

**BUILDING FOOTPRINT RECONSTRUCTION FROM SATELLITE
IMAGERY USING A DEEP LEARNING FRAMEWORK WITH
GEOMETRIC REGULARIZATION**

MUHAMMAD KAMRAN

A DISSERTATION SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE

YORK UNIVERSITY
TORONTO, ONTARIO

JANUARY 2025

© Muhammad Kamran, 2025

Abstract

Since the launch of Landsat-1 in 1972, Earth observation satellites have significantly evolved, now capturing vast amounts of high-resolution imagery. These satellites continuously transmit data that helps us monitor urban expansion, plan infrastructure, and respond to natural disasters. Satellite imaging plays a crucial role in creating accurate spatial maps, offering detailed insights into the built environment. However, reconstructing maps from satellite images is a complex challenge. The field of computer vision has made impressive progress in object recognition and representation, but accurately modeling buildings as geometric primitives remains difficult. Unlike traditional pixel-based approaches, primitive-based object representation requires understanding the spatial structure and relationships that existing methods often struggle with. In this thesis, we explore four different network architectures aimed at improving our baseline model, R-PolyGCN, by integrating novel modules that enhance building footprint reconstruction. The final chapter presents Decoupled-PolyGCN, our most advanced deep learning model, that leverages Graph Convolution Network (GCN) to enhance building footprint reconstruction. By incorporating geometric regularity, multi-scale feature fusion, and Attraction Field Maps (AFM), the model generates more structured and precise building outlines from a single satellite image. Evaluations on the Wuhan University (WHU) and SpaceNet-2 datasets show that Decoupled-PolyGCN outperforms existing approaches, improving Average Precision (AP) by Average Recall (AR)% and AR by 10%. These improvements enable more accurate and reliable mapping, benefiting applications in urban planning, disaster management, and large-scale spatial analysis.

January, 2025

To my beloved kids Armaghan and Momina

Acknowledgements

I sincerely thank the mentors and supporters who have been instrumental throughout my journey to get that prestigious prefix of Dr before my name. This journey has been littered with challenges and triumphs, highs and lows. Yet, it has been a rewarding and unforgettable adventure, made all the more meaningful by the presence of these extraordinary individuals.

First, I would like to express my profound appreciation to my supervisor, Prof Gunho Sohn, for giving me this opportunity and guiding and supporting me throughout this journey. He believed in me when even I stopped believing in myself.

I would also like to extend my gratitude to the supervisory committee members (Prof Mojgan Jadidi and Prof Costas Armenakis) for their contribution of the utmost importance.

Now it's time to mention several individuals from the AUSM lab. Firstly, Afnan Ahmad, who kindly showed me his apartment on the first day I visited the lab. His apartment was the go-to place whenever I needed something in the first few years at York. Zahra Arjmandi, who began her PhD a couple of months ahead of me, provided me with the assistance needed during all the initial setup and installation on my computer. Dr Jungwon Kang was consistently available to offer help whenever I needed it. Dr Ali Baligh shared his insights about the lab and life in Canada. Dr Amin Alizadeh, for his motivation throughout the journey. He was always there to boost my confidence. Dr Connie Ko and Dr Andreas Wichmann provided valuable inputs to the research discussions. Zhen Liu was my first friend in the lab, and we discussed all the frustrations of the research life together. Towards the end, Dr YJ also helped alot. Last but not by any means least, Moein Sheikholislami was there throughout the design and experimentation process of my thesis. He was kind enough to help me whenever I needed it.

I made some great friends while studying at York, and because of them, I never gave up and continued to work hard to get this prestigious degree. I am indebted to my friend Ali Mirza, who welcomed me to Canada. He and his family made me feel at home, and I never felt homesick because of them. Then came two very important people, Utkarsh Panday and Simran Kanda, who stopped me when I was about to pack my bag and leave Canada. These two precious people help me regain my confidence. Then came Prateeksha Pathak, who supported me with her words of motivation and made me stay focused. Also, Saroar played an essential role in the last couple of years of my PhD to not give up.

Then comes my elder brothers, Dr. Muhammad Irfan and Dr. Muhammad Nauman. Both held doctorate degrees and intimately understood my journey, so they guided me at each step of my PhD. I frequently had detailed discussions with them, and their insights were beneficial.

Special mention of my **Wife and Parents**, without whom I am nothing. They believed in and supported me through thick and thin.

Table of Contents

Abstract	ii
Acknowledgements	iv
Table of Contents	vi
List of Tables	x
List of Figures	xi
Abbreviations	xiii
1 Introduction	1
1.1 Motivation	3
1.2 Background	4
1.3 Applications of Building Footprint Extraction	7
1.4 Problem Domain	9
1.4.1 Spatial Learning	9
1.4.2 Semantic Learning	10
1.4.3 Geometric Learning	12
1.5 Challenges	13
1.6 Research Objectives	15
1.7 Contributions	16
1.8 Thesis Organization	18

2	Literature Review	19
2.1	Basic Deep Learning and Computer Vision	19
2.2	Object Detection	23
2.2.1	Traditional Methods for Object Detection	24
2.2.2	Deep Learning-Based Object Detection	24
2.3	Image Segmentation	27
2.3.1	Traditional Image Segmentation Method	27
2.3.2	Deep Learning-Based Image Segmentation Method	28
2.4	Building Footprint Extraction	29
2.4.1	Traditional Building Footprint Extraction	29
2.4.2	Deep Learning-Based Building Extraction	30
2.5	Building Boundary Regularization	33
2.5.1	Traditional Building Boundary Regularization	33
2.5.2	Deep Learning-Based Building Boundary Regularization	33
2.6	Open Datasets for Building Footprint Extraction	35
2.7	Summary	40
3	RO-POLYGCN: Region-Based Orthogonal Polygon Graph Convolution	
	Network	41
3.1	Overview	41
3.2	Backbone Network	42
3.2.1	Feature Encoding	43
3.2.2	Object Detection	45
3.3	Graph Convolution Network	49
3.3.1	Graph Initialization	51
3.3.2	Graph Implementation	52
3.4	Loss Design	54
3.5	Training Strategy	60
3.6	Summary	61

4	FAE-PolyGCN: Feature Augmentation-based Edge-preserved Polygon Graph Convolution Network	63
4.1	Overview	63
4.2	Feature Augmentation Module	65
4.2.1	Resolution-Based Stream Separation	66
4.2.2	Image Gradient Concatenation	67
4.3	Edge-Preserving Boundary Mask Head	68
4.4	Network Detail	69
4.5	Loss Design	71
4.6	Summary	73
5	AT-PolyGCN: Graph-Based Polygonal Segmentation of Building Footprints Using Attraction Field Maps	75
5.1	Overview	75
5.2	Attraction Field Maps	77
5.2.1	Impact of Attraction Field Map	79
5.2.2	Attraction Field Generation	79
5.3	Network Details	80
5.4	Loss Design	82
5.5	Summary	84
6	Decoupled-PolyGCN	85
6.1	Overview	85
6.2	Decoupled Object Detection	87
6.3	Network Details	90
6.3.1	Training Phase	90
6.3.2	Inference Phase	92
6.4	Loss Design	93
6.5	Summary	95

7	Dataset Analysis and Preparation Pipeline	97
7.1	Challenges in Datasets for Building Footprint Extraction	97
7.2	Dataset Description and Preprocessing Pipeline	98
7.2.1	SpaceNet-2 Dataset	99
7.2.2	WHU Dataset	101
7.3	Dataset Distribution	107
7.4	Summary	112
8	Results and Analysis	115
8.1	Network Configuration and Implementation Details	115
8.2	Performance Evaluation Metrics	116
8.2.1	Raster-Based Metrics	117
8.2.2	Vector-Based Metrics	120
8.3	Experimental Results and Discussion	122
8.3.1	Qualitative Results	122
8.3.2	Quantitative Results	135
8.4	Discussions	139
8.5	Summary	144
9	Conclusions and Future Work	145
9.1	Conclusion	145
9.2	Future Work	149
	Bibliography	152

List of Tables

6.1	Top-performing networks on MMCV for horizontal bounding box generation	89
7.1	Characteristics of Datasets used in our Experiments	99
8.1	Overview of Results on WHU Dataset	135
8.2	Overview of Results on SpaceNet-2 Dataset	136
8.3	Overview of Area distribution based Results on WHU Dataset.	137
8.4	Overview of Area distribution based Results on SpaceNet-2 Dataset.	137
8.5	Overview of Vertex Distribution based Results on WHU Dataset.	138
8.6	Overview of Vertex Distribution based Results on SpaceNet-2 Dataset.	138
8.7	Comparison of DeCoupled-PolyGCN with competing methods on SpaceNet-2 Dataset	139
8.8	Comparison of DeCoupled-PolyGCN with competing methods on WHU Dataset	139

List of Figures

1.1	Representation of building footprints	5
1.2	Advent of Artificial Intelligence (AI)	6
1.3	Building Footprints for Storm Water Management	8
1.4	Spatial Learning	10
1.5	Semantic Learning	11
1.6	Research Contribution Flowchart	17
2.1	Classification and Segmentation Explanation	20
2.2	Classical CNN structure	23
3.1	Feature Encoding	44
3.2	Region Proposal Network	46
3.3	RoIAlign	48
3.4	Typical GCN Structure	49
3.5	RO-PolyGCN Network Architecture	51
3.6	Residual Block	53
4.1	Resolution-Based Stream Separation	67
4.2	Feature Augmentation Module with Image Gradient Concatenated	68
4.3	Edge Preserved Mask Branch	69
4.4	FAE PolyGCN Network Architecture	70
5.1	AFM Visualization	78

5.2	AT-PolyGCN Network Architecture	81
6.1	Detection Transformer Network Architecture	88
6.2	Detection Transformer with Collaborative Hybrid Assignments Training . . .	90
6.3	Training Phase of DeCoupled-PolyGCN	92
6.4	Inference Phase of DeCoupled-PolyGCN	93
7.1	SpaceNet Dataset	100
7.2	Image input to ArcGIS for preprocessing	102
7.3	Setting extent for image cropping	103
7.4	Cropping satellite imagery	103
7.5	Point Features	104
7.6	Line Features	105
7.7	Polygon Features	105
7.8	Resolving Open Polygon Issues	106
7.9	COCO format for groundtruth annotations	107
7.10	COCO format for individual Image annotation	107
7.11	Vertex-wise building distribution	108
7.12	Diversity of the dataset	109
7.13	Building density chart	110
7.14	Area-wise building Distribution	111
7.15	Area-wise building distribution Chart	111
7.16	Adjacent Buildings with Shared Corners	113
7.17	Buildings with Shared Corners distribution Chart	113
8.1	Intersection of Union (IoU)	118
8.2	Understanding True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN)	119
8.3	Overview of results from all of our Networks with Baseline(R-PolyGCN) and Groundtruth	123

8.4	Comparison of our baseline, R-PolyGCN, and our first model, RO-PolyGCN	124
8.5	Comparison of Results between our first model, RO-PolyGCN and our second model, FAE-PolyGCN	125
8.6	Comparison of Results between our second model, FAE-PolyGCN and our third model, AT-PolyGCN	126
8.7	Comparison of Results between our third model, AT-PolyGCN and our last model, Decoupled-PolyGCN	127
8.8	Overview of Results From our best model, Decoupled-PolyGCN	128
8.9	Comparison of Results with SOTA Model, HiSup	129
8.10	Handling Diversity of Buildings	130
8.11	Capturing buildings of varying sizes	131
8.12	Detecting complex building shapes	132
8.13	Distinguishing adjacent buildings and buildings at the edges	133
8.14	Extracting occluded buildings	134

List of Abbreviations

AFM Attraction Field Maps

AI Artificial Intelligence

AP Average Precision

AR Average Recall

AT-PolyGCN Graph-Based Polygonal Segmentation of Building Footprints Using Attraction Field Maps

AWS Amazon Web Services

CNN Convolutional Neural Networks

CO-DETR Collaborative Detection Transformers

DETR Detection Transformers

DL Deep Learning

DNN Deep Neural Network

ESA European Space Agency

FAE-PolyGCN Feature Augmentation-based Edge-preserved Polygon Graph Convolution Network

FAM Feature Augmentation Module

FCN Fully Convolutional Network

FPN Feature Pyramid Network

GCN Graph Convolution Network

GDL Deep Geometric Learning

GIS Geographic Information Systems

GNN Graph Neural Networks

IoU Intersection over Union

MTA Max Tangent Angle

NASA National Aeronautics and Space Administration

NMS Non Maximum Suppression

NRCan Natural Resources Canada

OSM Open Street Maps

PoLiS Polygonal Line String Similarity

RO-PolyGCN Region-Based Orthogonal Polygon Graph Convolution Network

RPN Region Proposal Network

RoI Region of Interest

SOTA State of the art

UCS Union of Concerned Scientists

USGS United States Geological Survey

WHU Wuhan University

Chapter 1

Introduction

Maps have been vital to human civilization for centuries, especially for navigation, planning, and understanding the world. Traditionally, maps were created manually by explorers and cartographers who carefully measured and documented landscapes, producing renowned works like Ptolemy’s *Geographia* and the Mercator Projection of the 16th century [1, 2]. In the 20th century, before the introduction of satellites, cartographers relied on hand-drawn maps [3] using drafting tools to portray terrain features and contours, often including information from historical records and local knowledge. However, these methods were time-consuming and prone to inaccuracies. They frequently failed to cover remote areas comprehensively, highlighting the need for a transformative approach, one that satellite technology would eventually provide.

The advent of satellite technology revolutionized spatial mapping. When the Landsat 1 (originally called Earth Resources Technology Satellite), was launched in 1972 [4] by a joint National Aeronautics and Space Administration (NASA) and United States Geological Survey (USGS) program, it marked a new era for geospatial data collection. Over the decades, the deployment of satellites for Earth observation has grown exponentially [5]. According to the Union of Concerned Scientists (UCS) satellite database report[6], today, hundreds of operational satellites provide a continuous stream of data for mapping and analysis. These satellites fall into two primary categories: optical and radar. Optical satellites, such as Landsat[7] and Sentinel-2 [8], capture high-resolution imagery across various spectral bands,

enabling detailed visual analysis. Radar satellites, such as Sentinel-1 [9] and TerraSAR-X [10], use SAR (synthetic aperture radar) to penetrate clouds and capture data in all weather conditions, offering unique advantages in challenging environments[11, 12]. The resolution of these satellite imagery ranges from a few centimeters to meters.

Maps serve diverse purposes, from urban planning[13, 14] and disaster management[15, 16] to ecological monitoring and agricultural assessment [17, 18, 19]. The creation of maps typically involves a series of complex processes, including geometric registration (ensuring accurate spatial alignment of data), semantic conversion (assigning meaningful labels to mapped features), and vector extraction (representing features as abstract, simplified forms). While each step is crucial, this thesis focuses specifically on vector extraction, translating spatial features into abstract representations that encode their essential geometric and relational properties.

Abstract shape representations, such as building footprints or road networks, are key to simplifying and analyzing spatial data. These representations are particularly important for automation and scalability in map creation, as they allow for efficient storage, processing, and interpretation. Despite the decades of research devoted to map-making, the study of vectors and abstract representations remains underdeveloped compared to other components of spatial mapping. Advances in artificial intelligence and remote sensing technologies have addressed some challenges, but full automation of vector extraction and abstraction remains an open problem.

Automation in mapping [20, 21] is becoming increasingly vital due to rapid urban growth. Cities are expanding at unprecedented rates, with an estimated thousands of urban buildings constructed annually. Frequent updates to maps are essential for urban planning, infrastructure development, and environmental management. However, traditional methods of map creation, which rely on human labeling, cannot keep pace with the speed of urbanization. Mapping agencies like the USGS, Natural Resources Canada (NRCan) and the European Space Agency (ESA) provide high-quality maps, but their efforts are constrained by the time, labor, and costs associated with manual annotation. As a result, many regions, particularly

in developing countries, lack up-to-date or comprehensive maps, as documented in several reports highlighting global mapping gaps.

To address these challenges, researchers have explored both AI-driven and non-AI approaches to automate abstraction and vectorization. The emergence of the "new space era," characterized by advances in satellite technology, miniaturization, and commercial space ventures, has further transformed how spatial data is collected and utilized. This era has not only made easy access to Earth observation data but also enabled unprecedented insights into urban growth and land-use changes, impacting lives worldwide.

In this new space age, my thesis is dedicated to automating the process of vector extraction and abstract representation of buildings using the most powerful intelligence available. By leveraging advancements in deep learning, geospatial analysis, and computational efficiency, this work aims to bridge the gap between human-driven mapping and the demands of a rapidly urbanizing world, contributing to the development of scalable and automated mapping solutions for the future.

1.1 Motivation

The growing accessibility of satellite imagery has revolutionized geospatial analysis, offering unprecedented opportunities for creating accurate and detailed maps. However, significant challenges persist, particularly in the extraction of building footprints, which are vital for applications in urban planning, disaster management, and environmental monitoring. Despite advancements, much of the current process remains reliant on manual or semi-automatic methods, requiring substantial time and effort. Platforms like Open Street Maps (OSM) [22], which rely heavily on user contributions for manual corrections, highlight the limitations of existing systems in handling the vast quantities of available satellite imagery. These inefficiencies hinder the timely generation of accurate and up-to-date maps, which are critical in rapidly urbanizing areas.

A core challenge lies in the diversity of building designs, which vary dramatically across regions. This architectural variability makes it difficult for current models to generalize

effectively beyond specific localized datasets, often leading to a lack of scalability in footprint extraction techniques. Furthermore, balancing recognition and localization remains a persistent issue. While high-level abstract features aid in recognizing objects, they often lose the precise spatial details necessary for accurate localization. Conversely, low-level spatial features retain spatial precision but lack semantic depth, creating a trade-off that complicates accurate footprint extraction. The reliance on manually defined parameters for regularization further limits the adaptability and generalization of these techniques, particularly for irregular or complex building shapes.

Adding to the challenges, the preparation of high-quality labeled datasets is constrained by issues like imbalanced class distributions, noisy annotations, and the complexity of converting data into formats suitable for neural network training. These obstacles hinder the development of robust and scalable models for building footprint extraction.

Motivated by these challenges, my thesis aims to address the critical need for automation in building footprint extraction, focusing on overcoming architectural variability, enhancing generalization, and accurately capturing geometric representations. By leveraging advances in deep learning, computer vision, and satellite data processing, this research seeks to reduce human intervention, enhance model scalability, and bridge the gap between the demands of rapid urbanization and the limitations of current mapping practices. Through this work, I aspire to contribute to the automation of geospatial mapping in the context of the new space era, where the integration of cutting-edge technology is transforming the way we understand and interact with our built environment.

1.2 Background

Remote sensing techniques allow us to acquire information about Earth’s surface without physical contact [23]. Among these techniques, satellite images play a crucial role as a valuable data source. Their spatial, spectral, temporal, and radiometric resolutions and large-scale coverage make them indispensable for generating digital maps in Geographic Information

Systems (GIS), including Google Maps[24] and Bing Maps[25].

Buildings play a significant role in shaping cities, serving as the backbone of urban infrastructure. As urbanization accelerates, more people choose to stay in cities. At present, 55% of the world’s population is settled in urban areas, and this proportion is most likely to expand to 68% by 2050, according to the United Nations report [26].

The building footprint, as defined by [27], refers to the specific area on a project site that is occupied by the building structure. The perimeter of the building plan determines it. Building footprints do not encompass parking lots, landscapes, or other non-building facilities. Cartographers and planners more commonly define it as roof representations moved to the structure’s base. Figure 1.1 illustrates the three commonly used representation types, namely mask, boundary, and corner, for representing the building footprint.

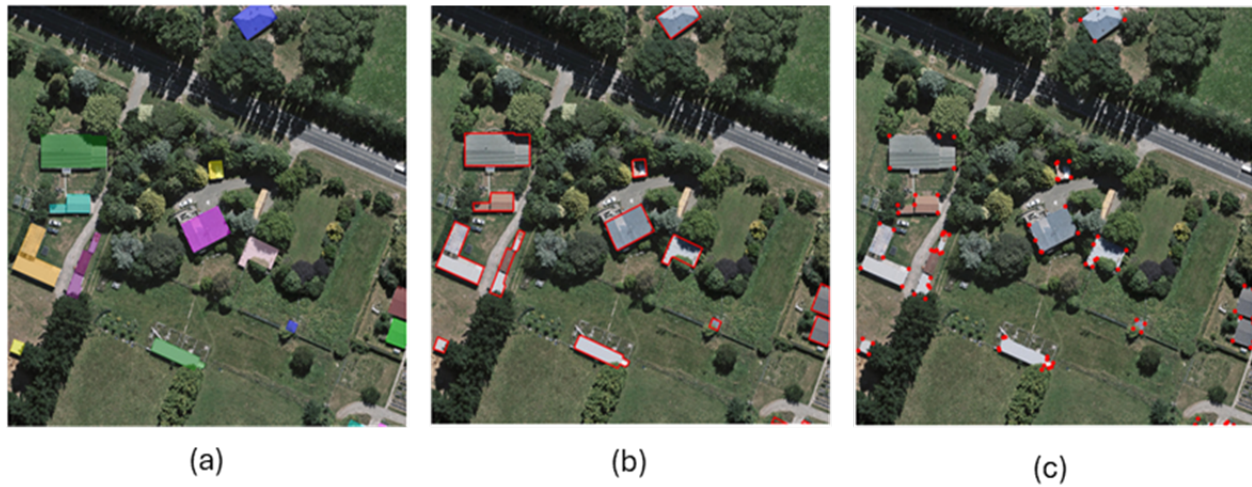


Figure 1.1 Representation of building footprints

The recent development of Machine Learning and Deep Learning [28] has significantly revolutionized the field of Artificial Intelligence in the last few years, outperforming numerous conventional methods in areas such as computer vision, speech recognition image classification, and natural language processing. Machine learning, deep learning, and neural networks are interconnected, but neural networks are specifically a part of machine learning, and deep learning is a specialized subset of neural networks, as depicted in Figure 1.2.

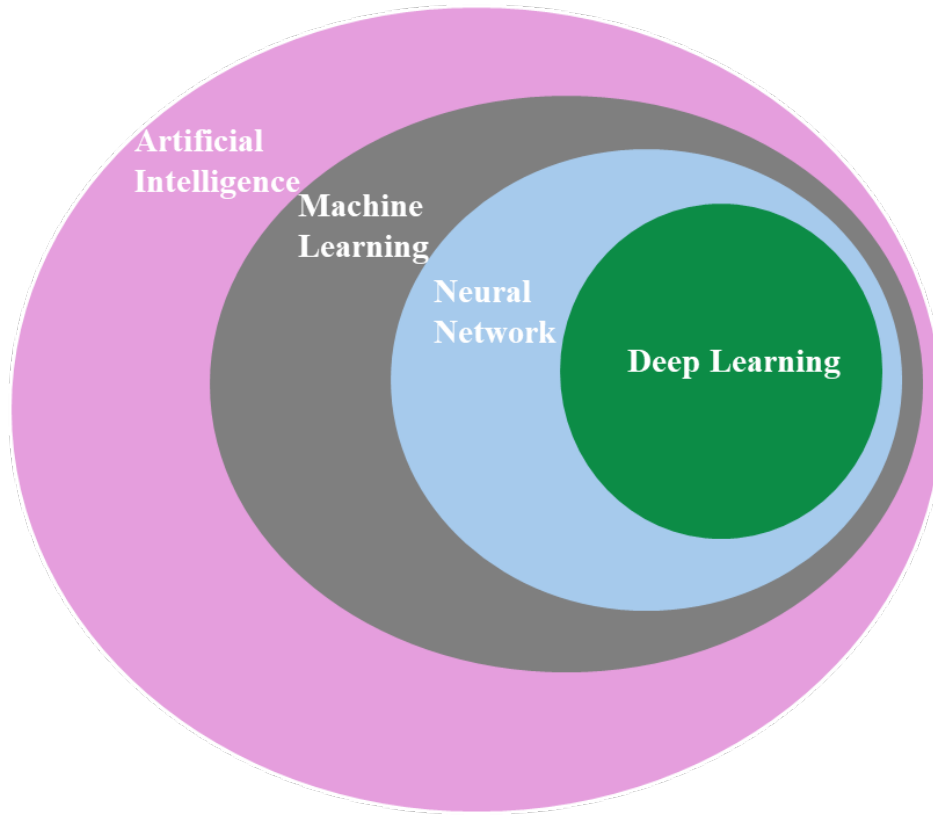


Figure 1.2 Advent of AI

Deep learning-based image classification models [29] are increasingly gaining attention and appeal within the remote sensing community as they learn and train from labeled data to find patterns within images, generalize their knowledge, and assign them to pertinent and suitable categories. The main objective is to assign a label to the entire image. Neural networks are composed of interconnected nodes arranged in layers. The layers consist of an input layer, one or more hidden layers, and an output layer. Every node, also known as an artificial neuron, establishes connections with other nodes using precise weights and thresholds. When the output is produced by a node beyond the predetermined threshold, it triggers data transmission to the subsequent layer. The term 'deep' in deep learning refers to the number of layers in a neural network. A deep neural network consists of more than three layers, which include the input and output layers. Convolutional Neural Networks (CNN)s [30] are a very powerful tool for image classification as they can automatically learn hierarchical features,

including spatial features such as wedges, textures, and shapes. It incorporates a logical machine-learning workflow. In the beginning, the data exploration is done to analyze and understand the dataset. Then an input pipeline is built to load and preprocess the images. After that, the model is trained using labeled data, followed by evaluating the performance of the model on test data, which ensures accuracy and other metrics. Then, multiple iterations are done to fine-tune the model by adjusting hyperparameters.

Since the last decade, there has been a significant increase in research on deep learning for remote sensing and building footprint extraction, and this trend continues to accelerate. Deep learning models can automatically learn rich and abstract features from immense training data and can be generalized to testing (new) data. Various datasets from the remote sensing domain in assorted formats await incorporation into deep learning research after being converted to an appropriate format. The geometric features, especially, are eminent in regularizing building footprints, increasing the quality of the building's footprints. These refined footprints are essential because they increase the usefulness of spatial maps across a broader spectrum of applications.

1.3 Applications of Building Footprint Extraction

Building footprint data is valuable for various crucial purposes, including urban planning and smart city development, which has made the extraction of building footprints a prominent subject of study in the scientific community. The building footprints are essential GIS data that allow for the estimation of energy consumption, standard of living, urban population, and property taxation. Similarly, urban planning highly depends on data regarding the physical area occupied by buildings. An accurate understanding of building footprints and 3-D building models becomes essential for deciphering complex city characteristics. Polygonal representations of regularized building footprints have high transferability across many GIS platforms and can be utilized for a wide range of applications, especially in the fields of digital city visualization, landscape modeling, virtual tourist systems, and

wireless communication. These can also be employed in the development and design of urban infrastructures, including water supply, sewage, electricity, and telecommunications. They are also useful in creating community facilities such as schools, hospitals, parks, and safety and transportation infrastructures like roads, police stations, and fire facilities. Additionally, they aid in the planning of various types of land use within the city, such as residential, commercial, industrial, and municipal areas. Regional planners might get advantages from multidimensional scenarios that provide practical insights into urban administration. Accurate knowledge of building footprints and 3D building models can be utilized to fully understand diverse urban features and provide urban planners with more realistic and multidimensional scenarios for managing urban areas [31].

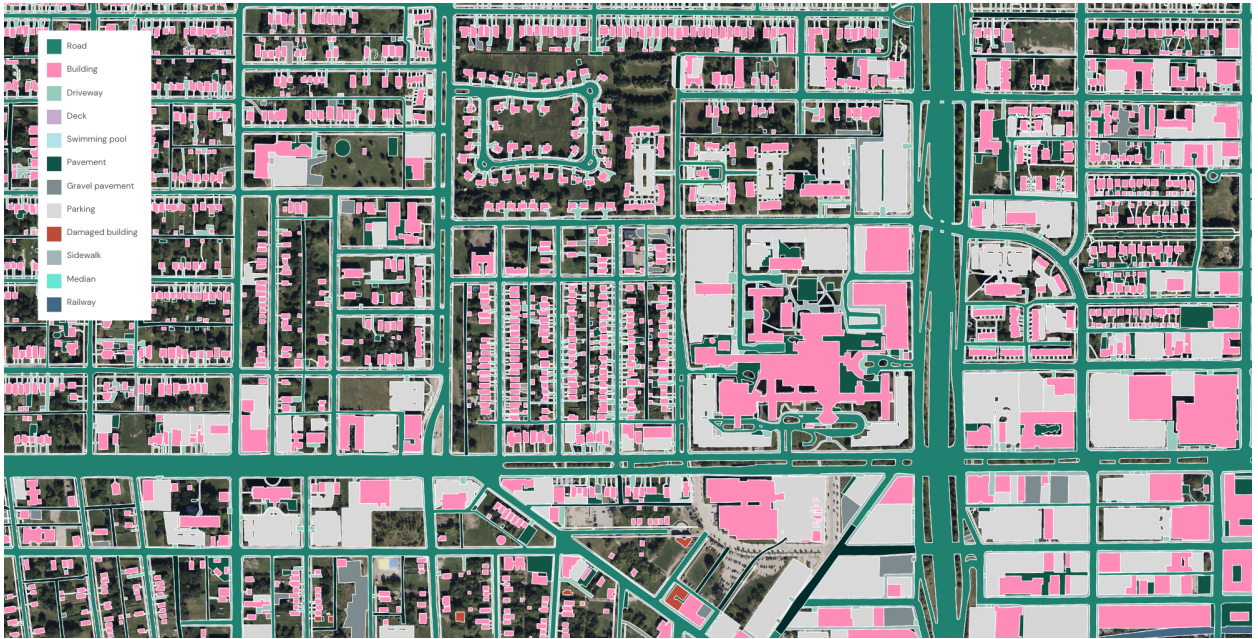


Figure 1.3 Building Footprints and other impervious surfaces extracted for the City of Detroit for Storm Water management [32]

We can also take the benefit of remotely accomplishing evaluations of real estate [33] with the building footprint data and other affiliated situational datasets. Furthermore, geo-marketing, such as market planning and development, site selection and forecasting, supply channels analysis, and modernization, can also be enhanced. Telecommunications companies strategically leverage the building footprint data to place signal towers, optimizing signal

transmission and coverage while minimizing the number of towers needed. Building footprint data is used in risk assessment tasks to improve the chance of preserving people’s assets and lives yearly. As standards upward thrust and prediction structures enhance planning and protection of the environment [34], microclimate studies such as urban heat island effect and rain-runoff modeling, or emergency response, which include estimation of destruction caused by earthquakes, rely on building models with higher accuracy and precision. Although some of these events are imminent, conducting a comprehensive risk assessment can mitigate potential risks and minimize overall damage.

1.4 Problem Domain

In order to create deep learning frameworks capable of autonomously extracting building footprints with boundary regularization from satellite images, we categorize the challenges into three primary components: spatial learning, semantic learning, and geometric learning.

1.4.1 Spatial Learning

The spatial features capture cardinal patterns related to edges, textures, shapes, and object boundaries. CNNs can create hierarchical representations that encode spatial information by analyzing and evaluating local pixel neighborhoods. Spatial learning’s main objective is to solve the object localization problem. Object localization is the process that aims to precisely classify and identify the location (bounding box) of specific objects inside an image.

Contrary to object classification, localization adds additional context about where the objects are situated. To accomplish object localization, deep neural networks extract spatial features from probable object regions. These regions are wrapped with features and then normalized to a fixed size. The resulting features are often referred to as Region of Interest (RoI) features. Spatial features contribute to handling variations in building sizes and scales. Understanding the contextual information around a building, such as neighboring structures and roads, enriches the process of accurate footprint extraction. Different buildings range



Figure 1.4 Spatial Learning

from small houses to large commercial structures, illustrating peculiar footprints. Spatial features take into account the surroundings to avoid false positives and are also crucial for robust extraction and reconstruction.

In essence, spatial learning plays a key role in object localization, enabling the detection of objects within images accurately. Neural networks leverage these features to accurately detect and localize building footprints within images. As deep learning models continue to grow, spatial reasoning remains very important for computer vision tasks.

1.4.2 Semantic Learning

The fundamental target of semantic segmentation is to efficiently label each pixel with a class based on the object type. Our research focuses on segmenting input images into binary maps, as we have two classes: the buildings and the background. Deep learning models excel at capturing semantic features at all levels, ranging from low-level edges to high-level object abstractions. At the same time, high-level semantics are crucial for recognizing all object



Satellite Image



Binary Segmentation Map

Figure 1.5 Semantic Learning

categories, and low-level and middle-level semantics impact pixel-level classification. Hence, efficiently combining semantic information from different levels of the networks is essential for accomplishing optimal segmentation performance.

It is crucial in diverse applications, including autonomous cars or trains, building footprint extraction, and identifying RoIs in various medical images. Given the high dimensionality of both input and output spaces, semantic segmentation in the computer vision field has conventionally been challenging. Deep learning-based methods have recently demonstrated robust semantic segmentation performance on numerous challenging benchmarks. The benefits of deep learning, ranging from representation learning to context modeling and scalability, contribute to significant improvements in semantic segmentation results. These approaches effectively utilize massive datasets paired with corresponding pixel-wise labels to train neural networks productively. During training, the model parameters are updated iteratively to improve the differentiable loss function, which measures the discrepancy between predictions and ground truth labels. During the process of inference, the network is given new data in order to generate a segmentation map that has the same spatial resolution as the input

image, where each pixel is assigned a label that corresponds to it.

1.4.3 Geometric Learning

Deep Geometric Learning (GDL) [35, 36] is an umbrella term that refers to a collection of methods that use symmetry and invariance to analyze machine learning challenges. It offers a standard design that enables the creation of various neural network designs, including Transformers, GCNs, and Convolutional Neural Networks (CNNs). Exposing regularities ingrained in the low dimensionality and structure of the physical world is the essence of GDL. The GDL attempts to alleviate the curse of dimensionality and improve performance in various applications by utilizing recognized symmetries.

The primary objective of our research is to utilize geometric learning techniques in order to predict the shapes of polygons accurately. Footprints that have regularized boundaries display polygonal shapes, while those with irregular boundaries do not possess geometric features. Based on our investigations and conclusions, it appears that utilizing the geometric information of polygons can provide a solution to the complex issue of boundary regularization. Geometric polygons can be depicted as a series of vertices or edges that do not conform to grid-based representations. Deep neural networks can learn geometric characteristics that adhere to grid-based structures and shape limitations. In addition, the non-grid-based characteristics possess more extensive geometric data and can also be acquired by deep learning models. Geometric learning allows for the creation of ideal polygon shapes, which leads to building footprints with regularized boundaries. Geometric learning allows for the creation of ideal polygon forms, which in turn leads to the development of building footprints with regularized boundaries.

Several existing image segmentation Deep Neural Network (DNN)s tend to produce irregularly shaped building segmentations. Experimental findings reveal that the segmentation predictions from widely used DNNs do not resemble real-life building footprints with straight edges and right-angled corners; mispredicted pixels often occur along the building edges. Specifically, mispredicted pixels are observed across all building types in encoder-decoder

networks. In contrast, for two-stage detectors where segmentations are made within traditional bounding boxes, mispredicted pixels tend to occur on buildings with rotated orientations (i.e., building edges are neither vertical nor parallel to the image edges), while accurate segmentations tend to occur on non-rotated buildings (i.e., building edges are either vertical or parallel to the image edges). These results highlight that the bounding-box-based approach may regularize non-rotated buildings, leading to higher-quality segmentation results. However, despite the assistance of bounding boxes, a significant proportion of buildings in satellite images exhibit randomly different orientations, resulting in substantial empty background spaces between the buildings and their bounding boxes, which contributes to mispredicted pixels.

Although attempts have been made to standardize the process of building segmentation into regular-shaped polygons, many commonly used approaches still depend on a large number of manually defined parameters and rules. Unfortunately, these methods rely significantly on human involvement and frequently encounter difficulties when attempting to generalize them to structures with complex shapes. Therefore, the resulting regularized polygons may not precisely depict the actual shapes of the buildings.

1.5 Challenges

Despite the growing availability and accessibility of satellite imagery, challenges still remain in accurate building footprint extraction. Creating accurate building footprints on GIS maps is constrained to manual or semi-automatic procedures, which demand considerable time and effort. For instance, OSM, which provides detailed building footprints that improve the precision of various open datasets, relies heavily on users for manual corrections and enhancements. Given the vast amount of available satellite imagery, processing these images remains time-consuming.

The wide variety in the design of the building roofs also poses challenges in the extraction of large-scale building footprints from satellite imagery. These roofs include but are not

limited to gable roofs, butterfly roofs, hip roofs, dutch roofs, mansard roofs, shed roofs, dormer roofs, M-shaped roofs, gambrel roofs, and flat roofs. As different areas have varied building architecture, so many models carry out the task of building footprint extraction in a smaller area more precisely. Still, they cannot generalize to areas with diverse building styles.

It is always difficult to create a balance between recognition and localization. Localization is the process of classifying and precisely locating an object within a particular image, typically by defining a tightly cropped bounding box centered on the instance. The abstract semantic information contributes to recognizing an object by high-level features contained in an image, but it lacks the spatial information required for localization. Similarly, the low-level features in an image contribute to spatial information for localization, but it lacks semantic information. Furthermore, the large size of satellite images, which contains buildings of variable scales, makes the process of collecting advanced characteristics for accurate identification and precise positioning more challenging.

The untapped capacity of geometric satellite images continues to provide a problem. Due to the predominant use of pixel-level or grid-based formats, extracting precise geometric information about polygon shapes can be challenging. Geometric characteristics are crucial in resolving the problem of irregular building boundaries, improving the accuracy of building footprints, and generating vector-based results for different uses. Nowadays, many popular methods for regularizing the segmentation of buildings into regularly shaped polygons still rely on a large number of manually specified parameters and rules. Unfortunately, these approaches depend heavily upon human intervention, which often leads to a lack of generalization for buildings with more complex shapes. As a result, the resulting regularized polygons may not accurately restore the original building shapes.

Hyperparameter optimization is crucial in deep learning for achieving high model performance [37]. Hyperparameters, like learning rate, batch size, number of layers, and dropout rate, are pre-set values that shape the model’s architecture and training process, affecting both convergence speed and accuracy. Selecting optimal values is challenging due to the non-linear, non-convex performance landscape, which makes manual tuning impractical.

Common methods for optimization [38, 39, 40] include grid search and random search. Grid search exhaustively tests all combinations within a predefined set, but it is computationally intensive. Random search, by sampling hyperparameter values randomly, is often more efficient and explores a broader range of configurations. Advanced techniques like Bayesian optimization and evolutionary algorithms use probabilistic models and iterative selection to balance exploration and exploitation, which is valuable for tasks where evaluating each configuration is costly. Interactions between hyperparameters, such as learning rate and batch size, further complicate optimization, requiring adaptive strategies to handle these dependencies. Hyperparameter optimization is computationally intensive, especially for large models, making techniques like early stopping and learning rate schedules essential for efficiency. Distributed frameworks enable parallelized search across multiple machines, though they require careful management of resources. Overall, hyperparameter optimization is vital for balancing efficiency and model performance, significantly impacting the success of deep learning models.

Previous research [31, 41] has observed the incorporation of additional light detection and ranging (LiDAR) point cloud data alongside images because LiDAR point clouds provide more evident geometric attributes. Nonetheless, creating large-scale maps only from LiDAR data is not practical due to its higher cost as compared to images. The challenges related to that include limited availability of high-quality labeled data, imbalanced class distribution, variability in building shapes and sizes, and noisy or incomplete annotations. Additionally, preparing the datasets and converting them into relevant formats for neural network training presents its own set of issues. Researchers and practitioners must tackle these challenges to enhance model accuracy and robustness.

1.6 Research Objectives

Given the advancements in remote sensing and deep neural networks, this research focuses on achieving the following objectives in the domain of abstract shape representation and

building boundary reconstruction.

1. To develop a deep neural network model capable of addressing the challenges of reconstructing regularized building boundaries and representing abstract shapes from remote sensing data, leveraging the strengths of deep geometric learning techniques.
2. To design a network that effectively integrates object detection and localization with pixel-level semantic segmentation, enabling the extraction of geometric contours and abstract boundary representations of structures.
3. To explore middle-level vision tasks by learning and predicting structured geometric and topological information through deep geometric learning approaches, facilitating accurate boundary reconstruction.
4. To enhance boundary detection accuracy and improve network training efficiency by introducing novel techniques for abstract representation and building boundary reconstruction.

1.7 Contributions

This study addresses key challenges in building footprint reconstruction and introduces advanced deep-learning frameworks focusing on boundary regularization. The main contributions of this work are summarized below.

Region-Based Orthogonal Polygon Graph Convolution Network (RO-PolyGCN):

This framework represents our initial deep learning-based approach for extracting building footprints from satellite imagery. Built upon the baseline R-PolyGCN[42], we enhanced the network by fine-tuning hyperparameters and integrating an orthogonality loss function. These improvements enable the network to generate precise building boundaries while ensuring well-regularized boundaries.

Feature Augmentation-based Edge-preserved Polygon Graph Convolution Network (FAE-PolyGCN): This model enhances RO-PolyGCN by incorporating two key

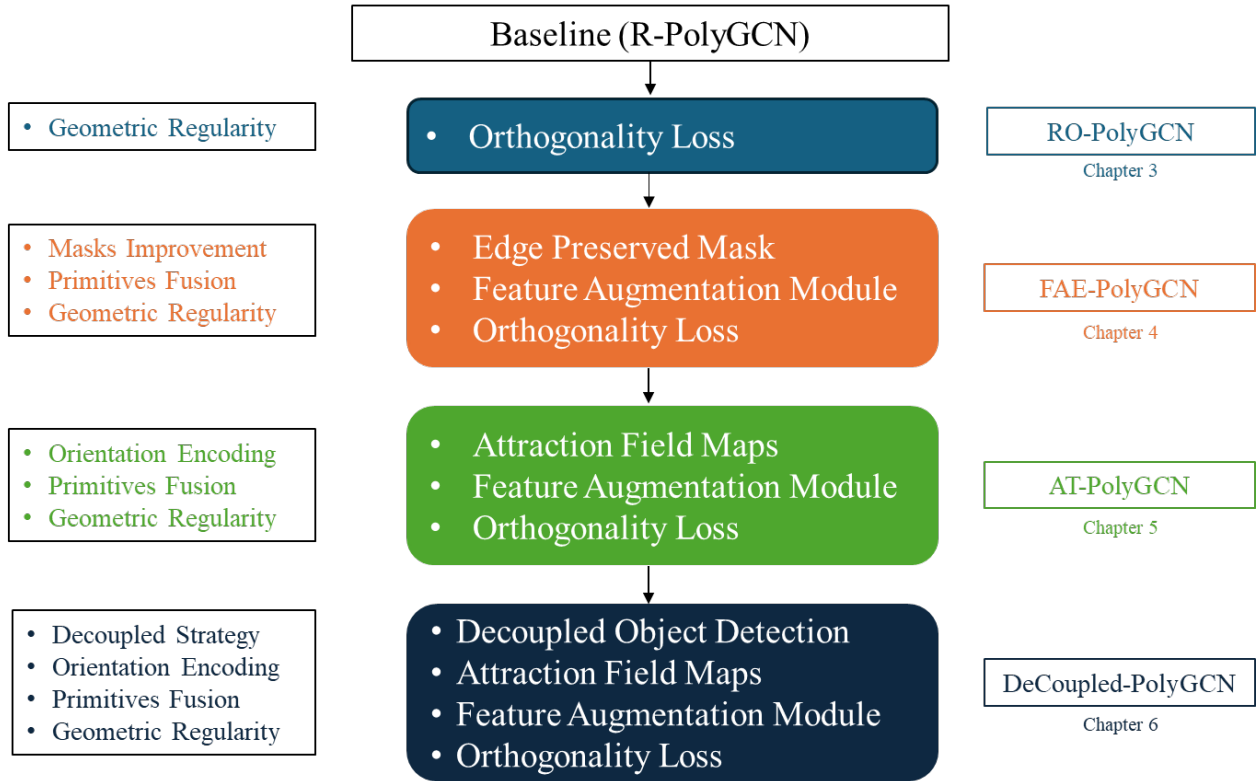


Figure 1.6 Research Contribution Flowchart

improvements: an edge-preserving mask head to retain sharp boundary edges and a Feature Augmentation Module to fuse features more effectively, resulting in better boundary predictions.

Graph-Based Polygonal Segmentation of Building Footprints Using Attraction Field Maps (AT-PolyGCN): This model leverages attraction field maps for more precise edge information and introduces innovative initialization techniques for graph networks, allowing the integration of orientation information.

Decoupled-PolyGCN: This model represents the pinnacle of our work, surpassing state-of-the-art methods across all evaluation metrics. This approach extracts RoIs using an offline object detection module. Apart from that, like AT-PolyGCN, it utilizes attraction field maps and variable initialization techniques for graph networks to refine predictions further and produce more accurate and regularized building polygons.

All of the proposed models combine GCNs with an instance segmentation backbone, enabling them to perform object recognition and localization, assign semantic labels to pixels, and generate regularized geometric polygon predictions simultaneously. Our custom training algorithms are specifically designed to handle multi-task loss functions, facilitating end-to-end training. Comprehensive experiments, including both raster- and vector-based evaluations, have been conducted to validate the effectiveness of these models.

To train our neural networks effectively, we prepared multiple datasets. Initially, we conducted experiments using the SpaceNet-2 dataset. To further generalize our approach, we later incorporated the WHU dataset. To ensure these datasets were properly formatted for training, we developed a comprehensive data preparation pipeline. This process involved an integrated approach, utilizing Python scripts along with tools like FME and ArcGIS to carefully preprocess and structure the data for our deep learning model.

1.8 Thesis Organization

This thesis consists of 8 chapters. Chapter 1 briefly introduces the study, covering the background, motivation, problem domain, research aims, and contributions. Chapter 2 of our study focuses on examining relevant studies and exploring accessible datasets for the purpose of extracting building footprints. Chapter 3 includes a comprehensive analysis of our initial network, including a detailed explanation of its architecture and innovative loss strategies. Chapter 4 covers our enhanced network, incorporating feature augmentation and boundary-preserving modules. In Chapter 5, we implemented Attraction field maps and included innovative initialization techniques. Chapter 6 discussed the decoupled approach for object detection and use of Attraction field maps and variable initialization methods. In Chapter 7, we discussed the SpaceNet-2 and WHU datasets in detail and our dataset pipeline for transforming these datasets into appropriate formats. In Chapter 8, we examined and compared our quantitative and qualitative results with the most current and advanced methods. In chapter 9, the study concludes and provides specific directions for future research.

Chapter 2

Literature Review

In this chapter, we will review the previous work and literature related to our research. We will divide the literature review into six sections. Section 2.1 will discuss deep learning and computer vision in general. In the subsequent sections, we will explore and discuss the comparison between conventional and deep learning-based models for object detection, segmentation, building footprint extraction, and building boundary regularization. In section 2.6, we will provide the relevant datasets that are available for our work.

2.1 Basic Deep Learning and Computer Vision

Recent advancements in deep learning and progress in device capabilities have significantly impacted the field of computer vision. Deep learning techniques in vision-based applications rapidly evolve because of increased computational power, energy efficiency, memory capacity, and image sensor resolution. The results are much better than before in terms of cost and performance. Basic deep learning models employ a multi-layer structure known as deep neural networks. These networks learn data representation with various levels of abstraction.

Deep learning has revolutionized computer vision by offering several benefits over traditional techniques. With deep learning, tasks such as image classification, semantic and instance segmentation, and object detection have achieved better accuracy.

Classification incorporates assigning a label or class to an entire image. The fundamental objective is to find whether an image contains a particular object. For instance, take an

example of the image in Fig; an image classifier would ascertain whether it belongs to the “balloon” class or, simply, whether this image has any balloons. Object detection transcends classification by identifying and localizing individual instances of objects within an image. In each image, object detection looks for the objects and their respective classes. In Figure 2.1, object detection would draw bounding boxes over all evident balloons, specifying their positions. Segmentation moves one step further by assigning labels to individual pixels to divide the image into relevant regions.

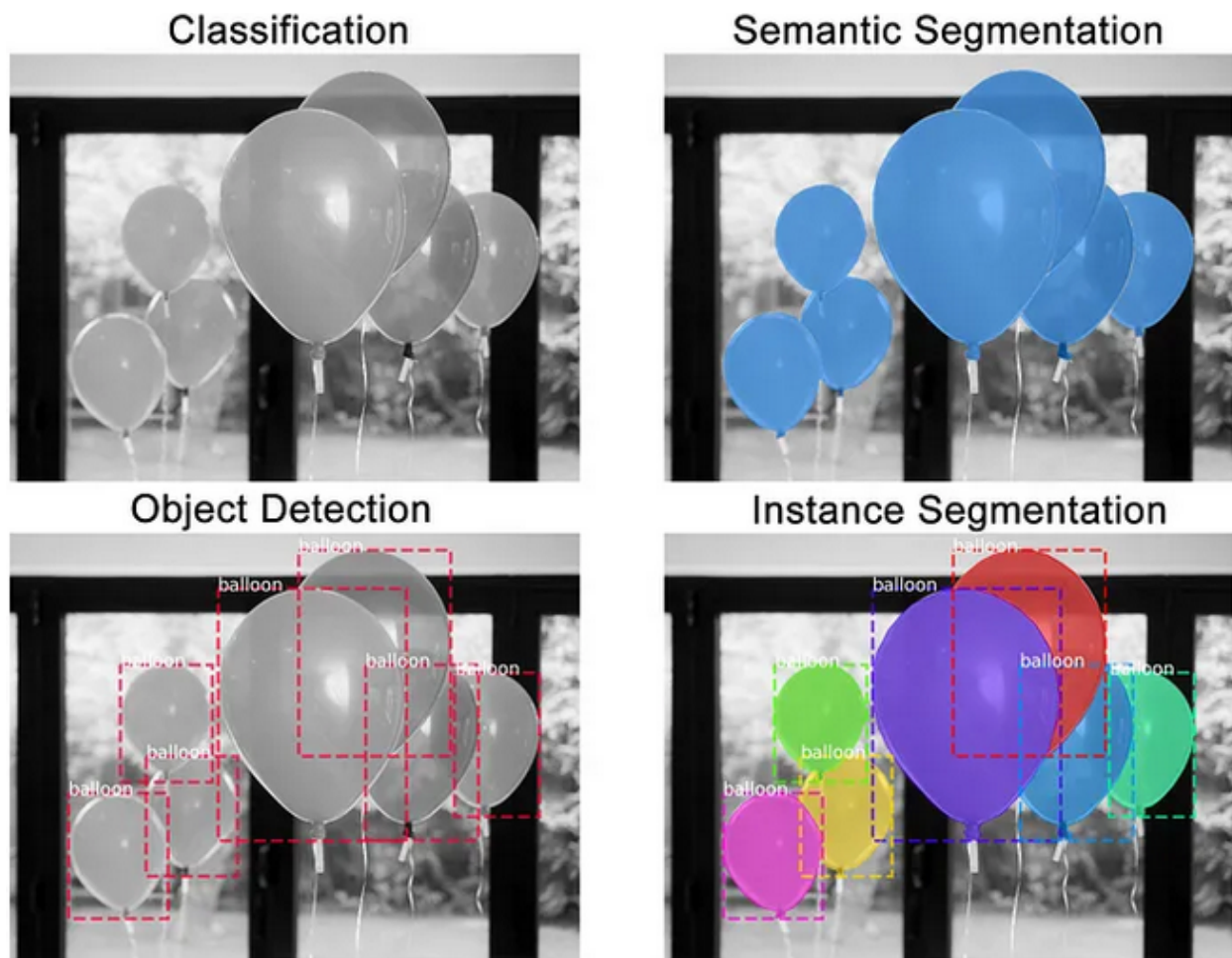


Figure 2.1 Classification and Segmentation Explanation [43]

Segmentation is further divided into two types. The first is Semantic Segmentation, an approach of delegating a classification label to every individual pixel. (e.g., “balloon,” “window,” “trees”). The pixels which belong to the same class are colored the same. In

Figure 2.1, at the top right, all balloons are assigned the same blue color. The second is Instance Segmentation, which expands the horizon of semantic segmentation by categorizing the individual instances of objects. All the instances of the same object get a unique color mask. In Fig, at the bottom-right, all the balloons are shown in different colors. Classification is helpful when looking for yes/ no answers about the occurrence of an object within an image. Object detection helps most in identifying vehicles and pedestrians for self-driving cars, being vigilant about intruders in surveillance activities, etc. Segmentation has its advantages in robotics, medical imaging, and scene understanding. In short, image classification as a whole is significant for understanding visual content, from primary classification to comprehensive localization and pixel-level segmentation.

Instead of traditional hard-coded programming-based solutions, deep learning neural networks are trained, requiring less fine-tuning and limited analysis from experts. Deep learning models can exploit video data, which is available in vast amounts in modern-day systems. Deep learning provides more flexibility than conventional computer vision, primarily domain-specific algorithms. Let's explore the differences between conventional computer vision techniques and deep learning-based approaches for object detection using a mobile robot as an example.

Before the deep learning era, computer vision techniques relied on feature descriptors such as SIFT [44], SURF [45], etc., to detect objects. Feature extraction was also a crucial step involving identifying small, descriptive, or essential patches in the images. Various computer vision algorithms, including edge, corner, and threshold detection, have contributed to feature extraction. Features used to be combined into a "bag of words" representation for each object class. During deployment, these "bag of words" definitions was searched against features in other images. If a substantial number of features matched, the image was to be used to classify as a specific object (e.g., balloon, donkey, etc.). This traditional approach had a few challenges because selecting relevant features for each image was subjective and complex. It was difficult to handle an increasing number of object classes. Additionally, fine-tuning each parameter required long trials and manual effort.

With the advent of deep learning, end-to-end learning was introduced, where models were trained on annotated image datasets instead of hand-crafted features. Deep Learning (DL) models are trained to find descriptive and salient features specific to each object class. DL models adapt to diverse object classes without domain-specific adjustments, resulting in superior performance compared to traditional algorithms with some computational trade-offs.

Deep neural networks take advantage of a complex combination of linear and nonlinear operations to create a connected architecture that automatically extracts deep features from input data. These features serve as the foundation for various tasks. Additionally, different frameworks can be utilized on these features for producing specific outcomes. In supervised learning, the network predictions are examined against the objective function, often called the loss function, that finds the difference between the actual result and the measured outcome. Networks are designed to enhance the measured results or predicts by minimizing the loss function. Specifically, the method involves calculating gradients at the final layer in relation to the parameters of the network. The gradients are subsequently backpropagated through the entire network using chain rules. The network parameters are modified based on the gradients during training iterations. This optimization technique, known as backpropagation, is a building block of all neural networks.

The convolutional neural network (CNN) is the most frequently used neural network in computer vision applications. Its significant components include convolution and pooling layers. A classic example [46] displayed in Figure 2.2 is LeNet-5. This network was designed to recognize and classify handwritten characters in the images. LeNet-5 is a neural network consisting of seven layers, including three layers of convolution, two sub-sampling (pooling) levels, along with one fully connected layer. The convolution layers use 5x5 filters with local connections and weight sharing. Feature maps are generated at each layer, capturing relevant information and patterns in the image. Pooling layers merge neighboring regions by averaging their values, creating abstract high-level features. Finally, the fully connected layer produces predictions based on these features.

Domain adaptation aims to transfer knowledge from a source domain to a target domain

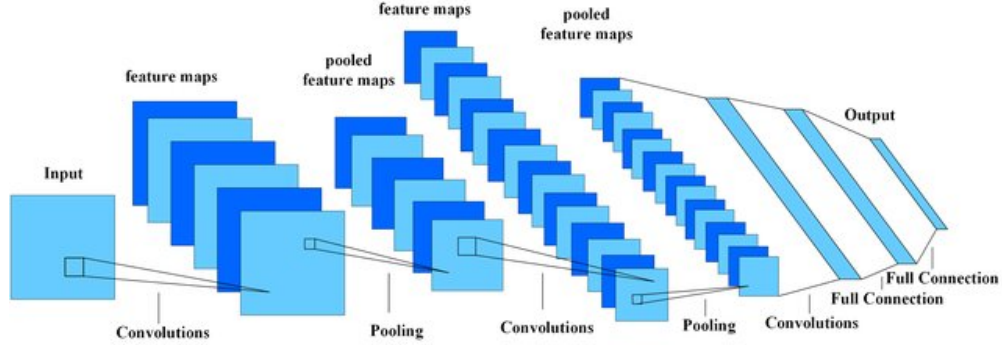


Figure 2.2 Classical CNN structure[46]

while avoiding the impact of domain shift. Within this particular context, the source-domain dataset has numerous samples that have been annotated. Nevertheless, the dataset specific to the target domain does not contain any labeled instances. Domain adaptation methods strive to enhance the efficacy of CNNs on the target domain. The source-domain dataset is utilized to align the data distribution between the two domains. Domain shift can be addressed using various approaches at various levels [86,87]. It can be able to tackle the domain-shift problem by exclusively aligning the distribution of images or address the issue of domain adaptation by focusing on both image and feature levels. Specific network architectures have been designed to balance computing costs with accuracy, resulting in the development of lightweight networks.

In the next sections, we will discuss the methods and networks used for traditional and deep learning-based object detection, image segmentation, building footprint extraction, and building boundary regularization.

2.2 Object Detection

An object detector is a computer vision algorithm or system designed to identify and locate specific objects within images or videos. Its main objective is to detect the presence and location of objects in visual input data accurately and efficiently.

2.2.1 Traditional Methods for Object Detection

If we look back to object detectors from two decades ago, we find that due to the limitations of effective image representation, early object detection algorithms relied on handcrafted features. In 2001, Viola et al [47]. developed the Viola-Jones detector, an object recognition framework for achieving real-time face detection of humans. The system utilized sliding windows to systematically examine every possible position and scale inside an image in order to determine whether any of the windows featured a human face. An integral image is used to speed up detection, making the computational complexity of each sliding window independent of its size. In 2005, Dalal et al. [48] proposed the histogram of Oriented Gradients (HOG) detector. It improved from previous methods of its time, such as SIFT and shape transforms. HOG works on a dense pixel grid with blocks, which is similar to a sliding window. Gradients are computed from pixel intensity changes, capturing local image patterns. HOG gained popularity for pedestrian detection. As an extension to the HOG detector, Felzenszwalb [[49], in 2009, introduced the Deformable Part-based Model (DPM). Later, R. Girshick [50] made a variety of improvements. It breaks down object detection, such as the problem of detecting a car, into a “divide and conquer” strategy by detecting object parts (e.g., window, body, wheels). During training, DPM learns how to decompose an object, and during inference, it combines detections of different parts.

2.2.2 Deep Learning-Based Object Detection

Object detection seemed to have reached a plateau after 2010 and faced limitations due to the saturation of performance in handcrafted features. However, in 2012, the evolution of convolutional neural networks (CNNs) breathed new life into the field [29]. Deep CNNs excelled at learning robust, high-level feature representations from images. The breakthrough came in 2014 with the proposal of Region-based CNNs (RCNNs) for object detection [51].

The RCNN typically utilizes a detection pipeline consisting of two stages. The system utilizes a search algorithm to generate approximately two thousand region proposals based

on the input image. Afterward, the recommended regions are fed into a Convolutional Neural Network (CNN) to gather relevant details. A Support Vector Machine (SVM) classifies these regions and makes predictions about bounding boxes using extracted information [52]. Fast R-CNN [53] improves the efficiency of R-CNN by utilizing a convolutional neural network (CNN) to analyze the entire image in a single pass. This eliminates the need to feed 2000 regions separately. Nevertheless, both R-CNN and Fast R-CNN necessitate a preprocessing stage to produce region recommendations or proposals utilizing a selective search technique. The process of generating this proposal can be somewhat time-consuming. Faster R-CNN [54] solves this issue by discarding the search algorithm. Instead, it utilizes a Region Proposal Network (RPN) to generate proposals directly from predetermined anchor boxes. The RPN utilizes the capabilities of the convolutional neural network to gain information on region suggestions, leading to improved speed and accuracy in detection.

In contrast, one-stage models bypass the phase of generating region proposals and instead employ a more straightforward technique. They immediately apply one-shot detection across densely sampled locations in the input image. One-stage models have a simplified and unified network design, making them more efficient. The YOLO [55] algorithm divides the image into grids of predetermined sizes and employs a Convolutional Neural Network (CNN) to estimate the bounding boxes and probability of different classes within each grid. SSD [56] extracts multi-scale features using a series of convolution layers. Recent works focus on directly detecting key points, such as corners or centers of objects, using CNNs. From these key points, bounding box predictions can be derived. Overall, the choice between two-stage and one-stage models depends on the trade-off between accuracy and efficiency in specific applications.

Deep learning methods use data-driven optimization to pass image features through neural network models. Marmanis et al. [57] applied an ensemble of networks pre-trained on large media image databases [58]. Then, they fine-tuned them on remote sensing imagery with a ground sampling distance of 0.1m. These models were originally designed for computer vision but are well generalized to remote sensing data. Fixing lower layer weights during early

training and later making them learnable improved computational efficiency. Maggiori et al. (2017) enhanced results by applying Fully Convolutional Network (FCN)s to remote sensing imagery [59]. Multi-scale processing captured contextual information by downsampling one branch while maintaining input resolution in the other branch. The branches were then combined, and an activation function was applied. The training involved initial training on inaccurate OSM ground truth and following a refinement step using hand-labeled ground truth. Using nDSM (normalized Digital Surface Model) as a data source, Bittner et al. (2017) extracted building footprints [60]. Height information proved valuable for this task. While FCRFs (Fully Connected Conditional Random Fields) improved fine details in building masks, we opted not to use them due to hyperparameter complexity. Adapting the FCN’s architecture, Bittner et al. (2018) addressed complex structures and spectral variations in building footprints, achieving promising results on VHR (Very High Resolution) remote sensing imagery [61]. Transformers, originally designed for natural language processing, have revolutionized deep learning by effectively capturing long-range dependencies and relationships in data. In the context of object detection, transformers replace traditional convolutional architectures by using self-attention mechanisms to process entire images at once. Unlike CNNs, which apply local filters, transformers allow each part of an image to interact with all other parts, helping the model understand contextual relationships between objects. This global perspective enables more accurate object detection, especially in complex scenes where multiple objects may be closely positioned or overlapping. Detection Transformers (DETR) is a specific application of transformers in object detection, offering a new approach compared to traditional models like Faster R-CNN or YOLO. Instead of relying on hand-crafted components such as anchor boxes or region proposals, DETR formulates object detection as a direct set prediction problem. It employs a Transformer encoder-decoder structure, where the encoder processes image features extracted by a convolutional backbone, and the decoder generates predictions using self-attention. Each object is directly predicted with a class label and bounding box, eliminating the need for post-processing steps like non-maximum suppression. While DETR provides a more streamlined and intuitive detection process, it can

be computationally expensive and may have slower inference times compared to optimized CNN-based detectors. However, its ability to model complex object relationships makes it particularly useful for challenging detection tasks.

2.3 Image Segmentation

One of the key problems in the research we are conducting is image segmentation. The deep learning researchers have also invested considerable effort in this area.

2.3.1 Traditional Image Segmentation Method

The majority of image segmentation algorithms proposed in recent decades can be primarily categorized into four groups: threshold-based, edge-based, region-based, and classification-based methods. Segmenting objects from images using threshold values is a straightforward and widely employed technique. This method [62] involves the classification of pixels into different groups based on manually or automatically defined thresholds, which are determined by the range of values of the pixels. However, the most significant drawback of image thresholding is its inability to distinguish between various regions that have comparable grayscale values. Edge-based approaches utilize edge-detection filters such as Sobel [63] and Canny [64] to identify the sharp transitions between adjacent pixels. This allows for the creation of boundaries that can be used for segmentation. Region-based approaches employ clustering [65, 66, 67], region-growing [68], shape analysis [69, 70], or graph-based methods [71] to segment distinct areas of an image. The drawback of the edge-based and region-based methods is that they often fail to provide stable and generalized results for images with highly varied illuminance and texture conditions. On the other hand, classification-based techniques regard image segmentation as the process of categorizing each individual pixel [72]. When appropriate feature extractors and classifiers are used, this method has a tendency to provide more accurate segmentations compared to the other three categories of methods.

2.3.2 Deep Learning-Based Image Segmentation Method

Given the substantial shift in building segmentation research from traditional approaches to deep learning-based methods, let us now delve into the topic of deep learning image segmentation.

A typical example of a deep learning model commonly used for semantic segmentation is a FCN. Within this network, the process of obtaining downsampled features involves passing the input image via conventional convolution and pooling layers. In addition, FCN utilizes deconvolution layers to boost the feature map’s resolution and match the original image’s size. The pixel-wise prediction technique produces semantic segmentation results on the up-sampled feature map. SegNet [73] and U-Net [74], which are built on the FCN architecture, employ an encoder-decoder structure. In this structure, the decoder component is specifically used to learn the process of up-sampling the feature map. To do this, [73] transfers the pooling indexes from the encoder to the decoder. The U-Net architecture [74] utilizes feature transfer from the encoder to the decoder to create a fusion of features from several scales. This enables the network to identify and classify small objects, as well as accurately separate and analyze images with a high density of objects.

Typically, there are two commonly used methods to accomplish instance segmentation. The first approach involves initially conducting semantic segmentation on the image, followed by the application of instantiation to group related pixels and identify distinct objects. Both DeepMask [75] and SharpMask [76] make use of this pipeline. Mask R-CNN [77] proposes an alternative method. Mask R-CNN is a method that is very similar to Faster R-CNN. It starts by performing instantiation and then uses a segmentation detection network to identify and locate items. The detection network can also produce RoI features that are accurately localized. In addition to RoI features, object masks are acquired by implementing FCN. The entire network possesses a comprehensive and end-to-end integrated design. Mask R-CNN’s segmentation achieves higher accuracy than models that utilized the first technique on the majority of benchmarks.

2.4 Building Footprint Extraction

Building footprint extraction and reconstruction have emerged as critical tasks in remote sensing and geospatial analysis owing to their diverse applications in urban planning, disaster management, environmental monitoring, and smart city development. Accurate building footprints provide essential insights into urban landscapes, enabling tasks such as population density estimation, infrastructure planning, and energy consumption analysis. The building footprint is a 2D graphical depiction that conveys accurate details regarding the exact position, dimensions, and configuration of a building on the ground. Building footprints are depicted using three conventional representation types: mask, boundary, and corner. Now, let's dig deep into examining both traditional approaches and deep learning-based techniques for building footprint extraction and reconstruction.

2.4.1 Traditional Building Footprint Extraction

Early approaches for generating building footprints from remote sensing images can be divided into three categories. The first category is based on geometrical primitives, such as edges and corners of the building, which are initially extracted from the image. These primitives are then assembled to form enclosed polygons corresponding to individual buildings. Overall, Geometric features play a vital role in this approach[78, 79, 80]. The second is the over-segmentation technique, which divides the image into superpixels or segments. Each segment represents a specific region, which can be used to delineate building boundaries. Common over-segmentation methods [81] include clustering, graph models, active contour models[82], and watershed segmentation [83, 84]. Last is classifier-based methods [72]. This method is divided into two stages: Initially, features are extracted from each pixel based on spatial, textual, or spectral characteristics. Machine learning models are then used to classify pixels as building or non-building. These methods leverage handcrafted features and effectively distinguish buildings from other objects. In addition to the three main categories, there is another method based on optical imagery called the index-based method. This

approach considers contrast and brightness information to extract buildings [85]. Next, the segmentation prediction of the building footprint is made based on the extracted features through techniques that include template matching [86], graph cut [69], adaptive boosting (AdaBoost) [87], random forests [88], support vector machines [89] or conditional random fields (CRF) [90].

Although there were many significant achievements, the traditional methods-based studies often concentrated on relatively small study regions. Therefore, their performance may need to improve when applied to complex areas with diverse building types. Additionally, they rely on handcrafted features guided by expert intervention, which is not always practical.

2.4.2 Deep Learning-Based Building Extraction

Over the past couple of decades, deep learning approaches have vastly improved the field of remote sensing. Because of their efficiency and accuracy, these approaches have outperformed traditional methods and revolutionized the way we analyze remote sensing data. The major breakthrough came when Krizhevsky in 2012 used convolutional neural networks (CNN) in the ImageNet classification [29]. CNNs combine feature extraction and classification in a single model, allowing adaptive feature learning from labeled training data. They offer improved generalization capabilities compared to conventional methods. Furthermore, they address the issue of explicit feature design in conventional approaches by enabling adaptive feature learning using labeled training data.

Researchers[91] explored patch-based CNN architectures for segmenting roads and buildings. These models integrate low-level features (e.g., asymmetry, compactness) during post-processing to enhance performance. However, considerable overlaps between patches can increase memory costs and reduce efficiency. For example, when a patch-based method processes an image in patches of 32×32 pixels, its memory cost is increased by thirty-two times compared to the conventional methods. Such approaches experience higher memory costs and substantially decrease processing efficiency when dealing with larger areas or patch sizes.

Early CNNs for building labeling showed promising results but were sensitive to hyperparameter settings. Current research employs increasingly complex convolutional neural network (CNN) architectures [92], such as multi-layer perceptrons (MLPs) with skip connections. In order to successfully integrate features from multiple scales, a multi-layer perceptron (MLP) structure with a skip connection modeled after the U-Net architecture is created in [93]. In [94], the U-Net is used as the basic model, incorporating a number of constraints. It limits the ability to compare feature map outputs from various scales with ground truth images from the same scales.

The Digital Surface Model serves as auxiliary data to the image data in [95]. Also, Generative Adversarial Networks and semantic segmentation models have significantly improved accuracy and efficiency in building footprint extraction. [96] proposes a matching-GAN architecture that introduces a matching network that replaces discriminators to modify the traditional GAN. This approach has been successfully applied to semantic segmentation tasks, including building footprint extraction.[97] improves the U-Net model using GANs and spatial and channel attention mechanisms. The goal is to generate prediction maps that are more discriminative and address the issue of uncertainty in segmentation.

A significant drawback of semantic segmentation methods is their struggle with instance-level distinctions. Buildings, even those classified in the same class, show variations in size, which presents difficulties for this task. For example, when several objects of the same class are clustered together, it is difficult to retrieve the number and boundary of the objects. As semantic segmentation networks are not very good at dealing with very small or big buildings, this limitation comes from that. Therefore, models that can produce instance-level segmentation have been proposed. For Example, Pedro et al. presented a two-step approach in [75]. Firstly, the network proposes segmentation candidates, and then it classifies them using an object detection algorithm. [98] enhanced U-Net, for instance segmentation. They added an output channel to predict borders between closely positioned or adjacent objects. Some networks, like [99], used instance segmentation networks for building boundary learning. In order to improve the accuracy of defining the boundaries of the buildings, [100]

also employed instance segmentation networks to refine the individual building boundaries. However, instance segmentation networks can be explored more, especially for closely situated or connected buildings in large-scale urban images. Graph convolution networks (GCN) combined with CNNs have become popular for building footprint extraction[42]. Graph convolution is a process that extends traditional convolution to graph-structured data. It enables nodes to aggregate information from their neighbors using a weighted transformation, followed by a non-linearity. A transformer-based approach that uses the combination of CNN and GCN called PolyBuilding [101] reduces vertex redundancy in the CNN–GCN paradigm. This approach focuses on learning building corner points from remote sensing images.

Boundary-based methods are used to directly learn building boundaries in a comprehensive manner, resulting in the generation of building footprints. Building footprint extraction faces several challenges, especially when dealing with buildings of varying sizes and complex shapes. Researchers have experimented with methods for extracting buildings from images at different scales. Most of the research [102, 103] is done on aggregating multiscale information. [104] focus on multiscale feature extraction, while others focus on designing dedicated architectures like Siamese [105] and multitask networks [106, 107]. Convolutional Neural Networks (CNNs) are efficient in their performance, yet they may lose certain intricate information due to their inherent spatial invariance properties. This frequently results in imprecise and inconsistent building boundaries. Therefore, in order to preserve geometrical details, enhanced approaches employ techniques such as signed distance transform [108], frame fields [109], and attraction field representation [110, 111, 112]. The modeling of attraction fields is highly effective for complex buildings and may accurately retain intricate structures in complicated buildings. The geometric characteristics of buildings are not easily apparent in imagery, which has a rather low spatial resolution. Graph models are able to capture the interactions between pixels and enhance their representation. Graph models have been integrated into end-to-end network learning frameworks [113, 114].

2.5 Building Boundary Regularization

Building boundary regularization is a central challenge in the geometric learning of polygon shapes. Although Convolutional Neural Networks (CNNs) perform well in segmentation, they often produce slightly mispredicted pixels along building boundaries. This results in the traced building polygons having irregular shapes. Researchers have prioritized regularizing building segmentations into precise, regular, and simple polygons.

2.5.1 Traditional Building Boundary Regularization

Before the advent of deep learning, the process of extracting building footprints relied more heavily on LiDAR point clouds or publicly available GIS data rather than images. Point clouds provide spatial locations that are more geometrically meaningful. Previous work like [31, 41] utilized point cloud data and employed Binary Space Partitioning (BSP) and Minimum Description Length (MDL) algorithms to produce and optimize building polygon shapes for the purpose of detecting footprints and regularizing building boundaries. These studies also addressed 3D building roof reconstruction. Similarly, from point clouds to extract regularized building polygons, [115] applied shape optimization algorithms. Despite the significant regularization effects, such conventional methods rely heavily on high human interventions and are specifically designed for buildings with pre-specified constraints. Therefore, they often fail to generalize to buildings with more complex shapes.

2.5.2 Deep Learning-Based Building Boundary Regularization

As conventional methods have many limitations, especially in terms of generalization, there has been a growing interest in deep learning-based building regularization techniques. These techniques aim to improve generalization capabilities and automate the process. Several segmentation models have been designed to focus on building boundaries and learning geometric information effectively. Researchers have explored multi-task learning strategies to

take full advantage of boundary information. For instance, [116] combined images and Digital Elevation Models in a SegNet model. This model integrates additional edge and boundary predictions from an FCN model by employing a multi-task learning approach. Similarly, [93, 117] employs the multi-task learning approach to incorporate an additional boundary loss during the training of FCN or U-Net models. Furthermore, [94] assert that they are capable of generating building boundaries with enhanced regularity.

Modern CNN architectures like Deep Active Ray Network (DARNet)[118] have integrated classic polygonal models, such as active contour models (ACM)[119], into their design. By doing so, these methods leverage richer geometric information, resulting in improved polygon contour predictions and the accurate delineation of building boundaries. Although there have been improvements, segmentation models continue to generate building polygons at the pixel level, which may require smoother and more regular boundaries. Along these lines, researchers have explored different novel approaches. The PSPNet [120] is a semantic segmentation network that identifies the initial building contour. Additionally, a modified PointNet is used to predict the coordinate offsets of the polygon vertices, resulting in the generation of regularized buildings. [121] proposed a fully convolutional neural network trained with adversarial and regularized losses for building boundary regularization in satellite images. [122] developed a generative adversarial network (GAN) that generates regularized segmentation masks. In the same year, [123] proposed a method that directly predicted regularized segmentations by combining edge features extracted using holistically nested Edge Detection with segmentation masks.

The earliest attempt at an end-to-end deep learning-based model for building regularization was made by [124]. They proposed a CNN that directly predicts the vertices of building polygons. However, this architecture suffers from a fixed prediction size. Despite its end-to-end structure, the network does not fully consider the simplicity and regularity of building polygons. There were a few more straightforward approaches to overcome this issue. PolyRNN [125] and PolyRNN++ [126] utilize recurrent neural networks (RNNs) to make successive predictions of polygon vertex locations, with the current prediction being influenced by the

prior ones. Nevertheless, [125, 126] are predominantly employed for semi-automatic annotation by utilizing bounding boxes but do not generate object detection outcomes. PolyMapper [127] integrates concepts from [125] and [77] to establish an integrated framework for detecting objects and predicting sequential polygon vertices. It has been utilized in extensive image collections to extract the outlines of buildings and roads. Nevertheless, PolyMapper does not possess the capability to predict objects with complex shapes. Furthermore, the computational cost of its convolutional Long Short-Term Memory [92] module is high. To address this problem, [128] introduced a comprehensive network that combines a segmentation CNN with a boundary regularization CNN. It investigates the utilization of graph convolutional networks (GCNs) to represent polygons in the form of graphs. Graphs are more effective and use supplementary geometric characteristics in contrast to RNN models.

Deep learning-based regularization methods have improved generalization. However, the resulting regularized building vectors may still need more simplicity and perfect regularity than conventional methods. Typically, building segmentation results are vectorized using conventional or deep learning-based regularization as a post-processing step. However, recent efforts aim to directly integrate the polygon regularization procedure into end-to-end deep learning-based models.

2.6 Open Datasets for Building Footprint Extraction

In the field of building footprint extraction and reconstruction using deep learning, the network’s performance depends on the datasets that provide diverse scenarios and challenges in building footprint extraction. These datasets serve as benchmarks for determining the performance of algorithms and models. High-resolution optical images, such as aerial photographs captured by satellites like QuickBird, IKONOS, and WorldView, have been and continue to be the main data sources for acquiring 2D and 3D building information. Images of high quality and high resolution provide excellent options for extracting information. Nevertheless, the effectiveness of building footprint extraction techniques may be significantly

hindered by insufficient image understanding, the presence of shadows, relief displacement, and occlusion effects in optical images lacking height data.

Multiple open datasets are accessible to the public and can be used to assess the performance of models for extracting building footprints. Typically, the datasets consist of aerial photographs and building footprints labeled with pixel or object-level annotations. These annotations indicate the coordinates of building polygons at the object level. This section provides an overview of the specifications of various cutting-edge datasets that are frequently utilized and openly available for the purpose of building footprint extraction and reconstruction tasks.

Wuhan University (WHU) [129] dataset comprises both aerial and satellite imagery for building footprint analysis. The aerial dataset is sourced from the website of New Zealand Land Information Services. The dataset consists of almost 220,000 individual structures that have been identified and extracted from high-resolution aerial images in Christchurch, New Zealand. These images have a spatial resolution of 0.075 meters and span an area of 450 square kilometers. The satellite imagery dataset comprises two distinct subgroups. For the first group, the data is gathered from multiple locations worldwide, such as Wuhan, Taiwan, Los Angeles, Ottawa, Cairo, Milan, Santiago, Cordoba, Venice, and New York. Various sources from the remote sensing domain are used to obtain this data. This includes ZY3, IKONOS, QuickBird, Worldview series etc. The resolutions of these resources range from 0.3 m to 2.5 m. The second sub-dataset of satellite images includes 6 adjacent photos spanning an area of 860 square kilometers in East Asia. This dataset contains a total of 34,085 buildings with a ground resolution of 2.7 meters. The entire dataset is delineated manually.

INRIA Aerial Image Labeling [130] dataset consists of aerial orthorectified color imagery. The imagery has a spatial resolution of 0.3 m and covers a total area of 810 km². The dataset is divided into two parts: 405 km² for training and 405 km² for testing. It includes imagery from ten different cities across the world. Five cities, namely Austin, Chicago, Kitsap, West Tyrol, and Vienna, are designated for training purposes and have a publicly available ground truth dataset of building footprints. The remaining portion of the dataset (Bellingham,

Bloomington, Innsbruck, San Francisco, East Tyrol) is exclusively utilized for evaluation purposes, with a hidden groundtruth. The dataset comprises 360 RGB tiles, with each tile having dimensions of 5000×5000 pixels. The collection was created by merging openly available images and official building footprints that are in the public domain. The images were obtained from many global GIS services, including the USGS National Map.

SpaceNet-2 [131] dataset provides a wide range of high-resolution satellite photos, making it a valuable resource for detecting building footprints across different geographical locations. The dataset includes more than 302,701 outlines of buildings derived from 3/8-band Worldview-3 satellite images with a spatial resolution of 0.3 meters. This dataset offers a comprehensive view, spanning 3011 km^2 and encompassing cities from four different continents: North America(Las Vegas), Europe(Paris), Asia (Shanghai), and Africa(Khartoum), providing a global perspective. The dataset includes 24,856 images, each measuring 650×650 pixels, ensuring a detailed view of the urban landscapes in these cities.

SpaceNet-4 [132] dataset was released as part of the SpaceNet Off-Nadir Building Detection Challenge. It comprises 27 high-resolution WorldView-2 satellite images of Atlanta, Georgia. These images vary in off-nadir angles from 7 to 54 degrees, offering diverse perspectives of the same urban area. The dataset includes over 120,000 building footprint annotations, and the complete collection of images was divided into tiles of $450m \times 450m$ with varying resolutions ranging from 0.46 to 1.67 meters. The images are categorized into three segments based on their off-nadir angles: Nadir (0-25 degrees), Off-nadir (26-40 degrees), and Very Off-nadir (40-55 degrees). This categorization aids in assessing the impact of viewing angles on building detection performance.

Massachusetts datasets spans approximately 340 km^2 of Boston, consisting of 151 RGB aerial images. Each image covers an area of 2.25 km^2 with a resolution of 100 cm per pixel, which is lower compared to some other datasets. The building annotations were derived from OpenStreetMap and converted into binary images, where building footprints are represented as pixel-wise labels. The availability of high-quality building footprint data is largely thanks to the City of Boston, which contributed its complete building footprint

dataset to OpenStreetMap. This dataset primarily covers urban and suburban areas and includes buildings of all sizes, from large structures to individual houses and even garages. The data was created using imagery provided by the state of Massachusetts, with all images standardized to a resolution of 1 meter per pixel. The building labels were derived from OpenStreetMap, ensuring comprehensive and detailed mapping.

Toronto city benchmark encompasses a much larger area of around 712.5 km^2 , covering the Greater Toronto Area (GTA). The dataset includes high-resolution aerial images with a 10 cm per pixel resolution. It provides detailed annotations, featuring roughly 400,000 building footprints and 8,439 km of road data. Unlike the Massachusetts dataset, the Toronto benchmark includes both pixel-wise labels and object-based vectorized building polygons. Additionally, it offers corresponding point cloud data, making it a rich resource for urban mapping and analysis. Beyond performance benchmarking, Toronto provides extensive geospatial data on building footprints. Through the city’s Open Data Portal, users can access datasets containing building outlines, which can be used to develop 3D city models. These resources play a crucial role in urban planning, environmental analysis, and a wide range of research and analytical projects.

ISPRS benchmark on urban object classification and 3D reconstruction [133] consists of aerial footage from Vaihingen and Potsdam. The imagery consists of 38 and 33 image patches, respectively, which have been taken from a large true orthophoto (TOP) mosaic. The TOP photos are 8-bit TIFF files, including three bands: near-infrared, red, and green. The DSM is a TIFF file that has a single band, which encodes DSM heights using 32-bit float values. For Vaihingen, the imagery has a ground sampling distance of 9 cm and dimensions of 6000×6000 pixels, while Potsdam images have a resolution of 5 cm and dimensions of 2500×2500 pixels. The Potsdam imagery includes an additional blue band, resulting in four bands (IRGB). The landcover categorization assignment involves labeling pixels with six distinct categories. This data is used to reconstruct 3D building roofs. The TOP and the DSM utilize the same grid, removing the necessity for further geocoding considerations during processing.

Bing Maps has recently released an open dataset containing building footprints worldwide. From 2014 to 2024, they detected approximately 1.3 billion buildings using imagery from sources such as Maxar, Airbus, and IGN France. The dataset includes building footprint polygon geometries represented in line-delimited GeoJSON format. The building extraction process involves two stages: Semantic Segmentation and Polygonization. The evaluation metrics comprise Intersection over Union and Dominant Angle Rotation Error. The metrics show that the quality of these matched building polygons is comparable to hand-digitized buildings in OpenStreetMap.

Available datasets for building footprint extraction tasks consist of aerial images in RGB or multispectral formats. Additionally, certain datasets may contain Remote Sensing data or point clouds as auxiliary information. The resolution of images varies from a few centimeters to one meter. There are two sorts of annotations for building footprints: pixel-wise labels and object-wise labels. Pixel-wise labels are binary images where building pixels are categorized as foreground, and all the nonbuilding pixels are categorized as background. They provide only semantic information. Object-wise labels represent individual building polygons. They include coordinates for the polygon vertices. These labels exhibit additional geometric attributes that are important for learning building shapes.

In many parts of the world, especially in remote or developing areas, LiDAR data is either too costly, difficult to obtain, or simply unavailable. This makes image-based networks a more practical solution for building boundary reconstruction. Satellite and aerial images, which are widely accessible, provide a more affordable alternative while still capturing important structural details like edges and shadows. These visual cues help deep learning models refine building outlines effectively. Unlike LiDAR, image-based methods can be easily scaled to cover large regions without requiring specialized equipment. Additionally, historical satellite imagery allows for long-term monitoring of urban growth and changes over time. With deep learning techniques like CNNs and GCNs, image-based approaches can achieve highly accurate building footprint extractions, making them a powerful choice in areas where LiDAR data is limited or unavailable.

Datasets with object-wise labels, such as the WHU and SpaceNet-2 building datasets, are highly recommended. These datasets cover different cities across continents. In this way, they provide a variety of building roof outlooks for experimentation work. Considering these facts, we have selected SpaceNet-2 and WHU datasets for our work.

2.7 Summary

In this chapter, we presented the literature review related to our work. We divided the research mainly into traditional and deep learning-based approaches. We started by initially discussing the research related to basic computer vision and deep learning. Then, we moved to object detection and segmentation models. We debated conventional segmentation and object detection methods and their comparison with deep learning-based methods. Similarly, we then discussed the conventional building footprint extraction and boundary regularization methods along with their deep learning counterparts. In the end, we discussed the open datasets available for building boundary extraction. All of the discussed research inspires our novel architectures.

Chapter 3

RO-POLYGCN: Region-Based Orthogonal Polygon Graph Convolution Network

This chapter presents RO-PolyGCN (Region-Based Orthogonal Polygon Graph Convolution Network), an innovative framework designed to address the challenges of building footprint reconstruction by integrating object detection, localization, and structured geometric learning into a unified pipeline. To achieve this, RO-PolyGCN leverages two core components: a backbone network and a graph neural network. The backbone network serves as a robust feature encoder, proficiently capturing spatial and semantic features for well-localized regions of interest. The graph neural network subsequently refines and processes these features to infer structured geometry and topology, resulting in regularized building boundary reconstruction. The upcoming sections detail the architecture, functionality, and contributions of each component within the network.

3.1 Overview

Feature extraction is a fundamental component of data analysis, typically achieved through statistical approaches. However, the advent of deep learning methods, especially neural networks, has significantly enhanced the quality of feature extraction. The choice of an appropriate backbone network plays a pivotal role in achieving this objective, influencing

both the effectiveness of the task and its complexity.

Numerous backbone architectures are commonly employed in neural networks to extract features and perform object detection and localization tasks. Our approach uses a fusion of Residual Networks (ResNet-101) and Feature Pyramid Network (FPN) to capture deep features across multiple scales effectively. In order to recognize and localize building objects, we employ a two-stage model that consists of a RPN and a localization layer that includes regression and bounding box classification layers. In addition, we utilize a RoIAlign layer to extract the accurately localized RoI features for every target building object. The localized features will be used as an input to the GCN.

Accurately representing polygonal shapes requires moving beyond conventional pixel-based representations to incorporate detailed geometric information. Graph-based models, which inherently include vertices and edges, provide a more precise framework for capturing the geometric properties of polygons. Graph Neural Networks (GNN)s further enhance this capability by performing convolution operations that facilitate the exchange of feature information among vertices, enabling advanced geometric learning. Leveraging these strengths, we integrate a graph convolutional network with an object detection backbone to exploit the geometric potential of graph models and GNNs. This approach culminates in developing RO-PolyGCN, a specialized model capable of detecting buildings in satellite images and predicting polygon vertex coordinates through implicit learning of their geometric shapes. By sequentially connecting vertices with straight edges, the model produces more regularized building footprint boundaries, enhancing both accuracy and representation quality.

3.2 Backbone Network

In this section, we present the backbone network, which serves as the foundation for feature encoding, object detection, and localization. The following subsections detail the techniques and methodologies employed to extract multi-scale features and accurately detect and localize objects, forming the basis for subsequent network components.

3.2.1 Feature Encoding

As shown in the figure, our feature encoding network is established into bottom-up and top-down pathways. In the bottom-up stage, we use ResNet [134], a Convolution Neural Network that is designed to process numerous convolution layers. The primary innovation brought by ResNet is its use of residual blocks. These blocks deal with solving vanishing gradient problems. ResNet focuses on fitting the residual (difference) between the desired and actual output. Its skip activations connect from one layer to another by skipping some intermediate layers. This enables efficient training of deep networks. The first ResNet architecture, ResNet-34, introduced shortcut connections to transform a plain network into its residual counterpart. The design of the plain network was influenced by VGG networks, such as VGG-16 and VGG-19, which use convolutional layers with 3×3 filters. However, unlike VGG networks, ResNets utilize fewer filters and have a lower computational complexity. ResNet-50 builds on the ResNet-34 architecture but incorporates a key modification: the introduction of a bottleneck block. This change was made to address concerns about training time for deeper networks. Instead of the 2-layer blocks used in ResNet-34, ResNet-50 replaces them with 3-layer bottleneck blocks, significantly improving accuracy compared to the 34-layer version. For larger ResNet models, such as ResNet-101 and ResNet-152, the architecture extends the depth by stacking more 3-layer bottleneck blocks. Despite the increased depth, the 152-layer ResNet retains relatively low complexity.

Our backbone network processes the satellite image through a five-stage ResNet-101 architecture (C1–C5). Each stage consists of multiple convolutional layers, with the final layer of each stage applying 2×2 pooling to downsample the feature map by half, progressively capturing high-level features while reducing spatial dimensions.

The top-down pathway as shown in Fig 3.1 of our backbone network utilizes the Feature Pyramid Network (FPN) [135]. FPN, in general, creates a feature pyramid with high-level semantics enabling object detection. It inputs images of different sizes and produces feature maps at various scales. In the same way, our network combines features from various scales generated by the bottom-up process. A 1×1 convolution kernel is applied to the current

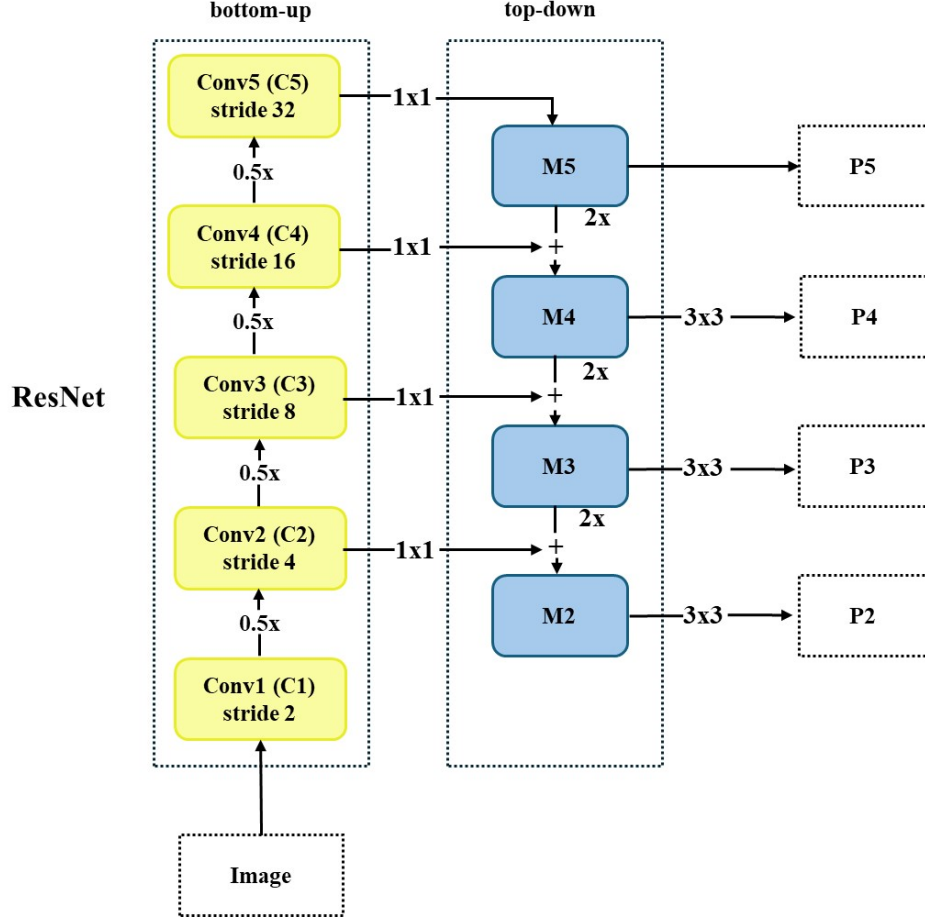


Figure 3.1 Feature Encoding[134]

feature map and combined element-wise with the upsampled feature map from the previous layer. To reduce the impact of upsampling distortion, a 3×3 convolution kernel is also learned, enabling the generation of feature maps across multiple scales. P5 (32×32), P4 (64×64), P3 (128×128), and P2 (256×256) from multiple levels of FPN from M2 to M5. These scale levels correspond to the hierarchical representation of the feature maps, as our datasets contain building objects of various sizes. So, detecting objects across different scales is crucial due to the large coverage area of our input satellite images. Utilizing the multi-scale feature maps generated by our encoding network enhances the ability to identify building objects in comparison to relying on feature maps at a single scale.

3.2.2 Object Detection

Our two-stage object detection network has three essential components: the Region Proposal Network (RPN), a localization layer, and the RoIAlign layer, which are seamlessly integrated with multi-scale feature maps. This integration facilitates the generation of localized RoI features. The RPN generates initial bounding boxes by utilizing multi-scale features and predefined anchor boxes. These proposed boxes are then used to extract corresponding features, resulting in cropped feature maps. Next, the RoI pooling technique is applied to these cropped features to produce RoI features. The localization layer, which is responsible for bounding box regression and classification, employs these RoI features to refine bounding box coordinates and assign class scores. Finally, the RoIAlign layer inputs the refined bounding boxes and multi-scale feature maps to produce precisely localized RoI features, ensuring accurate detection and localization. The following subsections will provide a comprehensive analysis of these three layers.

3.2.2.1 Region Proposal Network

As the name suggests, RPN generally generates proposals for regions where objects might be present. The algorithm employs a sliding window technique to detect and locate these locations on the feature maps. The system includes a classifier to determine the likelihood of an object's presence and a regressor to make precise modifications to the bounding box. The predefined anchor boxes and feature maps from multiple scales are given as input to the RPN. Anchor boxes of diverse sizes [32,64,128,256,512] and shapes [0,1,2] are created for distinct strides [4,8,16,32,64] on the input image. Within this particular context, the term "anchor shape" pertains to the proportion between the width and height of the box shown in Fig 3.2.

By adhering to this method, we generate multiple anchor boxes, with each one matching a distinct entry in the multi-scale feature maps. Within the region proposal network, each element of the feature maps has the task of predicting bounding box proposals for the

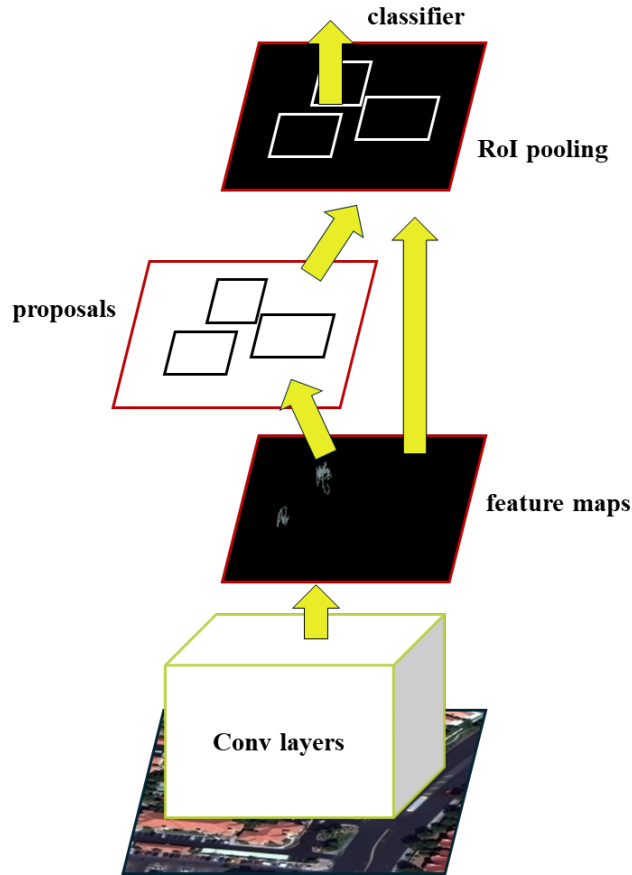


Figure 3.2 Region Proposal Network

corresponding anchor boxes through a box regression layer. Additionally, it determines class scores using a classification layer. These class scores, referred to as objectness scores, indicate the likelihood of an anchor box containing an object. After determining the threshold of the objectness score, the process of Non Maximum Suppression (NMS) is used to improve the accuracy of the bounding box recommendations. NMS is a key technique used after the region proposal step in object detection to filter out duplicate bounding boxes and keep only the most relevant ones. The concept behind NMS is fairly simple: it looks at all the proposed bounding boxes, compares their confidence scores, and removes those that significantly overlap with a higher-scoring box. This process helps improve detection accuracy by reducing false positives—cases where a bounding box is mistakenly placed in an area without an actual object. By keeping only the most confident and non-redundant bounding boxes, NMS ensures

that the final detections are more precise. Additionally, it enhances efficiency by cutting down the number of unnecessary calculations, making object detection algorithms faster and more streamlined.

In our case, NMS decreases the overall count of bounding box suggestions while preserving the ratio of positive to negative proposed boxes, often established as 3:1. Although the resulting proposals are closer to the actual object locations compared to the initial anchor boxes, further refinement is still required for precise localization.

3.2.2.2 Localization Layer

The box proposals generated by the RPN are utilized to extract multi-scale feature maps by cropping. The cropping method uses equation 3.1 to extract portions of the feature maps, taking into account the dimensions of the box proposals.

$$k = [k_0 + \log_2 \sqrt{wh}/224] \quad (3.1)$$

The width and height of the box proposal are denoted by w and h , respectively. The value of k_0 is set to 4, and k represents the scale level described in the previous section to crop the feature map. These cropped features exhibit diverse dimensions. For normalization, the cropped features are passed through the RoI pooling layer, resulting in RoI features of a uniform size, usually 14x14. In the cropping process, it is necessary to choose the right feature scale to align with the size of the identified items. This ensures that feature maps with higher resolutions are associated with smaller objects, while those with lower resolutions correlate to larger objects. Utilizing multi-scale feature maps allows us to extract more comprehensive and precise semantic information from various sizes. Like RPN, the classification and box regression layer is utilized by the localization layer to predict class scores and the coordinates of the bounding boxes based on the RoI characteristics. The class scores correspond to the predicted probability assigned to each class, namely building and nonbuilding.

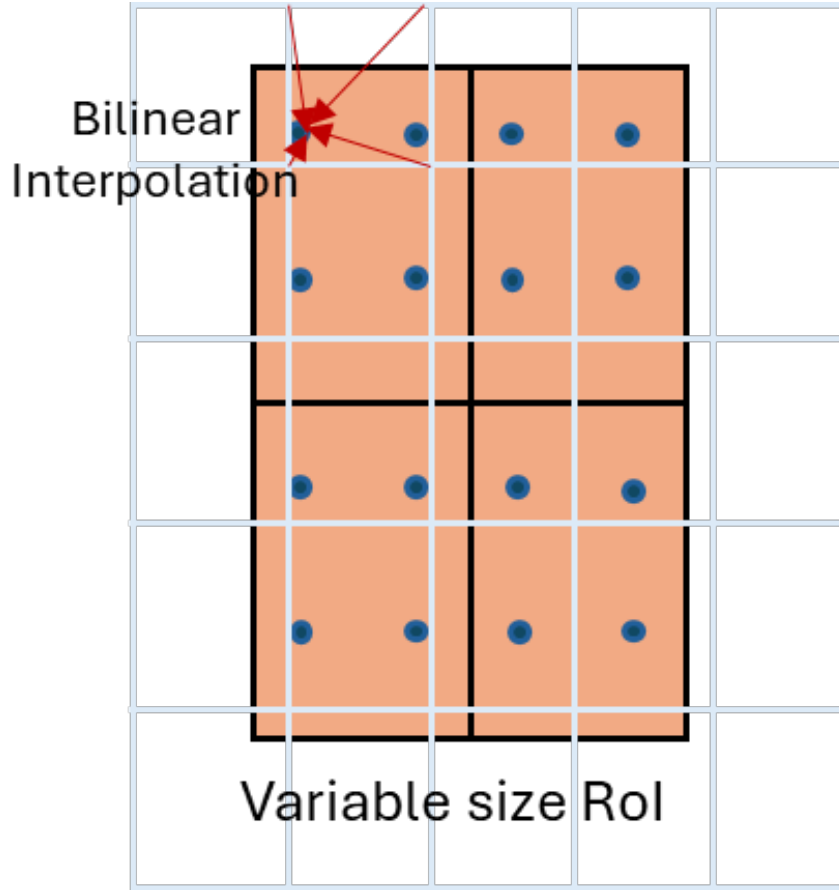


Figure 3.3 RoIAlign

3.2.2.3 RoIAlign

The RoIAlign layer addresses the limitations of traditional RoI Pooling. The box regression layer from the localization layer produces the RoI features, which typically have floating point coordinates. The traditional RoI Pooling layer quantizes the RoIs into fixed-size grids(14x14). This quantization can lead to misalignment of RoIs, resulting in information loss. Achieving accurate pixel-level segmentation involves more than just basic cropping and RoI pooling. It necessitates finely localized feature maps. The RoIAlign layer effectively resolves the problem of misalignment and significantly improves the precision of the RoI features. RoIAlign preserves the floating-point coordinates and uses differentiable bilinear interpolation to calculate the values at these coordinates, ensuring accurate and localized RoI

features. RoIAlign avoids potential rounding errors during cropping and pooling operations, unlike methods that round floating-point coordinates to integers. Accurate feature extraction enhances the process of identifying and precisely defining the location and boundaries of objects. RoIAlign is robust to variations in RoI sizes and aspect ratios.

3.3 Graph Convolution Network

GCN is a convolutional neural network applied directly to graphs to exploit their structural information. It solves the problem of classifying nodes in a graph. A typical Graph convolution network is shown in Fig 3.4

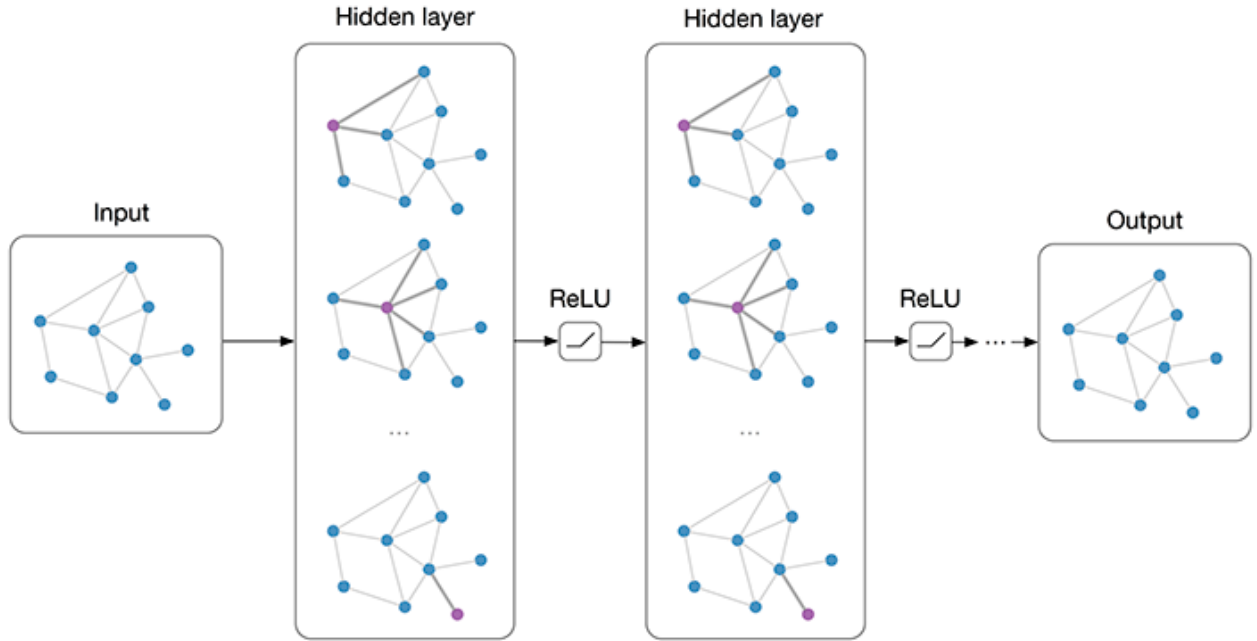


Figure 3.4 Typical GCN Structure

The fundamental distinction between CNNs and GCNs lies like the data they process and their approach to handling spatial relationships. CNNs are designed for structured grid-like data, such as images and time series. In contrast, GCNs operate on graph-structured data, where entities are represented as nodes connected by edges that define their relationships. In CNNs, convolutions involve fixed-size filters sliding over the structured input, such as a 3×3

kernel used in image processing. Conversely, GCNs perform graph convolutions by aggregating information from neighboring nodes based on the graph’s connectivity structure. While CNNs define a pixel’s neighborhood based on spatial adjacency, GCNs dynamically determine node neighbors based on the graph’s edges. CNNs are commonly used for applications like image classification, object detection, and image segmentation. In contrast, GCNs are applied in domains such as social network analysis, molecular property prediction, and building footprint refinement, including models like PolyGCN. Mathematically, CNNs perform standard convolution. Regarding computational efficiency, CNNs benefit from shared weights and optimized convolution operations, making them well-suited for parallel processing on GPUs. However, GCNs are computationally demanding due to the irregular nature of graph data, requiring operations involving adjacency matrices. In summary, CNNs excel at processing structured grid-based data like images. GCNs are specifically designed for graph-based tasks, making them well-suited for applications such as building footprint reconstruction from satellite imagery.

The main objective of the GCN in our architecture is to adjust the initial vertices of polygons, aligning them with the boundaries of the target building polygons. In order to accomplish this, we initially produce vertices according to a pre-defined sequence. For each vertex graph, features are then computed through interpolation using the enhanced feature representations. Afterward, the vertex offsets are predicted by inputting graph characteristics into multi-step and multi-layer graph convolutional networks. The ultimate regularized polygon is obtained by adding these predicted offsets to the initial vertex coordinates, enforcing orthogonality loss, and sequentially connecting the vertices to form the polygon.

Figure 3.5 presents the intricate architecture of our network RO-PolyGCN, a backbone network alongside a graph convolution network designed to deal with complex building boundary reconstruction tasks. As discussed, the backbone network identifies the target buildings and generates highly localized RoI information. Afterward, in these precisely localized areas, the GCN algorithm is used to acquire knowledge about the geometric shapes of building polygons.

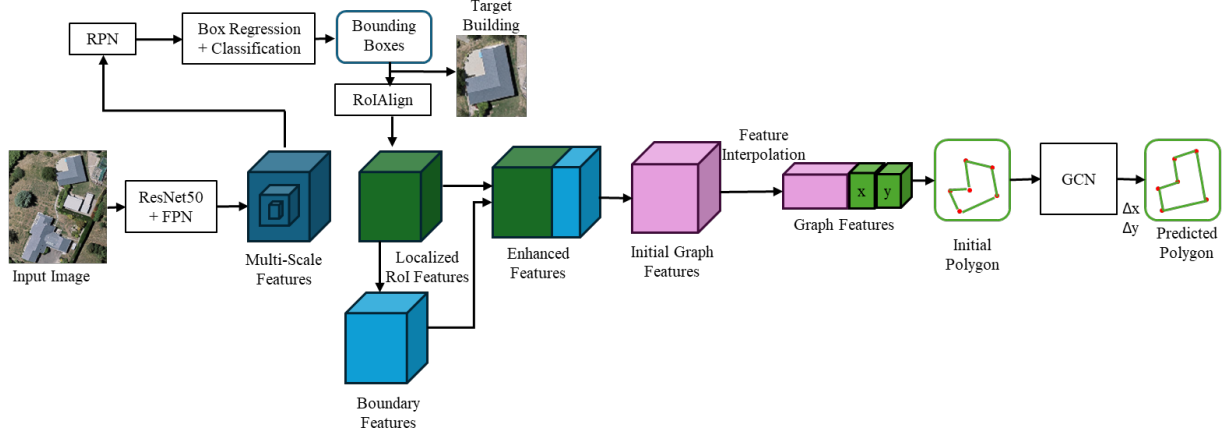


Figure 3.5 RO-PolyGCN Network Architecture

We train two fully connected layers with the explicit purpose of predicting polygon boundary masks. These masks include both edge and vertex masks for the target buildings. These masks are generated from the RoI feature maps obtained from the backbone network. The boundary prediction provides the likelihood of the building’s edges and vertices for each individual pixel.

3.3.1 Graph Initialization

The vertices of the target building polygon are initialized by allocating N points as the vertices of a regular polygon. The initial polygon is formed by connecting these initial vertices with a straight line. The number of vertices in each polygon is standardized and consistently matches the ground truth data. The vertices are arranged clockwise, ensuring a clearly defined sequence and preserving the polygon’s structure. In addition, the starting vertices are strategically placed in the feature map’s middle region. Let $V_i = [x_i, y_i]$ represent the coordinates of the i th vertex and $V = (v_1, v_2, \dots, v_N)$ designate the set of all polygon vertices, which serve as the initial nodes of our graph model. The graph edges, E , are generated by establishing connections between each node in V and its four adjacent nodes. In GCN, this connection enables the exchange of information and mutual influence across four adjacent nodes, leading to enhanced geometric coherence over a more considerable distance. The

initial graph is thus defined as $G = (V, E)$. Additionally, the node's current location (x_i, y_i) is concatenated with its node features in the following manner.

$$f_i = \text{concat}\{F(x_i, y_i), x_i, y_i\} \quad (3.2)$$

where f_i , the node V_i 's graph feature is a combination of feature maps and vertex position information. This feature will serve as the input for GCN.

3.3.2 Graph Implementation

Our methodology employs a multi-step framework to achieve precise and accurate polygon predictions. Initially, we commence by examining the initial features of the graph. The features are inputted into a GCN to calculate the initial offsets for the polygon vertices in the first round. According to the predicted offsets, we modify the vertices' positions. Next, we employ interpolation to derive new graph features from the improved characteristics. The interpolated features are inputted into another GCN. The second GCN generates a supplementary vertex offset. We perform this method repeatedly in the next few steps. Every step in the process enhances the precision of the vertex positions and increases the accuracy of the polygon prediction. Our iterative methodology gradually improves the placements of vertices, leading to a more precise prediction of polygons.

Our study employs a three-stage GCN, with each phase utilizing a multi-layer GCN. Let the v_i 's neighboring nodes in the graph are denoted by $N(v_i)$, and at layer l , the parameters of the network are defined as w_0^l, w_1^l . At this layer, we define the fundamental calculation for graph convolution for node v_i as:

$$f_i^{l+1} = w_0^l f_i^l + \sum_{v_j \in N(v_i)} w_1^l f_j^l \quad (3.3)$$

where f_i^l and f_i^{l+1} are the node v_i 's graph feature and the convolution operation's output at layer l respectively as shown in Fig 3.6.

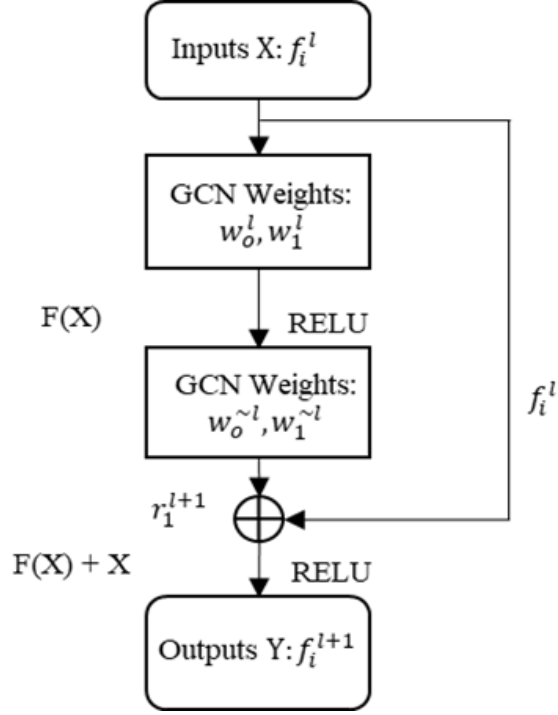


Figure 3.6 Residual Block

After that, we employ a residual block from ResNet. The residual block's calculations can be formulated as:

$$r_i^l = RELU(w_0^l f_i^l + \sum_{v_j \in N(v_i)} w_1^l f_j^l) \quad (3.4)$$

$$r_i^{l+1} = w_0^l r_i^l + \sum_{v_j \in N(v_i)} w_1^l r_j^l \quad (3.5)$$

$$f_i^{l+1} = RELU(r_i^{l+1} + f_i^l) \quad (3.6)$$

The equations 3.4 and 3.5 mentioned earlier are two-layer graph convolutions and are designed to produce the residual r_i^{l+1} . The convolution weights consist of four elements: w_0^l , w_1^l , w_0^l and w_1^l . In addition, we include the residual r_i^{l+1} and the identity of the input f_i^l . Next, we employ an activation layer called RELU, which is a Rectified Linear Unit, to acquire the ultimate output f_i^{l+1} .

3.4 Loss Design

We divide our network’s loss design into two branches. The first branch discusses the loss function for the backbone, which includes RPN loss, regression and classification loss, and localization loss. The second branch deals with losses for polygon vertex prediction and orthogonality. Additionally, we discuss the training strategies for optimizing the results of these loss functions.

Within the backbone network, losses for object detection are computed in two stages: the RPN and the subsequent box regression and classification steps. Both stages involve a multi-task training approach, incorporating two distinct loss components, box regression loss and classification score loss, to optimize the network’s performance in detecting and localizing objects.

Box Regression Loss: In accordance with Faster RCNN [136], instead of computing box coordinates for the input to box regression loss, we compute the box deltas, which indicate the required modifications to convert the initial anchor boxes into more precise proposals. The box deltas are defined by the following equations:

$$t_x = (x - x_a)w_a, t_y = (y - y_a)h_a \quad (3.7)$$

$$t_w = \log(w/w_a), t_h = \log(h/h_a) \quad (3.8)$$

$$t_x^* = (x^* - x_a)w_a, t_y^* = (y^* - y_a)h_a \quad (3.9)$$

$$t_w^* = \log(w^*/w_a), t_h^* = \log(h^*/h_a) \quad (3.10)$$

Here, x and y indicate the center point coordinates of the bounding box, while the variables h and w denote its height and width, respectively. Furthermore, x, x_a and x^* denote the predicted box, anchor box, and actual box, respectively (the same applies to y, w , and h). The bounding box loss values quantify the disparity between the real box parameters, specifically the (x, y) coordinates of the box position, width, and height, and the estimated

values. Our underlying network predicts the box deltas (t_x, t_y, t_w, t_h) , which represent the regression from a predetermined anchor box to a predicted box.

To evaluate the accuracy of these predictions, we use the Smooth L1 loss function. Smooth L1 loss is a hybrid between L1 loss and L2 loss, adjusting its behavior based on the magnitude of the error. When the error is small, it acts like L2 loss, creating a smooth, quadratic transition that prevents small differences from having too much impact. However, when the error becomes large, it behaves like L1 loss, maintaining robustness against extreme outliers. L1 loss, also known as mean absolute error, calculates the absolute difference between predicted and actual values. Smooth L1 loss introduces a threshold, often referred to as beta, to determine the transition between L1 and L2 behaviors. When the error is below this threshold, the loss function is less sensitive to minor fluctuations, making it more stable than the mean squared error. However, it switches to L1 loss for errors exceeding the threshold, making it more sensitive to large deviations. This balance helps improve model performance by handling both minor variations smoothly and more significant errors effectively.

Our Smooth L1 loss function compares the predicted box deltas to the ground truth regression values $(t_x^*, t_y^*, t_w^*, t_h^*)$

$$L_{\text{reg}}(t, t^*) = \sum_{i \in \{x, y, w, h\}} \text{Smooth}_{L1}(t_i - t_i^*) \quad (3.11)$$

Where a Smooth_{L1} loss [62] is defined as:

$$\text{Smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{otherwise.} \end{cases} \quad (3.12)$$

Classification Score Loss: The classification loss values are dependent upon the confidence score of the actual class. Therefore, the classification losses indicate the level of certainty the model has in its predictions of the class labels or, more precisely, how accurate the model is

in predicting the actual class.

Binary Cross-Entropy (BCE) loss is a way to measure how well a model predicts probabilities for binary classification tasks—where there are only two possible outcomes (e.g., cat vs. dog, spam vs. not spam, building vs. background). Think of it as a measure of how far off the model’s confidence is from the correct answer.

Imagine you’re guessing whether a coin flip will land on heads or tails, but instead of just saying "heads" or "tails," you give a confidence level (e.g., 90% heads, 10% tails). If the actual result is heads, then you’d want your confidence in heads to be as high as possible. If you were only 50% sure, that wouldn’t be a great prediction. Binary cross-entropy loss quantifies this idea mathematically—it gives a small loss when your confidence is correct and a large loss when your confidence is wrong.

Thus, in this case, we employ a binary cross-entropy loss to calculate the classification score loss, which is suitable for binary classification problems.

$$L_{cls}(p(y)) = -(y \log(p(y)) + (1 - y) \log(1 - p(y))) \quad (3.13)$$

where y is the predicted class label (0 or 1) and $p(y)$ is the class label’s probability score.

RPN Loss: During the training of the RPN, we assign negative and positive class labels to the anchor boxes, which are also known as objectness scores. Thus, we establish the RPN loss as:

$$L_{rpn}(p^{obj}, t^{rpn}) = \frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} L_{cls}(p_i^{obj}(y)) + \frac{1}{N_{box}} \sum_{i=1}^{N_{box}} L_{reg}(t_i^{rpn}, t_i^*) \quad (3.14)$$

where p^{obj} and t^{rpn} are the objectness scores and proposed box deltas predicted by RPN while N_{cls} is the total number of boxes, including both positive and negative boxes, after applying non-maximum suppression (NMS). N_{box} refers specifically to the number of positive boxes.

Localization Loss: We utilize box deltas with positive values for box regression since negative items, which do not have any object, do not contribute. While weight coefficients are

sometimes applied to balance the box regression and classification losses, prior studies [17] suggest that altering these coefficients has minimal effect on performance. Therefore, in our approach, we combine the two losses without introducing additional weighting coefficients. The localization layer classifies the boxes into two separate categories: building and non-building. For the bounding boxes, it gives highly accurate location predictions. Both objectives, classification and regression, are optimized by calculating a unified localization loss.

$$L_{\text{loc}}(p^{\text{class}}, t^{\text{loc}}) = \frac{1}{N_{\text{cls}}} \sum_{i=1}^{N_{\text{cls}}} L_{\text{cls}}(p_i^{\text{class}}(y)) + \frac{1}{N_{\text{box}}} \sum_{i=1}^{N_{\text{box}}} L_{\text{reg}}(t_i^{\text{loc}}, t_i^*) \quad (3.15)$$

The variables p^{class} and t^{loc} represent the class scores and predicted bounding box deltas, respectively. So, the total backbone loss is given by

$$L_{\text{backbone}} = L_{\text{rpn}} + L_{\text{loc}} \quad (3.16)$$

In addition to the backbone network losses, we have formulated loss functions specifically for polygon localization and orthogonality.

Polygon Localization Loss: The location of a vertex in a polygon is indicated by its coordinates $v(x, y)$ and the location of a polygon is represented by its N vertices: $p = v_i | i = 1, 2, \dots, N$. Let us consider that K polygons are extracted from our model, which we will label as $P = p_k | k = 1, 2, \dots, K$. These polygon vertices are also specified in clockwise order, similar to the ground truth vertices. The number of vertices in both the ground truth and predicted point sets is found to be the same. Therefore, the difference in position or distance between the ground truth p^{gt} and the predicted polygon p^{pre} can be calculated using the geometric $L1$ distance.

$$L_1(p^{\text{pre}}, p^{\text{gt}}) = \sum_{i=0}^N (|x_i^{\text{pre}} - x_i^{\text{gt}}| + |y_i^{\text{pre}} - y_i^{\text{gt}}|) \quad (3.17)$$

However, a challenge arises from the fact that the vertex correspondences between the

ground truth and predicted point sets are not aligned due to the yet-to-be-discovered starting vertices. To overcome this difficulty, we establish the originating vertex of the ground truth and identify the corresponding elements. In addition, we employ a thorough search to find the best matching starting vertex for the predicted point sets. Consequently, the predicted point sets will be expanded by utilizing each vertex as the initial point. Let us consider that there are N vertices in each polygon. By applying this method, N distinct sets of predicted points are formed for the polygon from the original point set. The point sets are arranged in a clockwise direction; however, they have N distinct starting vertices. The $L1$ distances are then calculated between all N extended predicted point sets and the groundtruth point sets. This leads to the calculation of N polygon distances. The polygon localization loss is calculated by identifying the shortest distance among the N distances of the polygons. Therefore, this method can also ascertain the best pairing of vertices. Given the set of K extracted polygons, there are K ground truth polygons, and the number of predicted polygons will be multiplied by $K \times N$. The mathematical representation of the loss function for all K polygons is as follows:

$$L_{\text{poly}}(p^{\text{pre}}, p^{\text{gt}}) = \frac{1}{K} \sum_{k=1}^K \min_{j \in \{0,1,\dots,N-1\}} (L_1(p_{k+j}^{\text{pre}}, p_k^{\text{gt}})) \quad (3.18)$$

Where L_{poly} an average polygon distance with vertex correspondences, denotes our polygon localization loss.

Orthogonality Loss: Geometric regularization in building footprint detection and reconstruction is a process aimed at ensuring that the predicted building shapes adhere to specific geometric constraints, often reflecting real-world building properties like straight edges, right angles, and symmetry. This can be achieved by designing loss functions or constraints that penalize deviations from these expected properties. In remote sensing and computer vision, where satellite or airborne imagery is used to identify and outline building structures, regularization techniques are crucial for improving the accuracy and reliability of model

predictions. Regularization helps in reducing noise around boundaries, ensuring that edges are smooth and continuous. This is especially important in satellite imagery, where irregularities due to shadows or occlusions might lead to jagged edges in unprocessed predictions. Many buildings have walls that meet at right angles. Incorporating orthogonalization techniques encourages predicted polygons to have right-angled corners where applicable, making results more consistent with actual urban architecture. By enforcing these geometric properties, regularization improves both the visual quality and quantitative performance of building footprint predictions, especially in metrics that reward accurate alignment with true shapes.

Our approach utilizes a multi-layer GCN to predict the vertex offsets of polygons, thus producing the building footprint. To address the problem of geometric orthogonality and to retain the right-angle characteristics and area as much as possible, a loss function is proposed to augment geometric regularity, particularly the orthogonality of building boundaries. The orthogonality loss can produce a marginal gain in object detection rate and shape regularity. The loss function is defined as:

$$L_{ortho} = \frac{1}{N} \sum_{j=1}^N L(P_j) \quad (3.19)$$

$$L(P) = \frac{1}{n} \sum_{i=1}^n \min_{\theta_{peak}} |\theta_i - \theta_{peak}| \quad (3.20)$$

where P_j is each polygon, and θ is each inner angle and θ_{peak} belongs to 0, 90, 180, 270. The inner angles are from polygons and do not consider predicted orientation vectors. N is the number of polygons, and n is the number of vertices (inner angles) in each polygon.

Overall, the total loss function for our whole network is:

$$L_{RO-PolyGCN} = L_{backbone} + L_{poly} + L_{ortho} \quad (3.21)$$

3.5 Training Strategy

This section examines training options for effectively managing numerous losses in our network. In order to achieve accurately localized regions, it is necessary to train both the RPN loss and the localization loss. In addition, we incur losses for predicting boundaries, polygon vertices, and orthogonality. Nevertheless, there are difficulties in training all the losses of RO-PolyGCN.

Most of our losses primarily consist of either entropy or smooth $L1$ loss, which can be effectively trained simultaneously. Nevertheless, using geometric point distance as a polygon localization loss may encounter multiple challenges when combined with other loss functions during training. Firstly, the geometric point distance lacks standardization and displays variability in scale. Moreover, achieving a reliable and consistent polygon localization loss is feasible when the target building areas are normalized. This often occurs when the RPN and localization losses reach a point of minimal size and stability. Before that, the generation of partial building polygons with reduced geometric significance will occur. This happens due to the need to fully optimize region localization in the backbone network. Due to the dependence of our GCN model on polygon geometric features, any inaccurate polygon localization can result in inadequate or incorrect training of the GCN. We employ polygon localization loss normalization to tackle these problems and carry out multistage training.

To standardize the polygon localization loss, we initially limit the coordinates of the polygon vertices to the range $[0,1]$. Subsequently, the distance of the polygon is divided by the total number of vertices of the polygon. In addition, a weight coefficient λ is incorporated into the polygon localization loss.

Our total loss function becomes:

$$L_{RO-PolyGCN} = L_{backbone} + \lambda \frac{1}{N} L_{poly} + \lambda L_{ortho} \quad (3.22)$$

Here, N is the number of vertices in each polygon. The polygon localization weight, denoted by λ , is a hyperparameter that is assigned a value from the set 0.25, 0.75, 1.25. This hyperparameter will undergo optimization during the training process. Regularizing the polygon localization loss to match the scale of other losses makes it possible to train balanced losses.

The training process is segmented into three main sections. In the initial stage, we primarily emphasize training the backbone network. In addition, we train the boundary prediction component using these losses while keeping the GCN polygon vertex prediction component fixed. During the backpropagation process, we will disable the GCN’s gradient update. We aim to create stable localized zones for the target buildings by exclusively training the backbone network. As we continue, we freeze the backbone network after a specific number of training iterations and incorporate the GCN component to enhance the accuracy of polygon prediction. Ultimately, we optimize the overall network by simultaneously training the GCN component and the backbone network. This multi-stage training strategy helps reduce the backbone network’s negative impact on the GCN polygon prediction.

3.6 Summary

This chapter introduces RO-PolyGCN, our proposed network for building footprint extraction. It begins with the backbone network, designed to extract multi-scale features for identifying and localizing target objects (buildings) from input images. The backbone follows a two-stage object detection pipeline, starting with a coarse representation and refining it into more detailed predictions. A multi-task loss function is employed during training to optimize the detection process. The chapter then explores the Graph Network and its regularized loss functions. The graph model, comprising vertices and edges, captures the geometric properties of polygons. A GCN is used to predict and refine the positions of polygon vertices iteratively, starting from an initial approximation and progressively enhancing accuracy. The vertices are arranged in a predefined sequence. Additionally, the chapter introduces a tailored loss

function for learning the precise positions of polygons and incorporates an orthogonal loss to ensure consistency in the predicted building shapes. The training process is further enhanced with a multi-stage approach for better convergence and performance.

Chapter 4

F AE-PolyGCN: Feature Augmentation-based Edge-preserved Polygon Graph Convolution Network

This chapter will introduce our second network, FAE-PolyGCN (Feature Augmentation-based Edge-preserved Polygon Graph Convolution Network). Initially, the network’s limitations from the previous chapter will be discussed. After that, our newly created Feature augmentation module and an added edge-preserved branch in the network will be explained. It will discuss how they both will help improve performance in multiple metrics.

4.1 Overview

Mask localization in instance segmentation, while effective for many applications, has several limitations that restrict its precision and applicability. One significant drawback is the lack of geometric precision, as masks provide pixel-level segmentation but often fail to capture the structural details required for tasks like building footprint reconstruction. This can result in imprecise and irregular boundaries, especially for objects with complex shapes or sharp edges. Moreover, the approaches that just focus on masks only struggle to maintain accuracy across varying object sizes, with smaller objects often poorly represented due to limited pixel resolution and larger objects losing internal details. Another limitation is that mask localization is computationally intensive, especially when generating dense masks for

high-resolution imagery, making it less efficient for large-scale applications. It also faces challenges in handling overlapping regions, where instances may merge or occlude each other, leading to ambiguous outputs.

Improving mask localization in instance segmentation requires a combination of advanced methodologies, architectural enhancements, and data augmentation techniques [137, 138]. One effective approach is to integrate boundary-aware modules into the segmentation network, which emphasize edge information and refine object contours for greater precision and accuracy. Incorporating multi-scale feature fusion can also improve the network’s ability to capture fine details across varying object sizes, ensuring better localization for both small and large instances. Using higher-resolution feature maps during the decoding process can help preserve finer details, particularly for intricate or irregularly shaped objects. Additionally, hybrid models that combine pixel-wise mask predictions with geometric representations, such as polygons or contours, can enhance localization accuracy by leveraging both raster and vector formats.

Our previous network architecture comprised a backbone network and a GCN, with the initial polygon predictions generated for GCN based on the features provided by the backbone. However, after extensive evaluation, we identified a critical limitation: the features provided by the backbone network for initializing the polygons were insufficient for enabling the GCN to refine polygon boundaries effectively. High-frequency details, essential for accurate polygon predictions, were not adequately captured. To address this issue, incorporating multi-scale features became a priority, as combining information across scales can significantly improve the precision and quality of final polygon predictions.

The backbone in our initial design was based on Mask R-CNN, a widely adopted and versatile model known for its performance across various tasks, including object detection, instance segmentation, and multi-object scene processing. Mask R-CNN employs pixel-level classification through FCNs to generate instance masks. However, it treats all pixels equally, overlooking critical boundary and shape information, which is essential for accurately delineating object contours. This limitation became evident in our application, where

precise boundary information is crucial for generating accurate polygons. As a result, the GCN struggled to refine polygons effectively, leading to coarse and overlapping boundaries. Additionally, the Mask R-CNN backbone struggled to identify small objects, as pixel-level features alone were insufficient for detailed localization. These shortcomings highlighted the need for an improved backbone capable of capturing both high-frequency and boundary-aware features to support the GCN and enhance polygon boundary predictions. We propose a new network architecture incorporating two novel modules to address these challenges.

The first is the Feature Augmentation Module (FAM), designed to enhance the feature maps by integrating predicted vertex and edge logits into the network. This module effectively combines high-frequency and low-frequency features, significantly improving the accuracy of polygon predictions. FAM is a key addition to the architecture, ensuring the GCN receives enriched features to refine polygon boundaries in a better way.

The second innovation is an Edge-Preserving Branch, which tackles the limitations of Mask R-CNN by explicitly incorporating object boundary information into the network. This branch includes a mask head specifically designed to preserve boundary details, enabling the simultaneous learning of object masks and boundaries. By emphasizing boundary features, this module enhances mask localization accuracy, ensuring the predicted masks align more precisely with object contours. This dual-module architecture demonstrates substantial improvements over existing methods, producing results with enhanced boundary precision and overall segmentation quality.

4.2 Feature Augmentation Module

We propose the Feature Augmentation Module (FAM) to enhance the GCN performance by providing enriched features, including better-defined corners and edges, beyond the semantic-based features derived from the backbone. The primary purpose of FAM is to augment the backbone’s output with predicted edge and vertex maps, explicitly guiding the GCN on where to place vertices and edges. This addresses a key issue in the RO-PolyGCN network,

where low-resolution semantic features were insufficient for small buildings and fine-grained details, resulting in suboptimal polygon predictions. By incorporating high-frequency details and fusing multi-scale features, FAM effectively preserves texture and color contrast, which is critical for accurate boundary and vertex predictions, especially in complex architectural structures.

Our experiments revealed that the previous network struggled to generate accurate edge and vertex maps because input features were optimized for semantic mask prediction, which lacked the high-frequency details necessary for precise edge or vertex prediction. Traditional methods often fail to handle such intricate details, leading to inaccuracies in object boundaries. To overcome this, FAM was designed to maximize the use of multi-resolution features from the backbone, ensuring minimal loss of high-frequency details. This capability allowed FAM to excel in preserving fine details, particularly in images with complex structures.

FAM is integrated into our current network to improve its precision in identifying corners and edges, which is beneficial for better building boundary reconstruction. By enriching feature maps with predicted vertex and edge logits, FAM significantly enhances the network’s ability to generate accurate polygon predictions. This module combines high-resolution boundary features with semantic context through a fusion block. The fusion block integrates multi-scale information, addressing limitations of previous low-resolution semantic features that failed to represent small buildings effectively. Inspired by [139], the module is implemented and evaluated in several design steps to understand and optimize the network’s performance. The backbone used as the baseline for these experiments is ResNet101 with FPN, providing a robust foundation for our research.

4.2.1 Resolution-Based Stream Separation

The feature augmentation module in the first step separates low-resolution (LR) and high-resolution (HR) feature streams shown in 4.1. The box and mask heads receive LR features from the backbone, while the Feature Augmentation Module, including the edge, vertex, and GCN heads, uses HR features. The fusion block, composed of three ResNet blocks,

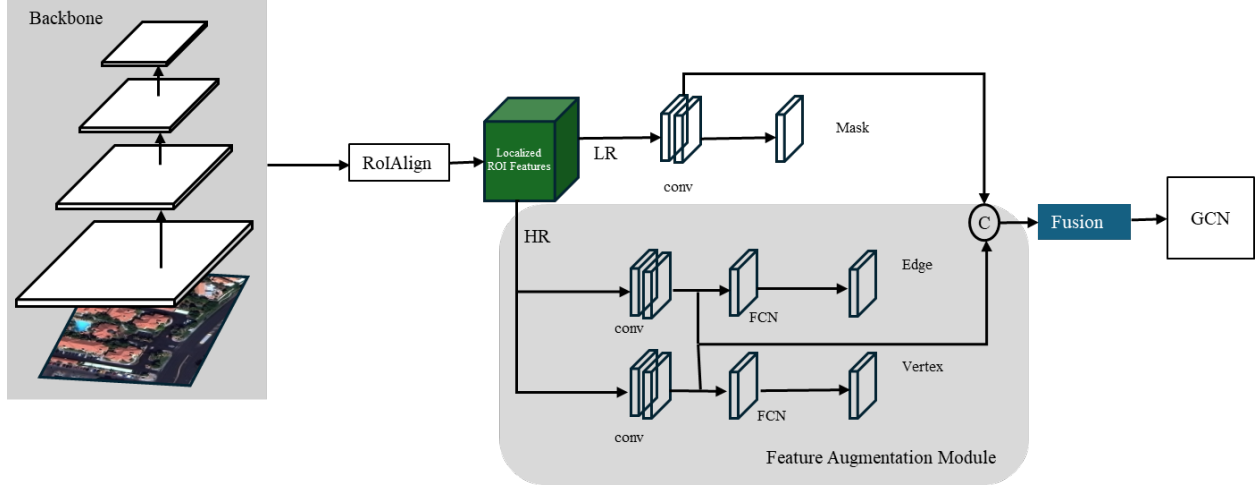


Figure 4.1 Resolution-Based Stream Separation

ensures the preservation of fine-grained details, particularly for small objects. This stream not only captures intricate edges but also processes broader semantic contexts, enabling better differentiation between objects and their backgrounds.

4.2.2 Image Gradient Concatenation

An image gradient refers to the alteration in intensity or hue in a specific direction within an image. The image gradient is an essential component in image processing. Here, we calculate the image gradient and concatenate it with the features from the heads to feed the Fusion block. The outputs are amalgamated in the fusion blocks. This integration combines the detailed edge information with the contextual understanding, resulting in refined segmentation masks that accurately delineate the building object’s overall shape. The architecture is illustrated in Fig 4.2.

As we observed in the results, most images predict masks, edges, and vertices remarkably well. As long as the masks are regularized and sufficiently simple, predicted polygons are as good, but when the masks suffer from rounded corners or curvy edges, polygons deteriorate, too. We need high-frequency details, so fusing multiscale features can improve the predictions.

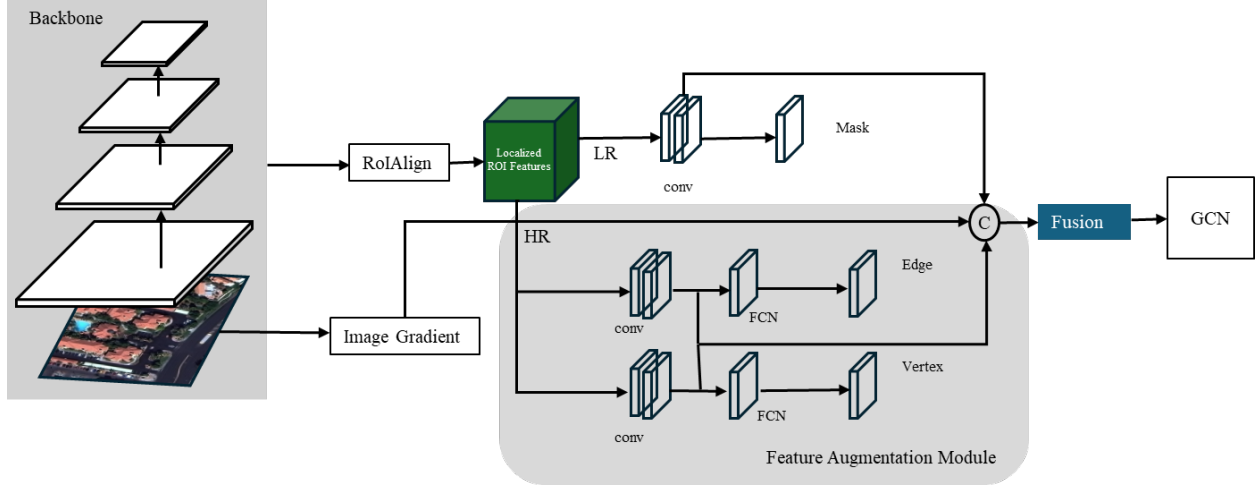


Figure 4.2 Feature Augmentation Module with Image Gradient Concatenated

4.3 Edge-Preserving Boundary Mask Head

In our previous design, Mask R-CNN served as the baseline for instance segmentation. However, its segmentation results limited the effectiveness of the GCN, a critical component that captures spatial relationships between objects in an image. We explored enhancements to the baseline network and proposed architectural modifications to address this limitation. While the FAM, discussed earlier, improved the initial features by incorporating edge and vertex maps with semantic masks, further enhancement of the semantic mask predictions was necessary. This revised module can be explained by Fig 4.3

Our experiments demonstrated that integrating an edge-preserved mask branch into the backbone significantly improved performance reduced computational complexity, and aligned better with our network’s requirements. This revised architecture outperformed the original Mask R-CNN baseline in semantic segmentation tasks, achieving more accurate and consistent results.

Mask R-CNN, which relies on pixel-level predictions derived from local information, aims to create a translation-invariant predictor for instance segmentation. However, its reliance on local features often results in coarse and indistinct masks, overlooking global shape information and potentially including false positives. Detailed analyses of raw mask

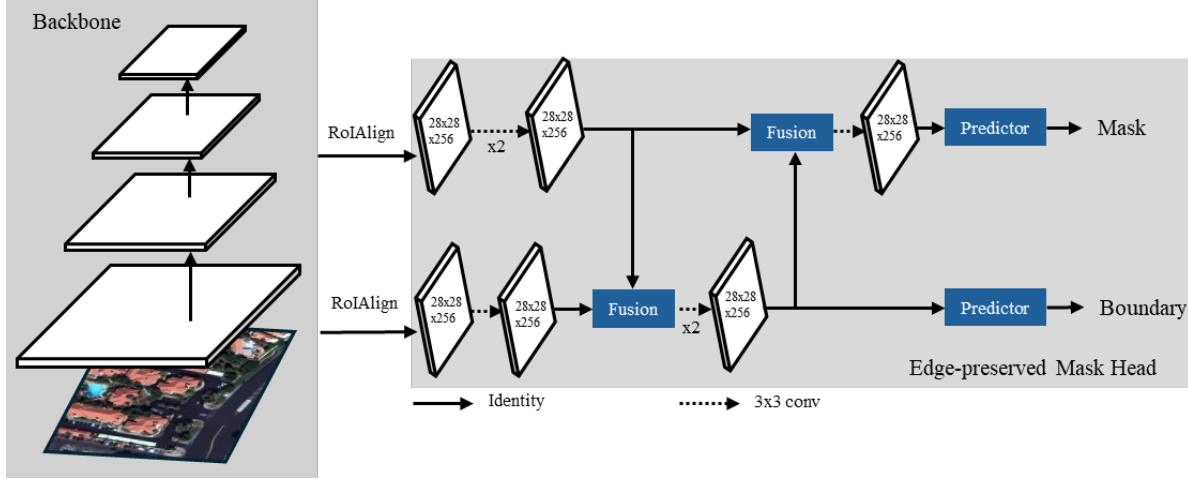


Figure 4.3 Edge Preserved Mask Branch [140]

predictions using ResNet-50 and FPN, as well as ResNet-101 and FPN, revealed a critical limitation: the inability of the baseline to incorporate shape and boundary information effectively. To address this, we developed a novel network that integrates instance boundary information directly into the segmentation process, unifying instance mask and boundary prediction. This approach not only enhances mask precision but also establishes a new framework for segmentation research.

4.4 Network Detail

The proposed network combines the edge-preserved mask branch with the FAM to explicitly predict instance-level boundaries, providing critical shape and localization information for improved mask accuracy. Unlike traditional semantic segmentation methods that treat boundaries and masks as separate entities, our network incorporates augmentation blocks to learn these properties simultaneously. The enhanced mask head, referred to as the edge-preserving mask head, operates in two stages.

. In the first stage, features are extracted using the combination of RESNet-101 and FPN for both the RPN and second-stage heads. The second stage performs classification, bounding box regression, and precise mask prediction through RoIAlign, which ensures

accurate alignment of features with the proposed RoIs. By leveraging boundary features, the network enriches shape and spatial information, enhancing mask predictions.

This integrated approach significantly improves the learning process for both boundary and mask features. The strong interconnection between boundaries and masks allows boundary features to refine mask predictions with additional shape and spatial context. The resulting network achieves precise localization and segmentation, effectively capturing boundaries while maintaining computational efficiency, making it suitable for large-scale applications. The Fig 4.4 shows the complete network architecture diagram.

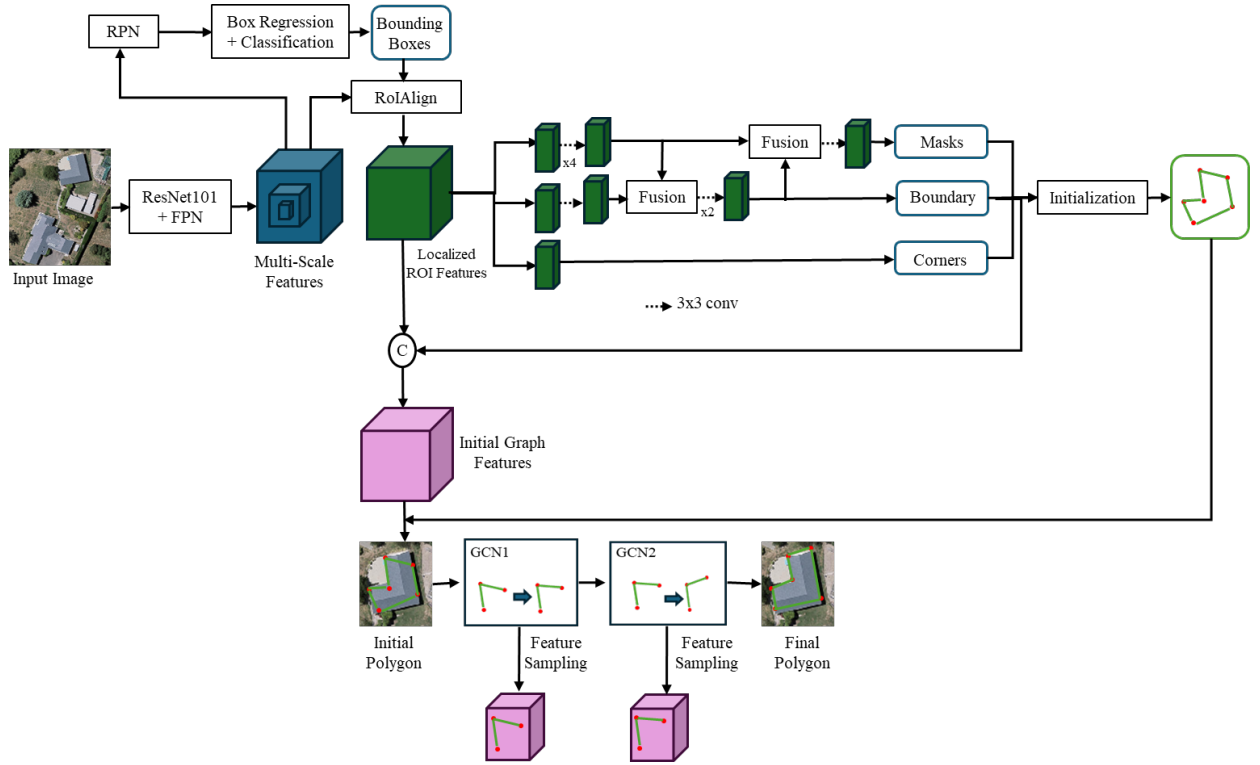


Figure 4.4 FAE PolyGCN Network Architecture

The integration of features from the FAM and the edge-preserving mask head plays a pivotal role in enhancing the initialization of polygons within the GCN. The enriched features from FAM, which combine high-frequency details with multi-resolution semantic context, provide the GCN with precise vertex and edge predictions, ensuring better alignment with object boundaries. Simultaneously, the edge-preserving mask head contributes accurate

boundary features and refined mask predictions, supplying the GCN with a robust foundation for polygon initialization. These improvements enable the GCN to generate more accurate and geometrically consistent polygons during the refinement process. Furthermore, the addition of orthogonality loss helps enforce the regularity of building footprints by promoting orthogonal relationships between polygon edges. This loss function further reduces irregularities, ensuring that the final building polygons exhibit clean, sharp corners and well-aligned edges, which are critical for applications requiring precise building footprint reconstruction. The combination of these advancements significantly enhances the overall performance and accuracy of the GCN, making it more effective in handling complex building geometries

4.5 Loss Design

Our network’s loss design is divided into two distinct branches, similar to RO-PolyGCN. The first branch focuses on enhancing object detection and boundary prediction. It incorporates multiple loss components, including the Region Proposal Network (RPN) loss and localization loss, along with an additional boundary loss specifically designed to refine boundary predictions. The second branch is dedicated to polygon refinement, incorporating losses for polygon vertex prediction and orthogonality loss to ensure the generation of precise and regularized building polygons. So, we define the loss function for localization and rpn in a similar way as described in Chapter 3.

$$L_{\text{rpn}}(p^{\text{obj}}, t^{\text{rpn}}) = \frac{1}{N_{\text{cls}}} \sum_{i=1}^{N_{\text{cls}}} L_{\text{cls}}(p_i^{\text{obj}}(y)) + \frac{1}{N_{\text{box}}} \sum_{i=1}^{N_{\text{box}}} L_{\text{reg}}(t_i^{\text{rpn}}, t_i^*) \quad (4.1)$$

$$L_{\text{cls}}(p(y)) = -(y \log(p(y)) + (1 - y) \log(1 - p(y))) \quad (4.2)$$

Boundary prediction presents unique challenges, particularly due to the issue of class imbalance caused by the disproportionately smaller number of boundary pixels compared to the background. Most traditional edge or boundary detection methods address this

imbalance using weighted binary cross-entropy loss, assigning higher weights to boundary pixels. However, this approach often results in coarse and indistinct boundaries, limiting its effectiveness in scenarios requiring high precision.

To overcome these challenges, we employ a combination of Dice loss and binary cross-entropy loss, as suggested in [140], to optimize boundary learning. Dice Loss is a crucial metric for evaluating how well a machine learning model performs in image segmentation tasks. It measures how similar the predicted segmentation is to the actual ground truth, helping to fine-tune the model by minimizing this loss. One of the most significant advantages of Dice Loss is its effectiveness in handling imbalanced datasets, where one class (such as the background) is much larger than the other (such as the object of interest). In these cases, a model might appear to have high accuracy simply by predicting the dominant class most of the time. However, Dice Loss ensures that both the foreground and background are considered relatively, preventing the model from favoring one class disproportionately. This makes Dice Loss especially valuable in computer vision applications, where accurate segmentation is key. By optimizing a model to minimize Dice Loss, we can improve its ability to detect and segment objects correctly, even when dealing with challenging datasets. Dice loss is particularly advantageous because it quantifies the overlap between predicted and ground truth boundaries, focusing on regions of disagreement. Unlike cross-entropy-based losses, Dice loss is unaffected by the foreground-to-background pixel ratio, effectively mitigating the class imbalance issue. This combination ensures the network learns to predict accurate and sharp boundaries while maintaining robust performance across diverse datasets and scenarios. The formulation of our boundary loss L_b is as follows.

$$L_b(p^{pre}, p^{gt}) = L_{DICE}(p^{pre}, p^{gt}) + \lambda L_{BCE}(p^{pre}, p^{gt}) \quad (4.3)$$

where dice loss is defined as

$$L_{Dice}(p^{pre}, p^{gt}) = 1 - \frac{2p^{pre}p^{gt} + 1}{p^{pre} + p^{gt} + 1} \quad (4.4)$$

so the overall backbone loss, for FAE-PolyGCN network becomes

$$L_{backbone} = L_{rpn} + L_{loc} + L_b \quad (4.5)$$

The integration of boundary loss into the first branch enhances the accuracy of edge detection, providing the GCN with improved initialization features. Meanwhile, the orthogonality loss in the second branch enforces geometric regularity, promoting well-aligned edges and sharp corners in the final polygon predictions. Together, these carefully designed loss components contribute to a more precise and reliable building footprint extraction process, addressing both semantic and geometric challenges in urban object reconstruction.

Our total loss function becomes:

$$L_{FAE-PolyGCN} = L_{backbone} + \lambda \frac{1}{N} L_{poly} + \lambda L_{ortho} \quad (4.6)$$

4.6 Summary

In this chapter, we tackled the limitations of existing instance segmentation networks, specifically Mask R-CNN, which often fails to capture fine boundary details and struggles with smaller objects. To address these challenges, we proposed an enhanced architecture incorporating a Feature Augmentation Module (FAM) and an edge-preserving boundary mask head. FAM enriches feature maps with high-frequency details, combining low- and high-resolution features to improve polygon prediction accuracy. We significantly enhance the GCN’s ability to detect object boundaries and shapes by integrating predicted vertex and edge logits. We also introduced an image gradient concatenation step and a high-resolution mask addition to refine features further and improve GCN initializations. The edge-preserving boundary mask head simultaneously learns object boundaries and masks, which allows the network to achieve precise localization and clearer mask delineations. Our novel design advanced over the traditional Mask R-CNN, demonstrating improved precision and reduced

computational complexity, making it more robust for real-world applications. The result across various metrics shows a substantial improvement compared to our previous model, producing projected masks that align more precisely with the boundaries of objects.

Chapter 5

AT-PolyGCN: Graph-Based Polygonal Segmentation of Building Footprints Using Attraction Field Maps

This chapter introduces our third network, AT-PolyGCN: a graph-based approach for polygonal segmentation of building footprints using Attraction Field Maps. We will begin by examining the limitations of the network discussed in the previous chapter. Following this, we will present the newly incorporated Attraction Field Maps and an innovative initialization technique for the graph convolution network, detailing how both contribute to performance improvements across multiple metrics.

5.1 Overview

The current backbone architecture in our network lacks the accuracy needed for optimal segmentation, resulting in initial polygons that are insufficiently precise for the GCN, thus limiting overall performance. Existing methods for building footprint extraction, particularly for large buildings, often yield incomplete or inaccurate representations, struggling with very large, complex, or closely adjacent buildings. Additionally, the feature augmentation module is inadequate for capturing these intricate details, as missed corners cannot be recovered in the polygonal representation. The edge-preserving mask branch offers some improvement in segmentation but has limitations, enhancing results only to a certain extent.

In this network, we will integrate Attraction Field Maps to greatly enhance edge detection, along with modifications to the graph initialization process to produce more accurate initial polygons. Together, these enhancements will enable more precise and comprehensive building footprint extraction.

Edge detection in remote sensing identifies sharp intensity changes in images, marking boundaries between features like urban areas, forests, or water bodies, enhancing the visibility of elements such as buildings and roads. Traditional techniques like gradient-based Sobel or Canny detectors and Laplacian-based methods locate edges by analyzing intensity changes. Line segment detection, meanwhile, focuses on linear features, such as roads and building outlines, using techniques like the Hough Transform or Line Segment Detector (LSD) to group pixels into lines. These methods enable feature extraction and change detection, aiding in applications like urban planning, environmental monitoring, and disaster response by highlighting significant structures in complex images. In region partition map, each pixel in the image is assigned to exactly one line segment, forming a region. This clear assignment eliminates ambiguity that often arises in traditional methods and ensures that all pixels contribute to a specific segment. A region partition map divides an image into distinct, homogeneous regions based on characteristics like color, texture, or intensity, making image analysis easier. Segmentation techniques such as thresholding, clustering, and region-growing help define these partitions by grouping similar pixels. In remote sensing, region partition maps aid in tasks like separating urban and non-urban areas or land and water bodies. For building boundary detection, the map segments buildings as consistent regions, allowing edge detection to refine their boundaries. This approach improves accuracy, scales to large datasets, and supports automated analysis, benefiting applications in urban planning, disaster response, and environmental monitoring. Challenges include handling complex scenes and resolution dependency, as high-resolution images yield better results.

Xu. et al., [110] introduces Attraction Field Maps to enhance building boundary detection in satellite imagery. The Attraction Field Map supervises boundary refinement, with a stronger attraction for closer pixels, tightening the predicted polygons. Together with region

partition maps, these maps support accurate polygonal mapping by combining structural awareness.

Towards this goal, we are inspired by the representation of edges with AFM [110, 111, 112] at the middle-level description. The attraction field map encodes the projection vector for each pixel relative to its associated line segment. This means that each pixel within a region is represented by a vector pointing towards the line segment, which aids in recovering line segments from the input image. By eliminating local ambiguity inherent in gradient magnitude-based methods, the attraction field representation allows for more accurate detection of line segments, particularly in low-contrast areas where traditional methods struggle. The approach helps to manage the imbalance between line and non-line classifications, which is a common challenge in line segment detection tasks. This is achieved by focusing on the attraction fields rather than relying solely on detecting edges. The method demonstrates scale invariance, meaning it can effectively detect line segments across varying scales, which is validated through precision and recall evaluations.

5.2 Attraction Field Maps

An attraction field map is a representation of the spatial distribution of attractive or repulsive forces within an image. It can be generated based on various image features, such as gradients, corners, or object boundaries. By incorporating attraction field maps into edge-guided feature enhancement, we can potentially improve feature extraction by emphasizing regions of high attraction. The primary purpose of an AFM is to assist in accurately detecting and delineating object boundaries in an image. It can be particularly useful in challenging scenarios where traditional edge detection methods might struggle, such as in the presence of noise, occlusions, or weak gradients. Attraction fields can be used to define regions of interest for building segmentation. By modeling buildings as attractive objects, attraction fields can aid in building shape reconstruction. Attraction fields can be used for object-based image analysis, where buildings are considered as individual objects. Concatenate attraction field maps with feature

maps at different neural network layers. This approach allows the network to learn to combine the information from both sources.

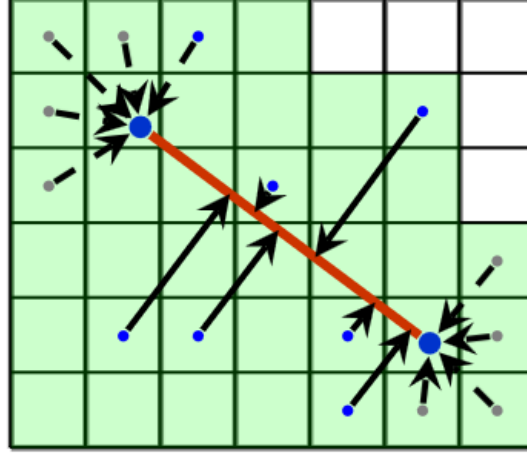


Figure 5.1 AFM Visualization[111]

Incorporating attraction field information into the loss function encourages the network to focus on regions of high attraction. This can be achieved by adding a term to the loss function that penalizes deviations from the attraction field map. The quality of the attraction field map is crucial for the success of this approach. Effective methods for generating accurate and informative attraction field maps need to be developed.

To visualize attraction field maps for building boundaries, we start with images and their ground truth annotations in the desired format. Then, by using the polygon coordinates in the segmentation data, we create a binary mask for each building, marking pixels inside the building boundary as 1 and outside as 0. A distance transform is applied to this mask to calculate each pixel's Euclidean distance to the nearest boundary such that boundary pixels have a distance of zero. For each pixel inside the building, compute an attraction field vector that points toward the nearest boundary with a magnitude inversely proportional to the distance. Finally, we use different plots to visualize these vectors as arrows overlaid on the original image, indicating the direction and strength of attraction toward building edges as

displayed in Fig 5.1.

5.2.1 Impact of Attraction Field Map

Attraction Field Maps (AFMs) bring major advantages to building boundary detection in remote sensing by enhancing feature representation, localization, and occlusion handling. AFMs generate a vector field that guides pixels toward their nearest boundary, enabling precise edge localization and detailed boundary delineation, even in cases of weak edges or shadowed areas. By refining complex structures and connecting disjoint segments, AFMs ensure continuity and clarity in edge detection. They are effective in segmentation by separating buildings from their surroundings and distinguishing closely packed structures, which is particularly valuable in dense urban areas. AFMs are also noise-robust, focusing on significant boundaries to reduce noise impact, which is essential for low-contrast images.

Furthermore, AFMs improve generalization by preserving typical building geometries, such as straight lines and right angles, across varying building types, and they help reduce overfitting by concentrating on structural information rather than image-specific textures or noise. Additionally, AFMs accelerate network training by focusing on key areas, boosting accuracy, generalization, and robustness for large-scale urban applications like urban planning and disaster management. Beyond remote sensing, AFMs have applications in refining object boundaries, assisting in segmentation, feature extraction, and object recognition. In medical imaging, AFMs can aid in accurately outlining structures like tumors or organs in complex images, showcasing their versatility in scenarios requiring precise boundary detection.

5.2.2 Attraction Field Generation

The Attraction Field Map (AFM) is a vector field that points each pixel toward its nearest point on the building boundary. This field is designed to "attract" each pixel to its closest boundary, thereby improving boundary refinement. Therefore, the AFM is calculated based on the Euclidean distance between each pixel in the image and its closest point on the building boundary.

For each pixel $p = (x, y)$ in the image, we first compute the Euclidean distance to the nearest boundary point $b = (x_b, y_b)$.

$$D(P) = \min_{b \in \text{boundary}} \|p - b\|^2 \quad (5.1)$$

where $\|p - b\|^2$ represents the Euclidean distance between p and b .

For each pixel p , we then calculate the vector $a(p)$ that points from p to its closest boundary point b . If p is a point within the building region, $a(p)$ is defined as:

$$a(p) = (x_b - x, y_b - y) \quad (5.2)$$

To standardize, vectors may be normalized to unit length, indicating direction without magnitude, ensuring consistent directional guidance. Normalization ensures the AFM captures only the direction toward the boundary rather than the magnitude.

$$a(p) = \frac{(x_b - x, y_b - y)}{\|(x_b - x, y_b - y)\|^2} \quad (5.3)$$

In our experiments, we have used the AFMs without normalization. Also, optional post-processing, such as smoothing, can further refine the AFM, reducing noise and enhancing boundary guidance. This process results in a vector field that effectively "pulls" pixels towards building boundaries, aiding in accurate boundary refinement.

5.3 Network Details

GCNs require an input graph, which can either be fixed, as seen with the arbitrary circle based on semantic features used in RO-PolyGCN, or dynamically based on prior predictions, such as masks and corners. The latter approach optimally leverages available information, initializing the GCN with a more accurate estimation of the building shape and allowing it to focus on fine-tuning details.

The Region Proposal Network (RPN) generates a set of RoI, from which we localize and crop bounding boxes from the backbone feature maps. However, to apply convolution layers on these RoIs, we need to resize them to a fixed size. This resizing can lead to a loss of detail, particularly for large objects. Additionally, resampling can produce inaccurate predictions with reduced confidence, especially around edges and corners, which then misguides the GCN module. This resampling issue affects not only predictions but also ground truth quality. Since our GCN relies on feature maps to accurately position vertices, this degradation can lead to confusion and create less precise, curvy polygons.

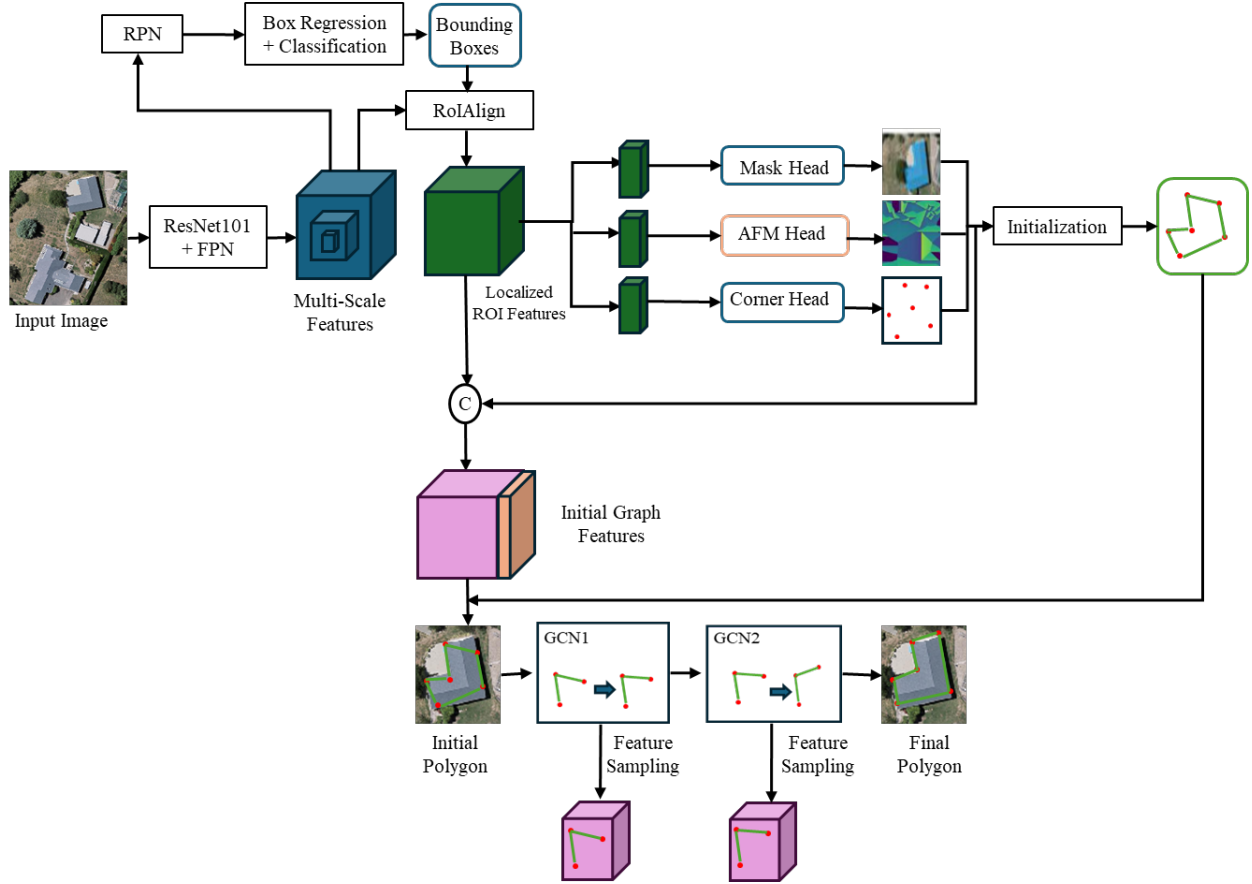


Figure 5.2 AT-PolyGCN Network Architecture

Our initialization module for GCN constructs this initial polygon by using predicted masks and corners as input. First, the contour of the mask is extracted via the marching squares algorithm [141] and simplified with the procedure by Douglas-Peucker Algorithm [142].

The Douglas-Peucker algorithm is a line simplification method that reduces the number of points in a curve while preserving its overall shape. It works by recursively removing points that deviate the least from an ideal straight line connecting key points. The process starts with the two endpoints and finds the point farthest from the line between them. If the distance exceeds a given threshold, the point is kept, and the curve is split into two segments for further simplification. This continues until all remaining points are within the tolerance, resulting in a simplified yet accurate representation of the original shape.

Next, a polygon is formed by matching predicted corners to their closest points on the contour, recovering any potentially missing corners from the contour points to ensure a complete shape. This results in a more geometrically accurate initial graph. Finally, 16 vertices are uniformly resampled from the graph to match the vertex count in the ground truth.

5.4 Loss Design

Our loss design and training strategy builds upon the loss functions used in RO-PolyGCN with the addition of AFM loss in this network. The incorporation of AFM loss enhances performance by improving the quality and accuracy of the resulting building polygons.

The loss function for AFM is typically designed to minimize the difference between the predicted attraction vectors and the ground truth attraction vectors. The L1 Loss is often used to calculate this difference, as it penalizes deviations in both direction and magnitude of the vectors. The loss function is defined as follows:

$$L_{\text{AFM}} = \frac{1}{N} \sum_{p \in P} \|\tilde{v} - v_p\|^2 \quad (5.4)$$

where N is the total number of pixels in the image, P represents the set of all pixels in the image. $\tilde{v}$ is the predicted attraction vector at pixel p . v_p is the ground truth attraction vector at pixel p . $\|\cdot\|$ denotes the Euclidean norm, ensuring the loss captures both the direction

and magnitude difference. The sum is taken over all pixels p within the building regions to focus on the areas of interest.

The loss function measures how closely the predicted vectors match the ground truth vectors. By minimizing this loss, the model is trained to produce attraction vectors that closely guide each pixel toward the actual building boundary, refining the boundary prediction.

The AFM provides explicit supervision for each pixel, guiding it toward the boundary through a vector field. Minimizing the difference between predicted and ground truth attraction vectors helps the model learn the geometry of building boundaries. It encourages the network to predict vectors that pull pixels precisely toward the closest edge, achieving more accurate polygonal boundary representation.

We have rpn and localization loss from equations (3.14) and (3.15), so the overall backbone loss, for AT-PolyGCN network, becomes

$$L_{backbone} = L_{rpn} + L_{loc} + L_{AFM} \quad (5.5)$$

Integrating AFM loss into the backbone improves edge detection and provides orientation information, resulting in improved Graph Convolution Network GCN initialization features. On the other hand, the orthogonality loss in the second branch ensures geometric regularity by enforcing well-aligned edges and corners in the final polygon predictions. These thoughtfully designed loss components work together to address semantic and geometric challenges in urban object reconstruction, delivering a more accurate and dependable building footprint reconstruction process.

Our total loss function becomes:

$$L_{AT-PolyGCN} = L_{backbone} + \lambda \frac{1}{N} L_{poly} + \lambda L_{ortho} \quad (5.6)$$

5.5 Summary

In this study, we proposed a deep learning framework for automatically extracting building footprints from satellite imagery with enhanced boundary regularization. Our approach integrates a backbone network for multi-scale feature encoding and object detection to produce well-localized RoI features, which are further refined using orientation information from Attraction Field Maps (AFMs) and a Graph Convolutional Network for building footprint reconstruction. The framework employs AFMs and GCNs to improve geometric learning and boundary precision. AFMs are particularly effective in handling diverse building structures by improving detectability for smaller buildings and providing consistent vector guidance for larger or more complex buildings, even in the presence of occlusions. They also address challenges with overlapping building edges by maintaining distinct fields for each structure, ensuring clear and separate boundaries. Our experimental results show that AT-PolyGCN surpasses the performance of both our baseline and previous methods, achieving high accuracy and producing regularized building boundaries. The results are competitive with state-of-the-art techniques, demonstrating the effectiveness of our approach. This advancement represents a significant step toward fully automated, precise building footprint extraction, minimizing the need for manual intervention. The proposed framework holds great potential for applications in urban planning, GIS mapping, and related fields, offering a scalable and accurate solution for spatial data analysis.

Chapter 6

Decoupled-PolyGCN

This chapter presents Decoupled-PolyGCN, a two-step network designed to address the limitations observed in our previous building footprint reconstruction networks. By separating object detection and polygonal segmentation into distinct stages, the network achieves improved accuracy and regularity in building footprint predictions. This chapter details the network architecture, training, and inference strategies, as well as the loss functions and hyperparameter tuning required to achieve state-of-the-art performance.

6.1 Overview

The results from our previous networks highlighted a critical limitation: the use of Mask R-CNN as the backbone. The baseline was inadequate, particularly for detection tasks, which directly affected subsequent processes like polygon reconstruction. The subpar performance of the detection module translated into challenges in generating accurate initial polygons. To improve the features provided to the Graph Convolution Network GCN for polygon initialization, we implemented several enhancements.

One of these improvements was the introduction of the Feature Augmentation Module (FAM), designed to incorporate high-frequency details essential for precise polygon predictions. By leveraging multi-scale features, FAM aimed to enhance the backbone’s output by adding predicted edge and vertex maps, providing explicit guidance to the GCN for vertex and edge placement. However, the inability of the baseline to effectively capture shape and boundary

information persisted, hindering the precision of mask predictions. To address this issue, we integrated instance boundary information into the segmentation process, combining mask and boundary predictions to improve accuracy. While these modifications improved results to some extent, they were still insufficient. Missed corners in polygonal representations could not be recovered, and the Feature Augmentation Module alone was inadequate for capturing intricate details.

To further enhance edge detection, we integrated Attraction Field Maps and modified the graph initialization process, which helped create better initial polygons. However, despite these refinements, the inability to provide the GCN with better features for polygon prediction remained a bottleneck. The underlying issue was that the RoI generated by the object detection module were suboptimal. It became evident that better RoIs could significantly enhance the final polygon predictions. To address this, we hypothesized that taking object detection decoupled and providing higher-quality RoIs could improve results substantially.

Deep learning approaches for polygonal building segmentation generally fall into two categories: two-step methods and direct polygonal segmentation techniques. Two-step methods typically involve raster segmentation followed by post-processing steps such as vectorization or simplification. They often utilize auxiliary representations like frame fields or directional indicators to facilitate vectorization. In contrast, direct polygonal segmentation methods predict building polygons directly from the input image, bypassing intermediate rasterization steps. While direct methods eliminate some complexities, they often face challenges like missing corners and inconsistent projections, and they are computationally intensive. On the other hand, two-step approaches, though more traditional, tend to produce better results due to their modular nature. In this work, we adopted a two-step approach. We first trained an object detection network decoupled on the complete image to generate high-quality bounding boxes. For this, we evaluated several object detection networks and selected CoDino Swin DETR, which produced the most accurate bounding boxes. The RoIs provided by this detection network were then fed into our backbone network to generate initial polygons. Using variable initialization points based on the mask and predicted corners, the GCN refined

these polygons through multiple iterative steps. The final predictions from the GCN included orthogonal boundaries, achieved through the incorporation of geometric supervision, which enhanced the geometrical correctness of both masks and polygon predictions. This two-step methodology allowed us to overcome the limitations of previous designs, providing more accurate and reliable building footprint reconstructions.

6.2 Decoupled Object Detection

Object detection is a fundamental task in our experiments, as it requires the localization of objects and the classification of their categories. In our network, the primary goal of object detection is to predict bounding boxes and corresponding category labels for buildings. Previously, we used Mask R-CNN and its variants for object detection, but these methods produced bounding boxes with insufficient information, which ultimately impacted the accuracy of the final polygon predictions. To address this issue, we decided to enhance the object detection network’s performance by performing object detection decoupled.

Modern object detectors often approach this task indirectly by designing surrogate regression and classification problems based on a large number of proposals [143, 54], anchors [144], or window centers [145, 146]. These methods rely heavily on post-processing steps, such as collapsing near-duplicate predictions, the design of anchor sets, and heuristics to assign target boxes to anchors [147]. This reliance on hand-crafted components can limit the simplicity and efficiency of the detection pipeline.

Transformers have significantly advanced deep learning by capturing long-range dependencies and intricate relationships within data. When applied to object detection, transformers move away from traditional convolutional architectures by utilizing self-attention mechanisms to analyze entire images simultaneously. Unlike CNNs, which rely on localized filters, transformers enable each region of an image to interact with all others. This broader contextual awareness helps the model recognize objects more accurately, particularly in complex scenes where multiple objects may be positioned close together or overlapping.

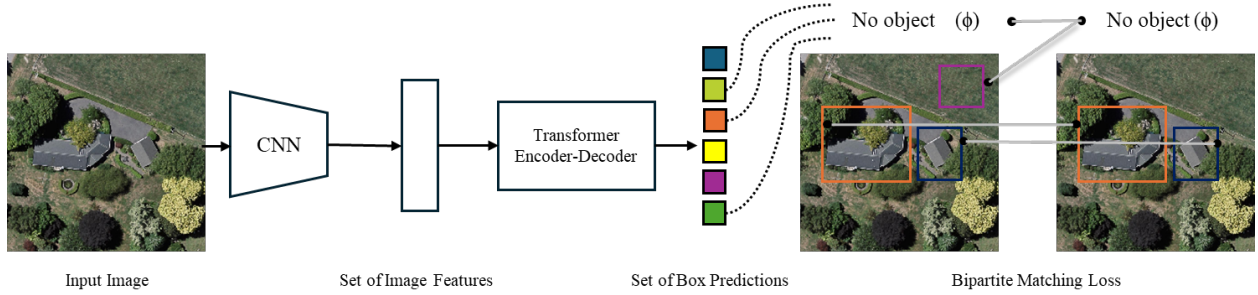


Figure 6.1 Detection Transformer Network Architecture [148]

To overcome the challenges in object detection, DETR [148] was introduced as a direct prediction approach, bypassing these surrogate tasks and streamlining the detection process. DETR predicts all objects simultaneously and uses a set-based loss function to perform bipartite matching between predicted and ground truth objects. This end-to-end approach eliminates the need for components like spatial anchors and non-maximum suppression, which are commonly used in traditional object detection methods. DETR is a deep learning model designed for object detection, utilizing the Transformer architecture originally developed for natural language processing. Unlike traditional detectors, DETR treats object detection as a set prediction problem, eliminating the need for proposal generation and post-processing steps. By using self-attention mechanisms, DETR captures complex relationships between objects within an image, leading to more precise predictions with fewer manually designed components. At its core, DETR relies on a Transformer encoder-decoder structure, allowing it to understand long-range dependencies within an image. Instead of using anchors or region proposals, it directly predicts objects as a set, assigning each one a bounding box and a class label. One of the key advantages of DETR is its simplicity—it has a streamlined architecture with fewer hand-crafted elements, making it easier to implement and interpret. Additionally, thanks to self-attention, it excels in detecting objects in complex scenes where interactions between objects are important.

Unlike most conventional detection methods that make predictions relative to initial guesses (e.g., two-stage detectors predict relative to proposals, while single-stage methods predict relative to anchors or a grid of potential object centers), DETR directly predicts detections

in absolute terms with respect to the input image. This eliminates the dependency on hand-crafted initial guesses and simplifies the detection pipeline. DETR can be implemented using standard components like CNNs and transformers, making it easily reproducible across different frameworks.

Recent advancements have focused on improving DETR-based detectors while retaining their end-to-end simplicity. We utilized codes and configurations provided by open-mmlab’s mmdet [149] for training different models. MMDet is a core library for training deep learning models, offering a flexible and robust framework. It features a universal open architecture, a streamlined user interface, and a highly customizable training process.

We analyzed different models available on MMDet, including top-performing networks like YOLOX-Nano, YOLOX-l [150], RTMDet-l [151], Grounding DINO-T-Square, Grounding DINO-B, Grounding DINO-T[152] and Collaborative Detection Transformers (CO-DETR) [153]. Table 6.1 shows the performance of these networks.

Table 6.1 Top-performing networks on MMDet for horizontal bounding box generation

Networks	AP(%)	AR(%)
YOLOX-Nano	61.4	66.4
RTMDet-l	74.6	79.1
YOLOX-l	76.3	80.8
Grounding DINO-T-Square	73.1	81.2
Grounding DINO-B	73.6	80.5
Grounding DINO-T	73.4	80.2
CO-DETR	78.4	83.0

After a thorough analysis of different DETR variants [152, 154, 155], we selected CO-DETR [153], which is built on the Deformable-DETR framework [156], for its superior training efficiency and performance. CO-DETR addresses the limitations of traditional one-to-one set matching by introducing a collaborative hybrid assignment training scheme. This approach leverages one-to-many label assignments to enhance the encoder’s feature learning and the decoder’s attention mechanisms. Auxiliary heads are integrated into the transformer encoder, and these heads are supervised using versatile one-to-many label assignments, significantly improving training efficiency.

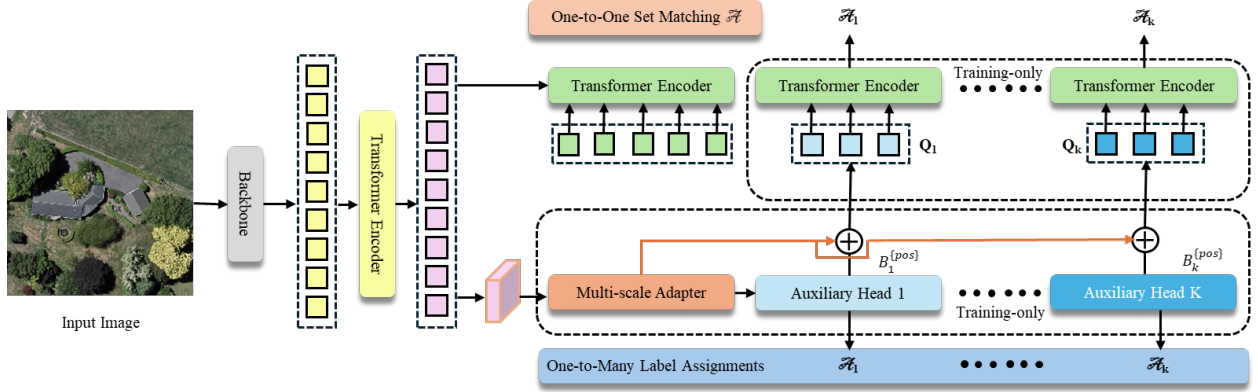


Figure 6.2 Detection Transformer with Collaborative Hybrid Assignments Training[153]

The CO-DETR framework follows the standard DETR protocol, where the input image is processed through a backbone and encoder to generate latent features. These features are then refined by predefined object queries in the decoder via cross-attention. The collaborative hybrid assignment training scheme improves feature learning in the encoder and attention learning in the decoder by generating customized positive queries and optimizing label assignments. CO-DETR has been shown to significantly improve both the qualitative and quantitative performance of DETR-based detectors, making it an ideal choice for our experiments.

6.3 Network Details

To address the challenges in our previous network, we adopted a two-step approach, performing decoupled object detection followed by using our specialized network for polygonal segmentation of building footprints. We refer to this method as Decoupled-PolyGCN. Below, we detail the network design and workflow for both the training and inference phases.

6.3.1 Training Phase

In the training phase, we utilize the ground truth bounding boxes to crop RoI from the input image. These RoIs are then fed into UResNet101, an architecture that combines the strengths

of U-Net and ResNet-101 to enhance image segmentation tasks. U-Net’s symmetric encoder-decoder design facilitates precise localization by combining contextual information from downsampling paths with detailed spatial information from upsampling paths. ResNet-101, with its deep residual learning and skip connections, ensures effective training by addressing vanishing gradient issues. The integration of ResNet-101 as the encoder in U-Net enables UResNet101 to capture complex features while maintaining localization accuracy. This results in high-resolution segmentation maps that are particularly beneficial for building footprint extraction. The output features from UResNet101 have the same dimensions as the RoIs, ensuring localized features that can be utilized for downstream predictions.

From these features, we use three prediction heads: the Segmentation Head, AFM Head, and Corners Head, similar to those in PolyAttractNet. These heads produce highly localized RoI information, which is then processed by the GCN for geometric building polygon predictions. To enhance the feature maps further, boundary masks generated from the RoI features are concatenated with the RoI feature map.

For polygon initialization, we employ dynamic points. The input for this process includes both masks and corner predictions. Corners are classified into two types: convex and concave. Contours of the masks are extracted and simplified, and thresholds are applied to convex and concave corners. Non-maximum suppression is then used to refine corner extraction. These corners are combined with the mask contours, and the closest points on the contour to each corner are identified to extract border information for proper connectivity. This process generates an initial polygon. Additionally, any missing corners from the mask contour that were not included in the initial polygons are identified and added in their appropriate positions, resulting in an enhanced initial polygon for further refinement by the GCN.

The Graph Convolutional Network GCN is employed to iteratively refine polygon vertices. Starting with the initial approximation, we use a three-step GCN process to progressively improve the accuracy of vertex positions. In the first step, the graph features are input into the GCN to compute vertex offsets. These offsets adjust vertex positions, and new graph features are interpolated from the updated positions. The interpolated features are then processed

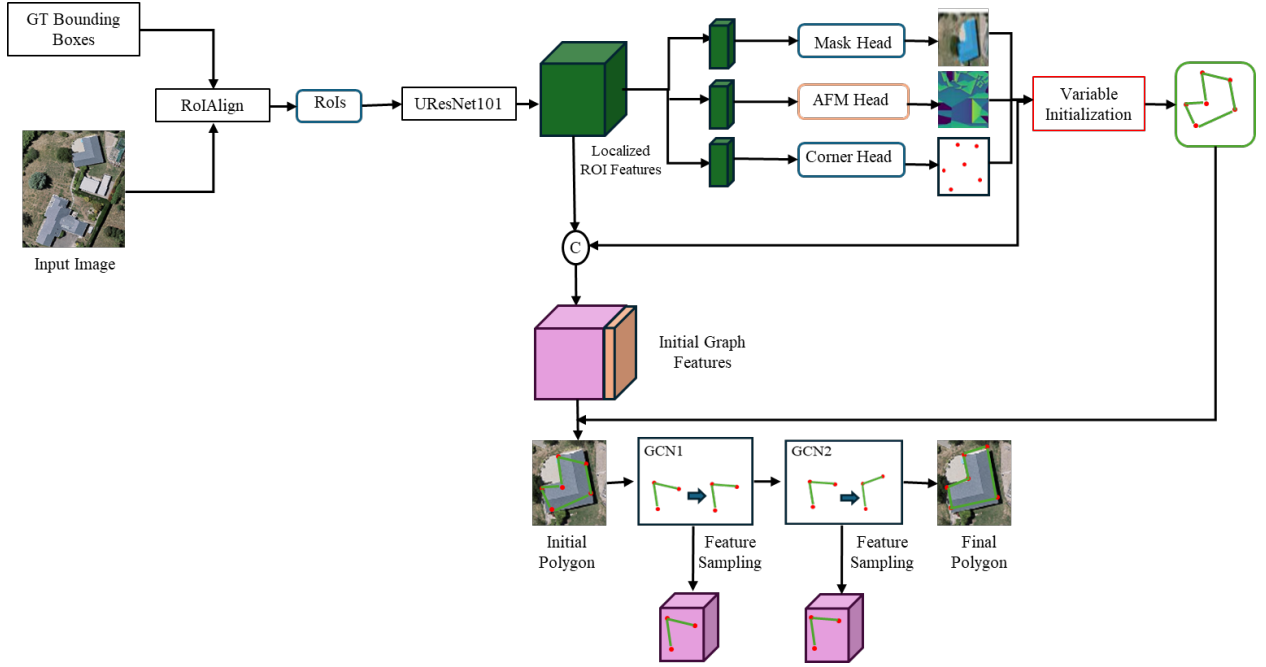


Figure 6.3 Training Phase of DeCoupled-PolyGCN

by a second GCN to compute additional offsets, further refining the vertex positions. This iterative process continues for three steps, progressively enhancing the precision of polygon predictions. The addition of orthogonality loss ensures that the final polygons are more regularized, with sharp and well-aligned edges.

6.3.2 Inference Phase

In the inference phase, the overall network design is similar to the training phase, with the exception that the input to UResNet101, the backbone of the network, comes from bounding boxes predicted by the decoupled object detection network. For object detection, we use CO-DETR, which was selected after extensive experimentation as the best-performing detection network. CO-DETR provides high-quality bounding boxes, which are used to generate RoIs from the input image. The output features are then concatenated with the predictions from the Mask Head, AFM Head, and Corners Head in the same manner as in the training phase. Polygons are initialized dynamically using the mask and corner features,

and a multi-step GCN is applied to refine the boundaries. The inclusion of orthogonality loss during this process ensures that the final polygons are geometrically regularized.

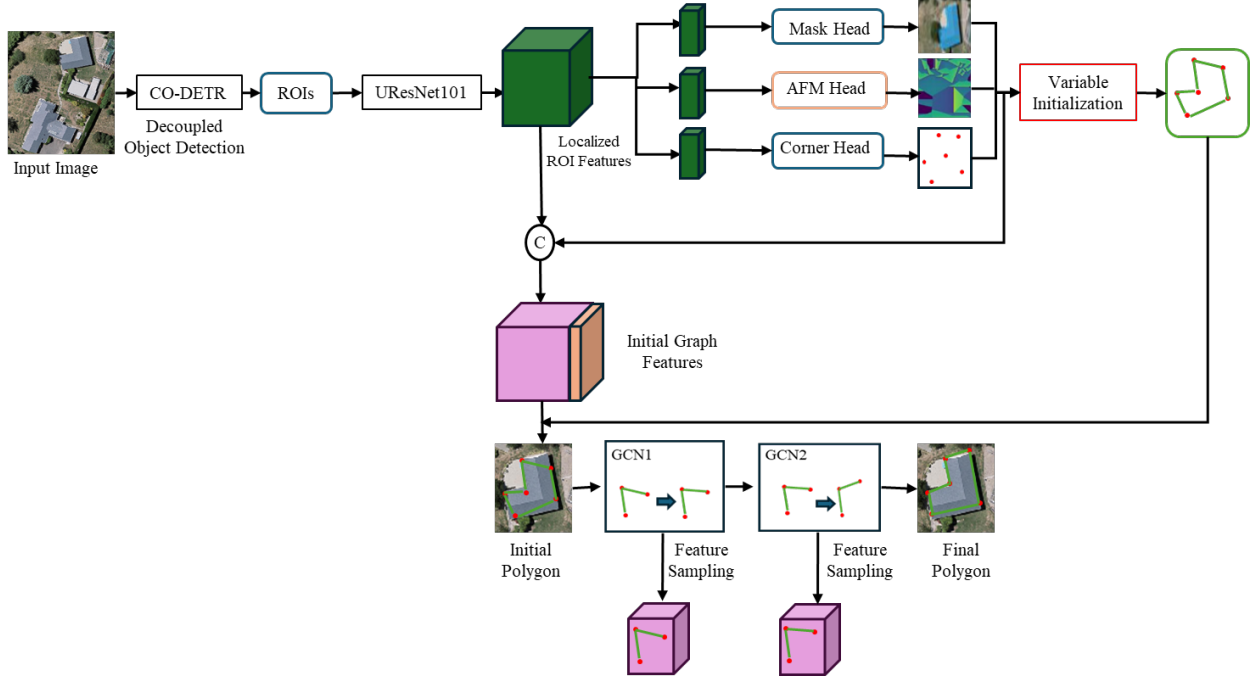


Figure 6.4 Inference Phase of DeCoupled-PolyGCN

This two-step approach demonstrates the critical importance of high-quality detection results and their impact on polygonal segmentation. By improving both raster-based and vector-based metrics, Decoupled-PolyGCN significantly enhances the accuracy and regularity of building footprint predictions.

6.4 Loss Design

Our network employs a multi-loss training strategy, as discussed in previous chapters, to optimize different components of the model. Cross-entropy loss is widely used for instance segmentation tasks, particularly for mask prediction, to quantify the difference between the predicted mask probabilities and the ground truth. For binary mask prediction, which classifies each pixel as either foreground or background, we use the binary cross-entropy

(BCE) loss.

$$L_{mask}(p(y)) = -(y \log(p(y)) + (1 - y) \log(1 - p(y))) \quad (6.1)$$

For the building corners, which are categorized into convex and concave corners, we also apply cross-entropy loss to optimize their predictions.

The Attraction Field Map (AFM) loss is designed to minimize the discrepancy between the predicted attraction vectors and the ground truth attraction vectors. If N is the total number of pixels in the image, P represents the set of all pixels in the image. $\tilde{v}$ is the predicted attraction vector at pixel p . v_p is the ground truth attraction vector at pixel p . Using the $L1$ loss, this is formulated as.

$$L_{AFM} = \frac{1}{N} \sum_{p \in P} \|\tilde{v} - v_p\|^2 \quad (6.2)$$

The features from the mask and corners are used to construct an initial polygon with a dynamic size, which is subsequently passed to the Graph Convolutional Network GCN for further refinement. The loss function for the GCN is designed to optimize polygon vertex prediction by minimizing the difference between the predicted vertex positions p^{pre} and the ground truth vertex positions p^{gt} . This is calculated using the geometric $L1$ distance.

$$L_1(p^{pre}, p^{gt}) = \sum_{i=0}^N (|x_i^{pre} - x_i^{gt}| + |y_i^{pre} - y_i^{gt}|) \quad (6.3)$$

To enforce geometric regularity and ensure that building boundaries retain their orthogonal properties, we introduce an orthogonality loss. This loss helps maintain right-angle characteristics and improves the shape regularity of building polygons. The orthogonality loss is defined as

$$L_{ortho} = \frac{1}{N} \sum_{j=1}^N L(P_j) \quad (6.4)$$

For training, we combined different loss function weights and conducted extensive hyperparameter tuning to achieve optimal network convergence. Hyperparameter tuning is crucial in deep learning as it directly impacts learning, generalization, and stability. Balancing the weights of loss components, such as mask loss, AFM loss, vertex prediction loss, and orthogonality

loss, was key to ensuring no single task dominated the optimization process, which could degrade performance. For example, overemphasizing mask loss could lead to well-segmented masks but poorly regularized polygons, while underweighting it could produce coarse masks, hindering downstream tasks. Tuning involved systematically experimenting with parameters like learning rates, batch sizes, and optimizer settings. This helped the network capture fine-grained boundary details while maintaining geometric regularity. Learning rates were particularly critical, as rates that were too high or low could hinder convergence. Similarly, batch size affected stability and computational efficiency, with smaller batches improving generalization but increasing noise.

Through this process, we stabilized the training and optimized various aspects of building footprint extraction, including mask prediction, corner detection, polygon refinement, and geometric regularity. This comprehensive approach significantly improved the network’s performance, resulting in more accurate segmentation and precise polygonal predictions for building footprint reconstruction.

6.5 Summary

This chapter introduced Decoupled-PolyGCN, a two-step framework designed to address the challenges faced by previous methods in building footprint reconstruction. The first stage involves decoupled object detection utilizing CO-DETR, an advanced transformer-based detection network, to produce high-quality bounding boxes. The second stage has multiple prediction heads and multi-step GCN for building polygon predictions. Our network design highlights the critical role of high-quality detection results in influencing polygonal information and improving the overall accuracy of building footprint reconstruction. The contribution of the Attraction Field Map is particularly noteworthy, as it enabled the generation of cleaner, simpler polygons with precise vertex and edge positioning. Additionally, the use of dynamic points for polygon initialization proved highly effective, resulting in more streamlined polygon shapes. The incorporation of orthogonality loss further enhanced the regularity of

building boundaries, ensuring well-structured and geometrically consistent polygons. By combining these innovations, the proposed method significantly improves both the accuracy and regularity of building footprint predictions, achieving performance that surpasses the state of the art.

Chapter 7

Dataset Analysis and Preparation Pipeline

Training accurate deep learning models requires a well-structured dataset preparation pipeline. We discussed our network architectures in previous chapters, but we must have optimized datasets to achieve optimal performance. This chapter will introduce the datasets, the challenges in preparing datasets, and the dataset pipeline used in our experiments. We have prepared and used SpaceNet-2 and WHU datasets. The dataset pipeline typically involves several steps, including data collection, preprocessing, annotation, and model training. By thoroughly following this process, we can create robust datasets. These datasets will expedite the development of practical building footprint reconstruction algorithms.

7.1 Challenges in Datasets for Building Footprint Extraction

Obtaining high-quality labeled datasets for building footprint extraction can be challenging. Similarly, manually annotating building footprints in satellite imagery is time-consuming and expensive. Existing labeled datasets may not cover diverse urban environments, leading to limited generalization. Converting the datasets into the correct format takes much work, even if they are available. Building footprints are often a smaller category class than other land cover types, including roads, vegetation, etc, causing imbalanced class distribution.

Imbalanced class distribution can affect model performance. Additionally, buildings are of various shapes and sizes, ranging from tiny houses to large commercial structures. Therefore, models must cater to this situation in a certain way to extract footprints accurately. Manual annotations may be inaccurate as they might contain errors, especially when dealing with complex building boundaries or occlusions. Noise in annotations also affects model training and generalization.

Converting raw data like satellite imagery into suitable input formats for neural networks requires preprocessing. The preprocessing involves many steps, including resizing, cropping, normalizing pixel values, and handling missing data, depending on the network. Neural networks typically work with numerical data. Therefore, choosing an appropriate representation, such as pixel intensities and spectral bands, is crucial. Ensuring that the neural network can handle multiscale features is also very important. In some situations, we must address the class imbalance problem by either oversampling or undersampling building footprints. This augmentation of the dataset often helps improve the robustness of the model. Annotating building footprints in a consistent and standardized format, such as bounding boxes and polygons, is essential. Converting annotations from one format to another, e.g., from shapefiles to pixel masks, might be inclined to have errors. Combining satellite imagery with GIS map data (e.g., OpenStreetMap, Google Maps) helps improve the process of identifying and extracting buildings. However, integrating different data sources is not a trivial task.

7.2 Dataset Description and Preprocessing Pipeline

A range of datasets must be prepared to train neural networks. In our initial experiments, we used the SpaceNet-2 dataset. After that, we used the WHU dataset to generalize our experiment. These datasets were chosen because they cover diverse areas across all continents, allowing us to design the network accordingly and ensuring that the results can be generalized to other datasets. We set up a comprehensive data preparation pipeline to prepare these datasets and make them trainable with our deep neural network. We used an integrated

strategy to carefully prepare the training dataset using Python scripts and software such as FME and ArcGIS.

Table 7.1 Characteristics of Datasets used in our Experiments

Dataset	SpaceNet-2	WHU
Geographic Area	Las-Vegas	Christchurch
Data format	GeoJSON, JSON(COCO format)	GeoJSON, JSON(COCO format)
Amount (GB)	1	4.5
Visualization Tool	FME, ArcGIS	FME, ArcGIS
Area	216 km^2	450 km^2
Building Annotations	108,320	220000
No of Images	3851	7150
Ground Resolution	30cm	30cm

7.2.1 SpaceNet-2 Dataset

The initial dataset used in our study is SpaceNet-2, an open dataset obtained from the building extraction challenge of the DeepGlobe workshop held at CVPR in 2018. This dataset consists of high-resolution satellite images, which also include accurate information about the boundaries of buildings. The dataset encompasses both urban and suburban areas and includes four specific cities: Las Vegas, Paris, Shanghai, and Khartoum. The dataset is distinctive due to the presence of four cities located on four separate continents. This ensures a wide range of different appearances for the roofs of the buildings.

The DigitalGlobe Worldview-3 Satellite took the photos in GeoTiff data format. The image dimensions are 650x650 pixels, and the resolution is 30 centimeters per pixel, enabling the image to cover an area of 200 meters by 200 meters. The dataset contains 24,586 satellite photos that have been labeled, and it includes 302,701 building footprint polygons that have been thoroughly annotated. The annotations are organized by objects and are in the GeoJSON format. The experimental data were obtained from Amazon Web Services (AWS) under a license granted by the DeepGlobe workshop. From the dataset, we have chosen the Pan-sharpened RGB images for our research out of the several types of imagery from satellites



Figure 7.1 SpaceNet Dataset

available. We performed a series of preprocessing steps on the dataset before utilizing it for training and testing our neural network model.

1. Initially 48 bits long, the raw RGB images were converted to 24-bit RGB. This conversion was necessary to ensure the images could be viewed by the most common image viewers, enhancing the practicality of our data processing.
2. The building footprint annotations are originally presented in the following format: ImageId, BuildingId, PolygonWKT-Pix, PolygonWKT-Geo. Since the geographic coordinates were not needed for the purposes of research, we eliminated the latitude and longitude coordinates of the building polygon vertices in geographical space.
3. We transformed the initial GeoJSON files to standard JSON format, which made loading the data more accessible. As a requirement of our design assumptions, we organized

the polygon vertices in clockwise order.

4. Next, we generated bounding boxes and pixel-wise labels based on the preexisting building footprint annotations. We employed OpenCV, a freely available computer vision toolkit, to create polygon masks at the pixel level using the annotated polygon vertices to get pixel-level labels.
5. We excluded empty images that did not contain any building footprints. In addition, we established a threshold of 50 pixels for the polygon area. Consequently, we excluded building polygons with an area below this level.
6. We resized the original images to accommodate our neural networks. Similarly, the dimensions of the ground truth, which encompassed the vertex coordinates of the polygons, bounding boxes, and polygon masks, were also modified.
7. We employed two distinct methodologies for normalizing the images. In the initial method, the pixel intensities were subtracted by the average RGB values, but in the subsequent method, they were subtracted by the standard deviation.

The dataset is randomly partitioned into 80% training, 10% validation, and 10% testing samples, including their corresponding ground truth data.

7.2.2 WHU Dataset

There needed to be more than one dataset to generalize the results of our network. We decided to add the WHU dataset as the second dataset we used to train and test with our network. The dataset contains around 220,000 independent buildings obtained from aerial images with a spatial resolution of 0.075 m and a coverage area of 450 km in Christchurch, New Zealand. The data was acquired from several cities around the world utilizing remote sensing tools, including QuickBird, Worldview series, IKONOS, and ZY-3. Fig 6.2 shows some images from the WHU dataset. We needed to preprocess this data because it was not in the desired COCO format. We prepared this dataset for our experiments by using ArcGIS,

writing different Python scripts, and rigorously validating them.

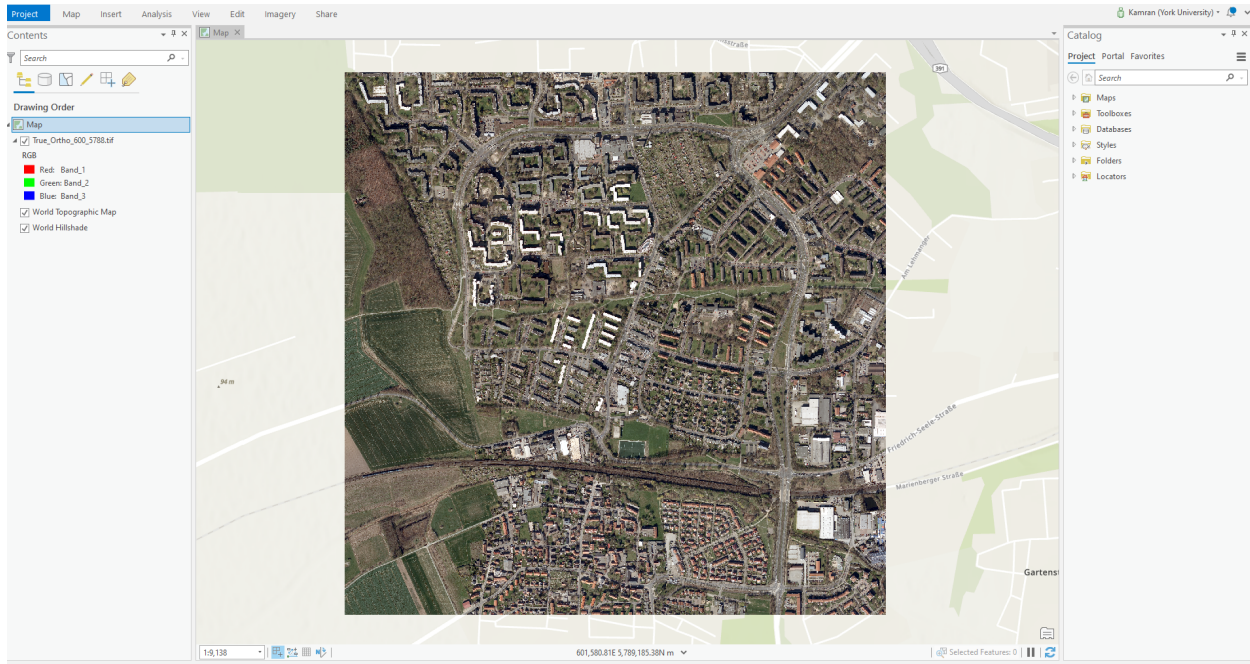


Figure 7.2 Image input to ArcGIS for preprocessing

The preprocessing involved the following steps.

1. The first step involves cropping satellite imagery into 512x512 pixel tiles using the Create Fishnet toolset in ArcGIS. This tool generates polygon cells by defining the output's coordinate system and specifying the cell size and grid dimensions. The geometry type is set to Polygon, and the template extent is aligned with the image's maximum and minimum coordinates to ensure accurate tiling for building polygon generation.
2. Next, the Split Raster tool is used to divide the images into tiles based on the selected grid dimensions. The resulting tiled images inherit most of the attributes of the original raster, including the spatial reference, source type, pixel type, pixel depth, and cell sizes.
3. To generate the required annotation files, initially, we need to generate a GeoJSON File, so we create line, point, and polygon features for each file using multiple tools

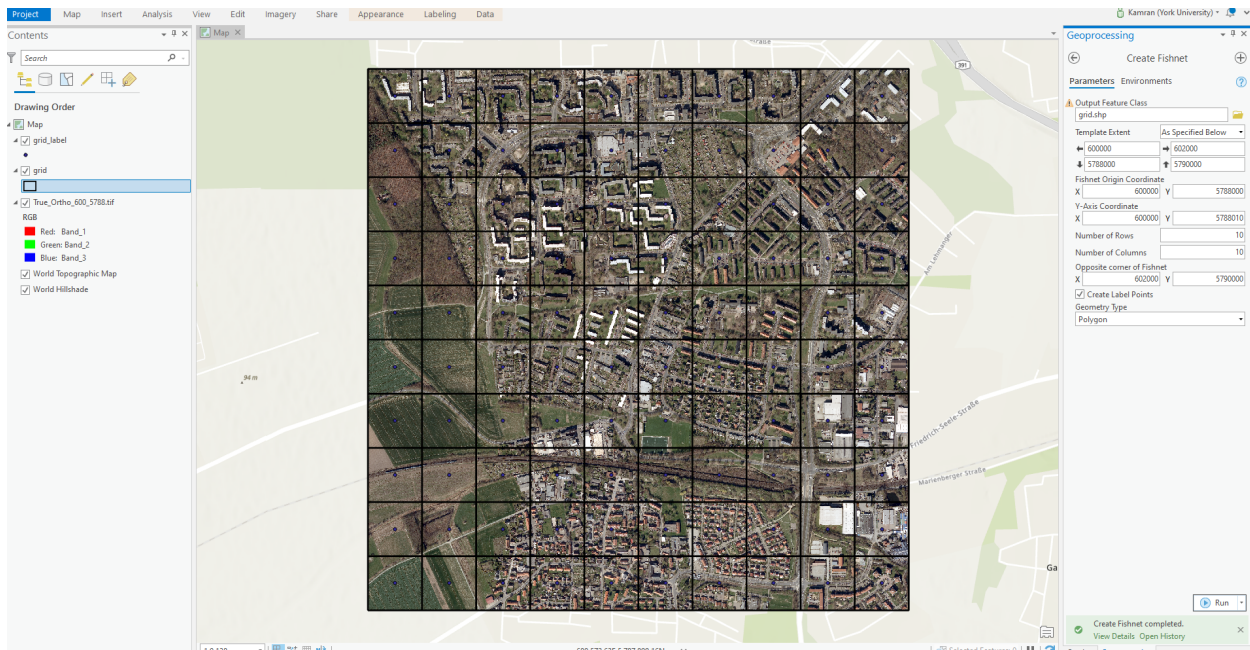


Figure 7.3 Setting extent for image cropping

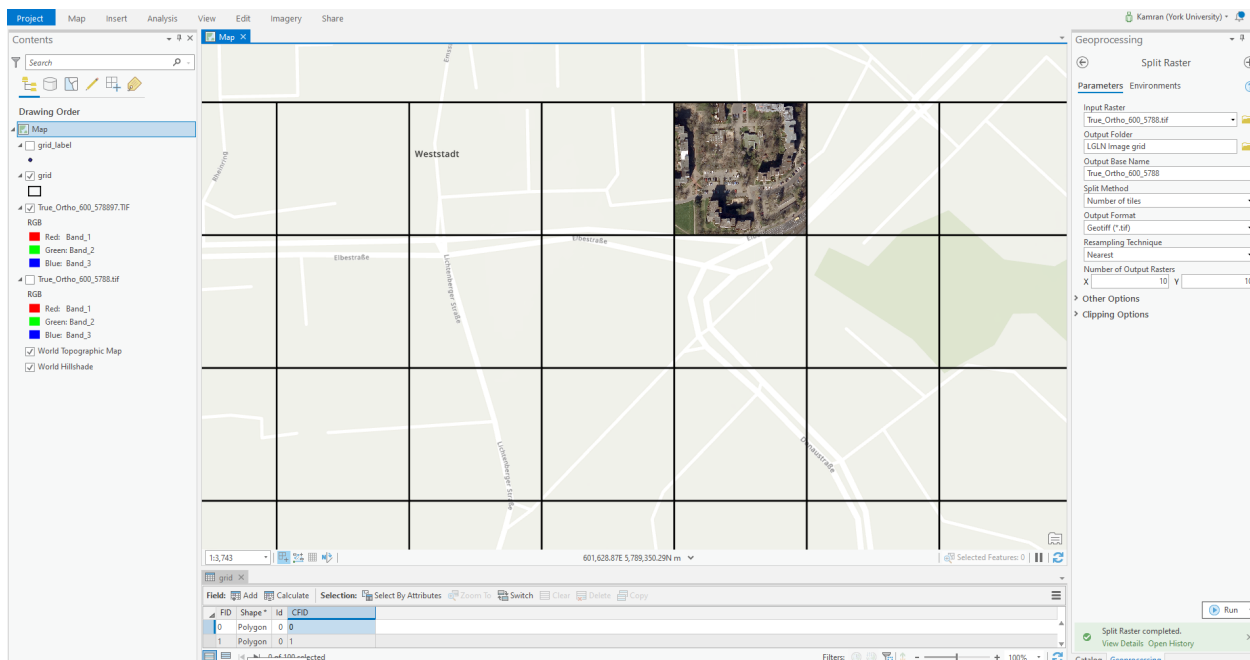


Figure 7.4 Cropping satellite imagery

from ArcGIS.

The available tools include "Feature Vertices to Points," which generates a feature class

of points derived from designated vertices or locations of the input features as shown in Fig 7.5.

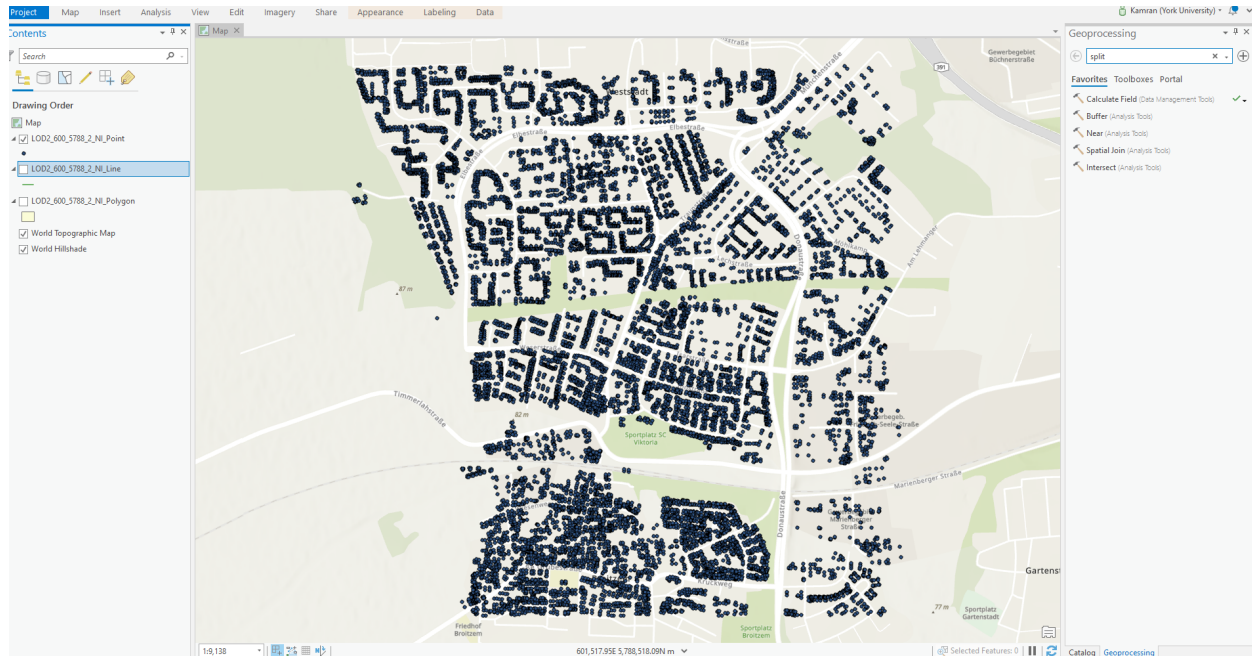


Figure 7.5 Point Features

The "Feature to Line" tool generates a feature class that consists of lines created by transforming the boundaries of polygons into lines as displayed in Fig 7.6.

As shown in Fig 7.7, the "Feature to Polygon" tool generates a feature class that consists of polygons created from the areas covered by input line or polygon features. On the other hand, the "Simply Polygon" tool simplifies polygons by deleting unnecessary vertices while maintaining their fundamental shape.

4. The Polygon.shp file generated in the previous step is loaded back into ArcGIS for further refinement. Using the Create Fishnet toolset, we ensure the annotation files align with the maximum and minimum image coordinates. The rows and columns are set to 10, and the geometry type remains as Polygon to create grids for annotations. This step mirrors the image cropping process to maintain alignment between images and annotations.

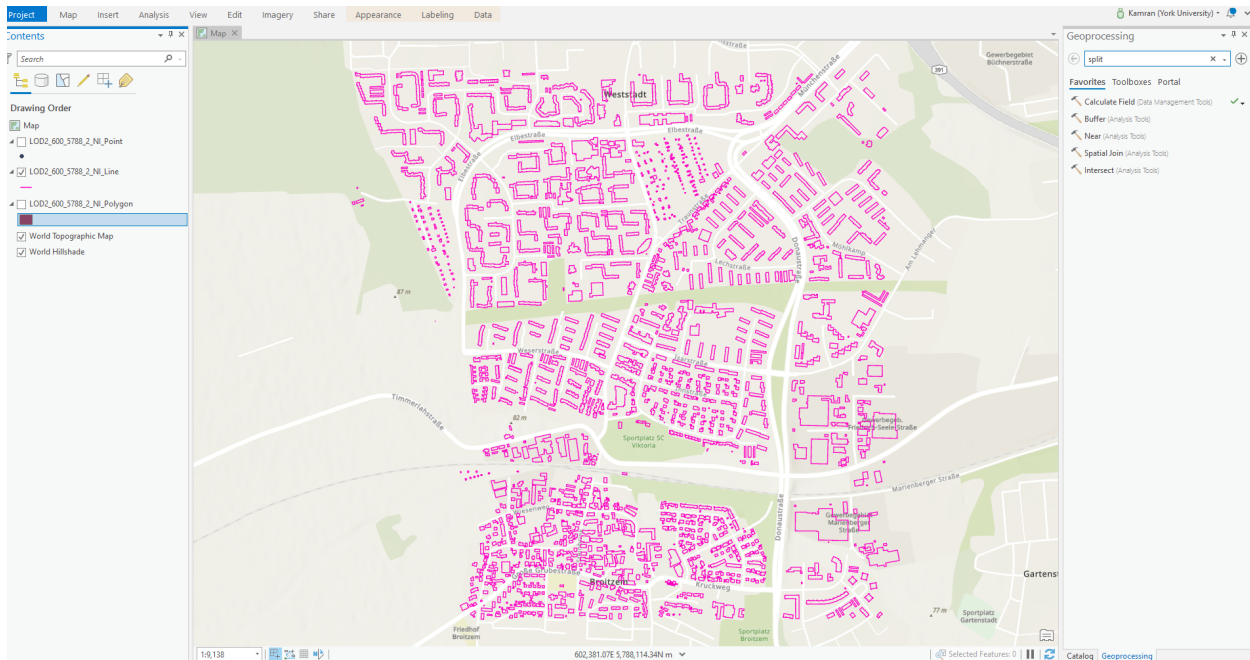


Figure 7.6 Line Features

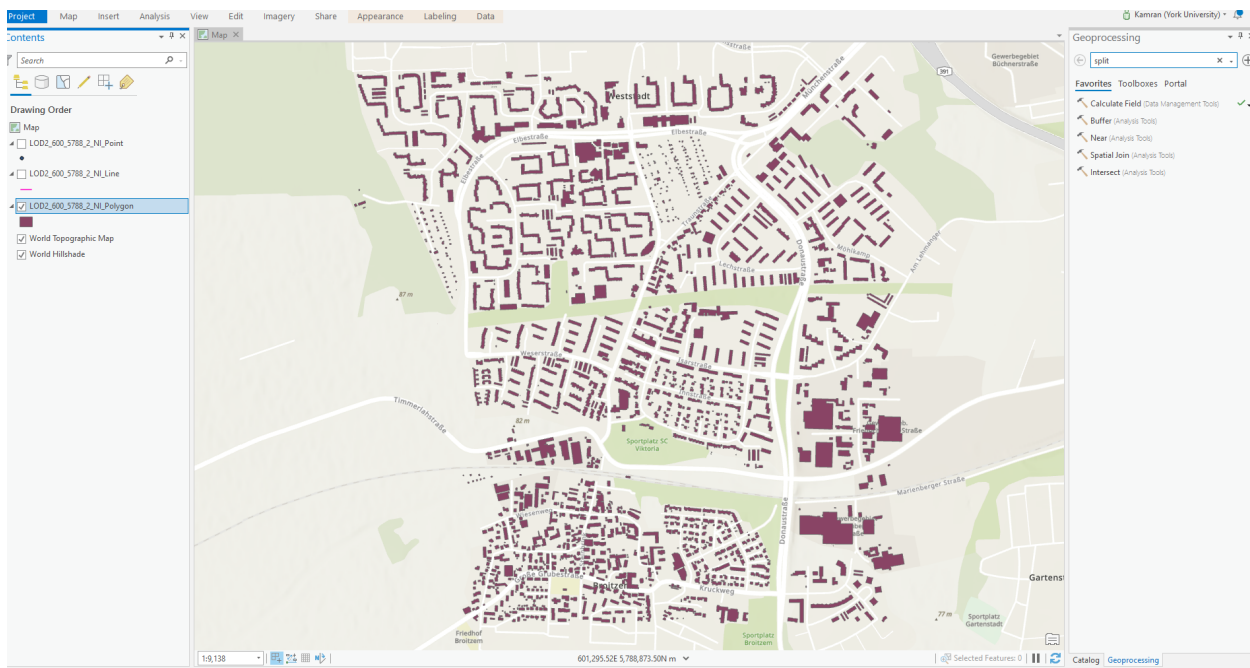


Figure 7.7 Polygon Features

5. To address issues with open polygons, we add a text field to the grid attribute table and copy the FID (Feature Identifier) to the new field. This allows us to split the grids into

individual polygons using the Split Tool in ArcGIS. The tool automatically resolves open polygon issues, ensuring all polygons are closed and ready for further processing.

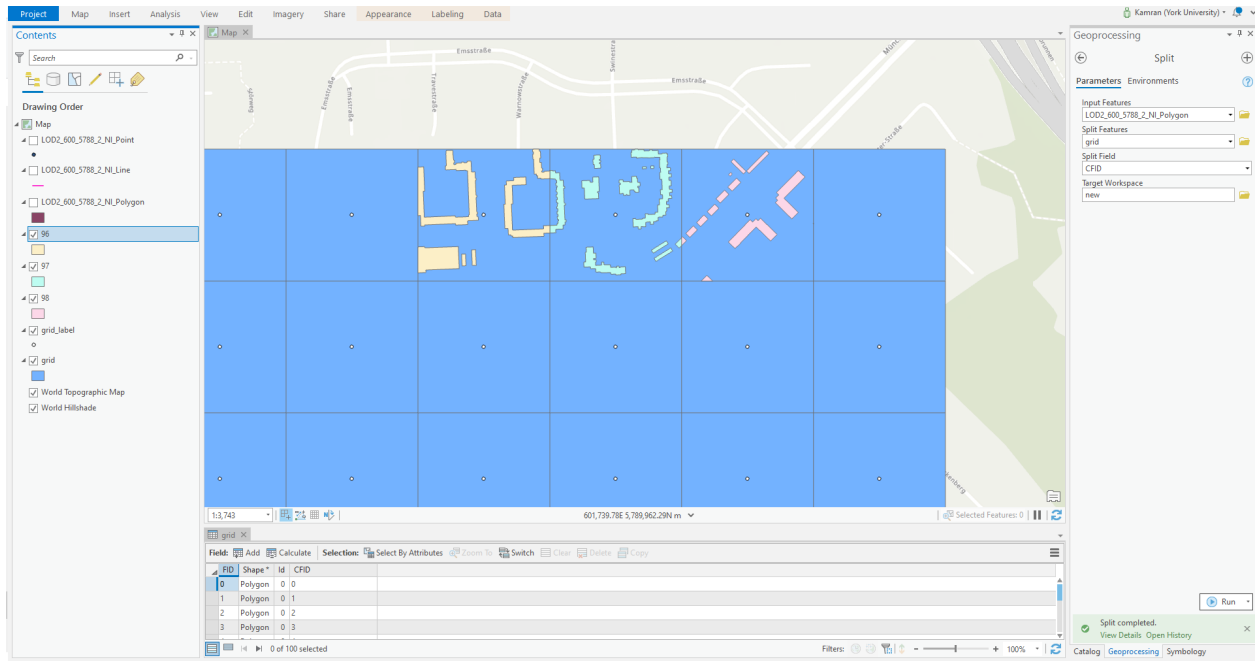


Figure 7.8 Resolving Open Polygon Issues

6. The Batch Feature to JSON tool is used to convert the polygon features into GeoJSON format. This tool transforms the shape, attributes, and coordinate system of each feature into JSON format, saving them with a .geojson extension. This standardized format is compatible with a wide range of deep learning frameworks, making it ideal for training and evaluation.
7. To prepare the annotations for use in COCO-compatible networks, each GeoJSON file is converted into the COCO format JSON file. This involves renaming the GeoJSON files and corresponding images to ensure consistent naming. Then Writing a Python script to convert the GeoJSON data into COCO format. The COCO format includes key fields such as {id, image_id, category_id, segmentation, area, bbox, iscrowd}. Each polygon is mapped to the COCO format, with its segmentation, bounding box, and other properties extracted from the GeoJSON data. This step ensures compatibility

with neural networks that require COCO-formatted annotations for training.

```
{
  "id": 22999,
  "image_id": 6783,
  "category_id": 1,
  "segmentation": [
    [35.0, 186.5, 25.5, 186.0, 25.0, 165.5, 58.0, 164.5, 59.0,
     167.5, 63.5, 168.0, 63.0, 185.5, 35.0, 186.5]
  ],
  "area": 787.75,
  "bbox": [25.0, 164.5, 38.5, 22.0],
  "iscrowd": 0
}
```

Figure 7.9 COCO format for groundtruth annotations

```
{
  "licence": 1,
  "file_name": "6783.TIF",
  "coco_url": "none",
  "height": 512,
  "width": 512,
  "date_captured": "2022-06-01 14:18:37.660165",
  "flickr_url": "none",
  "id": 6783
}
```

Figure 7.10 COCO format for individual Image annotation

8. After converting the annotations into COCO format, the dataset is partitioned into training, validation, and test folders. Separate JSON files are created for each split, containing the corresponding image annotations.

7.3 Dataset Distribution

Our experiments are based on two highly diverse datasets: the WHU Dataset and the SpaceNet Dataset, both of which offer a wide range of building structures and environmental

contexts. To better understand the characteristics of these datasets, we analyzed the data across three key aspects: the distribution of buildings by vertex count, the density of buildings per image, and the classification of buildings by size. These analyses highlight the diversity and complexity of the datasets, providing valuable insights into their suitability for training deep learning models for building footprint reconstruction.

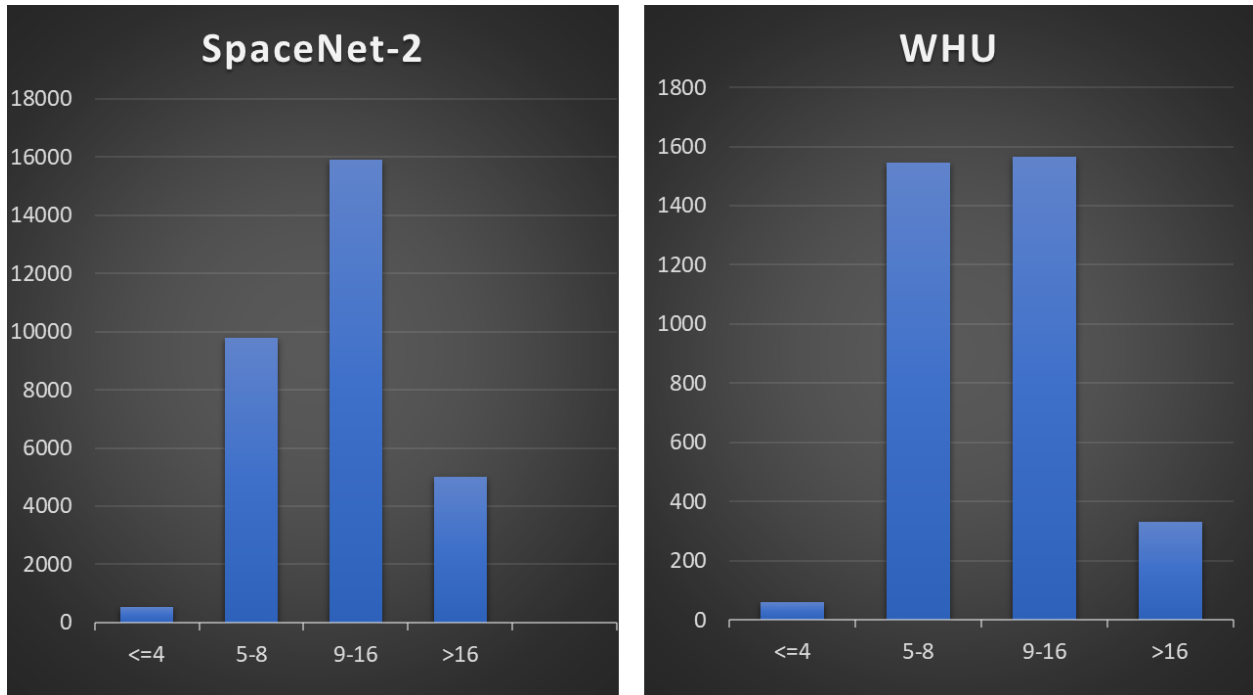


Figure 7.11 Vertex-wise building distribution

Buildings in both datasets vary significantly in complexity, which is reflected in the number of vertices defining their polygonal footprints. To categorize these buildings, we divided them into four groups based on the number of vertices (Buildings with four or fewer vertices, with 5 to 8 vertices, with 9 to 16 vertices, and with more than 16 vertices). Figure 7.11 illustrates the distribution of buildings across these vertex-based categories. The chart shows that the majority of buildings in both datasets have 16 or fewer vertices, with a large portion falling into the 9 to 16 vertices range. This distribution reflects a mix of both simple and moderately complex building structures, providing a balanced dataset for training and evaluation.

The images in the WHU and SpaceNet datasets capture a variety of urban and suburban



Figure 7.12 Diversity of the dataset

scenes, ranging from sparsely populated areas to dense cityscapes. To quantify this variability, we divided the images into four categories based on the number of buildings present in each image (Less than 5 buildings, 5 to 10 buildings, 10 to 25 buildings and more than 25 buildings). Figure 7.12 presents the building density distribution in our datasets, while Figure 7.13 provides a pie chart summarizing the percentage of images in each category. The analysis

reveals that most images in both datasets are dense, with a large portion containing more than 25 buildings per image. This density introduces variability in building configurations, occlusions, and overlaps, making these datasets particularly valuable for training models that need to generalize across different urban and suburban layouts.

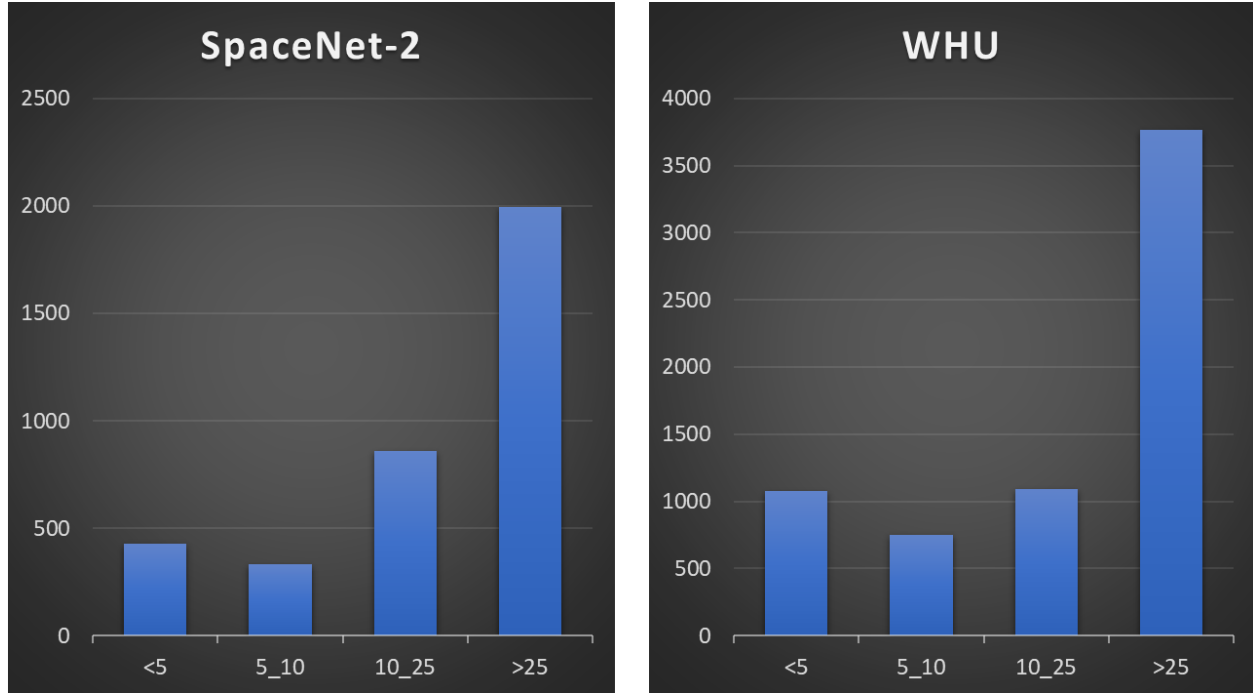


Figure 7.13 Building density chart

Another critical aspect of dataset diversity is the distribution of buildings by size. Building size significantly impacts the performance of segmentation models, as smaller buildings are more prone to being missed or misclassified, while larger buildings pose challenges in terms of boundary regularity. Following the MS COCO classification guidelines, we categorized buildings into three size-based groups.

1. Small buildings: Area less than 32 square pixels.
2. Medium buildings: Area between 32 and 96 square pixels.
3. Large buildings: Area greater than 96 square pixels.



Figure 7.14 Area-wise building Distribution

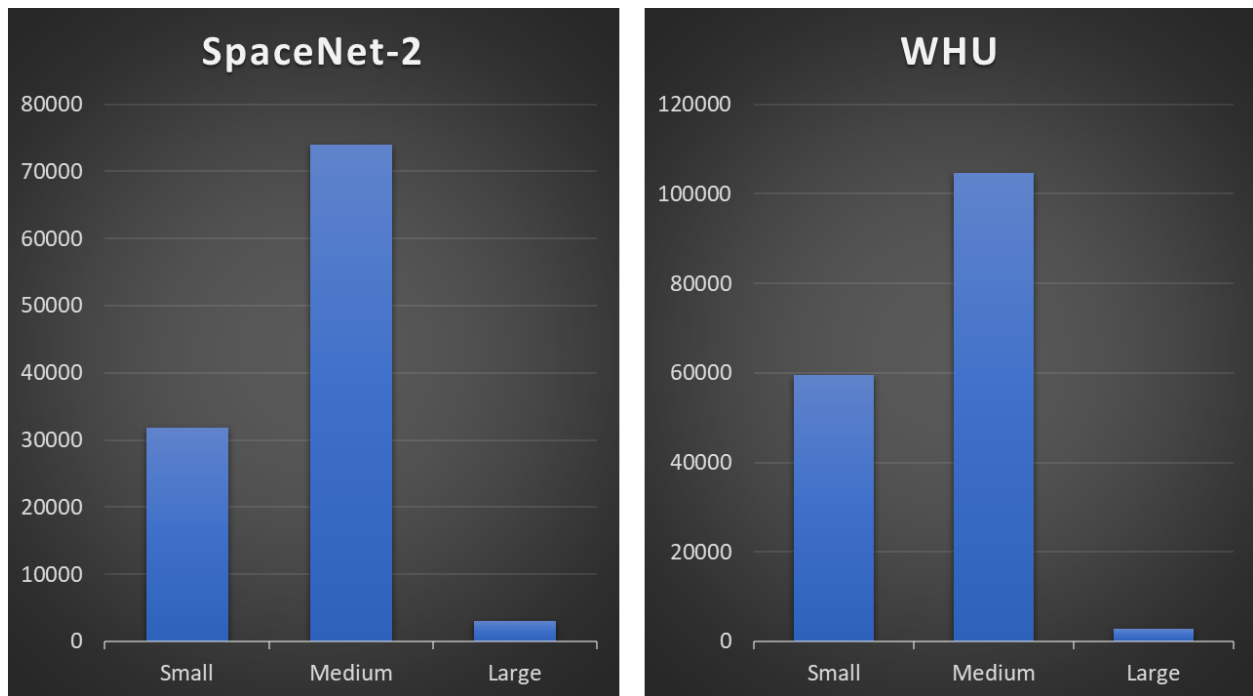


Figure 7.15 Area-wise building distribution Chart

Figure 7.15 shows the distribution of buildings by size, while Figure 7.14 provides visual examples of small, medium, and large buildings in a single image. The analysis indicates that the datasets contain a significant number of medium and large buildings, with small buildings comprising a smaller fraction. The presence of large buildings introduces challenges related to shape regularization, while small buildings test the network’s ability to detect fine details and maintain precision.

An essential aspect of our research focuses on evaluating the performance of our network in extracting building footprints for adjacent buildings, particularly those with shared corners, as illustrated in Fig 7.16. Adjacent buildings with shared corners present a unique challenge in building footprint extraction, as accurately distinguishing and separating their boundaries is crucial for precise segmentation and polygon reconstruction. This issue becomes even more critical in dense urban environments, where buildings are often situated close to one another. To address this, we analyzed the distribution of buildings with shared corners in both datasets, as shown in Fig 7.17. These buildings constitute a significant portion of the datasets, making it imperative to enhance the network’s ability to handle such scenarios effectively.

7.4 Summary

This chapter provides an overview of the WHU and SpaceNet-2 datasets, along with the preprocessing pipelines used to prepare them for training our building footprint reconstruction networks. These datasets, characterized by their diversity in vertex count, building density, and size, present both challenges and opportunities for evaluating deep learning models. The chapter highlights the complexities of preparing high-quality datasets, including annotation difficulties, class imbalance, data format inconsistencies, and preprocessing needs. Systematic transformations were applied to ensure the datasets were compatible with our neural network designs, facilitating robust training and evaluation. The variability in vertex count tests the model’s ability to handle a range of structures, from simple to complex polygons. Building



Figure 7.16 Adjacent Buildings with Shared Corners

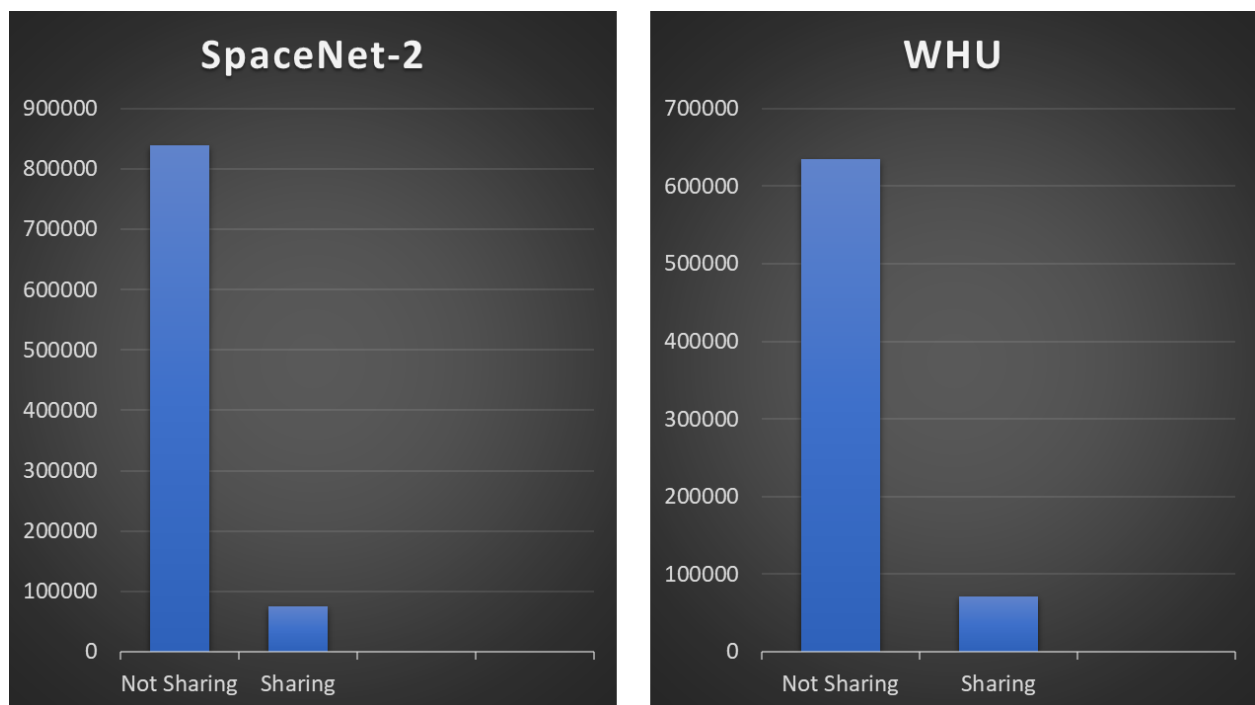


Figure 7.17 Buildings with Shared Corners distribution Chart

density challenges the segmentation of overlapping and occluded structures, while the range of small, medium, and large buildings ensures scalability across different spatial dimensions. By analyzing these characteristics, we gain a deeper understanding of the challenges and opportunities provided by these datasets, which are crucial for advancing the development of accurate and reliable building footprint reconstruction models.

Chapter 8

Results and Analysis

This chapter will discuss the configuration and implementation details of all our networks, as well as the quantitative and qualitative results and their analysis. We will also explain the different performance metrics used to measure the accomplishments of our networks.

8.1 Network Configuration and Implementation Details

In the beginning, we employed a multi-step data pipeline to prepare the SpaceNet and WHU datasets for our experiments. The final annotations for both datasets were formatted in COCO JSON. The SpaceNet dataset has images of size 650x650 pixels, while the WHU dataset has images of size 512x512 pixels. For the experiments of our first three networks, the SpaceNet dataset was trained with a batch size of 1 for 35 epochs, while the WHU dataset was also trained with a batch size of 1 but for 30 epochs. To achieve optimal results with Decoupled-PolyGCN, we trained it for 10 epochs on the WHU and SpaceNet datasets. In all experiments, the images were resized to 800x800 pixels before input into the network. In the first three networks, the RoI size was fixed at 28x28, while in Decoupled-PolyGCN, we increased the RoI size to 64x64 to capture more detailed features.

In RO-PolyGCN, we used ResNet50 and FPN for feature extraction. For FAE-PolyGCN and AT-PolyGCN, we utilized a deeper architecture with ResNet101 and FPN. In our final network, Decoupled-PolyGCN, we adopted UResNet101, a hybrid architecture combining ResNet101 and UNet, to improve feature extraction and segmentation accuracy.

Significant architectural enhancements were introduced in our networks. In the first three models, we utilized GCN, with the number of vertices per building polygon fixed at 16 and each vertex connected to 4 neighbors. However, in Decoupled-PolyGCN, we advanced our GCN to produce more refined polygon predictions. Additionally, we implemented a variable initialization technique based on corner prediction, enabling the number of vertices per polygon to adapt to the specific geometry of each building dynamically.

All networks were trained using the Adam optimizer with a learning rate and weight decay applied. Careful calibration and hyperparameter tuning were performed to select optimal weights for the loss functions. Training time per epoch varied depending on the network and dataset. For SpaceNet, one epoch took approximately 60-80 minutes, while WHU took 55-85 minutes on average. Inference times also varied across the models. RO-PolyGCN and FAE-PolyGCN achieved an inference time of approximately 0.20 seconds per image, while AT-PolyGCN was slightly faster at 0.15 seconds per image. Decoupled-PolyGCN demonstrated a significant reduction in inference time, achieving 0.05 seconds per image due to the integration of vectorization techniques and the inclusion of optimized modules in the visualization pipeline. This improved efficiency highlights the advancements made in our final network.

We trained our models using an NVIDIA GeForce RTX 2080 GPU with 12GB memory, and the CUDA version is 11.4. For RO-PolyGCN, the experiments were done using Python 3.6 and Pytorch 0.4.0. The FAE-PolyGCN and AT-PolyGCN utilized Python 3.9 and Pytorch 1.6.1, while for DeCoupled-PolyGCN, we used Python 3.7 and Pytorch 1.7.0. All the codes will be soon publicly available on our GitHub site.

8.2 Performance Evaluation Metrics

The results of our study are qualitatively analyzed using a combination of raster-based and vector-based metrics, which allow for a comprehensive evaluation of our models' performance in building extraction tasks. By leveraging these dual metric approaches, we effectively

assess our model’s capability to automate the extraction process while managing the complex variability present in urban environments. These include automating the extraction process, managing diverse building roof appearances, balancing recognition and localization accuracy, differentiating closely spaced buildings, detecting buildings of varying sizes, and precisely capturing the geometric shapes of building polygons. Our metrics showcase the robustness of our approach across a range of complex urban scenarios. Together, these metrics highlight the versatility and robustness of our method, validating its adaptability and precision across a broad spectrum of urban scenarios, from densely packed areas to regions with mixed building scales and sizes. We have evaluated our results based on raster- and vector-based metrics, which are discussed in the sections below.

8.2.1 Raster-Based Metrics

Raster-based building evaluation metrics are quantitative measures used to assess the performance of building extraction models by comparing the predicted building masks with the ground truth masks at the pixel level. These metrics focus on how accurately the model identifies building pixels within raster images (pixelated data), providing insights into the pixel-wise classification performance of the model. Raster-based metrics focus on pixel-level accuracy and provide insights into how well our model captures building extents and shapes in a continuous format. For that purpose, we need to discuss IoU, Precision, and Recall.

Intersection over Union (IoU) is a widely used metric in object detection and segmentation tasks to evaluate the accuracy of object localization by measuring the overlap between a model’s predicted bounding box (or segmentation mask) and the ground truth box (or mask). It is defined as:

$$\text{IoU} = \frac{\text{Area of Overlap } (A \cap B)}{\text{Area of Union } (A \cup B)} \quad (8.1)$$

Where A and B represent the predicted and ground truth building polygons, respectively, the Area of Overlap is the region where both polygons intersect, and the Area of Union is the

combined area covered by both polygons without double-counting the overlap. Let's suppose Predicted Box is A and Groundtruth is B then IOU can be explained by Figure 8.1

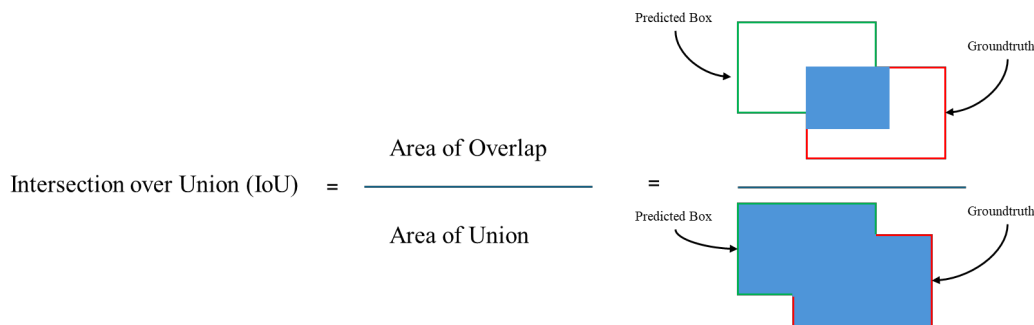


Figure 8.1 Intersection of Union (IoU)

An IoU of 1 indicates perfect overlap, meaning the model has precisely localized the object, while an IoU of 0 indicates no overlap. IoU is often used as a threshold to classify a prediction as a True Positive (TP). The typical threshold is $IoU \geq 0.5$, which is often used as a baseline threshold; predictions are considered TPs if their IoU with the ground truth is 0.5 or higher.

Understanding True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) help assess model performance in detection and classification tasks:

True Positive (TP): A correct positive prediction, e.g., detecting an object with an IoU that meets or exceeds the threshold.

False Positive (FP): An incorrect positive prediction, e.g., detecting an object that either doesn't exist or doesn't meet the IoU threshold.

True Negative (TN): A correct negative prediction, e.g., correctly identifying the absence of an object (less relevant in object detection).

False Negative (FN): A missed positive prediction, where an actual object is either undetected or has an IoU that is too low.

From these terms, we derive Precision and Recall:

Precision measures the accuracy of positive predictions. High precision means that most detected objects are correct, with fewer false positives. This is especially important in fields

		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Figure 8.2 Understanding True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN)

where false positives are costly. It is calculated as:

$$Precision = \frac{TP}{TP + FP} \quad (8.2)$$

Recall measures the model's ability to capture all relevant positives. High recall means the model detects most actual objects with fewer false negatives. This is essential in scenarios where missing a detection has serious consequences. It is calculated as:

$$Recall = \frac{TP}{TP + FN} \quad (8.3)$$

Together, precision and recall provide a balance in evaluating a model's performance. Precision reflects the accuracy of positive predictions, while recall shows how many actual objects were detected.

F1 score is a machine learning evaluation statistic that measures the predictive ability of a model by analyzing its performance on individual classes rather than considering total performance like accuracy does. The F1 Score combines precision and recall into a single value, which is particularly useful when both false positives and false negatives are costly. It is calculated as the harmonic mean of precision and recall:

$$F1 - Score = \frac{2 \times (Precision + Recall)}{Precision \times Recall} \quad (8.4)$$

The F1 Score ranges from 0 to 1, where 1 indicates perfect precision and recall. This metric balances both values, penalizing models with low precision or recall, and is valuable in cases of class imbalance or when both types of errors (false positives and false negatives) carry significant costs.

8.2.2 Vector-Based Metrics

Vector-based building evaluation metrics assess the quality of building extraction by analyzing the geometric accuracy of predicted building polygons rather than pixel-based masks. These metrics focus on evaluating the alignment, shape, and structural accuracy of predicted polygons against ground truth polygons, making them particularly valuable for applications requiring precise building boundaries and shapes, such as urban planning and mapping. Vector-based metrics are especially effective in applications where boundary precision, shape preservation, and structural fidelity are important. These metrics allow for a nuanced understanding of model performance, revealing how well the extracted polygons align with real building shapes, edges, and structures in complex urban landscapes.

PoLiS (Polygonal Line String Similarity) is a vector-based metric designed to measure the similarity between predicted and ground truth building polygons, particularly in building extraction tasks. It emphasizes shape and structural alignment over pixel accuracy, making it ideal for assessing the geometric precision of building boundaries.

The Polygonal Line String Similarity (PoLiS) metric calculates the shortest distances

between each point on the predicted polygon and the nearest point on the ground truth polygon, and vice versa. This two-way distance calculation ensures that both shapes are well-represented, minimizing bias if one polygon is more complex than the other. PoLiS quantifies alignment and shape similarity between polygons by evaluating discrepancies in spatial arrangement and structure. This sensitivity to shape differences makes PoLiS especially valuable in applications requiring accurate boundary reproduction for building outlines.

PoLiS measures both the positional and boundary similarity of the predicted and ground truth polygons, considering the entire perimeter of both polygons. It is calculated as the average distance between the matched vertices in polygons P and Q:

$$\text{PoLiS}(P, Q) = \frac{1}{2N_P} \sum_{p_j \in P} \min_{q \in Q} \|p_j - q\| + \frac{1}{2N_Q} \sum_{q_k \in Q} \min_{p \in P} \|q_k - p\| \quad (8.5)$$

MaxTangent (Maximum Tangent Angle Error) is a vector-based evaluation metric used in building extraction tasks to measure the angular alignment between predicted and ground truth building polygon boundaries. By comparing the tangent angles of corresponding segments in both polygons, MaxTangent evaluates how accurately the predicted shape aligns with the true geometry, especially regarding edge orientation and corner preservation.

The metric calculates the absolute difference in tangent angles for each boundary segment, capturing deviations in segment orientation. The maximum angle difference among all segments is recorded as the MaxTangent value, representing the worst-case angular misalignment. This focus on boundary orientation and alignment makes MaxTangent effective for evaluating structural fidelity, as it is sensitive to corner preservation, edge orientation, and boundary regularity.

Unlike area-based or distance-based metrics, MaxTangent specifically targets angular alignment, highlighting misaligned edges and discrepancies at sharp corners. This sensitivity to angle accuracy is particularly useful for ensuring that models maintain architectural features and the structural layout of buildings with defined regular edges. For each edge in P

and Q , calculate the tangent vector as the difference between two consecutive vertices: The tangent error at each point p_i is given by the distance between the tangent vectors of p_i and its closest point q . The MaxTangent metric is then the maximum of these distances over all vertices p_i in P .

8.3 Experimental Results and Discussion

As discussed in the previous sections, we proposed four models, which were trained and evaluated on two distinct datasets, SpaceNet-2 and WHU. We utilized two diverse datasets to ensure the robustness and generalization of our algorithms and networks. In this section, we first compare the performance of all proposed models with the ground truth and the State of the art (SOTA) building reconstruction network, HiSup [110]. Next, we will discuss the step-by-step progression in the development of our models and how the performance of our network improved incrementally with the addition of each module. Following this, we analyze and present the qualitative results of our most effective model, Decoupled-PolyGCN. We examine its performance across diverse scenarios, including variations in area types, closely situated buildings, structures of different sizes, and geometric shapes. We also display cases involving occlusions and complex building structures. The results for both the WHU and SpaceNet-2 (Vegas) datasets will be discussed and visually demonstrated. In the end, we will discuss the performance of all our networks and their contributing factors towards performance enhancement.

8.3.1 Qualitative Results

Let us delve deeper into the qualitative results. Figure 8.3 provides a comprehensive analysis and comparison of the outcomes from all our models against the ground truth and the baseline R-PolyGCN model. The figure highlights the progression and improvement in performance across our models. Our top-performing model, Decoupled-PolyGCN, produces building footprints that are more accurate, regularized, and closely aligned with the ground truth. In

some of the cases, our best model can predict building boundaries even though there are incorrect data in the groundtruth annotation.



Figure 8.3 Overview of results from all of our Networks with Baseline(R-PolyGCN) and Groundtruth

Now, we showcase the progression and performance improvements achieved as we systematically integrated various modules into our models. Starting with incorporating orthogonality loss. Then, we introduced the feature augmentation module and edge-preserving mask head. After that, we integrated attraction field maps. Finally, we adopted decoupled object detec-

tion to enhance performance further. Each module contributed significantly to the overall accuracy and robustness of the network and showed measurable impact at every stage.

The baseline model, R-PolyGCN, produced irregular building boundaries. To produce more regularized building boundaries, we embedded geometric regularity by adding orthogonality loss in our first network, RO-PolyGCN. We further enhanced the baseline results by carefully tuning the hyperparameters. This network produced much more regularized boundaries, and the performance of the medium-sized building also improved. The figure 8.4 shows the comparison of baseline, R-PolyGCN, and our first network, RO-PolyGCN. However, we still faced issues in detecting small and large buildings.

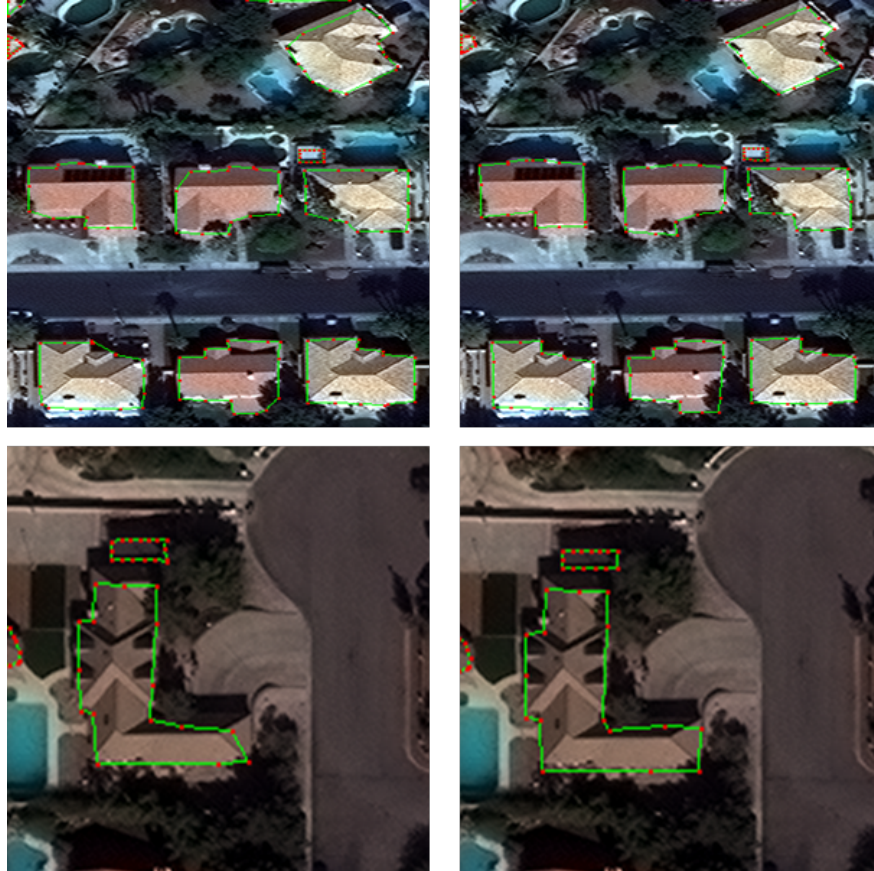


Figure 8.4 Comparison of our baseline, R-PolyGCN, and our first model, RO-PolyGCN. The left side shows Results from R-PolyGCN, and the right side shows Results from RO-PolyGCN

To further improve our network’s performance and to predict the small buildings more

efficiently, we introduced our second network, FAE-PolyGCN. This network incorporates the Feature Augmentation Module to integrate high and low-resolution features. Also, it adds an edge-preserved mask branch to capture boundary details better. FAE-PolyGCN demonstrated significant improvements, particularly in detecting small and medium-sized buildings, leading to an overall enhancement in performance. The Fig 8.5 highlights the advancements achieved through these enhancements. We can see that the small buildings not predicted in our first network are detected in our second network. However, challenges remained with accurately handling large buildings, which continued to affect the overall results.



Figure 8.5 Comparison of Results between our first model, RO-PolyGCN and our second model, FAE-PolyGCN. The left side shows Results from RO-PolyGCN and the right side shows Results from FAE-PolyGCN

To address the challenges associated with large buildings, we developed AT-PolyGCN. This network integrates Attraction Field Maps to provide orientation information to the model. Combined with the Feature Augmentation Module and Orthogonality Loss, AT-PolyGCN effectively enhances the network’s ability to accurately predict large building footprints. This integration leads to significant improvements, as the network now handles large buildings more efficiently. Additionally, overall performance improved across both vector-based and raster-based metrics. The figure8.6 illustrates the comparison between FAE-PolyGCN and AT-PolyGCN, showcasing how AT-PolyGCN achieves superior results and enhances overall accuracy. It clearly shows improved performance over large buildings, and the building polygons are much more regularized.

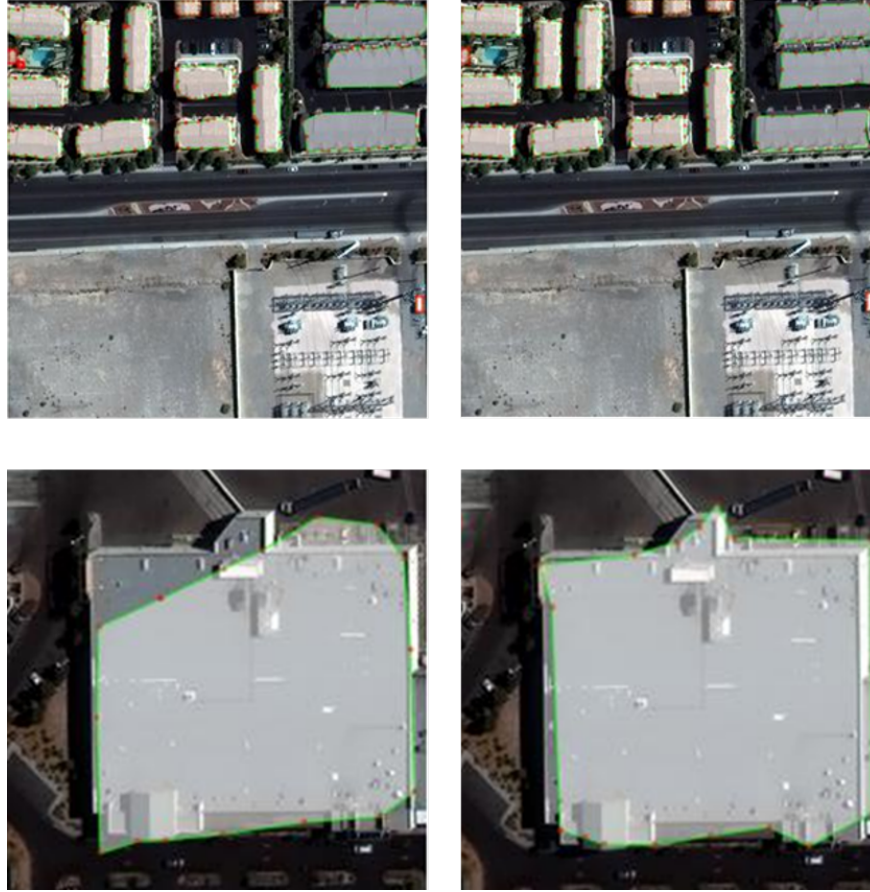


Figure 8.6 Comparison of Results between our second model, FAE-PolyGCN and our third model, AT-PolyGCN. The left side shows Results from FAE-PolyGCN and the right side shows Results from AT-PolyGCN

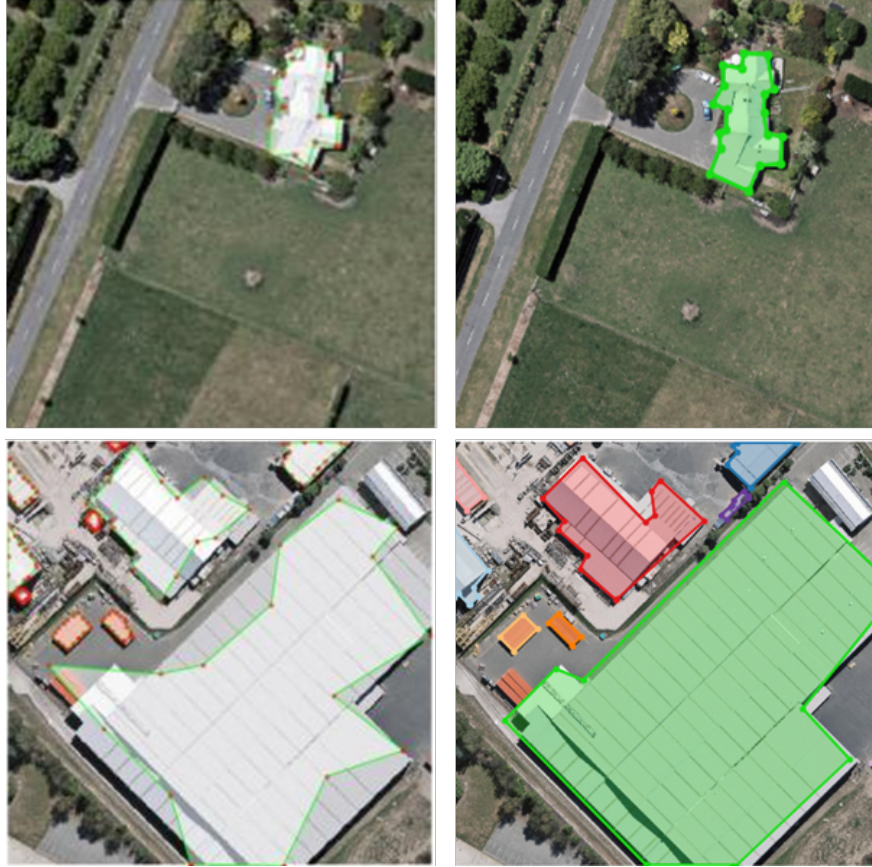


Figure 8.7 Comparison of Results between our third model, AT-PolyGCN and our last model, Decoupled-PolyGCN. The left side shows Results from AT-PolyGCN and the right side shows Results from DeCoupled-PolyGCN

Despite incorporating modules such as orthogonality loss, the Feature Augmentation Module (FAM), and the Attraction Field Map (AFM), our network encountered challenges. These challenges primarily occur due to the backbone network’s inability to provide the Graph Convolutional Network GCN with optimal initial features. To overcome this limitation, we adopted a decoupled approach for object detection using the CO-DETR network. Furthermore, the fixed number of vertices used in previous models restricted flexibility. Therefore, we implemented a dynamic vertex initialization strategy. This approach allows the number of vertices to adapt to the complexity of each building. With these improvements, our final network, Decoupled-PolyGCN, achieved a significant boost in performance. Figure 8.7 illustrates the comparison between AT-PolyGCN and Decoupled-PolyGCN, highlighting the

latter’s superior accuracy and regularity, delivering the best results overall.



Figure 8.8 Overview of Results From our best model, Decoupled-PolyGCN

Our final model, Decoupled-PolyGCN, demonstrates exceptional performance across various scenarios. Figure 8.8 provides an overview of results obtained from Decoupled-PolyGCN in different scenes, clearly showcasing the high quality and robustness of the outcomes. In the figure, it is displayed how precisely and accurately Decoupled -PolyGCN predicts building boundaries for both small and large buildings at the same time. The prediction for complex shapes of buildings has been improved as well. Along with that, the model is efficient in a way to capture the adjacent buildings and buildings on the corners as well. The detailed visual results are shown and discussed below. We start with its comparison with the SOTA model Hisup [110] and then explain the model’s effectiveness over different complex scenarios.



Figure 8.9 Comparison of Results with SOTA Model, HiSup. The top row indicates prediction from HiSup Model, middle row displays the boundary polygons predicted by DeCoupled-PolyGCN, and the last row is the ground truth.

Figure 8.9 presents a comparison between our proposed Decoupled-PolyGCN and the SOTA model HiSup [110]. The results clearly show that DeCoupled-PolyGCN significantly outperforms HiSup, delivering superior performance across various scenarios. The HiSup model struggles with complex-shaped buildings and fails to effectively distinguish closely

located structures. DeCoupled-PolyGCN, on the other hand proves to be more robust and versatile, performing consistently well under diverse conditions and addressing these challenges with greater accuracy and precision.

We thoroughly analyzed and refined our network to ensure it performs robustly across diverse scenarios, addressing various challenges presented by different datasets and building characteristics. Below, we discuss the key capabilities of our model and how it effectively handles diverse conditions, as demonstrated in the results.



Figure 8.10 Handling Diversity of Buildings

Handling the Diversity of Buildings: The datasets used in our study include a wide range of urban and rural scenarios, including scenes with sparse building distributions to highly dense areas containing numerous buildings in a single image. This diversity posed a

challenge in designing a model that could generalize well across different conditions. Our network, as demonstrated in Figure 8.10, performs consistently across all types of scenes. The model efficiently detects and segments building footprints, maintaining high accuracy and reliability in both low-density rural and highly populated urban areas.



Figure 8.11 Capturing buildings of varying sizes

Capturing Buildings of All Sizes: Detecting buildings of varying sizes, particularly smaller buildings, has traditionally been challenging. In our initial models, smaller buildings were often missed entirely or detected partially due to limited feature representation. However, as shown in Figure 8.11, our enhanced network has significantly improved in this aspect. Smaller buildings are now detected with greater efficiency and precision, thanks to improvements such as better feature augmentation and the inclusion of high-resolution features. Additionally,

the detection rate for larger buildings has also improved substantially with the addition of Attraction field maps, with the model consistently ensuring accurate boundary delineation.

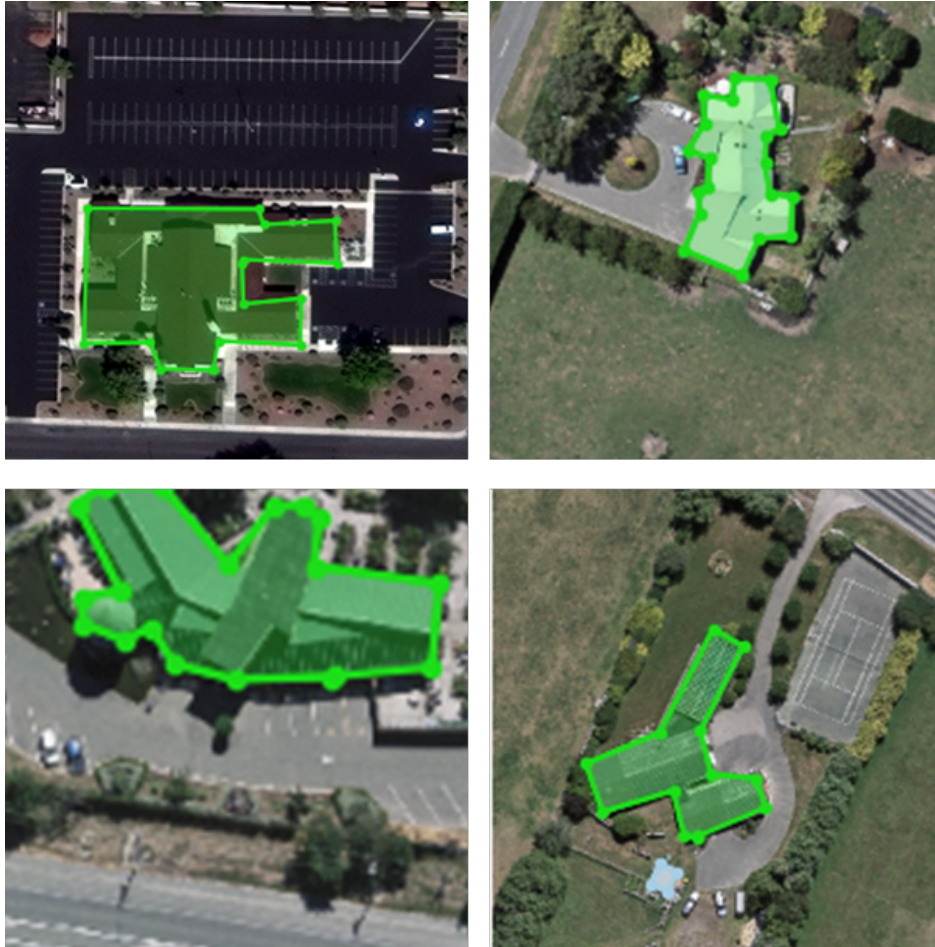


Figure 8.12 Detecting complex building shapes

Detecting Buildings with Complex Geometric Shapes: Buildings often exhibit diverse geometric shapes, ranging from simple rectangular structures to complex polygons with multiple vertices. Our model, equipped with geometric learning capabilities and an orthogonality loss function, excels at detecting buildings with intricate shapes and varying numbers of vertices. As illustrated in Figure 8.12, we have displayed buildings of very complex shapes and our model accurately captures the overall structure of buildings while maintaining regularized boundaries. This is a major improvement as our previous networks performed well over the buildings with less vertices, but the prediction rate dropped over the complex building shapes.

Distinguishing Adjacent buildings: One of the major challenges in building segmentation is distinguishing buildings that are located very close to each other or extracting the boundaries of buildings situated on the edges of images. Overlapping or closely positioned buildings often lead to misclassification. That eventually results in poor performance. However, as demonstrated in Figure 8.13, our network, Decoupled-PolyGCN effectively overcomes these challenges. By leveraging Attraction Field Maps and high-quality feature representations, the model maintains distinct boundaries for adjacent buildings and accurately segments structures on image edges. This ensures that all buildings are detected as separate entities with clear and precise boundaries.



Figure 8.13 Distinguishing adjacent buildings and buildings at the edges

Extracting occluded buildings: Occlusion is a common problem in satellite imagery,

where parts of buildings may be obscured by trees, shadows, or other structures. Detecting the complete boundaries of occluded buildings is critical for achieving accurate building footprint extraction. Our network, as shown in Figure 8.14, successfully handles occluded buildings by predicting the full structure even when parts are not visible. This is achieved through the model’s ability to learn from contextual information and its geometric regularization techniques.



Figure 8.14 Extracting occluded buildings

These results from our best model, visually demonstrate the versatility and robustness of our network in handling a wide range of scenarios. From diverse building distributions and varying sizes to complex shapes and occlusions, the model consistently delivers high-quality results. Its ability to distinguish closely located buildings, accurately capture edge cases and

reconstruct occluded structures highlights its adaptability and reliability. These advancements make our network a powerful tool for building footprint extraction, addressing both common and complex challenges in urban and rural mapping applications.

Similarly, We have thoroughly analyzed and presented the results of all our networks and showed how the different modules enhanced the performance sequentially and eventually our best model, Decoupled-PolyGCN proved to go past the State of the art methods. We have displayed all the results visually. In the next section we will demonstrate our quantitative results.

8.3.2 Quantitative Results

To evaluate the performance of our proposed models, we conducted a comprehensive quantitative analysis using both raster-based and vector-based metrics. This dual evaluation approach highlights the effectiveness of our models not only in pixel-level segmentation but also in accurately capturing the geometric regularity of building footprints. Our results demonstrate the consistent progression of performance across the proposed models, with each new module contributing to improved accuracy and boundary regularization.

Table 8.1 Overview of Results on WHU Dataset. Lower values of POLIS and Max Tangent Angle (MTA) are better.

Networks	AP(%)	AR(%)	PoLiS ↓	MTA ↓
R-PolyGCN	45.7	57.5	2.58	45.23
RO-PolyGCN	46.1	58.3	1.96	43.88
FAE-PolyGCN	53.0	61.4	1.90	43.35
AT-PolyGCN	55.3	62.5	1.46	42.16
DeCoupled-PolyGCN	65.7	74.5	1.22	37.95

Table 8.1 presents the performance of our models on the WHU dataset, highlighting consistent improvements across all evaluated metrics. Each successive module contributes to enhancing the network’s overall effectiveness. Our final model, Decoupled-PolyGCN, achieves the best results, delivering highly regularized building boundaries and setting new benchmarks for state-of-the-art performance. There is a 20% improvement in AP and 17% in

AR in DeCoupled-PolyGCN as compared to the baseline model, R-PolyGCN. Notably, lower values of PoLiS and MTA indicate better performance. The results clearly show a progressive decline in these values with each model iteration, highlighting the steady advancements in our network’s geometric capabilities capabilities.

Similarly, Table 8.2 shows the quantitative results for our models on the SpaceNet-2 (Vegas) dataset. As with the WHU dataset, our models consistently improve with each added module. DeCoupled-PolyGCN, in particular, stands out by delivering highly accurate segmentation results and regularized polygonal boundaries, making it our most robust and efficient model. There is a 15% improvement in AP and 12% in AR in DeCoupled-PolyGCN as compared to the baseline model, R-PolyGCN. As discussed earlier, lower values of PoLiS and MTA indicate better performance. The results clearly show a continuous decrease in these values with each model iteration. This showcases the improvement in geometric capabilities of our network.

Table 8.2 Overview of Results on SpaceNet-2 Dataset. Lower values of POLIS and MTA are better.

Networks	AP(%)	AR(%)	PoLiS ↓	MTA ↓
R-PolyGCN	42.1	53.5	3.51	44.65
RO-PolyGCN	44.5	55.1	3.25	41.27
FAE-PolyGCN	50.4	57.0	3.20	41.19
AT-PolyGCN	52.0	58.2	3.12	40.35
DeCoupled-PolyGCN	57.8	65.5	2.78	36.51

Our experiments were conducted on highly diverse datasets, encompassing buildings of all sizes: small, medium, and large. As discussed in Section 7.3, and following the COCO definition, small buildings are those with an area of less than 32 square pixels, medium buildings have areas between 32 and 96 square pixels, and large buildings have areas greater than 96 square pixels. A detailed analysis was performed based on building size categories. After incorporating the Feature Augmentation Module (FAM), our network showed improved performance on small buildings. The addition of Attraction Field Maps (AFM) introduced oriented information, significantly enhancing performance on large buildings, although it had

no notable effect on small and medium buildings. In our final network, the adoption of a decoupled strategy led to considerable improvements for both small and large buildings. This indicates that better segmentation and the reduction in missed corners significantly enhanced the overall performance of our network.

Table 8.3 Overview of Area distribution based Results on WHU Dataset.

Networks	AP_S	AP_M	AP_L	AR_S	AR_M	AR_L
RO-PolyGCN	37.4	55.5	40.6	42.2	76.5	56.6
FAE-PolyGCN	46.1	62.9	41.3	47.9	77.6	60.4
AT-PolyGCN	45.9	66.9	48.6	48.8	79.5	69.4
DeCoupled-PolyGCN	55.0	77.2	64.7	64.8	85.9	78.3

Table 8.3 provides an overview of the results for different building size distributions on the WHU dataset. Similar experiments and analyses were conducted on the SpaceNet-2 dataset, as shown in Table 8.4. The observed patterns were consistent: FAM improved performance for smaller buildings, AFM enhanced performance for larger buildings, and the decoupled strategy resulted in significant improvements across all size categories, particularly for small and large buildings.

Table 8.4 Overview of Area distribution based Results on SpaceNet-2 Dataset.

Networks	AP_S	AP_M	AP_L	AR_S	AR_M	AR_L
RO-PolyGCN	18.7	58.3	31.8	28.6	67.7	46.8
FAE-PolyGCN	25.8	61.9	32.8	33.1	68.2	47.6
AT-PolyGCN	25.2	63.8	46.8	33.2	69.7	57.5
DeCoupled-PolyGCN	28.7	68.9	59.5	41.2	76.7	69.2

Both datasets used in our experiments feature buildings with a wide range of complexities, represented by as few as 3 vertices and up to 73 vertices. To analyze performance across varying building shapes, we categorized the buildings into four groups based on the number of vertices: ≤ 4 , $5 - 8$, $9 - 16$, and > 16 .

In our initial models, we used a fixed initialization of 16 vertices for the graph, which led to suboptimal performance, particularly for buildings with more than 16 vertices in the ground truth. To address this limitation, our final network, Decoupled-PolyGCN, introduced

Table 8.5 Overview of Vertex Distribution based Results on WHU Dataset.

Networks	$AP_{\leq 4}$	AP_{5-8}	AP_{9-16}	$AP_{>16}$	$AR_{\leq 4}$	AR_{5-8}	AR_{9-16}	$AR_{>16}$
FAE-PolyGCN	40.1	57.7	60.2	49.9	52.7	63.6	64.0	57.6
AT-PolyGCN	39.8	57.1	61.6	54.8	52.0	64.6	67.5	60.6
DeCoupled-PolyGCN	50.0	60.9	66.7	79.0	61.3	70.1	79.5	81.0

a variable initialization technique. This adjustment significantly improved performance for buildings with more than 16 vertices and also enhanced results for buildings with 4 or fewer vertices. Tables 8.5 and 8.6 present the vertex-wise performance comparisons on the WHU and SpaceNet-2 datasets, respectively, demonstrating the impact of this dynamic approach.

Table 8.6 Overview of Vertex Distribution based Results on SpaceNet-2 Dataset.

Networks	$AP_{\leq 4}$	AP_{5-8}	AP_{9-16}	$AP_{>16}$	$AR_{\leq 4}$	AR_{5-8}	AR_{9-16}	$AR_{>16}$
FAE-PolyGCN	36.9	50.4	67.3	38.2	35.6	53.6	66.2	57.6
AT-PolyGCN	36.2	51.4	68.6	43.9	34.9	53.8	66.5	59.5
DeCoupled-PolyGCN	41.9	52.7	69.4	68.4	50.0	61.7	72.2	69.6

We conducted a comprehensive evaluation of our models alongside a comparison with several SOTA methods from the literature. For this, we selected both two-step and direct polygonal segmentation approaches to compare against our model. Table 8.7 provides a detailed overview, showcasing how our Decoupled-PolyGCN model surpasses existing methods. Notably, our model achieves a 5% improvement in AP and a 7% increase in AR, representing a significant performance improvement. This comparison clearly shows the advancements made by our model in terms of both accuracy and geometric regularity. Furthermore, it highlights that our approach consistently outperforms both two-step and direct polygonal segmentation SOTA methods, demonstrating its effectiveness and robustness.

Alongside the Vegas dataset, we carried out experiments on the WHU dataset. We compared our results with the two-step and direct polygonal methods. The comparison is shown in table 8.8 , which clearly demonstrates our network to be superior to the SOTA method (HiSup). It improves 6% in AP and 10% in AR.

Table 8.7 Comparison of DeCoupled-PolyGCN with competing methods on SpaceNet-2 Dataset

Networks	AP(%)	AR(%)
MaskRCNN [77]	43.2	52.5
R-PolyGCN [42]	42.1	53.5
PolyMapper [127]	49.6	55.0
FrameField [109]	52.1	57.3
HiSup [110]	52.5	58.3
Ours(DeCoupled-PolyGCN)	57.8	65.5

Table 8.8 Comparison of DeCoupled-PolyGCN with competing methods on WHU Dataset

Networks	AP(%)	AR(%)
MaskRCNN [77]	45.9	55.4
R-PolyGCN [42]	45.7	57.5
HiSup [110]	59.9	64.0
Ours(DeCoupled-PolyGCN)	65.7	74.5

8.4 Discussions

We began our research by utilizing R-PolyGCN as the baseline model, systematically building upon it to improve network performance. Over time, multiple modules were added to address challenges and enhance the network’s capability. Continuous analysis of the network was conducted under diverse experimental settings. The datasets chosen for our experiments were highly diverse, featuring buildings of varying sizes and shapes, with scenes ranging from dense rural areas to crowded urban environments. These presented significant challenges, requiring a robust solution capable of generalizing across all conditions.

Size-Based Analysis: The performance of our network varied significantly depending on the size of the buildings. Smaller buildings presented notable challenges for segmentation due to their fine details, often leading to reduced average precision scores. Errors were most frequent in buildings ranging from 50 to 200 square pixels, where structures were either missed entirely or incorrectly merged with adjacent buildings. Medium-sized buildings, which represent nearly 70% of the dataset, exhibited the best overall performance across all models. However,

larger buildings, though less common in the datasets, still demonstrated a performance gap.

Our baseline model, RO-PolyGCN, faced significant difficulty in detecting smaller buildings. The integration of the Feature Augmentation Module (FAM) in FAE-PolyGCN addressed these challenges by enhancing high-resolution feature representation. This resulted in a marked improvement in the detection of small buildings, with an 8% increase in AP and a 5% increase in AR for smaller buildings compared to RO-PolyGCN, as shown in Tables 8.4 and 8.3. However, FAE-PolyGCN still struggled with larger buildings. The introduction of Attraction Field Maps (AFMs) in AT-PolyGCN further enhanced the network’s capability by encoding orientation information, which proved crucial for improving the segmentation of large buildings. While AFMs had little impact on small and medium buildings, their contribution significantly elevated the performance for larger structures. AT-PolyGCN achieved a 14% improvement in AP and a 10% improvement in AR for large buildings compared to FAE-PolyGCN. Finally, the adoption of a decoupled object detection strategy in Decoupled-PolyGCN led to substantial advancements for both small and large buildings. The dynamic graph initialization in this model effectively reduced missed corners, ensuring better segmentation. For small buildings, Decoupled-PolyGCN demonstrated a 3% improvement in AP and an 8% improvement in AR over AT-PolyGCN. Similarly, for large buildings, it achieved a 13% increase in AP and a 12% increase in AR, showcasing its ability to handle diverse building sizes effectively. These results are visually represented in Figure 8.11, highlighting the network’s capability to segment both small and large buildings with improved precision and accuracy.

Vertex-Based Analysis: The datasets in this study primarily consisted of buildings with 5–8 or 9–16 vertices, with fewer examples having less than 4 or more than 32 vertices. For our experimental evaluation, we categorized the buildings into four vertex-based categories (≤ 4 , $5 - 8$, $9 - 16$, and > 16). In our first three models, we used a fixed vertex count of 16 for graph initialization. While this approach worked well for buildings with 9–16 vertices, it often resulted in distortions due to over-sampling or under-sampling for other categories, negatively impacting shape accuracy. For instance, buildings with fewer than 4 vertices were particularly

susceptible to IoU degradation when corners were missed. Similarly, smaller buildings with more than 16 vertices suffered from inaccuracies in polygon predictions, as the fixed vertex count failed to capture their geometric complexity. Conversely, buildings with higher vertex counts required precise handling to prevent the accumulation of vertex placement errors, which could increase PoLiS values. We experimented with different initialization values like 8, 32 and 96. These values were chosen as they are commonly used for fixed initialization in different scenarios. Although initializing with 96 vertices proved very good for complex buildings but in general for fixed number of vertices, we found out that 16 was the best choice

Our final model, Decoupled-PolyGCN, introduced dynamic graph initialization. This method allows the number of vertices to adapt based on the complexity of the building. This results in polygons with more accurate and regularized boundaries. This improvement is evident in the quantitative as well as qualitative results. As shown in Figure 8.12, our dynamic approach significantly enhanced the model’s performance for buildings with both lower and higher vertex counts. Table 8.5 highlights the improvements observed in the WHU dataset. For buildings with ≤ 4 vertices, the Average Precision AP and Average Recall AR increased by 10% when comparing our model FAE-PolyGCN to Decoupled-PolyGCN. Similarly, for buildings with > 16 vertices, AR showed a substantial improvement of 24%. On the SpaceNet-2 dataset, as detailed in Table 8.6, Decoupled-PolyGCN delivered even more remarkable improvement, with AP for buildings with > 16 vertices increasing by 30% and AR by 12% compared to FAE-PolyGCN. These results show the effectiveness of our dynamic graph initialization strategy. This enables Decoupled-PolyGCN to handle buildings with diverse vertex counts more accurately and efficiently. This adaptability ensures improved performance across varying building complexities and shapes.

Proximity-Based Analysis: Dense urban environments, characterized by closely spaced buildings, presented another layer of complexity. In many such cases, the boundaries of adjacent buildings frequently overlapped, resulting in segmentation errors where two buildings were incorrectly merged into one. Fig 7.17 shows that there is ample amount of building who share corners with each other or the buildings which lie at the boundary of the image. This

issue was particularly found in earlier models, which lacked the precision required for fine-grained boundary delineation and affected the overall performance of the network. However, Our final model, Decoupled-PolyGCN, leveraged the CO-DETR detection network, which significantly improved bounding box predictions. These enhanced bounding boxes provided better initialization features for the Graph Convolutional Network GCN, enabling the network to distinguish closely located buildings with much greater accuracy. This improvement in graph initialization directly translated to more precise and well-separated polygon predictions, even in highly dense urban scenes. In Figure 8.13, our model DeCoupled-PolyGCN demonstrates great performance and capturing individual buildings effectively. Dense urban environments, characterized by closely spaced buildings, introduced an additional layer of complexity to our segmentation task. In such cases, the boundaries of adjacent buildings often overlap, leading to segmentation errors where two or more buildings are incorrectly merged into one structure. Figure 7.17 illustrates the prevalence of buildings that share corners or lie along the image boundaries. This issue was particularly evident in our earlier models, which lacked the precision necessary for fine-grained boundary delineation. It negatively impacted the overall performance of the network.

Our third model, AT-PolyGCN tackled this problem by adding orientation information. Ultimately, our final model, Decoupled-PolyGCN, effectively addressed these challenges further by leveraging the CO-DETR detection network. This significantly enhanced the quality of bounding box predictions. These high-quality bounding boxes provided superior initialization features for the Graph Convolutional Network GCN, allowing it to accurately distinguish between closely located buildings. This improvement in graph initialization directly translated into more precise and well-separated polygon predictions, even in densely packed urban scenes. As shown in Figure 8.13, Decoupled-PolyGCN demonstrates exceptional performance in accurately segmenting individual buildings even they are in close proximity with other buildings or buildings with shared corners. The model effectively resolves issues of overlapping boundaries, capturing distinct structures even in highly complex and dense urban settings, showcasing its robustness and reliability in real-world scenarios.

Incomplete or Erroneous Annotations: A recurring challenge in our experiments was the presence of very small, truncated buildings near the image boundaries. These structures often led to degraded performance due to incomplete or incorrect annotations, where both the network and human annotators struggled to accurately define the buildings’ shapes. To evaluate the impact of such annotations, we conducted several experiments by excluding these truncated buildings from evaluation. This approach led to noticeable improvements in performance metrics, such as PoLiS, which measures the geometric distance between predicted and ground truth polygons, as well as AP and AR scores. There were also quite a few errors of a large building in close proximity with a very small buildings and being annotated in the ground truth as one building. However, the results presented in all the tables in this thesis include evaluations conducted with the incorrect annotations to provide a fair comparison across all models and scenarios. Despite this, our final model, Decoupled-PolyGCN, demonstrated remarkable robustness, correctly predicting many building footprints even when the ground truth annotations were inaccurate. This highlights the reliability and adaptability of our model in handling challenging and imperfectly annotated datasets.

We performed extensive analysis and experimentation over different datasets. We demonstrated the progressive development of our models, resulting in the robust and highly accurate performance of our final network, Decoupled-PolyGCN. Starting with the baseline model, R-PolyGCN, we systematically identified key limitations. We then step-by-step addressed the limitations through the integration of innovative modules and advanced techniques. Each iteration of the network incrementally improved its ability to accurately extract building footprints while maintaining geometric regularity and adapting to diverse urban and rural scenarios. The integration of orthogonality loss, the Feature Augmentation Module, Attraction Field Maps, and a decoupled object detection strategy contributed to consistent performance gains. This helped in overcoming challenges associated with varying building sizes, complex geometries, and densely packed urban scenes. The final model, Decoupled-PolyGCN, surpassed the performance of existing SOTA methods. This systematic progression and iterative refinement have resulted in a robust and highly efficient framework for building footprint

extraction. This paves the way for future advancements in urban mapping, GIS applications, and remote sensing technologies.

8.5 Summary

This chapter presented the configuration and implementation details of our networks, evaluated their performance using quantitative and qualitative metrics, and compared them with SOTA models. We utilized two diverse datasets, SpaceNet-2 and WHU, to ensure the robustness and generalization of our proposed models. We employed both raster-based and vector-based metrics for performance evaluation. These dual metrics allowed us to comprehensively assess the capability of our networks in handling complex urban scenarios, including occlusions, varying building sizes, and adjacent structures. Quantitative results demonstrated consistent improvements across our models. The final model, Decoupled-PolyGCN, achieved significant gains, with a 6% improvement in mean Average Precision AP and a 10% improvement in mean Average Recall AR over the SOTA model, HiSup on the WHU dataset. On the SpaceNet-2 dataset, Decoupled-PolyGCN outperformed competing SOTA methods by 5% in AP and 7% in AR delivering accurate segmentation results with well-regularized building polygons. Qualitative analysis further highlighted the robustness of our models in diverse scenarios. Decoupled-PolyGCN excelled in detecting buildings of varying sizes and shapes, distinguishing adjacent structures, handling occlusions, and segmenting buildings on image edges. Advanced features such as Attraction Field Maps, dynamic polygon initialization, and orthogonality loss proved crucial in improving geometric regularity and segmentation accuracy.

Chapter 9

Conclusions and Future Work

9.1 Conclusion

This thesis presented a comprehensive study on building footprint extraction and reconstruction from satellite imagery. It addresses the key challenges of boundary regularization, geometric accuracy, and scalability in diverse urban environments. By integrating novel deep learning techniques and graph-based approaches, we developed a series of progressively refined models that significantly enhance the accuracy and reliability of automated building footprint segmentation.

We began by identifying the limitations of existing methods. We proposed a series of incremental improvements across multiple network designs to address these issues. Each iteration introduced innovative modules and techniques, including orthogonality loss, the Feature Augmentation Module FAM, edge-preserving mask head, and Attraction Field Maps, which collectively enhanced the networks' geometric learning and segmentation capabilities.

Our first proposed model, RO-PolyGCN, employed a Mask R-CNN-based backbone combined with an orthogonal loss function and a Graph Convolutional Network GCN to iteratively refine polygon predictions. This model improved Average Precision and Average Recall over the baseline by nearly 1%, with particularly significant gains in vector-based metrics. Despite these improvements, the reliance on Mask R-CNN constrained its ability to produce highly accurate initial polygonal representations, prompting further enhancements

in subsequent models.

To overcome the limitations observed in RO-PolyGCN, we developed our second network, FAE-PolyGCN, which introduced a more advanced feature extraction mechanism designed to enhance polygon regularization and boundary precision, particularly for smaller buildings. This model leveraged a combination of multi-resolution features and an enhanced feature extraction pipeline to refine the accuracy of building footprint segmentation. We specifically focused on improving mask quality and strengthening boundary reconstruction, ensuring that finer details were preserved even in densely packed urban areas. A key innovation in FAE-PolyGCN was the integration of the Feature Augmentation Module FAM and a boundary-preserving mask branch, which collectively contributed to better feature representation and segmentation performance. To further enhance feature extraction, we adopted a hybrid backbone architecture, combining ResNet-101 and Feature Pyramid Networks FPN, which provided rich, multi-scale contextual information. This strategic enhancement significantly improved the network’s ability to detect and reconstruct smaller buildings with greater accuracy. The impact of these improvements was evident in the performance metrics. On the SpaceNet-2 dataset, FAE-PolyGCN achieved a 7% increase in Average Precision AP and a 5% improvement in Average Recall AR for smaller buildings. Similarly, on the WHU dataset, AP and AR improved by 9% and 6%, respectively. These results demonstrated the effectiveness of our modifications in refining building boundaries and maintaining geometric consistency. However, despite these advancements, challenges persisted in accurately reconstructing larger and more complex buildings, indicating the need for further refinement in feature representation and regularization techniques.

Building upon the previous improvements, our third network AT-PolyGCN introduced Attraction Field Maps AFMs to incorporate orientation information directly into the segmentation process. The integration of AFMs significantly enhanced the model’s capability to handle diverse building structures by ensuring more precise delineation, particularly for larger buildings and occluded structures. By preserving object boundaries and refining initial vertex placements, AT-PolyGCN achieved notable performance gains over previous models,

specifically in the segmentation of large buildings, where previous approaches struggled with geometric regularity and completeness. On the SpaceNet-2 dataset, AT-PolyGCN improved Average Precision AP and Average Recall AR for large buildings by 14% and 10%, respectively, demonstrating its ability to refine segmentation accuracy in complex urban environments. Similarly, on the WHU dataset, AP and AR improved by 7% and 9%, respectively, confirming the effectiveness of AFMs in enhancing boundary adherence and structure preservation. The improved results underscore the potential of AFMs in refining footprint extraction, particularly in cases involving occlusions, densely packed structures, and irregularly shaped buildings. However, despite these significant advancements, challenges remained in handling buildings with highly complex structures. The reliance on Mask R-CNN for object detection proved to be a limiting factor, as the quality of the initial bounding boxes was not always sufficient to support precise polygon reconstruction. The segmentation quality suffered when complex architectural designs required further refinements in object detection and feature extraction. This limitation motivated the transition toward a decoupled approach in subsequent models to enhance the robustness and scalability of building footprint reconstruction.

To further refine building footprint reconstruction, we developed Decoupled-PolyGCN, a two-step framework that effectively separates object detection from polygon segmentation to enhance accuracy and scalability. We conducted extensive experiments with various object detection architectures before selecting CO-DETR, a transformer-based detection network, as the optimal choice for generating high-quality bounding boxes. These bounding boxes were then fed into a UResNet101 backbone, which extracted rich multi-scale features to facilitate precise polygon reconstruction. The Graph Convolutional Network GCN played a critical role in iteratively refining the polygons, leveraging dynamic point initialization and orthogonality loss to ensure highly regularized and geometrically consistent building boundaries. This innovative approach led to significant improvements in both segmentation accuracy and computational efficiency, making it a practical solution for real-world applications where high precision and low inference time are crucial. Our evaluations showed consistent enhancements across both raster-based and vector-based metrics, validating the effectiveness

of Decoupled-PolyGCN. The model demonstrated robust performance across diverse datasets and architectural complexities, reinforcing its applicability in large-scale building footprint extraction tasks.

To ensure robustness and generalization, we evaluated our models on two diverse datasets, WHU and SpaceNet-2, each presenting unique challenges in terms of building density, vertex count, and structural complexity. Our systematic preprocessing pipeline addressed dataset inconsistencies and annotation challenges, enabling reliable training and evaluation. The variability in building sizes and densities tested the adaptability of our models, reinforcing their ability to handle complex urban environments.

Quantitative and qualitative evaluations further validated the effectiveness of our proposed methods. We demonstrated that Decoupled-PolyGCN consistently outperformed the state-of-the-art HiSup model, achieving a 6% improvement in Average Precision AP and a 10% improvement in Average Recall AR on the WHU dataset. On the SpaceNet-2 dataset, our model achieved a 5% improvement in AP and a 7% improvement in AR. Performance gains were not limited to raster-based evaluations, the vector-based metrics also saw significant improvements. The PoLiS metric decreased by 1.35 on the WHU dataset and 0.73 on the SpaceNet-2 dataset, indicating improved boundary alignment. Similarly, the MTA metric decreased by 7 on WHU and 8 on SpaceNet-2, highlighting the network’s ability to enhance geometric consistency and structural integrity.

Beyond the architectural advancements, this thesis underscores the importance of high-quality datasets in training and evaluating deep learning models for geospatial applications. Our work highlights how dataset diversity, annotation quality, and preprocessing techniques play a crucial role in developing reliable models for large-scale building footprint extraction.

In conclusion, the innovations introduced in this thesis, including Attraction Field Maps, dynamic polygon initialization, orthogonality loss, and two-step detection frameworks, represent significant advancements in automated building footprint reconstruction. By integrating graph-based learning with multi-scale feature extraction and geometric regularization techniques, we have laid a strong foundation for future research in this field. Our proposed

methods offer a scalable, precise, and computationally efficient solution for urban planning, GIS mapping, disaster response, and other geospatial applications, paving the way for enhanced automation and accuracy in large-scale spatial data analysis.

9.2 Future Work

Our proposed model has demonstrated significant improvements over existing approaches for building instance detection and reconstruction, there remain several critical areas for future exploration and refinement. Addressing these challenges will not only enhance the accuracy of building footprint reconstruction but also contribute to the broader goal of improving geometric consistency, robustness, and generalization in urban scene analysis. Below, we outline key directions for future research and development.

Improving Object Detection for Building Instance Segmentation The performance of building footprint reconstruction is inherently linked to the accuracy of object detection. As new object detection models emerge, particularly those based on transformer architectures and advanced convolutional backbones, integrating these state-of-the-art techniques into our pipeline could significantly enhance initial instance localization. Exploring self-supervised learning and domain adaptation techniques may also improve detection robustness across diverse geographic regions and imaging conditions, leading to more precise building segmentation and reconstruction.

Refining Geometric Regularization and Hyperparameter Tuning Another promising direction is optimizing the orthogonality function and fine-tuning hyperparameters to enhance geometric regularity in reconstructed buildings further. We can identify configurations that yield more stable and accurate reconstructions by systematically exploring hyperparameter search strategies, including Bayesian optimization and evolutionary algorithms. Additionally, incorporating differentiable constraints within the model architecture could facilitate better adherence to real-world building geometry.

Addressing Missing Corners in Reconstructed Buildings A prevalent issue in building

footprint extraction is the occasional absence of critical corner points, which can result in imprecise and incomplete building boundaries. This issue particularly appears in densely packed urban environments, where adjacent structures may obscure key vertices. Developing more sophisticated corner detection mechanisms, such as attention-based vertex refinement or reinforcement learning-based corner inference, could mitigate this challenge. Further research into corner consistency constraints and edge completion models may also help generate more accurate and topologically correct building outlines.

Exploring Graph Attention Networks for Improved Feature Aggregation Another potential improvement involves replacing the current Graph Convolution Network (GCN) with a Graph Attention Network (GAT). GATs introduce attention mechanisms that dynamically assign different importance levels to neighboring nodes, which can lead to more effective feature aggregation and improved learning of complex building geometries. By leveraging attention-based graph architectures, we may be able to enhance the model’s ability to capture spatial dependencies and refine polygonal reconstructions, particularly in cases where buildings exhibit irregular structures or occlusions.

Enhancing Reconstruction of Complex Geometries One of the primary challenges in building footprint reconstruction is accurately capturing highly intricate building geometries. While our model exhibits notable improvements over previous methods, complex structures with multiple edges, non-convex shapes, and fine-grained details still pose difficulties. Future work should focus on advancing geometric deep learning techniques, such as higher-order graph neural networks, to improve the representation and prediction of intricate building shapes. Additionally, leveraging multi-scale feature fusion and more robust regularization techniques could help refine polygonal outputs and better capture subtle geometric variations in building footprints.

Advancing End-to-End Learning for Building Footprint Extraction Currently, many frameworks, including ours, rely on separate modules for object detection, feature extraction, and polygon prediction. A promising avenue for future research is the integration of these components into a fully end-to-end trainable network. End-to-end architectures offer several

advantages, including improved efficiency, reduced error propagation between stages, and the potential for joint optimization of all sub-tasks. Investigating transformer-based architectures, graph-based end-to-end pipelines, or neural implicit representations could lead to more robust and seamless building reconstruction frameworks.

Expanding the Evaluation Metrics Beyond standard evaluation metrics such as Intersection over Union Intersection over Union (IoU) and Average Precision AP, incorporating additional shape-aware metrics could provide deeper insights into model performance. One such metric is the chamfer-IoU (c-IoU), which considers the distance between predicted and ground truth boundaries, offering a more comprehensive assessment of shape accuracy.

Standardizing and Utilizing Different Datasets Finally, a critical step in advancing building reconstruction research is the standardization and utilization of large-scale geospatial datasets. The recently acquired Large-Scale Geospatial Linked Network (LGLN) dataset, provided by the German government, presents a valuable opportunity for benchmarking and training deep learning models on diverse geospatial scenarios. Properly formatting and curating this dataset will enable more robust cross-region evaluations. It will also facilitate transfer learning approaches, and contribute to the development of generalizable building reconstruction models. Furthermore, leveraging LGLN’s extensive geospatial annotations could enhance multi-modal learning frameworks by integrating elevation data, material properties, and other contextual features.

In summary, while our current model has achieved significant advancements in building instance detection and reconstruction, several key areas remain open for further research. Enhancing geometric representation, improving detection capabilities, refining corner inference, advancing end-to-end architectures, and standardizing large-scale datasets are all crucial steps toward more accurate and reliable building footprint reconstruction. By pursuing these research directions, it is probable to contribute to the broader goal of developing scalable, efficient, and highly precise urban mapping technologies for real-world geospatial applications.

Bibliography

References

- [1] James Oliver Thomson. *History of ancient geography*. Biblo & Tannen Publishers, 1965.
- [2] John P Snyder. *Flattening the earth: two thousand years of map projections*. University of Chicago Press, 1997.
- [3] Paul R Wolf. “Surveying and mapping: History, current status, and future projections”. In: *Journal of Surveying Engineering* 128.3 (2002), pp. 79–107.
- [4] Samuel N Goward, Jeffrey G Masek, Thomas R Loveland, John L Dwyer, Darrel L Williams, Terry Arvidson, Laura EP Rocchio, and James R Irons. “Semi-centennial of Landsat observations & pending Landsat 9 launch”. In: *Photogrammetric Engineering & Remote Sensing* 87.8 (2021), pp. 533–539.
- [5] Raviv Levy, Jeffery A Miller, Julia A Barsi, Kurtis J Thome, Edward Kaita, Matthew Montanaro, and Brian L Markham. “Landsat 8 decade of on-orbit operation summary of events and current state”. In: *Earth Observing Systems XXVIII*. Vol. 12685. SPIE. 2023, pp. 182–196.
- [6] Ming Lin, Meng Jin, Juanzi Li, and Yuqi Bai. “GEOSatDB: global civil earth observation satellite semantic database”. In: *Big Earth Data* (2024), pp. 1–18.
- [7] Darrel L Williams, Samuel Goward, and Terry Arvidson. “Landsat”. In: *Photogrammetric Engineering & Remote Sensing* 72.10 (2006), pp. 1171–1178.

- [8] Francois Spoto, Omar Sy, Paolo Laberinti, Philippe Martimort, Valerie Fernandez, Olivier Colin, Bianca Hoersch, and Aime Meygret. “Overview of sentinel-2”. In: *2012 IEEE international geoscience and remote sensing symposium*. IEEE. 2012, pp. 1707–1710.
- [9] Dirk Geudtner, Ramón Torres, Paul Snoeij, Malcolm Davidson, and Björn Rommen. “Sentinel-1 system capabilities and applications”. In: *2014 IEEE geoscience and remote sensing symposium*. IEEE. 2014, pp. 1457–1460.
- [10] Rolf Werninghaus and Stefan Buckreuss. “The TerraSAR-X mission and system design”. In: *IEEE Transactions on Geoscience and Remote Sensing* 48.2 (2009), pp. 606–614.
- [11] Floyd M Henderson. “Principles and applications of imaging radar”. In: *Manual of remote sensing* 2 (1998).
- [12] Michael F Goodchild. “Citizens as sensors: the world of volunteered geography”. In: *GeoJournal* 69 (2007), pp. 211–221.
- [13] Praveen Kumar Rai and VK Kumra. “Role of geoinformatics in urban planning”. In: *J Sci Res* 55 (2011), pp. 11–24.
- [14] Craig Hancock, Gethin Roberts, and Ahmad Taha. “Satellite mapping in cities: how good can it get?” In: *Proceedings of the Institution of Civil Engineers-Civil Engineering*. Vol. 162. 3. Thomas Telford Ltd. 2009, pp. 122–128.
- [15] Serdar Kose, Enes Koytak, and Yusuf S Hascicek. “An overview on the use of satellite communications for disaster management and emergency response”. In: *International Journal of Emergency Management* 8.4 (2012), pp. 350–382.
- [16] AC Teodoro and L Duarte. “The role of satellite remote sensing in natural disaster management”. In: *Nanotechnology-based smart remote sensing networks for disaster prevention*. Elsevier, 2022, pp. 189–216.
- [17] David J Mulla. “Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps”. In: *Biosystems engineering* 114.4 (2013), pp. 358–371.

- [18] Marie Weiss, Frédéric Jacob, and Grgory Duveiller. “Remote sensing for agricultural applications: A meta-review”. In: *Remote sensing of environment* 236 (2020), p. 111402.
- [19] Clement Atzberger. “Advances in remote sensing of agriculture: Context description, existing operational monitoring systems and major information needs”. In: *Remote sensing* 5.2 (2013), pp. 949–981.
- [20] Federico Cugurullo. “Urban artificial intelligence: From automation to autonomy in the smart city”. In: *Frontiers in Sustainable Cities* 2 (2020), p. 38.
- [21] Seng W Loke and Andry Rakotonirainy. *Automated City*. Springer, 2021.
- [22] Mordechai Haklay and Patrick Weber. “Openstreetmap: User-generated street maps”. In: *IEEE Pervasive computing* 7.4 (2008), pp. 12–18.
- [23] James B Campbell and Randolph H Wynne. *Introduction to remote sensing*. Guilford press, 2011.
- [24] Heeket Mehta, Pratik Kanani, and Priya Lande. “Google maps”. In: *International Journal of Computer Applications* 178.8 (2019), pp. 41–46.
- [25] Chris Pendleton. “The world according to Bing”. In: *IEEE Computer Graphics and Applications* 30.4 (2010), pp. 15–17.
- [26] UN DESA. “United Nations, Department of Economic and Social Affairs, Population Division. 2018. World Urbanization Prospects: The 2018 Revision”. In: *Online Edition* (2018).
- [27] Transportation Problem. “McGraw-Hill Dictionary of Scientific & Technical Terms, 6E.(2003)”. In: *Retrieved April 6* (2020).
- [28] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.
- [29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

- [30] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.
- [31] Jaewook Jung, Yoonseok Jwa, and Gunho Sohn. “Implicit regularization for reconstructing 3D building rooftop models using airborne LiDAR data”. In: *Sensors* 17.3 (2017), p. 621.
- [32] *Top 10 Applications for Building Footprint Data — ecopiatech.com*. <https://www.ecopiatech.com/resources/blog/top-10-applications-for-building-footprint-data>. [Accessed 29-01-2025].
- [33] Yingjie Ji, Weiguo Wu, Guangtong Wan, Yindi Zhao, Weilin Wang, Hui Yin, Zhuang Tian, and Song Liu. “Segment Anything Model-Based Building Footprint Extraction for Residential Complex Spatial Assessment Using LiDAR Data and Very High-Resolution Imagery.” In: *Remote Sensing* 16.14 (2024).
- [34] Ying Zhang and Peter Crawford. “Automated extraction of visible floodwater in dense urban areas from RGB aerial photos”. In: *Remote Sensing* 12.14 (2020), p. 2198.
- [35] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. “Geometric deep learning: going beyond euclidean data”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.
- [36] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. “Geometric deep learning: Grids, groups, graphs, geodesics, and gauges”. In: *arXiv preprint arXiv:2104.13478* (2021).
- [37] James Bergstra and Yoshua Bengio. “Random search for hyper-parameter optimization.” In: *Journal of machine learning research* 13.2 (2012).
- [38] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

- [39] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. “Hyperband: A novel bandit-based approach to hyperparameter optimization”. In: *Journal of Machine Learning Research* 18.185 (2018), pp. 1–52.
- [40] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. “Practical bayesian optimization of machine learning algorithms”. In: *Advances in neural information processing systems* 25 (2012).
- [41] Jaewook Jung and Gunho Sohn. “A line-based progressive refinement of 3D rooftop models using airborne LiDAR data with single view imagery”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 149 (2019), pp. 157–175.
- [42] K Zhao, M Kamran, and G Sohn. “Boundary regularized building footprint extraction from satellite images using deep neural networks”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2 (2020), pp. 617–624.
- [43] Priya Dwivedi. “Semantic segmentation popular architectures”. In: *Semantic segmentation—popular architectures* (2019).
- [44] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60 (2004), pp. 91–110.
- [45] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. “Surf: Speeded up robust features”. In: *Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I* 9. Springer. 2006, pp. 404–417.
- [46] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
- [47] Paul Viola and Michael Jones. “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Vol. 1. Ieee. 2001, pp. I–I.

- [48] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. Ieee. 2005, pp. 886–893.
- [49] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. “Object detection with discriminatively trained part-based models”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2009), pp. 1627–1645.
- [50] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. “Deformable part models are convolutional neural networks”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2015, pp. 437–446.
- [51] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.
- [52] Corinna Cortes. “Support-Vector Networks”. In: *Machine Learning* (1995).
- [53] R Girshick. “Fast r-cnn”. In: *arXiv preprint arXiv:1504.08083* (2015).
- [54] Shaoqing Ren. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *arXiv preprint arXiv:1506.01497* (2015).
- [55] J Redmon. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [56] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. “Ssd: Single shot multibox detector”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 21–37.
- [57] Dimitrios Marmanis, Jan D Wegner, Silvano Galliani, Konrad Schindler, Mihai Datcu, and Uwe Stilla. “Semantic segmentation of aerial images with an ensemble of CNSS”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016 3* (2016), pp. 473–480.

- [58] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. “Convolutional neural networks for large-scale remote-sensing image classification”. In: *IEEE Transactions on geoscience and remote sensing* 55.2 (2016), pp. 645–657.
- [59] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. “High-resolution aerial image labeling with convolutional neural networks”. In: *IEEE Transactions on Geoscience and Remote Sensing* 55.12 (2017), pp. 7092–7103.
- [60] Ksenia Bittner, Shiyong Cui, and Peter Reinartz. “Building extraction from remote sensing data using fully convolutional networks”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 42 (2017), pp. 481–486.
- [61] Ksenia Bittner, Fathallahman Adam, Shiyong Cui, Marco Körner, and Peter Reinartz. “Building footprint extraction from VHR remote sensing images combined with normalized DSMs using fused fully convolutional networks”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11.8 (2018), pp. 2615–2629.
- [62] Chris A Glasbey. “An analysis of histogram-based thresholding algorithms”. In: *CVGIP: Graphical models and image processing* 55.6 (1993), pp. 532–537.
- [63] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. “Design of an image edge detection filter using the Sobel operator”. In: *IEEE Journal of solid-state circuits* 23.2 (1988), pp. 358–367.
- [64] John Canny. “A computational approach to edge detection”. In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), pp. 679–698.
- [65] Zhenyu Wu and Richard Leahy. “An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation”. In: *IEEE transactions on pattern analysis and machine intelligence* 15.11 (1993), pp. 1101–1113.
- [66] Keh-Shih Chuang, Hong-Long Tzeng, Sharon Chen, Jay Wu, and Tzong-Jer Chen. “Fuzzy c-means clustering with spatial information for image segmentation”. In: *computerized medical imaging and graphics* 30.1 (2006), pp. 9–15.

- [67] TN Pappas. “An adaptive clustering algorithm for image segmentation”. In: *IEEE Transactions on Signal Processing* 40.4 (1992), pp. 901–914.
- [68] Alain Tremeau and Nathalie Borel. “A region growing and merging algorithm to color segmentation”. In: *Pattern recognition* 30.7 (1997), pp. 1191–1203.
- [69] Ali Ozgun Ok. “Automated detection of buildings from single VHR multispectral images using shadow information and graph cuts”. In: *ISPRS journal of photogrammetry and remote sensing* 86 (2013), pp. 21–40.
- [70] Konstantinos Karantzas and Nikos Paragios. “Recognition-driven two-dimensional competing priors toward automatic and accurate building detection”. In: *IEEE Transactions on Geoscience and Remote Sensing* 47.1 (2008), pp. 133–144.
- [71] Pedro F Felzenszwalb and Daniel P Huttenlocher. “Efficient graph-based image segmentation”. In: *International journal of computer vision* 59 (2004), pp. 167–181.
- [72] Wei Li, Saurabh Prasad, James E Fowler, and Lori Mann Bruce. “Locality-preserving discriminant analysis in kernel-induced feature spaces for hyperspectral image classification”. In: *IEEE Geoscience and Remote Sensing Letters* 8.5 (2011), pp. 894–898.
- [73] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.
- [74] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18. Springer. 2015, pp. 234–241.
- [75] Pedro O O Pinheiro, Ronan Collobert, and Piotr Dollár. “Learning to segment object candidates”. In: *Advances in neural information processing systems* 28 (2015).

- [76] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. “Learning to refine object segments”. In: *Computer Vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 75–91.
- [77] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.
- [78] Hugo Ledoux and Martijn Meijers. “Extruding building footprints to create topologically consistent 3D city models”. In: *Urban and Regional data management*. CRC Press, 2009, pp. 51–60.
- [79] Norbert Haala and Martin Kada. “An update on automatic 3D building reconstruction”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 65.6 (2010), pp. 570–580.
- [80] Shuang He, Guillaume Moreau, and Jean-Yves Martin. “Footprint-based 3d generalization of building groups for virtual city visualization”. In: *Proceedings of the GEO Processing* (2012).
- [81] Aidan Finn and Nicholas Kushmerick. “Multi-level boundary classification for information extraction”. In: *European Conference on Machine Learning*. Springer. 2004, pp. 111–122.
- [82] Mengjia Liu, Peng Liu, Bingze Song, Yuwei Zhang, and Luo Zhang. “Active Contour Building Segmentation Model based on Convolution Neural Network”. In: *IOP Conference Series: Earth and Environmental Science*. Vol. 1004. 1. IOP Publishing. 2022, p. 012015.
- [83] Youssef El Merabet, Cyril Meurie, Yassine Ruichek, Abderrahmane Sbihi, and Raja Touahni. “Building roof segmentation from aerial images using a line-and region-based watershed segmentation technique”. In: *Sensors* 15.2 (2015), pp. 3172–3203.

- [84] Gang Li, Jinliang An, and Youchuan Wan. “Automatic building extraction based on improved watershed segmentation, mutual information match and snake model”. In: *International journal of computer applications in technology* 43.2 (2012), pp. 147–154.
- [85] Martino Pesaresi, Andrea Gerhardinger, and François Kayitakire. “A robust built-up area presence index by anisotropic rotation-invariant textural measure”. In: *IEEE Journal of selected topics in applied earth observations and remote sensing* 1.3 (2008), pp. 180–192.
- [86] Beril Sirmacek and Cem Unsalan. “Urban-area and building detection using SIFT keypoints and graph theory”. In: *IEEE transactions on geoscience and remote sensing* 47.4 (2009), pp. 1156–1167.
- [87] Örsan Aytekin, U Zöngür, and Ugur Halici. “Texture-based airport runway detection”. In: *IEEE Geoscience and Remote Sensing Letters* 10.3 (2012), pp. 471–475.
- [88] Patrick Aravena Pelizari, Kristin Spröhnle, Christian Geiß, Elisabeth Schoepfer, Simon Plank, and Hannes Taubenböck. “Multi-sensor feature fusion for very high spatial resolution built-up area extraction in temporary settlements”. In: *Remote sensing of environment* 209 (2018), pp. 793–807.
- [89] Mustafa Turker and Dilek Koc-San. “Building extraction from high-resolution optical spaceborne images using the integration of support vector machine (SVM) classification, Hough transformation and perceptual grouping”. In: *International Journal of Applied Earth Observation and Geoinformation* 34 (2015), pp. 58–69.
- [90] Ariadna Quattoni, Michael Collins, and Trevor Darrell. “Conditional random fields for object recognition”. In: *Advances in neural information processing systems* 17 (2004).
- [91] Rasha Alshehhi, Prashanth Reddy Marpu, Wei Lee Woon, and Mauro Dalla Mura. “Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 130 (2017), pp. 139–149.
- [92] S Hochreiter. “Long Short-term Memory”. In: *Neural Computation MIT-Press* (1997).

- [93] Benjamin Bischke, Patrick Helber, Joachim Folz, Damian Borth, and Andreas Dengel. “Multi-task learning for segmentation of building footprints with deep neural networks”. In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 1480–1484.
- [94] Guangming Wu, Xiaowei Shao, Zhiling Guo, Qi Chen, Wei Yuan, Xiaodan Shi, Yongwei Xu, and Ryosuke Shibasaki. “Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks”. In: *Remote Sensing* 10.3 (2018), p. 407.
- [95] Weiwei Sun and Ruisheng Wang. “Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with DSM”. In: *IEEE Geoscience and Remote Sensing Letters* 15.3 (2018), pp. 474–478.
- [96] Gellért Mátyus and Raquel Urtasun. “Matching adversarial networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8024–8032.
- [97] Xuran Pan, Fan Yang, Lianru Gao, Zhengchao Chen, Bing Zhang, Hairui Fan, and Jinchang Ren. “Building extraction from high-resolution aerial imagery using a generative adversarial network with spatial and channel attention mechanisms”. In: *Remote Sensing* 11.8 (2019), p. 917.
- [98] Vladimir Iglovikov, Selim Seferbekov, Alexander Buslaev, and Alexey Shvets. “Ternausnetv2: Fully convolutional network for instance segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 233–237.
- [99] Myungchul Kim, Sanghyun Woo, Dahun Kim, and In So Kweon. “The devil is in the boundary: Exploiting boundary representation for basis-based instance segmentation”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 929–938.

- [100] Wufan Zhao, Claudio Persello, and Alfred Stein. “Building instance segmentation and boundary regularization from high-resolution remote sensing images”. In: *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2020, pp. 3916–3919.
- [101] Yuan Hu, Zhibin Wang, Zhou Huang, and Yu Liu. “PolyBuilding: Polygon transformer for building extraction”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 199 (2023), pp. 15–27.
- [102] Shunping Ji, Shiqing Wei, and Meng Lu. “A scale robust convolutional neural network for automatic building extraction from aerial and satellite imagery”. In: *International journal of remote sensing* 40.9 (2019), pp. 3308–3322.
- [103] Shiqing Wei, Shunping Ji, and Meng Lu. “Toward automatic building footprint delineation from aerial images using CNN and regularization”. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.3 (2019), pp. 2178–2189.
- [104] Wenjing Deng, Qian Shi, and Jun Li. “Attention-gate-based encoder–decoder network for automatic building extraction”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), pp. 2611–2620.
- [105] Dengji Zhou, Guizhou Wang, Guojin He, Ranyu Yin, Tengfei Long, Zhaoming Zhang, Sibao Chen, and Bin Luo. “A large-scale mapping scheme for urban building from Gaofen-2 images using deep learning and hierarchical approach”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), pp. 11530–11545.
- [106] Haonan Guo, Qian Shi, Bo Du, Liangpei Zhang, Dongzhi Wang, and Huaxiang Ding. “Scene-driven multitask parallel attention network for building extraction in high-resolution remote sensing images”. In: *IEEE Transactions on Geoscience and Remote Sensing* 59.5 (2020), pp. 4287–4306.

- [107] Zenghui Zhang, Weiwei Guo, Wenhao Yu, and Wenxian Yu. “Multi-task fully convolutional networks for building segmentation on SAR image”. In: *The Journal of Engineering* 2019.20 (2019), pp. 7074–7077.
- [108] Jiangye Yuan. “Learning building extraction in aerial scenes with convolutional networks”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.11 (2017), pp. 2793–2798.
- [109] Nicolas Girard, Dmitriy Smirnov, Justin Solomon, and Yuliya Tarabalka. “Regularized building segmentation by frame field learning”. In: *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2020, pp. 1805–1808.
- [110] Bowen Xu, Jiakun Xu, Nan Xue, and Gui-Song Xia. “HiSup: Accurate polygonal mapping of buildings in satellite imagery with hierarchical supervision”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 198 (2023), pp. 284–296.
- [111] Nan Xue, Song Bai, Fudong Wang, Gui-Song Xia, Tianfu Wu, and Liangpei Zhang. “Learning attraction field representation for robust line segment detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1595–1603.
- [112] Nan Xue, Song Bai, Fu-Dong Wang, Gui-Song Xia, Tianfu Wu, Liangpei Zhang, and Philip HS Torr. “Learning regional attraction for line segment detection”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.6 (2019), pp. 1998–2013.
- [113] Yilei Shi, Qinyu Li, and Xiaoxiang Zhu. “Building footprint extraction with graph convolutional network”. In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 5136–5139.
- [114] Qingyu Li, Yilei Shi, Xin Huang, and Xiao Xiang Zhu. “Building footprint generation by integrating convolution neural network with feature pairwise conditional random field (FPCRF)”. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.11 (2020), pp. 7502–7519.

- [115] Shaobo Xia and Ruisheng Wang. “Extraction of residential building instances in suburban areas from mobile LiDAR data”. In: *ISPRS journal of photogrammetry and remote sensing* 144 (2018), pp. 453–468.
- [116] Dimitrios Marmanis, Konrad Schindler, Jan Dirk Wegner, Silvano Galliani, Mihai Datcu, and Uwe Stilla. “Classification with an edge: Improving semantic image segmentation with boundary detection”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 135 (2018), pp. 158–172.
- [117] Michele Volpi and Devis Tuia. “Deep multi-task learning for a geographically regularized semantic segmentation of aerial images”. In: *ISPRS journal of photogrammetry and remote sensing* 144 (2018), pp. 48–60.
- [118] Dominic Cheng, Renjie Liao, Sanja Fidler, and Raquel Urtasun. “Darnet: Deep active ray network for building segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 7431–7439.
- [119] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. “Snakes: Active contour models”. In: *International journal of computer vision* 1.4 (1988), pp. 321–331.
- [120] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. “Pyramid scene parsing network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2881–2890.
- [121] Stefano Zorzi and Friedrich Fraundorfer. “Regularization of building boundaries in satellite images using adversarial and regularized losses”. In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 5140–5143.
- [122] Stefano Zorzi, Ksenia Bittner, and Friedrich Fraundorfer. “Machine-learned regularization and polygonization of building segmentation masks”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 3098–3105.
- [123] Hoin Jung, Han-Soo Choi, and Myungjoo Kang. “Boundary enhancement semantic segmentation for building extraction from remote sensed image”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), pp. 1–12.

- [124] Nicolas Girard and Yuliya Tarabalka. “End-to-end learning of polygons for remote sensing image classification”. In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2018, pp. 2083–2086.
- [125] Lluís Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. “Annotating object instances with a polygon-rnn”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5230–5238.
- [126] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. “Efficient interactive annotation of segmentation datasets with polygon-rnn++”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 859–868.
- [127] Zuoyue Li, Jan Dirk Wegner, and Aurélien Lucchi. “Polymapper: Extracting city maps using polygons”. In: *arXiv preprint arXiv:1812.01497* 2 (2018).
- [128] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. “Fast interactive object annotation with curve-gcn”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5257–5266.
- [129] Shunping Ji, Shiqing Wei, and Meng Lu. “Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set”. In: *IEEE Transactions on geoscience and remote sensing* 57.1 (2018), pp. 574–586.
- [130] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. “Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark”. In: *2017 IEEE International geoscience and remote sensing symposium (IGARSS)*. IEEE. 2017, pp. 3226–3229.
- [131] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. “Deepglobe 2018: A challenge to parse the earth through satellite images”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 172–181.
- [132] Adam Van Etten and Daniel Hogan. “The spacenet multi-temporal urban development challenge”. In: *arXiv preprint arXiv:2102.11958* (2021).

- [133] Franz Rottensteiner, Gunho Sohn, Jaewook Jung, Markus Gerke, Caroline Baillard, Sebastien Benitez, and Uwe Breitkopf. “The ISPRS benchmark on urban object classification and 3D building reconstruction”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences; I-3* 1.1 (2012), pp. 293–298.
- [134] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
- [135] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.
- [136] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.
- [137] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. “Path aggregation network for instance segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8759–8768.
- [138] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. “Mask scoring r-cnn”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6409–6418.
- [139] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. “Deep interactive thin object selection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 305–314.

- [140] Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. “Boundary preserving mask r-cnn”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV* 16. Springer. 2020, pp. 660–676.
- [141] Will Schroeder, Kenneth M Martin, and William E Lorensen. *The visualization toolkit an object-oriented approach to 3D graphics*. Prentice-Hall, Inc., 1998.
- [142] Urs Ramer. “An iterative procedure for the polygonal approximation of plane curves”. In: *Computer graphics and image processing* 1.3 (1972), pp. 244–256.
- [143] Zhaowei Cai and Nuno Vasconcelos. “Cascade R-CNN: High quality object detection and instance segmentation”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.5 (2019), pp. 1483–1498.
- [144] T-YLPG Ross and GKHP Dollár. “Focal loss for dense object detection”. In: *proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2980–2988.
- [145] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. “Objects as points”. In: *arXiv preprint arXiv:1904.07850* (2019).
- [146] Zhi Tian, Xiangxiang Chu, Xiaoming Wang, Xiaolin Wei, and Chunhua Shen. “Fully convolutional one-stage 3d object detection on lidar range images”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 34899–34911.
- [147] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. “Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9759–9768.
- [148] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. “End-to-end object detection with transformers”. In: *European conference on computer vision*. Springer. 2020, pp. 213–229.

- [149] MMCV Contributors. *MMCV: OpenMMLab Computer Vision Foundation*. <https://github.com/open-mmlab/mmcv>. 2018.
- [150] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. “YOLOX: Exceeding YOLO Series in 2021”. In: *arXiv preprint arXiv:2107.08430* (2021).
- [151] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. *RTMDet: An Empirical Study of Designing Real-Time Object Detectors*. 2022. arXiv: 2212.07784 [cs.CV].
- [152] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. “Dino: Detr with improved denoising anchor boxes for end-to-end object detection”. In: *arXiv preprint arXiv:2203.03605* (2022).
- [153] Z Zong, G Song, and Y Liu. “DETRs with collaborative hybrid assignments training. arXiv 2022”. In: *arXiv preprint arXiv:2211.12860* 4 (2022).
- [154] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. “Dab-detr: Dynamic anchor boxes are better queries for detr”. In: *arXiv preprint arXiv:2201.12329* (2022).
- [155] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.
- [156] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. “Deformable detr: Deformable transformers for end-to-end object detection”. In: *arXiv preprint arXiv:2010.04159* (2020).