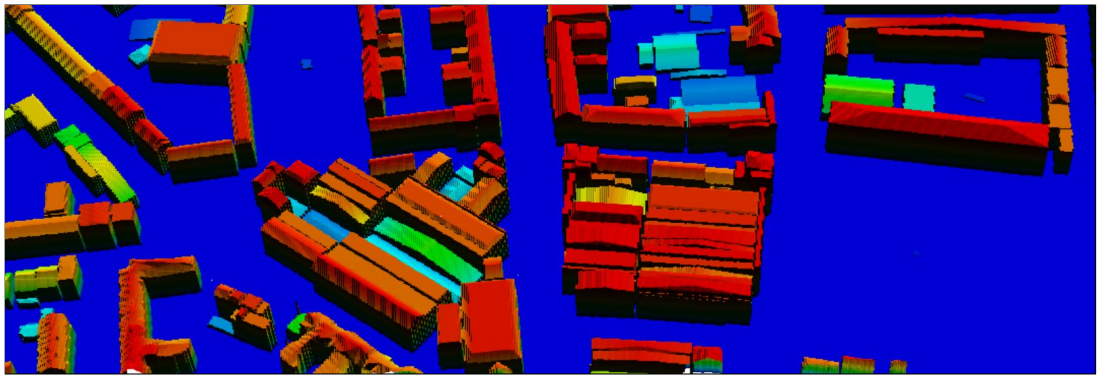
segment thin and topologically correct lines and junctions. On the other hand, building segmentation profits more from thicker lines, which avoids gaps better and hence leads to less missed separations between resulting building section polygons.

Evaluating the LoD-2 DSM with reference to the ground truth raster, our method achieves better values than SAT2LOD2 on all metrics, indicating more accurate geometrical results. In addition, our PLANES4LOD2 has accurate presentation about roof planes. Furthermore, PLANES4LOD2 recognizes inner yards and can properly handle such topological structure of buildings, whereas SAT2LOD2 regards them as parts of the buildings. All these factors demonstrate that PLANES4LOD2 performs superior.

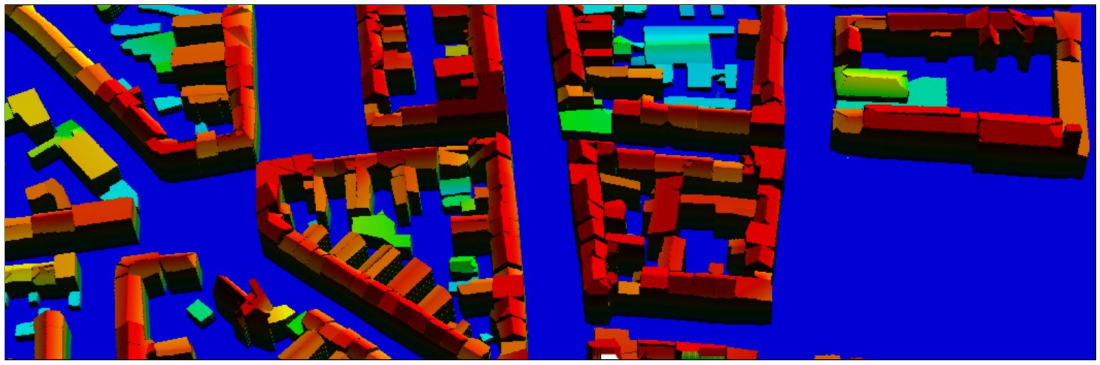
Figure 5.5 visualizes the predictions of models with different modalities and fusion strategies. We highlight multiple places where we noted significant visual deviations. In row (e), DepthAtt-UResNet34 channel & spatial produces separation lines in the raster segmentation. The results are more complete than in the other rows, which leads to more accurate and regular building sections and roof plane polygons than UResNet34 in row (c) and Fuse-UResNet34 in row (d). Furthermore, Fuse-UResNet34 sometimes produces false positives. Figure 5.6 presents the comparisons of the results obtained from models with the backbone architectures ResNet-34 (row (c)) and EfficientNetB3 (rows (d) and (e)) and under the addition of the topology loss (row (e)). In the RGB image, we highlight a rectangle by rescaling it to the lowest 30 % of pixel values, which



Ground Truth



SAT2LOD2



PLANES4LOD2

Figure 5.8: Visualization of the results of our and a reference method for LoD-2 reconstructions of the test region of Roof3D. For the visualization of height features, we use a color mapping from blue (low) to red (high).

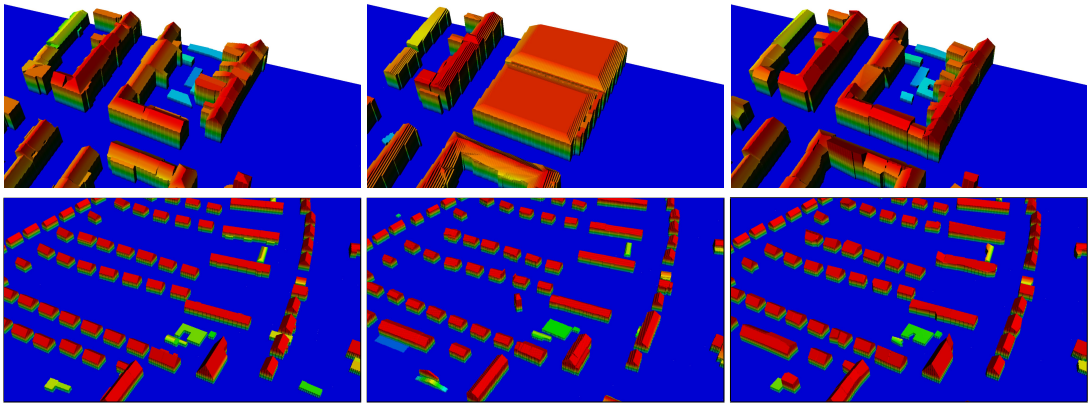


Figure 5.9: An example building in the test region of Roof3D (first row) that is reconstructed as a large block by a reference method and reconstructed in detail in our reconstruction and some example results in the test area in Braunschweig, Germany (second row). Both methods achieve practically identical results for simple roof shapes, as can be seen in the second row. However, PLANES4LOD2 can handle more complex buildings as visualized in the first row.

correspond to shadows in the original RGB image. In the highlighted box, regarding the low corresponding elevation in the DSM, the visible building structure in the middle most likely corresponds to garages. This structure is detected as buildings by DepthAtt-EfficientUnetB3 channel & spatial, whereas DepthAtt-UResNet34 channel & spatial segments it as background. In most parts of the visualization, DepthAtt-EfficientUnetB3-Topo channel & spatial produces thinner and more complete lines than the other two models, though it sometimes fails to detect building segments. Overall, the two models with EfficientNetB3 as backbones produce slightly more complete separation lines.

In Figure 5.7, a resulting LoD-2 model is visualized in vector format. From Figure 5.8 it becomes clear that the 3D building model of our method looks more similar to the ground truth than the one from SAT2LOD2. In the first row of Figure 5.9, we provide a more detailed visualization of the reconstruction performed by SAT2LOD2, our method and the reference ground truth. The rooftops generated by SAT2LOD2 look very regular because they are based on roof type reconstruction. This induces symmetry into the resulting roof of the building model. Our method generates building models that are visually much closer to the ground truth, but does not enforce symmetric properties similar to SAT2LOD2. Furthermore, SAT2LOD2 cannot reconstruct buildings with inner yards correctly, because it is based on binary building segmentation. In contrast, we reconstruct buildings based on individual sections and directly segment their roof planes. Hence, our method can capture inner yards well, which is an advantage in scenarios with complex building structure, as it is typical in European cities.

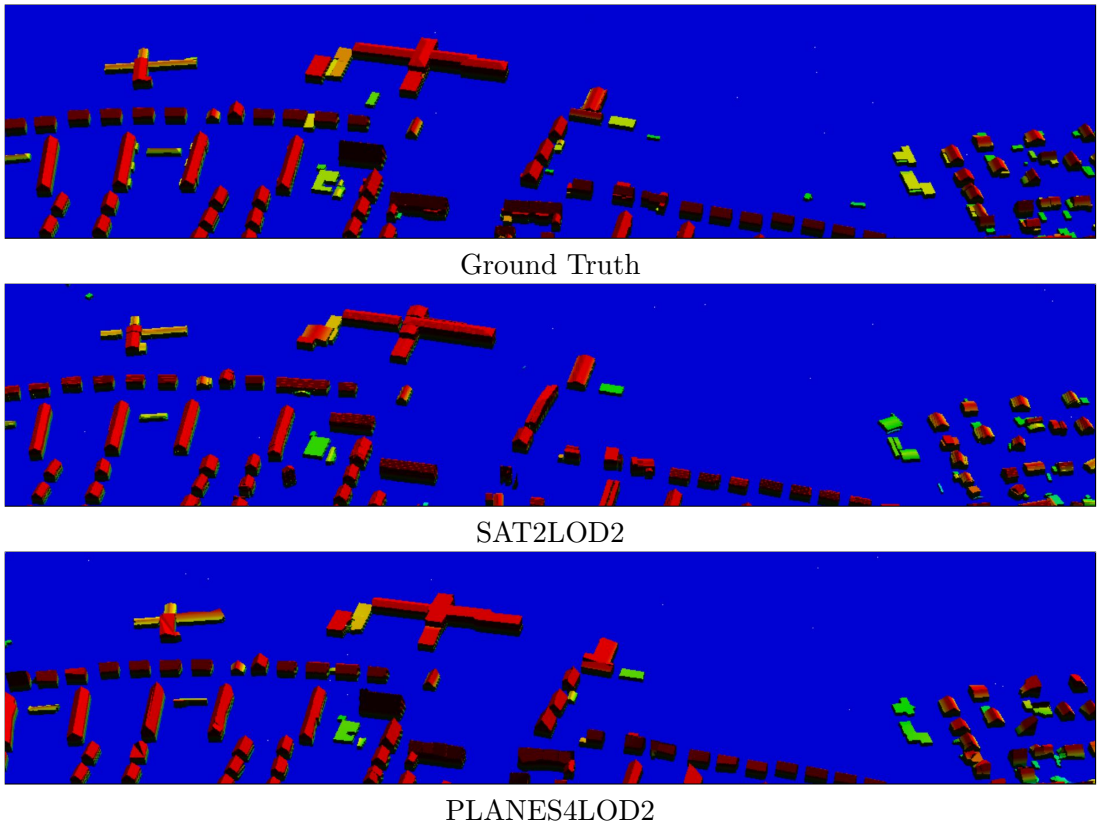


Figure 5.10: Visualization of the results of our and a reference method for LoD-2 reconstructions of buildings in Braunschweig. For the visualization of height features, we use a color mapping from blue (low) to red (high). The model is trained with data from Cologne and Berlin. PLANES4LOD2 profits most from the high resolution RGB image, allowing it to separate connected or close building sections. Furthermore, it is capable to filter noise from the DSM.

5.2.4.2 Generalization

To test the capability of our LoD-2 reconstruction method to adapt to an entirely new scene with different lighting conditions and different architectural styles, we evaluated it on a test region in Braunschweig, Germany. We also evaluated the SAT2LOD2 method on the same data for comparison. Quantitatively, Table 5.3 shows that our method scores RMSE 1.39 m, MAE 0.24 m, T_1 0.04 and T_3 0.02, whereas SAT2LOD2 achieves RMSE 2.52 m, MAE 0.71 m, T_1 0.10 and T_3 0.08. Hence, our method quantitatively outperforms the reference method compared by a factor of $\sim$2 to 3. Since SAT2LOD2 fits roof tops based on roof type primitives, it does not produce rooftops that are structurally accurate. On the other hand, our method fits a plane to each segmented roof plane polygon, which leads to more accurate, but less mathematically symmetric roof tops. Visually, in Figure 5.10, both our method and SAT2LOD2 show a reconstruction that looks quite similar to the ground truth. Taking a closer look in the second row of Figure 5.9, the impression remains that both results are similar to the ground truth. Even though our

method also outscores SAT2LOD2 on the simple Braunschweig test area, the advantages of PLANES4LOD2 are most significant when studying more complex scenes like the test region of Roof3D.

5.2.5 Discussion

The quality of the LoD-2 resulting from PLANES4LOD2 is affected by multiple factors. If there is high vegetation covering the roof plane, the accuracy of the associated plane parameters might be decreased. One possible way to address this issue would be to use a separate network to remove trees from the DSM [111–113]. Furthermore, we do not enforce symmetry between roof planes for any roof type other than for a single plane. Since we assume the roof type to be gable for buildings with two predicted roof planes, roof tops that have vertical gaps between roof planes will be modeled as if they intersect at the ridge line. Buildings with roof types like hip and half-hip will not be reconstructed in a regularized style, since we do not assure either symmetry or intersection of the roof planes at the identical height at junctions between them. On the other hand, primitive-based approaches like that of Li *et al.* [84] fit roof models that are a-priori symmetrical, but are less flexible than PLANES4LOD2. In practice, one could combine a primitive-based approach for simple roof types with PLANES4LOD2 for the remaining roof structures.

Further restrictions are induced by the GSD. We decided to use 0.3 m. A smaller GSD leads to better visibility of the roof lines and it would be easier to distinguish roof planes. On the other hand, it would cause more noise, since more details are visible, which the network would have to learn. Regarding a larger GSD, it would cause blurrier lines and PLANES4LOD2 is sensitive to the visibility of separation lines.

We also observed that PLANES4LOD2 predicts the instances of roof planes and building sections more accurate than what the metrics suggest. The reason for this is that the common objects in context (COCO)-metrics, including AP, AR and $F1_{INST}$ we are using, are very sensitive. For one ground truth polygon, if the highest overlap with a predicted polygon is 0.4999, it will not be recognized as a true positive, but as false negative. Another effect, that makes metrics underestimate, is possible ambiguous ground truth. Many small roof planes that exist in the ground truth can hardly be seen by bare eyes, or are so small that even a fine-grained neural network cannot detect them as a separate object. Moreover, the COCO-metrics compute the average precision AP and AR not only for the threshold 0.5, but also for much higher thresholds up to 0.95. While this is a reasonable threshold for large buildings or large objects on multi-media imagery, it is hard to achieve a good score in building section or roof plane segmentation. On the other hand, those metrics are commonly used in instance segmentation task and we argue that they are sufficient and realistic to compare different experimental setups.

5.3 SAT2BUILDING: LoD-2 Building Reconstruction from Satellite Imagery using Spatial Embeddings

5.3.1 Contributions

Our first main contribution of this section aims on improved instance segmentation. Since PLANES4LOD2 segments building sections and roof planes based on line features, it is prone to incomplete delineations in the case of occluded or low-contrast object boundaries. On the contrary, Neven *et al.* [114] present a method for instance segmentation that is based on spatial embeddings. Spatial embeddings use 2D direction vectors, which point to the center of an instance. We use this idea to segment building sections and roof planes, but add several skip-connections to the respective decoders and add hierarchical skip-connections [115] to allow flow of information from the building section to the roof plane task. Overall, we leverage spatial-embedding based instance segmentation in the much more challenging realm of remote sensing, where objects are tiny and often occluded.

Our second main contribution of this section is to predict building heights, sharing the encoder with the instance segmentation network. Building heights are a nDSM, but without vegetation. We leverage a regularization loss, which enforces surface normal consistency of the predicted depth with the ground truth.

This section presents SAT2BUILDING, introducing the following contributions:

- Building section and roof plane segmentation based on spatial embeddings.
- Building height estimation with an encoder that is shared with the instance segmentation network.
- Experimental evaluation on test areas in Bonn, Germany and Lyon, France with varying lighting conditions, architectural styles and GSDs.
- Comparative evaluation with three baseline methods for LoD-2 reconstruction.

5.3.2 Methodology

SAT2BUILDING consists of three stages. First, it segments building sections and roof planes by a single U-shape FCN alongside LoD-2 building heights from the concatenation of an orthorectified panchromatic image (PAN) and a patch of a photogrammetric DSM. Second, it vectorizes the segments. Third, it generates an LoD-2 model based on the vectorized segments and the building heights.

5.3.2.1 Architecture

We utilize ResNet50 [35] as the backbone network (yellow layers in Figure 5.11) together with two decoders for the building section task, two decoders for the roof plane task and one decoder for the building height task (blue layers in Figure 5.11). Each of the decoders gradually up-samples the feature map from the last encoder layer, using higher-resolution

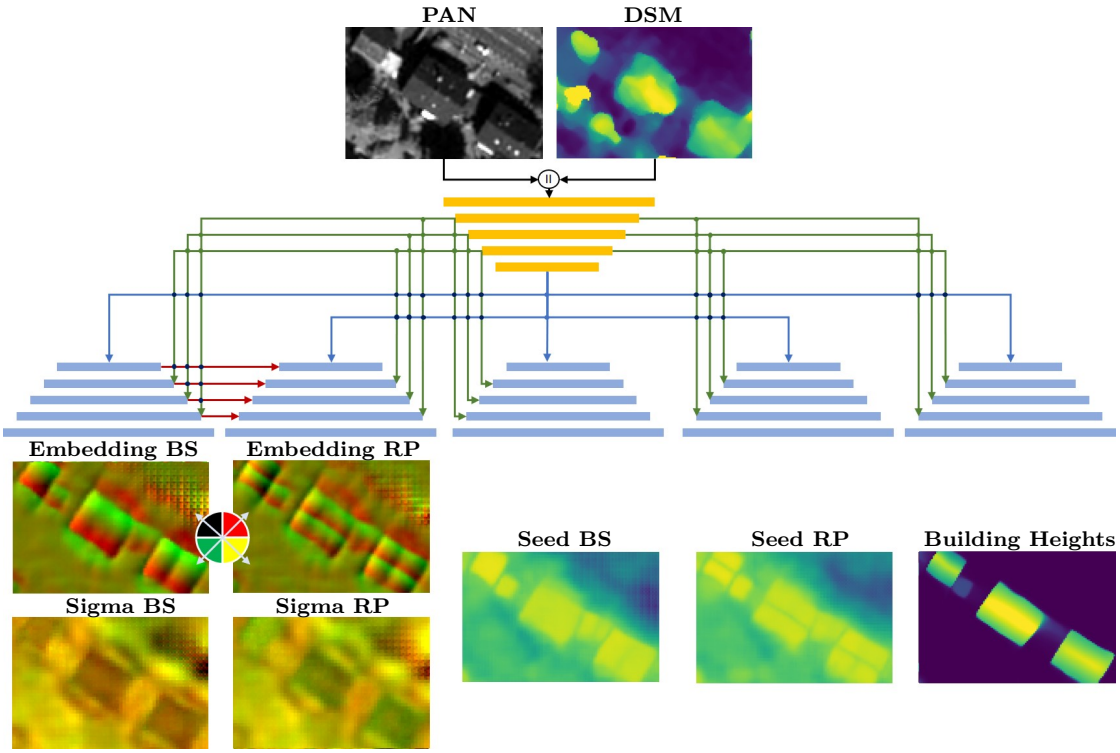


Figure 5.11: The structure of our proposed network architecture. **BS** and **RP** are abbreviations for building section and roof plane. The yellow and blue rectangles indicate encoder and decoder layers. "||" is the concatenation operation, the black arrows show the flow of the input data, blue arrows show the flow of the final encoder feature map to the decoder, green arrows are skip connections from the encoder to the decoders, and red arrows are hierarchical skip connections. The circle diagram is the legend of offset directions in the embedding map. The roof planes in Sigma RP are longer than the building sections in Sigma BS. Hence, the sections have a reddish color, whereas the planes have a green tint.

features maps from the encoder as guidance (indicated by green arrows in Figure 5.11). This helps in combining fine geometrical details with deep semantic features. Further information is passed from the building section decoders to the roof plane decoders (red arrows in Figure 5.11). This helps in introducing knowledge from the coarse building section segmentation to the finer roof plane segmentation, which is similarly done for hierarchical plant segmentation in Roggiolani *et al.* [115]. In all places where multiple feature maps flow to the same skip-connection, we use summation to aggregate them, which is more memory efficient then concatenation.

5.3.2.2 Instance Segmentation

We have two instance segmentation sub-tasks, which are building section and roof plane segmentation. For each of the tasks, our network produces three outputs, similar to Neven *et al.* [114]. We formulate the losses only for a single instance segmentation task, but all losses are computed once for each building section and roof plane.

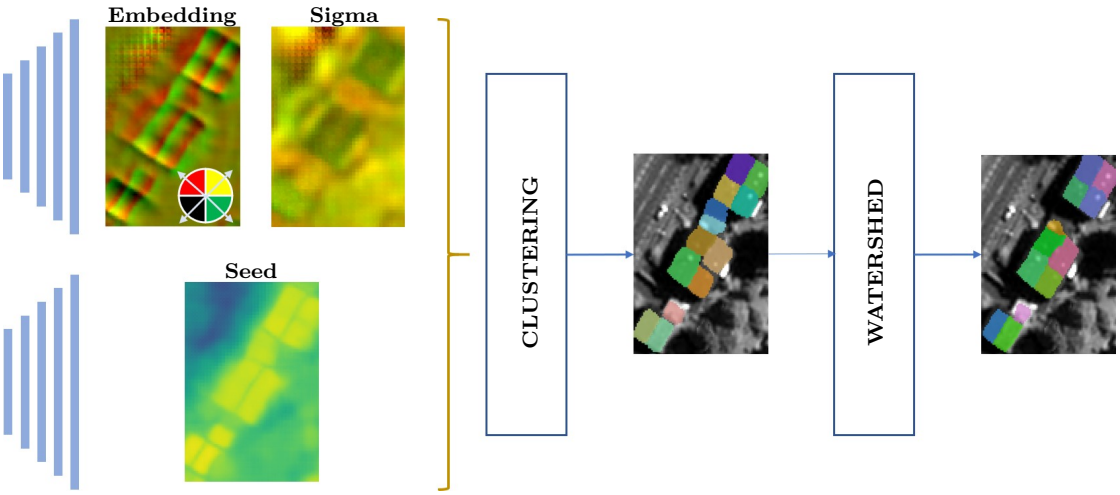


Figure 5.12: The process of spatial embedding-based roof plane segmentation. Two decoders output spatial embeddings, cluster shape parameters sigma and a seed map. These elements are passed to a simple clustering algorithm. Gaps between instances are closed using the watershed transformation.

The first output is the 2D direction vector $o_i \in \mathbb{R}^2$, pointing to the center of an instance $S_k \in \{S_1, S_2, ..., S_K\}$. A vanilla loss function to guide the training of o_i would be

$$\mathcal{L}_{mse} = \sum_{i=1}^{n} \|o_i - \hat{o}_i\|^2, \tag{5.12}$$

where $\hat{o}_i = C_k - x_i$ for $x_i \in S_k$. $C_k = \frac{1}{n}\sum_{x \in S_k} x$ is the centroid of all pixels x belonging to instance S_k.

However, during inference, we need to determine the cluster centers $C = \{C_1, C_2, ..., C_K\}$ and assign all pixels x to one of the cluster centers. A common method to do that is density-based clustering, where the density of the embeddings $e_i = x_i + o_i$ is computed and local maxima are selected as cluster centers. Then, pixels are assigned to the instance with the shortest distance from their corresponding embedding to the center of the instance. But this includes post-processing, which we want to avoid.

Hence, we use the Lovasz-Hinge loss [116] $\mathcal{L}_{lh}$, which is a continuous and differentiable extension of the hinge loss

$$\mathcal{L}_{hinge} = \sum_{k=1}^{K} \sum_{e_i \in S_k} max(\|e_i - C_k\| - \delta, 0), \tag{5.13}$$

where δ controls the cluster size. Instead of using a fixed cluster size, we replace $\|e_i - C_k\| - \delta$ by $2 \times \phi_k(e_i) - 1$, where

$$\phi_k(e_i) = exp(-t_{kx} \times (e_{ix} - C_{kx})^2 - t_{ky} \times (e_{iy} - C_{ky})^2), \tag{5.14}$$

and $t_k = exp(10 \times \sigma_k)$. Now, the cluster size is parameterized by $\sigma_k = \frac{1}{|S_k|}\sum_{\sigma_i \in S_k} \sigma_i$

and σ_i is the second output of our network for instance segmentation. Each σ_i has two components, σ_{ix} for the horizontal and σ_{iy} for the vertical direction. This makes it easier to learn non-square instances. To enforce smoothness of the σ_i, we use

$$\mathcal{L}_{smooth} = \frac{1}{|S_k|} \sum_{\sigma_i \in S_k} \|\sigma_i - \sigma_k\|^2. \tag{5.15}$$

During inference, we can get a hint on where pixels belonging to instances are located by using the seed s_i of pixel i, which is the third output of our network. The seed score indicates how close a pixel is to the center of any instance. This distance is equivalent to ϕ_k and should be zero in the background bg. Hence, we use the loss function

$$\mathcal{L}_{seed} = \frac{1}{N} \sum_{i=1}^{N} 1_{\{x_i \in S_k\}} \|s_i - \phi_k(e_i)\|^2 + 1_{\{x_i \in bg\}} \|s_i - 0\|^2, \tag{5.16}$$

where the gradient of $\mathcal{L}_{seed}$ is only computed with respect to s_i. The indicator functions $1_{\{x_i \in S_k\}}$ and $1_{\{x_i \in bg\}}$ constitute a mask of pixels belonging to any instance S_k or to the background bg. In the term $\|s_i - \phi_k(e_i)\|^2$, k denotes the number of the cluster S_k that pixel x_i belongs to according to $1_{\{i \in S_k\}}$. The final loss function for both instance segmentation tasks is

$$\mathcal{L}_{inst} = \mathcal{L}_{lh} + \mathcal{L}_{smooth} + \mathcal{L}_{seed}. \tag{5.17}$$

To obtain instances during inference time (see Figure 5.12), we sequentially select the pixels with the highest seed values s_k as the cluster centers C_k. We furthermore select σ_k at those pixels as the sigma value. Then, we assign all pixels i to cluster S_k if

$$e_i \in S_k \iff \phi_k(e_i) < 0.35. \tag{5.18}$$

Note that the value of $\phi_k(e_i)$ is specific to the center C_k of cluster S_k. The threshold in the implementation of Neven *et al.* [114] was 0.5, but we find that the smaller threshold 0.35 leads to more complete instances in our task. After the assignment of each cluster, we mask out the pixels assigned to cluster S_k and proceed with the new highest seed score. If a cluster contains less then 12 pixels, we discard it as noise. This limit lays far below the 128 pixels used in the original implementation, which is too high for small instances like roof planes in satellite imagery. The clustering process proceeds until there are less than 128 pixels which are not yet clustered. This clustering procedure leaves gaps between adjoining instances open. To solve this issue, we apply a variant of the watershed transformation [3], that makes each instance grow inside the limits of the binary building mask, obtained by thresholding the predicted building heights at 2 m, until it meets another instance.

5.3.2.3 Building Height Estimation

To obtain an LoD-2 model, the building height is required. We define the predicted building height as function $f(p) \in \mathbb{R}$ at some pixel $p \in P$, where P is the set of all

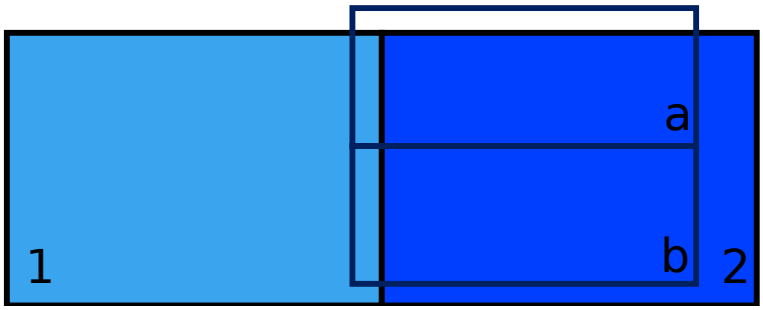


Figure 5.13: Visualization of the assignment of building section ids (numbers) to individual roof planes (letters). Since building section "2" has the higher intersection over union (IoU) with both roof planes "a" and "b", the ID "2" will be assigned to them.

pixels in an image. The function f is implemented by the shared encoder (yellow, top in Figure 5.11) and a decoder (right side in Figure 5.11). The decoder consists of stacked feature fusion modules (FFMs) [117], which has a high-resolution feature map from the encoder and a low-resolution feature map from the previous decoder layer or the bottleneck as inputs. Bilinear up-sampling brings the low-resolution feature map to the same resolution as the high-resolution feature map. In parallel, the high-resolution feature map is passed to a residual block. Consecutively, the sum of the output of the residual block and the up-sampled feature map is passed to another residual block. The ground truth height $\hat{f}(p)$ serves as the learning target in the mean squared error loss

$$\mathcal{L}_{mse} = \frac{1}{|P|} \sum_{p \in P} \|f(p) - \hat{f}(p)\|^2. \tag{5.19}$$

To improve regularity of the predicted height, we enforce it to have similar normals like the ground truth by utilizing the loss

$$\mathcal{L}_{normal} = \frac{1}{|P|} \sum_{p \in P} \|\nabla_{x,y} f(p) - \nabla_{x,y} \hat{f}(p)\|^2. \tag{5.20}$$

The final loss for depth estimation is

$$\mathcal{L}_{depth} = \mathcal{L}_{mse} + \mathcal{L}_{normal}. \tag{5.21}$$

5.3.2.4 Vectorization

To reconstruct an LoD-2 model, it is necessary to obtain vectorized roof structure information. We accomplish this by extracting all border pixels of building sections and roof planes. The border pixels are considered vertices and are connected by starting at an initial pixel. From there on, a search finds the closest neighbors among the vertices iteratively along both paths until a cycle exists. We then refine by utilizing the douglas peucker polygon simplification algorithm [105]. Hence, we remove pixels that,

if excluded, lead to an error of less than 1.0 m with respect to the initial polygon. In CityGML, the representation of an LoD-2 building is a collection of 3D roof planes with a single building ID. We assign building ids to roof planes by selecting the ID of the building section with the highest IoU with that roof plane. This approach is outlined in Figure 5.13.

5.3.2.5 LoD-2 Reconstruction

We add 3D information to the 2D polygons by sampling elevations inside roof plane polygons from the predicted building heights. We then leverage RANSAC [107] to estimate the parameters of a plane for each roof plane. RANSAC is particularly useful in this case because it is insensitive to outliers. RANSAC selects a random subset of the given samples and fits a model to it. Then, RANSAC checks whether the selected samples conform with the model. Those samples that conform are considered inliers. The higher the number of inliers, the higher the quality of the model is considered. The steps from random subset selection to inlier detection are repeated multiple times and the model with the most inliers is select as the final plane. We sample the elevation of the final plane at the vertices of the corresponding roof plane to obtain height values. The above vectorization and LoD-2 reconstruction procedure was originally presented in our previous work [11].

5.3.3 Experiments

5.3.3.1 Data

For training and validation, we use a World View-1 panchromatic image and photogrammetric DSM of Berlin, Germany of size 30733 × 45999 pixels (see Figure 5.14). We split the image into five vertical stripes of equal size and use the middle one for validation and the remaining four for training. As the ground truth for building sections, roof planes and building height we use public data provided by the senate of Berlin [3]. In the study area, it contains overall 479,626 building sections with 729,524 roof planes. Some samples of the test area are visualized in Figure 5.15.

We use two separate datasets for evaluation, one from Bonn, Germany of size 1023 × 896 pixels from Pleíades and the other from Lyon, France of size 1387 × 994 pixels from World View-1. For metric computation, we use public ground truth of both Bonn [4] and Lyon [5] in vector format. The ground truth of Bonn contains 508 building sections with 1,141 roof planes, and that of Lyon 778 building sections with 2,575 roof planes. In Figure 5.16, samples from Bonn and Lyon are visualized. In Lyon, a building section contains on average more than 3 roof planes, which is much higher than the ratio of ∼ 2 in Bonn and ∼ 1.5 in Berlin. Looking at the bottom row of Figure 5.16, it becomes clear that the ground truth in Lyon contains more details than the other areas.

[3] https://daten.berlin.de/tags/geodaten
[4] https://www.opengeodata.nrw.de/produkte/geobasis
[5] https://data.grandlyon.com/

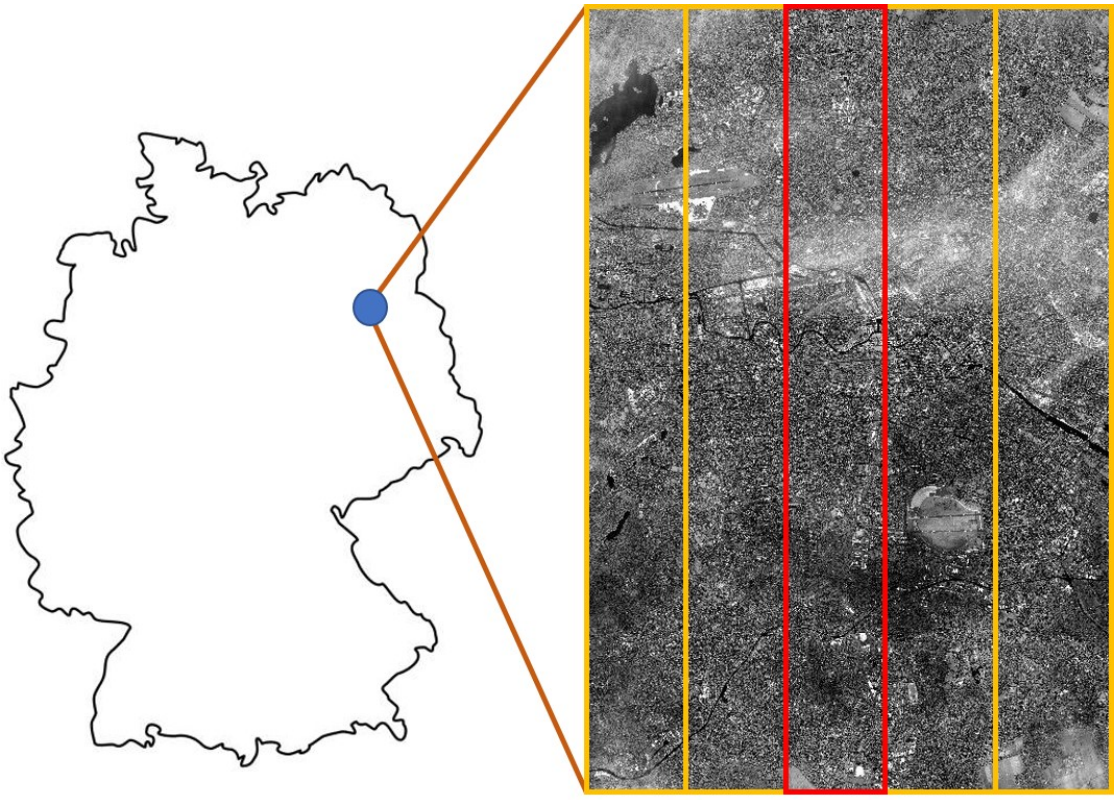


Figure 5.14: Visualization of the whole Berlin dataset and how it is split into training and validation areas. The red vertical box contains the validation data, the orange boxes contain the training data.

During training, we crop patches of size 256 × 256 pixels without overlap. We do random window shifting of up to 256 pixels in horizontal and vertical direction to increase the data diversity during training. In the validation phase, patches of size 256 × 256 pixels are cropped without overlap. While testing, the crop size is also 256 × 256 pixels and the overlap is 128 pixels horizontally and vertically. The network predicts per-patch and a large map is created by averaging the patches at the overlapping areas.

The data from Berlin and Lyon has GSD 0.5 m, whereas that of Bonn has GSD 0.7 m. Hence, we up-sample the data from Bonn to GSD 0.5 m. We generate all ground truth in GSD 0.5 m. During evaluation, we use two kinds of ground truth, which is raster ground truth of building heights and vector ground truth of roof planes.

5.3.3.2 Training Details

We use random initialization of the network parameters and train them using Adam optimizer [32] with learning rate 0.0002, momenta 0.5 and 0.999. The model is trained for 300 epochs and the learning rate is multiplied by 0.1 after the 100th and 200th epoch. We use batch size 8 and combine the loss functions from Equations (5.17) and (5.21) to

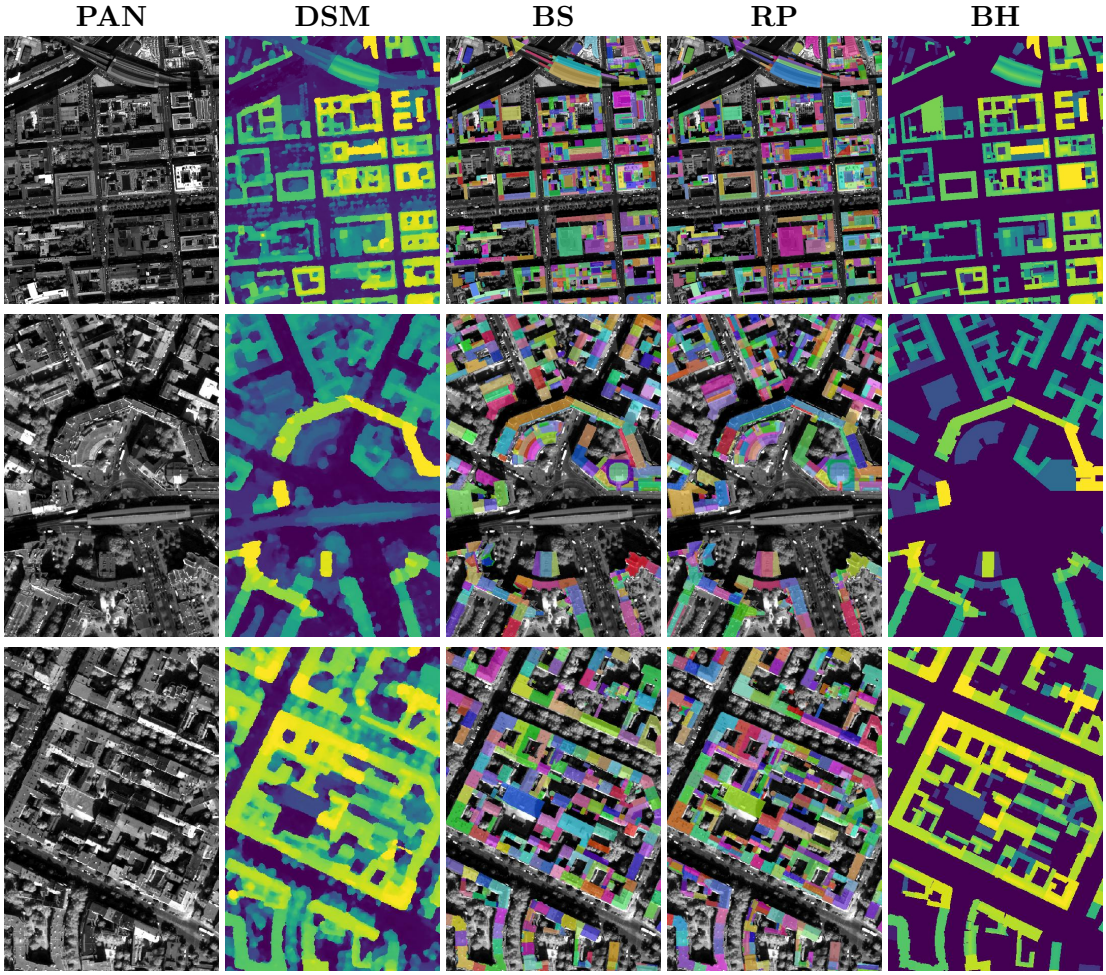


Figure 5.15: Parts from our training data in Berlin. **PAN** abbreviates panchromatic image, **BS** building sections, **RP** roof planes and **BH** building heights.

the final multi-task loss

$$\mathcal{L}_{total} = \mathcal{L}_{inst,bs} + \mathcal{L}_{inst,rp} + \mathcal{L}_{depth}. \tag{5.22}$$

5.3.3.3 Evaluation Metrics

To quantify the performance of the trained models, we evaluate both the vectorized roof planes in 2D and the rasterized predicted LoD-2 model in 3D. One commonly used metric for segmentation is the IoU. Since it doesn't take into account individual instances, it is not suitable for roof plane segmentation. On the other hand, the COCO metrics are too challenging for tiny objects like roof planes in satellite imagery. Hence, we provide the new metric

$$IoU^{gt}_{inst} = \frac{1}{|\hat{P}|} \sum_{\hat{p} \in \hat{P}} \max_{p \in P} IoU(p, \hat{p}), \tag{5.23}$$

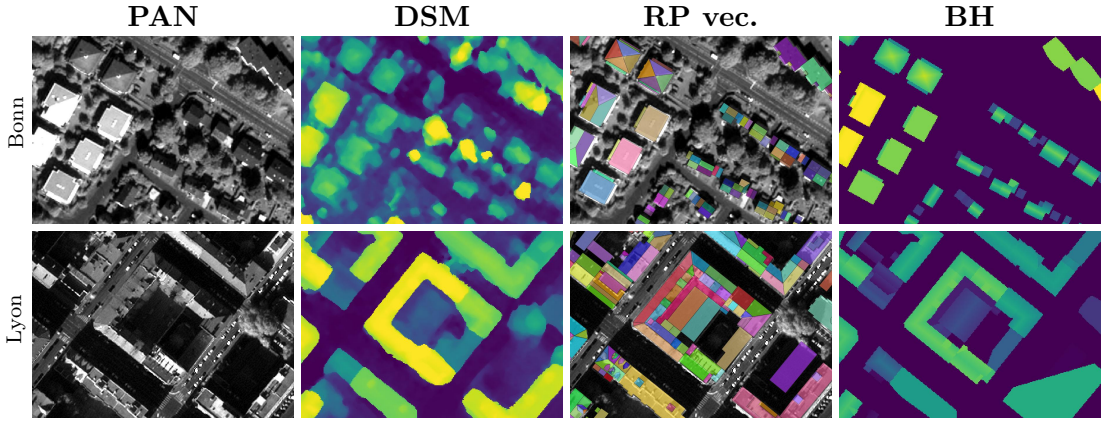


Figure 5.16: Parts from our test data in Bonn (top row) and Lyon (bottom row). **RP vec.** abbreviates vectorized roof planes.

where $p \in P$ is a predicted polygon and $\hat{p} \in \hat{P}$ is a ground truth polygon. In IoU^{gt}_{inst}, we iterate over the ground truth polygons and select the respective predicted polygon with the highest IoU. Then, these IoUs are averaged.

For the evaluation of the 3D models, we rasterize their height values and use the RMSE

$$\text{RMSE} = \sqrt{\frac{\sum_i |\hat{h}_i - h_i|^2}{N}}, \tag{5.24}$$

where i is a specific pixel, N is the number of pixels, $\hat{h}_i$ is the ground truth height at pixel i and h_i is the predicted height at pixel i. Furthermore, we use MAE

$$\text{MAE} = \frac{\sum_i |\hat{h}_i - h_i|}{N}. \tag{5.25}$$

for evaluation. Note that the MAE is less sensitive to outliers than the RMSE. To gain more insight into the obtained results, we also carry out qualitative inspection on both the roof plane polygons and the rasterized LoD-2 model.

5.3.3.4 Experiments

Several experiments are done to show the superiority of SAT2BUILDING. As the reference method, we use PLANES4LOD2 [11], which was originally trained for LoD-2 reconstruction of aerial imagery. To improve comparability, we re-train PLANES4LOD2 on the same satellite data of Berlin that we use for SAT2BUILDING. For PLANES4LOD2, we use an external DTM together with the DSM to derive building heights instead of predicting them. The next experiment is the method of Gui *et al.* [17] (SAT2LOD2-LineSep). SAT2LOD2-LineSep requires normalized building height information and building sections. We use the building height from our proposed SAT2BUILDING method and the building sections from PLANES4LOD2, which are obtained based on the prediction of separation lines between sections. Moreover, we feed the building

Table 5.4: Quantitative results of comparison between a single shared, and two separate encoders for two test areas. ↑ indicates that higher values are superior, ↓ indicates that lower values correspond to higher accuracy.

Test Area	Shared Encoder	IoU^{gt}_{inst} ↑	MAE ↓	RMSE ↓
Bonn		0.300	0.56 m	1.92 m
Bonn	X	**0.323**	**0.53** m	**1.83** m
Lyon		0.199	1.81 m	5.22 m
Lyon	X	**0.205**	**1.74** m	**4.98** m

height and sections from SAT2BUILDING to the method of Gui *et al.* [17] and call that experiment SAT2LOD2-Embed, because the input building sections are obtained using spatial embedding-based instance segmentation. We compare the above approaches to SAT2BUILDING.

To showcase the effectiveness of using a shared encoder for both instance segmentation and height estimation, we compare that to a setting with two separate networks without inter-connection.

5.3.4 Results

5.3.4.1 Quantitative Results

In this subsection, we analyse the quantitative results of the experimental study. In Table 5.4 we can see that it is advantageous to use a shared encoder instead of two separate networks. The unified network performs better in both instance segmentation and LoD-2 model geometrical accuracy on both test areas, showing that one encoder can effectively learn features that are more useful for both tasks as compared to two separate encoders. We attribute this advantage to the regularizing effect of the multi-task setting. Since both models are trained on data from Berlin that has different lighting conditions and architectural styles, the results highlight their capability to generalize to unseen data.

In Table 5.5, we observe the quantitative results of SAT2BUILDING with a shared encoder in comparison to three baseline methods. The comparison between SAT2LOD2-SepLine and SAT2LOD2-Embed shows that instance segmentation based on spatial embeddings is either an equally as good (Bonn) or a better (Lyon) basis for the LoD-2 reconstruction using the SAT2LOD2 method. The advantage of spatial embeddings over separation lines becomes particularly clear in Lyon, where the buildings are densely built. A higher density of buildings and a larger quantity of buildings with joint borders makes it more critical to discern building sections. Furthermore, separating buildings based on a thin line is very challenging in satellite imagery, as compared to aerial imagery with smaller GSDs below 0.3 m.

The comparison between SAT2LOD2-SepLine and PLANES4LOD2 shows that the LoD-2 reconstruction becomes geometrically more accurate if it focuses on reconstructing

Table 5.5: Comparative results on Bonn and Lyon. ↑ indicates that higher values are superior, ↓ indicates that lower values correspond to higher accuracy.

Test Area	NAME	IoU^{gt}_{inst} ↑	MAE ↓	RMSE ↓
Bonn	SAT2LOD2-SepLine	-	0.89 m	2.66 m
Bonn	SAT2LOD2-Embed	-	0.88 m	2.67 m
Bonn	PLANES4LOD2	0.1826	0.64 m	2.14 m
Bonn	SAT2BUILDING	**0.323**	**0.53** m	**1.83** m
Lyon	SAT2LOD2-SepLine	-	4.00 m	8.60 m
Lyon	SAT2LOD2-Embed	-	3.16 m	7.50 m
Lyon	PLANES4LOD2	0.162	2.66 m	6.35 m
Lyon	SAT2BUILDING	**0.205**	**1.74** m	**4.98** m

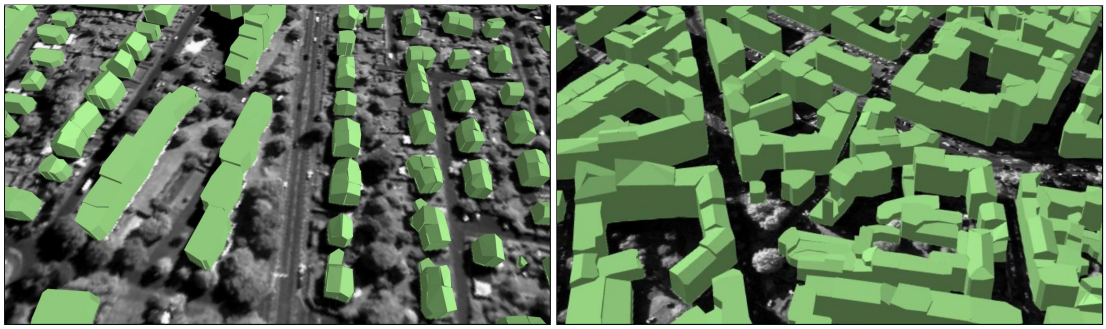
roofs based on individual roof planes instead of primitives. PLANES4LOD2 outperforms SAT2LOD2-SepLine with a large margin, remarkably in Lyon. The complex building roofs in Lyon make it even more important to accurately reconstruct each single roof plane.

Comparing PLANES4LOD2 with SAT2BUILDING puts the spotlight on the effect of using spatial embeddings as opposed to separation lines for the instance segmentation of roof planes. SAT2BUILDING clearly outperforms PLANES4LOD2 on all metrics and both test areas. The difference in IoU^{gt}_{inst} is bigger in Bonn than in Lyon. This can be explained by the GSDs of both test areas. The GSD in Lyon is lower than the one in Bonn, which is 0.7 m. A higher GSD makes it harder to segment the thin line that separates adjoining roof planes instances. This lead to poorer instance segmentation performance. On the other hand, instance segmentation based on spatial embeddings groups pixels to instances based on the center of the instance, which is easier to detect on lower-resolution satellite imagery than the separation line, which favors SAT2BUILDING over PLANES4LOD2.

Another advantage of SAT2BUILDING is its independence from external DTM information. The strong performance of SAT2BUILDING is achieved using its own predicted building height map, whereas PLANES4LOD2 includes external terrain information to obtain heights above ground. The improved values of MAE and RMSE indicate that using the predicted building heights has no negative effect on the performance as compared to using an external DTM.

5.3.4.2 Qualitative Results

In Figure 5.17, it can be seen that SAT2BUILDING generates geometrically accurate building models at LoD-2. Even under highly challenging conditions like large shadows, high GSDs and complex building structures, SAT2BUILDING correctly identifies individual roof planes. Nevertheless, in some places, roof planes are misaligned. The LoD-2 reconstruction pipeline can sometimes estimate incorrect plane parameters, if RANSAC randomly selects an inadequate subset of points on the plane.



Bonn **Lyon**

Figure 5.17: A 3D visualization of the results on our two test sites in Bonn and Lyon.

As we inspect Figure 5.18, we see that SAT2BUILDING reconstructs roof more similar to the ground truth than all other tested methods in Bonn. In the middle of the upper sample, the positive effect of the improved instance segmentation leads to correctly generated gable roofs. PLANES4LOD2 incorrectly generates flat roofs, which is caused by missing separation lines. SAT2LOD2-Embed and SAT2LOD2-SepLine produce regularized buildings, but they often do not accurately reflect the structure of the roof. SAT2LOD2 reconstructs building roofs based on fixed roof templates, which are often incorrectly inferred in case of complex buildings. Furthermore, SAT2LOD2 does not keep seamless neighboring relations while refining the boundaries of building sections, which leads to incorrect gaps between them. On the other hand, SAT2BUILDING and PLANES4LOD2 do not have such a gap because they refine outlines of adjoining building sections and roof planes jointly. In the lower example of Figure 5.18, SAT2BUILDING is the only method that gets the pyramid shape of the four squared building in the middle correct. Since we use training data from a public source, which contains many inconsistencies, this can sometimes cause incorrect predictions.

In Figure 5.19 we can see that SAT2BUILDING is more accurate than the other three methods even in highly complex scenarios like the city of Lyon. For example, the hipped roof in the middle of the scene is only accurately reconstructed by SAT2BUILDING. Furthermore, SAT2LOD2 generates even more incorrect gaps than in Figure 5.18.

5.3.5 Discussion

Although SAT2BUILDING outperforms the other methods in quantitative evaluation, the metric IoU^{gt}_{inst} does not exceed 0.323 in Bonn and 0.205 in Lyon. Particularly the value for Lyon is very low. We can observe in the third column bottom row of Figure 5.16, that the roof plane ground truth in Lyon contains many details, that are very hard to recognize in the panchromatic image (first column) and impossible to detect in the DSM (second column). On the other hand, SAT2BUILDING extracts roof plane polygons, representing larger planar structures of building sections. But the IoU^{gt}_{inst} metric averages IoU scores across ground truth instances, leading to a low score.

Moreover, Table 5.5 shows much higher MAE and RMSE in Lyon than in Bonn. The

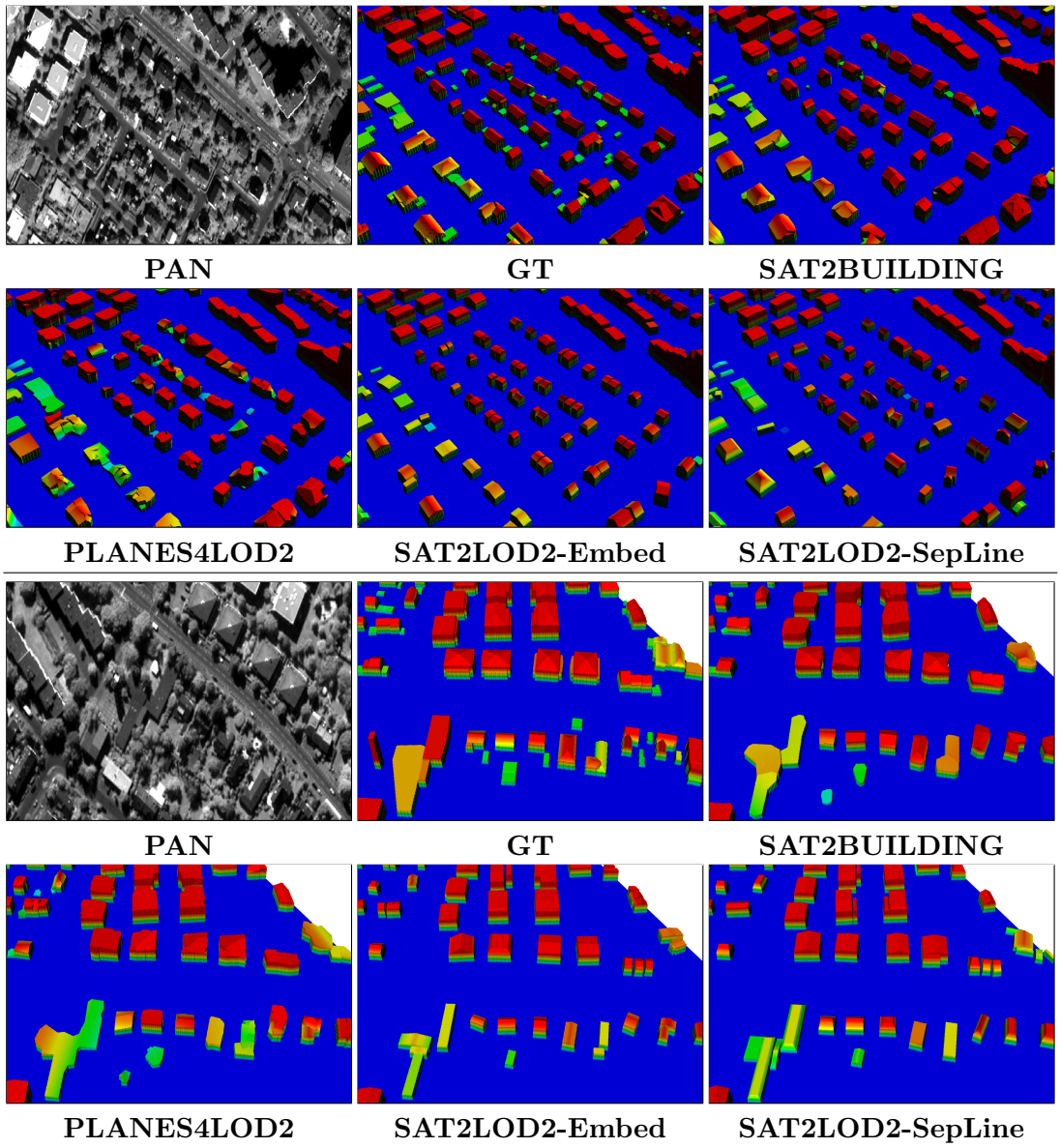


Figure 5.18: Visual results of the comparative study from Bonn.

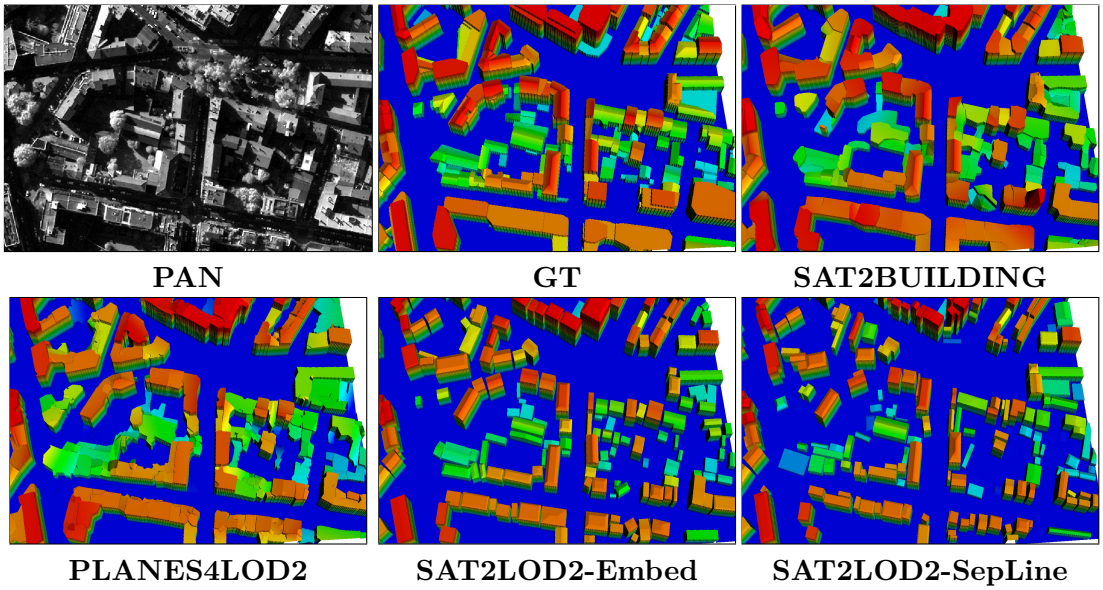


Figure 5.19: Visual results of the comparative study from Lyon.

evaluation of the generated 3D models was done based on rasterized height maps. Since our test area in Bonn is more sparse, it leads to a lot of background pixels, which causes better metrics. In Lyon on the other hand, a high building density, with buildings at various heights, and complex building structures have more room for error and cause worse 3D metrics.

5.4 Summary

We presented PLANES4LOD2, a method that uses planar roof components to reconstruct buildings as level of detail (LoD)-2 models. The PLANES4LOD2 pipeline relies on deep learning as well as conventional approaches to implement a full 3D reconstruction pipeline from an RGB image, a photogrammetric digital surface model (DSM) and a digital terrain model (DTM). The method makes use of the DSM in the novel depth attention module (DAM) to enhance building plane prediction and in the roof surface reconstruction. PLANES4LOD2 robustly interpolates roof surfaces from sampled height values and initial roof planes. The resulting LoD-2 building model appears visually similar, or close to be identical to the ground truth, even when the test region contains very complex building structures and is densely built. Furthermore, we demonstrated the advantages of our method for LoD-2 reconstruction compared to other software. We also evaluated PLANES4LOD2 on a test region in a different city. The results reveal superior generalization capability of our method being adaptive to lighting conditions and architectural styles different from the ones the model is trained.

Furthermore, we presented SAT2BUILDING, a novel method for LoD-2 reconstruction based on the segmentation of main planar roof components. SAT2BUILDING utilizes

deep learning and conventional methods to build a complete 3D reconstruction workflow, only based on panchromatic satellite imagery and photogrammetric DSM. Our method predicts normalized building heights, which makes it independent from external terrain information. SAT2BUILDING leverages spatial embeddings for robust roof plane segmentation. The resulting LoD-2 model is geometrically accurate, even when facing difficulties such as high ground sampling distances (GSDs) of 0.5 m to 0.7 m, large shadows, densely built areas, and complex roof structures. We showed how SAT2BUILDING improves in comparison to existing methods, obtaining considerable performance gains. Furthermore, SAT2BUILDING generalizes well when evaluated on cities with different lighting conditions, architectural styles, and GSDs then are contained in the training area.

6 Conclusion

6.1 Summary

Nowadays, remote sensing sensors are big providers of data that can help gain understanding about objects on the surface of the earth. Among others, optical sensors provide detailed structural and spectral information. With the help of deep learning, image data can be used to detect buildings and reconstruct them. Yet, there is a lack in methods that can do so with enough detail, i.e. detect parts and components of buildings. Furthermore, there is a lack in methods that can accurately reconstruct buildings in level of detail (LoD)-2. Another issue is the absence of methods that effectively and efficiently make use of digital surface models (DSMs) to separate buildings from other objects on the ground, such as roads. In conclusion, this thesis presents a comprehensive exploration into automated building information extraction and reconstruction techniques. Through a detailed analysis, novel methods were developed. In the fields of building section instance segmentation and LoD-2 reconstruction, two objectives are targeted.

The first part of the study proposes a solution for building section instance segmentation. It can separate building section in complex scenarios, which was showcased for various scenarios with different complexity of the buildings, geographical location, different lighting conditions, and architectural styles. Moreover, based on a case study on data from Medellín, Columbia, we showed how this method can support humanitarian aid, crisis management, and disaster response by providing detailed information about the building inventory in informal settlement.

The method does not produce any overlaps and does not leave gaps between directly neighboring building sections, which sets it apart from other AI-based instance segmentation methods such as Mask-RCNN. It achieves that by segmenting the separation line between directly neighboring sections as a separate class and then utilizing the watershed transform. The method uses a SkipFuse-UNet-3+, which has a strong, hierarchical interconnection between encoder and decoder, leading to well-segmented buildings and separation line, even under the presence of shadows. Furthermore, DSM information is injected by using a separate branch in the network architecture. This additional information makes it easier for the network to separate buildings from roads and highlight separation lines between sections if the neighboring buildings have a height difference in between them. The additional topological loss term lead to more complete separation lines and more regular building section appearance. This contribution is even helpful for the next step, which is vectorization of building sections, because the resulting polygons will be simpler. The method simply extracts initial boundary polygons for each section and then simplifies them using Douglas-Peucker. Hence, the resulting section

polygons are useful for applications that require memory-efficient and easily editable objects. Then, those polygons are enriched with the average height from the corresponding area of the input DSM, leading to LoD-1 models that are suitable for flow simulations, 3D mapping, an many more.

Despite the importance of building outlines and LoD-1 models, they lack detailed roof information that is pertinent for more accurate simulations and realistic graphical representations of cities. Hence, we propose two methods that extend our approach for 2D instance segmentation to LoD-2 reconstruction. The first method, PLANES4LOD2, utilizes a deep learning model, that extracts robust features from imagery and DSM data. This allows the method to extract building features in simple, rural scenarios like our test area in Braunschweig, Germany and in more complex, urban scenarios like in our test area in Cologne, Germany. Those features are building segment, building section separation line and roof plane separation line. They allow the gap-less and overlap-free segmentation of directly neighboring instances, which leads to LoD-2 models that share these properties if looked on from the top-view in the later part of the method. PLANES4LOD2 produces geometrically accurate, topologically consistent and semantically correct LoD-2 models in vector shape. It uses random sample consensus (RANSAC) to project roof planes polygons into 3D space. RANSAC is robust to outliers, which avoids the collapse of height values along the boundary of the building due to small misalignments between predicted roof planes and the building edge in the DSM. The second method we investigate for LoD-2 reconstruction is called SAT2BUILDING. It successfully predicts roof planes by using spatial embeddings, which focuses on the center of each roof plane to generate instances. Spatial embeddings are more robust in satellite imagery, where high ground sampling distance (GSD) of over 0.5 m and atmospheric distortion make it hard to recognize the separation line for both the eye and the neural network. Another important contribution of SAT2BUILDING is the prediction of the building height in the same deep learning model as the spatial embeddings. It overcomes the challenge of normalizing height values by extracting a digital terrain model (DTM), which is no longer necessary for LoD-2 reconstruction.

In summary, this thesis contributes significant advancements in the field of automated building information extraction and reconstruction, providing not only valuable insights but also innovative methodologies that are applicable across a wide range of domains. These advancements have the potential to impact areas such as disaster management, urban planning, environmental monitoring, and more, where precise and efficient building information is critical. By addressing complex challenges in this domain, this work lays a foundation for future research and practical applications in various interdisciplinary fields.

6.2 Future Work

Future work contains end-to-end methods for both building section instance segmentation and LoD-2 reconstruction. Even though the methods proposed in this thesis are robust, they still contain learning-free steps that are incapable of adapting to training

data. They learn how to extract image features that are suitable for semantic segmentation. Some post-processing is applied to obtain instances, and 3D reconstruction is based on the assumption that building models have planar components. To achieve end-to-end 3D reconstruction, a method that predicts multi-polygons with 3D coordinates is required. Then, the method could learn the criteria for building models like planarity and symmetry from the data or a specific loss could enforce them. Moreover, there is a lack of accurately labeled training data and satellite data at global scale. Future methods should be trained on a larger variety of satellite imagery and accurately labeled building models. Combining a large synthetic dataset can provide perfectly accurate labels in a pre-training step. In a finetuning process, a small set of various, accurately labeled, real imagery can be used to adapt a model to the target domain. Another future direction of work would be to target higher LoDs. For instance, modelling windows, chimneys, and doors from street view imagery can enrich 3D models of buildings. Even though there is existing work for these building components, they are yet to be integrated with the LoD-2 information in a single end-to-end trainable workflow.

Acronyms

AFM attraction field map.

AP average precision.

AR average recall.

ASIP approximating shapes in images with polygons.

BM building mask.

CityGML city geography markup language.

CNN convolutional neural network.

COCO common objects in context.

DAM depth attention module.

DSM digital surface model.

DTM digital terrain model.

FCN fully convolutional neural network.

FFM feature fusion module.

FN false negative.

FP false positive.

FSA full scale aggregated skip-connections.

GAN generative adversarial network.

GCP ground control point.

GIS geoinformation system.

GNN graph neural network.

GPS global positioning system.

GSD ground sampling distance.

GT ground truth.

IoU intersection over union.

KIBS keypoint inference by segmentation.

LiDAR light detection and ranging.

LoD level of detail.

LSM least squares matching.

LSTM long short term memory.

MAE mean absolute error.

mAP mean average precision.

mAR mean average recall.

MDL minimum description length.

nDSM normalized digital surface model.

NDVI normalized difference vegetation index.

OSM open street map.

POL primary orientation learning.

RANSAC random sample consensus.

ReLU rectified linear unit.

RMSE root-mean-squared error.

RNN recurrent neural network.

RoI regions of interest.

RPC rational polynomial coefficients.

RPN region proposal network.

SAR synthetic apperture radar.

SGM semi-global matching.

SOTA state-of-the-art.

TB touching borders.

TN true negative.

TP true positive.

UAV unmanned aerial vehicle.

VHR very high resolution.

List of Figures

List of Tables

A Related Publications

A.1 Journals

- [8]: ***P. Schuegraf***, *S. Zorzi, F. Fraundorfer, and K. Bittner, "Deep learning for the automatic division of building constructions into sections on remote sensing images," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 16, pp. 7186–7200, 2023*

- [11]: ***P. Schuegraf***, *S. Shan, and K. Bittner, "Planes4lod2: Reconstruction of LoD-2 Building Models using a Depth Attention-Based Fully Convolutional Neural Network," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 211, pp. 425-437, 2024*

- [12]: ***P. Schuegraf***, *S. Gui, R. Qin, F. Fraundorfer, and K. Bittner, "Sat2-building: Lod-2 Building Reconstruction from Satellite Imagery using Spatial Embeddings," Submitted to ISPRS Journal of Photogrammetry and Remote Sensing, 2024.*

A.2 Conferences

- [2]: ***P. Schuegraf***, *J. Schnell, C. Henry, and K. Bittner, "Building section instance segmentation with combined classical and deep learning methods," vol. V-2-2022, 2022, pp. 407–414*

- [16]: ***P. Schuegraf***, *M. Fuentes Reyes, Y. Xu, and K. Bittner, "Roof3d: A real and synthetic data collection for individual building roof plane and building sections detection," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. X1W1, pp. 971–979, 2023*

- [9]: ***P. Schuegraf***, *D. Stiller, J. Tian, T. Stark, M. Wurm, H. Taubenböck, K. Bittner, "Ai-based building instance segmentation in formal and informal settlements," IEEE International Geoscience and Remote Sensing Symposium, 2024.*

- [10]: ***P. Schuegraf***, *Z. Li, J. Tian, J. Shan, and K. Bittner, "Rectilinear building footprint regularization using deep learning," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2024*

- [17]: *S. Gui,* ***P. Schuegraf***, *K. Bittner, and R. Qin, "Unit-level lod2 building reconstruction from satellite-derived digital surface model and orthophoto," Ac-*

cepted for publication at ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2024

Bibliography

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE International Conference on Computer Vision*, pp. 2961–2969, 2017.

[2] P. Schuegraf, J. Schnell, C. Henry, and K. Bittner, "Building section instance segmentation with combined classical and deep learning methods," vol. V-2-2022, 2022, pp. 407–414.

[3] S. Beucher and F. Meyer, "The morphological approach to segmentation: The watershed transformation," *Mathematical Morphology in Image Processing*, 2018.

[4] T. H. Kolbe, G. Gröger, and L. Plümer, "Citygml: Interoperable access to 3d city models," in 2005, pp. 883–899.

[5] F. Nex and F. Remondino, "Automatic roof outlines reconstruction from photogrammetric dsm," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. I-3, pp. 257–262, 2012.

[6] H. Arefi and P. Reinartz, "Building reconstruction using dsm and orthorectified images," *Remote Sensing*, vol. 5, pp. 1681–1703, 2013.

[7] K. Bittner, S. Zorzi, T. Krauß, and P. d'Angelo, "Dsm2dtm: An end-to-end deep learning approach for digital terrain model generation," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-1/W1-2023, pp. 925–933, 2023.

[8] P. Schuegraf, S. Zorzi, F. Fraundorfer, and K. Bittner, "Deep learning for the automatic division of building constructions into sections on remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 7186–7200, 2023.

[9] P. Schuegraf *et al.*, "Ai-based building instance segmentation in formal and informal settlements," *IEEE International Geoscience and Remote Sensing Symposium*, 2024.

[10] P. Schuegraf, Z. Li, J. Tian, J. Shan, and K. Bittner, "Rectilinear building footprint regularization using deep learning," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2024.

[11] P. Schuegraf, J. Shan, and K. Bittner, "Planes4lod2: Reconstruction of lod-2 building models using a depth attention-based fully convolutional neural network," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 211, pp. 425–437, 2024.

[12] P. Schuegraf, S. Gui, R. Qin, F. Fraundorfer, and K. Bittner, "Sat2building: Lod-2 building reconstruction from satellite imagery using spatial embeddings," *Submitted to ISPRS Journal of Photogrammetry and Remote Sensing*, 2024.

[13] H. Huang *et al.*, "Unet 3+: A full-scale connected unet for medical image segmentation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1055–1059, 2020.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., 2015, pp. 234–241.

[15] A. Mosinska, P. Márquez-Neila, M. Koziński, and P. Fua, "Beyond the pixel-wise loss for topology-aware delineation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3136–3145, 2018.

[16] P. Schuegraf, M. Fuentes Reyes, Y. Xu, and K. Bittner, "Roof3d: A real and synthetic data collection for individual building roof plane and building sections detection," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X1W1, pp. 971–979, 2023.

[17] S. Gui, P. Schuegraf, K. Bittner, and R. Qin, "Unit-level lod2 building reconstruction from satellite-derived digital surface model and orthophoto," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2024.

[18] S. Gui and R. Qin, "Automated lod-2 model reconstruction from vhr resolution satellite-derived digital surface model and orthophoto," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 181, pp. 1–19, 2021.

[19] P. d'Angelo and P. Reinartz, "Semiglobal matching results on the isprs stereo matching benchmark," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, pp. 79–84, 2012.

[20] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 328–341, 2008.

[21] P. d'Angelo and G. Kuschk, "Dense multi-view stereo from satellite imagery," *IEEE International Geoscience and Remote Sensing Symposium*, pp. 6944–6947, 2012.

[22] Z. M. Moratto, M. J. Broxton, R. A. Beyer, M. Lundy, and K. Husmann, "Ames stereo pipeline, nasa's open source automated stereogrammetry software," *Lunar and Planetary Science Conference*, vol. 41, p. 2364, 2010.

[23] O. C. Ozcanli, Y. Dong, J. L. Mundy, H. Webb, R. Hammoud, and V. Tom, "A comparison of stereo and multiview 3-d reconstruction using cross-sector satellite imagery," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 17–25, 2015.

[24] K. Gong and D. Fritsch, "A detailed study about digital surface model generation using high resolution satellite stereo imagery," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III-1, pp. 69–76, 2016.

[25] M. Lehner and P. Reinartz, "Stereo evaluation of cartosat-1 data summary of dlr results during cartosat-1 scientific assessment program," *International Society for Photogrammetry and Remote Sensing (ISPRS)*, 1207 ff. 2016.

[26] R. Müller, T. Krauß, M. Lehner, and P. Reinartz, "Automatic production of a european orthoimage coverage within the gmes land fast track service using spot 4/5 and irs-p6 liss iii data," *ISPRS Conference Proceedings*, vol. 46, p. 6, 2007.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[28] T. Bartz-Beielstein, J. Branke, J. Mehnen, and O. Mersmann, "Evolutionary algorithms," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, 2014.

[29] P. J. M. v. Laarhoven and E. H. L. Aarts, "Simmulated annealing: Theory and applications," 1987.

[30] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, pp. 462–466, 1952.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014.

[33] N. Morgan and H. Bourlard, "Generalization and parameter estimation in feedforward nets: Some experiments," *Advances in Neural Information Processing Systems*, vol. 2, 1989.

[34] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *Proceedings of the 36th International Conference on Machine Learning*, pp. 6105–6114, 2019.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[36] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *ArXiv*, vol. abs/1207.0580, 2012.

[37] C. Henry, J. Hellekes, N. Merkle, S. M. Azimi, and F. Kurz, "Citywide estimation of parking space using aerial imagery and osm data fusion with deep learning and fine-grained annotation," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B$2_2$021, pp. 479–485, 2021.

[38] K. Bittner, P. Reinartz, and M. Korner, "Late or earlier information fusion from depth and spectral data? large-scale digital surface model refinement by hybridcgan," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1471–1478, 2019.

[39] P. Schuegraf and K. Bittner, "Automatic building footprint extraction from multi-resolution remote sensing images using a hybrid fcn," *ISPRS International Journal of Geo-Information*, vol. 8, 2019.

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

[41] K. Bittner, F. Adam, S. Cui, M. Körner, and P. Reinartz, "Building footprint extraction from vhr remote sensing images combined with normalized dsms using fused fully convolutional networks," *IEEE J. of Select. Topics in Appl. Earth Observ.s and Remote Sens.*, vol. 11, pp. 2615–2629, 2018.

[42] A. Huertas and R. Nevatia, "Detecting buildings in aerial images," *Computer Vision Graphicas Image Processing*, vol. 41, pp. 131–152, 1988.

[43] R. Guercke and M. Sester, "Building footprint simplification based on hough transform and least squares adjustment," 2011.

[44] F. Rottensteiner, J. Trinder, S. Clode, K. Kubik, and B. Lovell, "Building detection by dempster-shafer fusion of lidar data and multispectral aerial imagery.," pp. 339–342, 2004.

[45] N. Ekhtari, M. J. V. Zoej, M. R. Sahebi, and A. Mohammadzadeh, "Automatic building extraction from lidar digital elevation models and worldview imagery," *Journal of Applied Remote Sensing*, vol. 3, 2009.

[46] A. Turlapaty, B. Gokaraju, Q. Du, N. Younan, and J. Aanstoos, "A hybrid approach for building extraction from spaceborne multi-angular optical imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, 2012.

[47] Q. Li, L. Mou, Y. Hua, Y. Shi, and X. X. Zhu, "Building footprint generation through convolutional neural networks with attraction field representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022.

[48] Y. Zhang, W. Li, W. Gong, Z. Wang, and J. Sun, "An improved boundary-aware perceptual loss for building extraction from vhr images," *Remote Sensing*, vol. 12, 2020.

[49] P. Liu *et al.*, "Building footprint extraction from high-resolution images via spatial residual inception convolutional neural network," *Remote Sensing*, vol. 11, 2019.

[50] Q. Zhu, C. Liao, H. Hu, X. Mei, and H. Li, "Map-net: Multiple attending path neural network for building footprint extraction from remote sensed imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, pp. 6169–6181, 2021.

[51] A. Abdollahi, B. Pradhan, S. Gite, and A. Alamri, "Building footprint extraction from high resolution aerial images using generative adversarial network (gan) architecture," *IEEE Access*, vol. 8, pp. 209 517–209 527, 2020.

[52] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, 2015.

[53] M. Amo-Boateng, N. E. N. Sey, A. A. Amproche, and M. K. Domfeh, "Instance segmentation scheme for roofs in rural areas based on mask r-cnn," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 25, pp. 569–577, 2022.

[54] K. Zhao, J. Kang, J. Jung, and G. Sohn, "Building extraction from satellite images using mask r-cnn with building boundary regularization," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 224–242, 2018.

[55] Q. Wen *et al.*, "Automatic building extraction from google earth images under complex backgrounds based on deep instance segmentation network," *Sensors*, vol. 19, 2019.

[56] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, 2016.

[57] C. Szegedy *et al.*, *Going deeper with convolutions*, 2014. arXiv: 1409.4842 [cs.CV].

[58] S. Chen, Y. Ogawa, C. Zhao, and Y. Sekimoto, "Large-scale individual building extraction from open-source satellite imagery via super-resolution-based instance segmentation approach," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 195, pp. 129–152, 2023.

[59] A. Dosovitskiy *et al.*, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV].

[60] W. Zhao, C. Persello, and A. Stein, "Extracting planar roof structures from very high resolution images using graph neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 187, pp. 34–45, 2022.

[61] Q. Li *et al.*, "Instance segmentation of buildings using keypoints," *IEEE International Geoscience and Remote Sensing Symposium*, pp. 1452–1455, 2020.

[62] Z. Li, J. D. Wegner, and A. Lucchi, "Topological map extraction from overhead images," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1715–1724, 2019.

[63] S. P. Mohanty *et al.*, "Deep learning for understanding satellite imagery: An experimental survey," *Frontiers in Artificial Intelligence*, vol. 3, 2020.

[64] M. Li, F. Lafarge, and R. Marlet, "Approximating shapes in images with low-complexity polygons," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8633–8641, 2020.

[65] N. Girard, D. Smirnov, J. Solomon, and Y. Tarabalka, "Polygonal building extraction by frame field learning," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5887–5896, 2021.

[66] X. Sun, W. Zhao, R. V. Maretto, and C. Persello, "Building polygon extraction from aerial images and digital surface models with a frame field learning framework," *Remote Sensing*, vol. 13, 2021.

[67] S. Zorzi, S. Bazrafkan, S. Habenschuss, and F. Fraundorfer, "Polyworld: Polygonal building extraction with graph neural networks in satellite images," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1838–1847, 2022.

[68] S. Zorzi and F. Fraundorfer, "Re: Polyworld-a graph neural network for polygonal scene parsing," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16 762–16 771, 2023.

[69] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5221–5229, 2017.

[70] F. H. Wagner, R. Dalagnol, Y. Tarabalka, T. Y. F. Segantine, R. Thomé, and M. C. M. Hirye, "U-net-id, an instance segmentation model for building extraction from satellite images—case study in the joanópolis city, brazil," *Remote Sensing*, vol. 12, 2020.

[71] V. Iglovikov, S. Seferbekov, A. Buslaev, and A. Shvets, "Ternausnetv2: Fully convolutional network for instance segmentation," *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 233–237, 2018.

[72] Z. Li, B. Xu, and J. Shan, "Geometric object based building reconstruction from satellite imagery derived point clouds," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W13, pp. 73–78, 2019.

[73] L. Zebedin, J. Bauer, K. Karner, and H. Bischof, "Fusion of feature-and area-based information for urban buildings modeling from aerial imagery," *Proceedings of the European conference on computer vision (ECCV)*, pp. 873–886, 2008.

[74] S. Cui, Q. Yan, and P. Reinartz, "Complex building description and extraction based on hough transformation and cycle detection," *Remote Sensing Letters*, vol. 3, pp. 151–159, 2012.

[75] J. Tian and P. Reinartz, "Fusion of multi-spectral bands and dsm from worldview-2 stereo imagery for building extraction," *Joint Urban Remote Sensing Event (JURSE)*, pp. 135–138, 2013.

[76] D. Marcos *et al.*, "Learning deep structured active contours end-to-end," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8877–8885, 2018.

[77] S. Gur, T. Shaharabany, and L. Wolf, "End to end trainable active contours via differentiable rendering," *arXiv preprint arXiv:1912.00367*, 2019.

[78] A. Hatamizadeh, D. Sengupta, and D. Terzopoulos, "End-to-end trainable deep active contour models for automated image segmentation: Delineating buildings in aerial imagery," *Proceedings of the European conference on computer vision (ECCV)*, pp. 730–746, 2020.

[79] W. Zhao, C. Persello, and A. Stein, "Building outline delineation: From aerial images to polygons with an improved end-to-end learning framework," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 175, pp. 119–131, 2021.

[80] B. Dukai, H. Ledoux, and J. Stoter, "A multi-height lod1 model of all buildings in the netherlands," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-4/W8, pp. 51–57, 2019.

[81] R. Peters, B. Dukai, S. Vitalis, J. van Liempt, and J. Stoter, "Automated 3d reconstruction of lod2 and lod1 models for all 10 million buildings of the netherlands," *Photogrammetric Engineering and Remote Sensing*, vol. 88, pp. 165–170, 2022.

[82] H. Bagheri, M. Schmitt, and X. Zhu, "Fusion of multi-sensor-derived heights and osm-derived building footprints for urban 3d reconstruction," *ISPRS International Journal of Geo-Information*, vol. 8, 2019.

[83] D. Yu, S. Ji, J. Liu, and S. Wei, "Automatic 3d building reconstruction from multi-view aerial images with deep learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 171, pp. 155–170, 2021.

[84] Z. Li and J. Shan, "Ransac-based multi primitive building reconstruction from 3d point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 185, pp. 247–260, 2022.

[85] F. Alidoost, H. Arefi, and F. Tombari, "2d image-to-3d model: Knowledge-based 3d building reconstruction (3dbr) using single aerial images and convolutional neural networks (cnns)," *Remote Sensing*, vol. 11, 2219 ff. 2019.

[86] J. Lussange, M. Yu, Y. Tarabalka, and F. Lafarge, "3d detection of roof sections from a single satellite image and application to lod2-building reconstruction," 2023. arXiv: 2307.05409 [cs.CV].

[87] Z. Chen, D. Li, W. Fan, H. Guan, C. Wang, and J. Li, "Self-attention in reconstruction bias u-net for semantic segmentation of building rooftops in optical remote sensing images," *Remote Sensing*, vol. 13, 2021.

[88] X. Dai, M. Xia, L. Weng, K. Hu, H. Lin, and M. Qian, "Multiscale location attention network for building and water segmentation of remote sensing image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–19, 2023.

[89] X. Pan *et al.*, "Building extraction from high-resolution aerial imagery using a generative adversarial network with spatial and channel attention mechanisms," *Remote Sensing*, vol. 11, 2019.

[90] Z. Sun, W. Zhou, C. Ding, and M. Xia, "Multi-resolution transformer network for building and road segmentation of remote sensing image," *ISPRS International Journal of Geo-Information*, vol. 11, 2022.

[91] L. Wang *et al.*, "Unetformer: A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 190, pp. 196–214, 2022.

[92] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 240–248, 2017.

[93] R. Qin, J. Tian, and P. Reinartz, "Spatiotemporal inferences for use in building detection using series of very-high-resolution space-borne stereo images," *International Journal of Remote Sensing*, pp. 3455–3476, 2016.

[94] J. J. Betancur, "Approaches to the regularization of informal settlements: The case of primed in medellin, colombia," *Global Urban Development*, vol. 3, pp. 1–15, 2007.

[95] J. Claghorn and C. Werthmann, "Shifting ground: Landslide risk mitigation through community-based landscape interventions," *Journal of Landscape Architecture*, vol. 10, pp. 6–15, 2015.

[96] M. Sapena, M. Kühnl, M. Wurm, J. E. Patino, J. C. Duque, and H. Taubenböck, "Empiric recommendations for population disaggregation under different data scenarios," *Plos one*, vol. 17, 2022.

[97] M. Kuffer, K. Pfeffer, and R. Sliuzas, "Slums from space—15 years of slum mapping using remote sensing," *Remote Sensing*, vol. 8, p. 455, 2016.

[98] M. Wurm, T. Stark, U. of o Xiang Zhu, M. Weigand, and H. Taubenböck, "Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 59–69, 2019.

[99] N. J. Kraff, M. Wurm, and H. Taubenböck, "Uncertainties of human perception in visual image interpretation in complex urban environments," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4229–4241, 2020.

[100] T. Stark, M. Wurm, X. X. Zhu, and H. Taubenböck, "Detecting challenging urban environments using a few-shot meta-learning approach," *Joint Urban Remote Sensing Event (JURSE)*, pp. 1–4, 2023.

[101] H. Taubenböck, N. J. Kraff, and M. Wurm, "The morphology of the arrival city-a global categorization based on literature surveys and remotely sensed data," *Applied Geography*, vol. 92, pp. 150–167, 2018.

[102] M. Wurm *et al.*, "Revealing landslide exposure of informal settlements in medellín using deep learning," *Joint Urban Remote Sensing Event (JURSE)*, pp. 1–4, 2023.

[103] N. J. Kraff, H. Taubenböck, and M. Wurm, "How dynamic are slums? eo-based assessment of kibera's morphologic transformation," *Joint Urban Remote Sensing Event (JURSE)*, pp. 1–4, 2019.

[104] M. Wurm and H. Taubenböck, "Detecting social groups from space–assessment of remote sensing-based mapped morphological slums using income data," *Remote Sensing Letters*, vol. 9, pp. 41–50, 2018.

[105] D. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, pp. 112–122, 1973.

[106] B. Baheti, S. Innani, S. Gajre, and S. Talbar, "Eff-unet: A novel architecture for semantic segmentation in unstructured environment," *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1473–1481, 2020.

[107] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," vol. 24, 381–395, 1981.

[108] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *International Conference on Learning Representations*, 2019.

[109] S. Gui, R. Qin, and Y. Tang, "Sat2lod2: A software for automated lod-2 building reconstruction from satellite-derived orthophoto and digital surface model," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B2-2022, pp. 379–386, 2022.

[110] S. Woo, J. Park, J.-Y. Lee, and I. Kweon, "Cbam: Convolutional block attention module," *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

[111] K. Bittner, P. Reinartz, and M. Körner, "Late or earlier information fusion from depth and spectral data? large-scale digital surface model refinement by hybridcgan," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1471–1478, 2019.

[112] K. Bittner, L. Liebel, M. Körner, and P. Reinartz, "Long-short skip connections in deep neural networks for dsm refinement," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B2-2020, pp. 383–390, 2020.

[113] C. Stucker and K. Schindler, "Resdepth: A deep residual prior for 3d reconstruction from high-resolution satellite images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 183, pp. 560–580, 2022.

[114] D. Neven, B. Brabandere, M. Proesmans, and L. Van Gool, "Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8837–8845, 2019.

[115] G. Roggiolani, M. Sodano, T. Guadagnino, F. Magistri, J. Behley, and C. Stachniss, "Hierarchical approach for joint semantic, plant instance, and leaf instance segmentation in the agricultural domain," *IEEE International Conference onand Automation*, pp. 9601–9607, 2023.

[116] J. Yu and M. Blaschko, "Learning submodular losses with the lovasz hinge," *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[117] V. Patil, C. Sakaridis, A. Liniger, and L. Van Gool, "P3depth: Monocular depth estimation with a piecewise planarity prior," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1610–1621, 2022.