Master's Thesis of Engineering

Keypoint-based Deep Learning Approach for Building Footprint Extraction Using Very High Resolution Satellite and Aerial Images

고해상도 위성영상 및 항공영상에서의 건물경계추출을 위한 특징점 기반의 딥러닝 접근

February 2021

Department of
Civil and Environmental Engineering
Seoul National University

Doyoung Jeong

Keypoint-based Deep Learning Approach for Building Footprint Extraction Using Very High Resolution Satellite and Aerial Images

Advisor Yongil Kim

Submitting a master's thesis of Science

December 2020

Department of
Civil and Environmental Engineering
Seoul National University

Doyoung Jeong

Confirming the master's thesis written by
Doyoung Jeong
January 2021

Chair _____ (Seal)
Vice Chair _____ (Seal)
Examiner _____ (Seal)

Abstract

Building footprint extraction is an active topic in the domain of remote sensing, since buildings are a fundamental unit of urban areas. Deep convolutional neural networks successfully perform footprint extraction from optical satellite images. However, the semantic segmentation approach produces coarse results, such as blurred and rounded boundaries in the output, which are caused by the use of convolutional layers with large receptive fields and pooling layers.

Recently, a series of studies has been conducted to directly create polygon representations that describe geometric objects of vector structures through an end-to-end learnable approach. The objective of this thesis is to derive visually improved building objects by directly extracting vertices of independent buildings, which is accomplished by combining instance segmentation and keypoint detection. The target keypoints in building extraction are points of interest based on the local image gradient direction, that is, the vertices of a building polygon. The proposed framework follows a two-stage, top-down approach that that is divided into object detection and keypoint estimation. Keypoints between instances are distinguished by merging the rough segmentation masks and local features of regions of interest. A building polygon is created by grouping the predicted keypoints through a simple geometric method.

In this study, public datasets, namely SpaceNet 2 and Open Cities AI Challenge dataset were used for building footprint extraction. SpaceNet 2 contains satellite images of WorldView-3, which are not orthoimages, while Open Cities AI consists of orthorectified aerial images where annotations match roof outlines and building footprints.

The most widely used semantic segmentation model (EfficientNet–U-Net) and an instance segmentation network (Mask R-CNN) were implemented here to validate the performance of the proposed framework. The framework

was evaluated with three metrics, namely, F1 score, intersection over union (IoU), and structural similarity index measure (SSIM).

The results demonstrated that the proposed framework exhibited better segmentation performance compared with Mask R-CNN in terms of both qualitative and quantitative results under keypoint estimation. However, compared with the state-of-the-art EfficientNet–U-Net, which is based on semantic segmentation, the proposed network performed poorly. This is because the performance of the framework largely depends on the performance of the object detector. Nevertheless, the proposed framework, limited to the detected object in the preceding network, directly predicts the corner points of the building polygon to derive vectorized objects only from the output of the end-to-end learnable network. The proposed framework trains the geometric coordinates of the polygon's keypoints and demonstrates the potential to directly generate vectorized representations of segmented objects in the satellite images.

Keyword : Building Footprint Extraction, Keypoint Detection, Instance Segmentation, Deep Learning, Satellite image
Student Number : 2019-20867

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1. Background and Motivation

Buildings, which are a key piece of cadastral information related to populations and cities, are fundamental to urban planning and disaster management. Organizations, including governments, nongovernmental organizations, and the United Nations, need full access to comprehensive accurate assessment, especially for efficient disaster response, to allocate limited resources during building damage assessment. To this end, very-high-resolution satellite imagery offers a spatial resolution of up to 0.3 m, thus providing an unprecedented range of visual information about the location of key infrastructure and the level of damage caused by disasters. Therefore, such images have become increasingly important tools for building damage assessment. A building can be considered a basic unit in disaster response; thus, it is the core target feature extracted from satellite images. In disaster management, building footprint extraction precedes damage assessment; hence, the accuracy of building footprint extraction has a great influence on the accuracy of the assessment.

Building footprint extraction, which is an actively researched topic in the domain of remote sensing, is challenging due to the variability of building shapes, materials, and dimensions and the different types of backgrounds against which they are located [1]. In early works, building footprints were often delineated with multistep, bottom-up approaches and a combination of multispectral satellite imagery and airborne light detection and ranging (LiDAR) [2]. However, these methods have poor generalization abilities. Recently, deep neural networks (DNNs) have shown successful performance in building footprint extraction [3] by using only optical satellite images. DNNs with multiple nonlinear layers can automatically learn high-level abstract features from large amounts of training data and outperform

conventional algorithms, thus becoming a dominant method for building segmentation tasks.

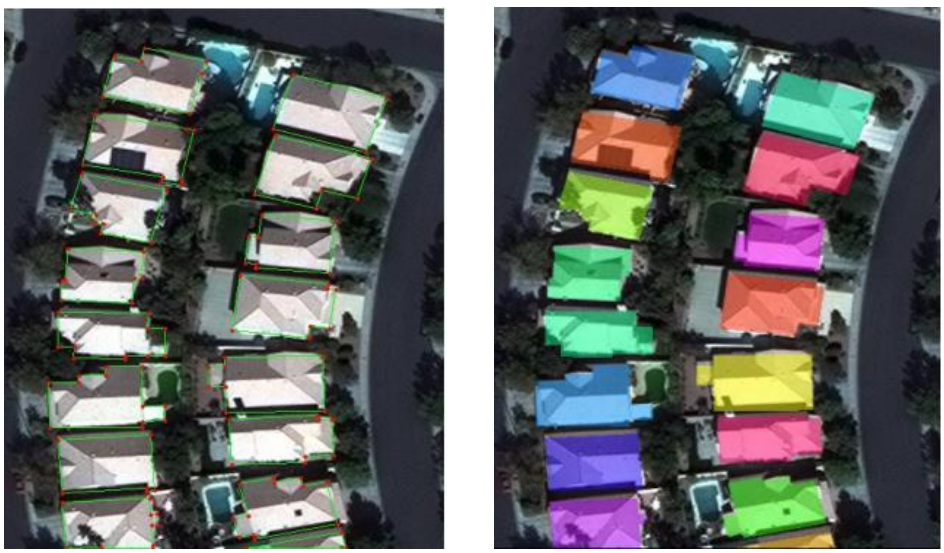
Research on the automatic extraction of building footprints has rapidly grown in recent years, resulting in the development of public datasets for machine-learning methods. Organizations, including the Defense Innovation Unit and Microsoft, in collaboration with OpenStreetMap (OSM) [4], provide vectorized building footprint datasets with various factors, such as spatial resolution and number of spectral bands.

Building footprint extraction is normally considered by the combination of two tasks: (i) segmentation, which is the extraction of building regions from the given area, and (ii) instantiation, which is the identification of individual buildings [3]. Most studies aim to extract individual, vectorized buildings by integrating the two tasks together. The result can be classified into two types of approaches, depending on which method is performed first. One approach is *semantic segmentation* (segmentation before instantiation) [5, 6], which classifies image pixels into building and nonbuilding pixels. Through post-processing, each individual pixel in building regions is identified by grouping connected pixels. The second approach is called *instance segmentation*, where each building is detected from within a bounding box. Then, each detected object is segmented into building and nonbuilding pixels [7].

Building footprint extraction is essentially a binary classification problem in that there are only two categories, namely, buildings and nonbuildings. In several challenges and papers, semantic segmentation was used to classify each pixel class by using deep features through U-Net-based deep learning networks [8], resulting in superior performance and wins in several challenges.

However, the semantic segmentation approach produces coarse segmentation results, such as nonsharp boundaries in the output, which are caused by the use of convolutional layers with large receptive fields and by the pooling layers in deep convolutional neural networks (DCNNs), which

fail to detect fine local details because they do not consider the interactions occurring between pixels. Moreover, the segmentation result is rasterized into a binary classification image, which is not a desirable output from a user's point of view for many applications [9]. Recently, a series of studies has been conducted to create polygon representations that describe geometric objects of vector structures in an end-to-end learnable approach [3, 7, 9]. Instead of pixel-wise segmentation maps, instance segmentation approaches have been introduced to directly generate polygons in an end-to-end network. After each instance is defined using detection modules, recurrent neural networks can be used to predict the vertices and edge masks of an instance. This approach can also produce a visually qualitative segmentation mask with sharp boundaries while connecting each vertex with its nearest neighbors.



a) Polygon annotation b) Pixel-wise mask annotation

Figure 1.1. Two General Annotation Forms of the Building Footprint Extraction Dataset [3]

In Figure 1.1, the label of the building footprint dataset is annotated by a group of points obtained by the user. For general semantic segmentation, the polygon annotations are preprocessed, that is, rasterized, into pixel-wise mask

annotations, but geometric details may be lost in rasterizing. As in the human annotation process, the geometric details that may be lost in the rasterizing process may be minimized by directly extracting polygon vertices using deep learning networks. Mask R-CNN also has the advantage of enabling easy vectorization by grouping the vertices.

In this thesis, each polygon point is regarded as a keypoint, and keypoint extraction is performed for instantiation after object detection. Keypoints are synonymous with interesting points, which are used as different definitions as targets. Keypoints in building extraction are points of interest based on the local image gradient direction, that is, the vertex of a building polygon. Scale-invariant feature transform (SIFT) has been used to extract good keypoint candidates from urban-area and building detection in satellite images [10]. The neighborhood between different vertices is summarized as edges, and several vertices are grouped together to identify an independent building. Beyond conventional keypoint descriptors, keypoint detectors based on end-to-end learning with deep learning perform well in various fields, such as image matching, human pose estimation, and instance segmentation [11, 12]. Through keypoint detection networks deployed after region proposal networks (RPNs), additional information, such as pose estimation, is extracted within the detection result. Therefore, the objective of this thesis is to derive visually improved building objects by directly extracting the vertices of independent buildings. Such extraction is conducted by combining instance segmentation and keypoint detection.

1.2. Research Objectives

This thesis aims to establish a deep learning framework that provides solutions to the problems of automatic building footprint extraction with keypoint detection and grouping. The tasks of detection, segmentation, and geometric learning for keypoint estimation are combined in the proposed framework, which will be introduced in Section 3.

1.2.1 Workflow

The developed framework employs the typical supervised learning mechanism illustrated in Figure 1.2. The framework is based on a combination of the instance segmentation model with the prediction and grouping of keypoints. The input datasets consist of satellite and aerial images with ground truth annotation of the building footprints, which are regarded as training data. The datasets are preprocessed properly to make them suitable for training deep learning networks for a low computational cost.

The proposed network is based on the instance segmentation approach, which can predict the segmentation mask of each instance extraction through an RPN. Unlike conventional approaches, keypoint detection modules take the place of fully convolutional networks (FCNs), which predict the segmentation mask on localized region-of-interest (RoI) features. The proposed framework is evaluated by comparing its performance against that of two representative models, namely, semantic segmentation and instance segmentation, which are the two axes of deep learning building footprint extraction. The performance is assessed using three metrics, namely, F1 score, interest of union (IoU), and structural similarity index measure (SSIM). For the generalization of model performance, the framework is trained, and the metrics are compared independently on two datasets with different characteristics

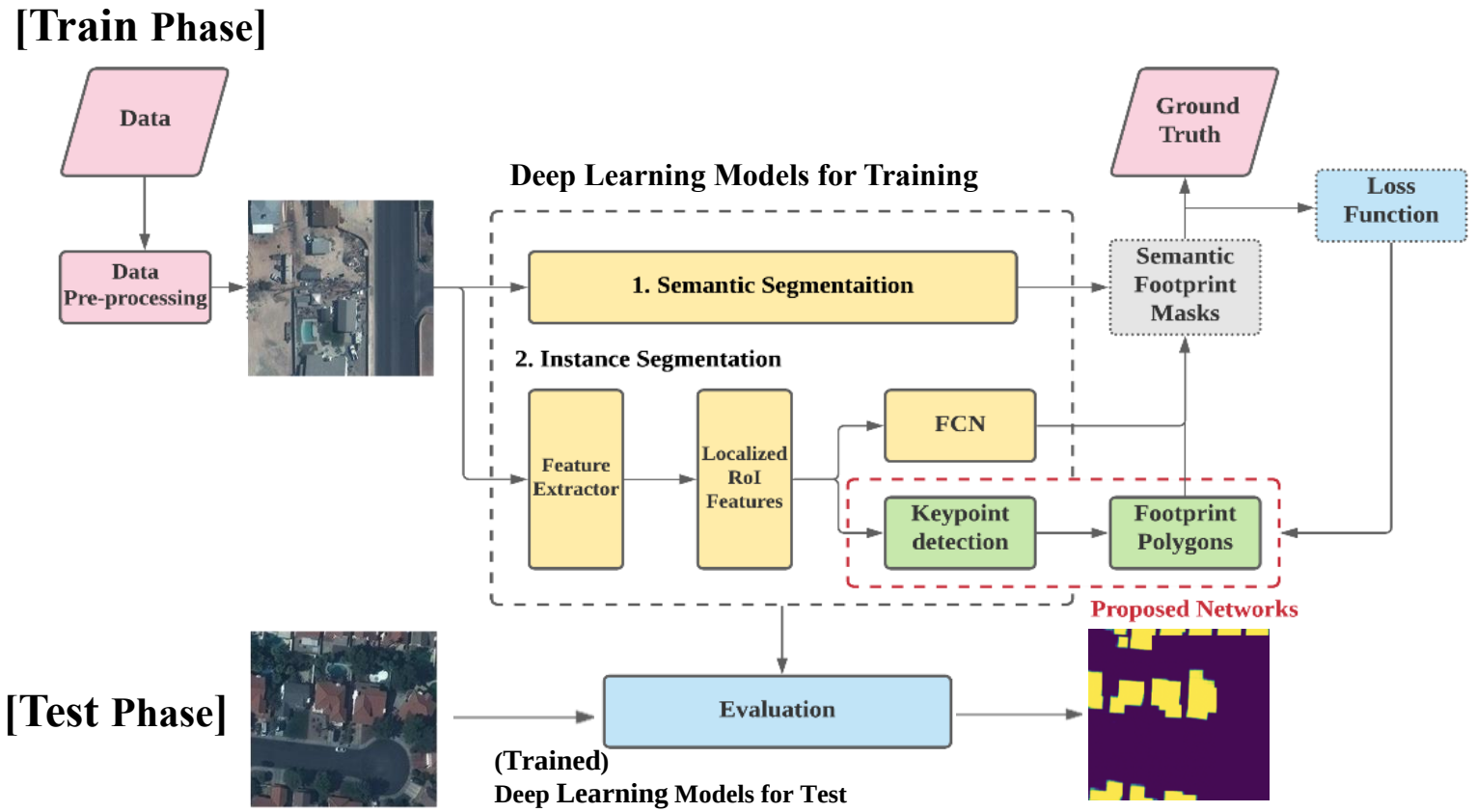


Figure 1.2. The General Workflow of the Proposed Framework

1.2.2 Contribution

The contributions of this thesis are as follows:

- A novel framework for building footprint extraction is proposed by the introduction of a keypoint detection module. Building polygons with improved visibility are obtained by replacing the segmentation task in instance segmentation with keypoint detection.
- The proposed network operates by simply adding the detection network module to the common two-stage instance segmentation networks in end-to-end learning, thereby predicting vectorized building polygons without heavy post-processing algorithm.
- At a satellite image spatial resolution of 30 cm, the proposed keypoint-based building polygon extraction is confirmed to be feasible through experiments and analysis.

1.3. Organization of Thesis

The thesis is organized as follows: Chapter 2 reviews relevant works and public dataset for building footprint extraction. Chapter 3 describes the backbone network for building footprint extraction. In Chapter 4, the proposed methodology is explained in three steps, instance segmentation, keypoint estimation and keypoint grouping. Chapter 5 describes the experimental details containing adopted dataset and implementation details for training the networks. Chapter 6 shows experimental results and discussion. Finally, the conclusion of the thesis is given in Chapter 7.

Chapter 2. Related Works

2.1. Deep Learning for Object Detection and Semantic Segmentation

2.1.1. Semantic Segmentation

Semantic segmentation classifies the pixels of an image into meaningful classes that are semantically interpretable and correspond to specific categories. An FCN [13] is a typical deep learning model for semantic segmentation, which plays a key role in modern models. The input image is fed into convolutional and pooling layers to extract and interpret the contextual information. The FCN, located at the end of the network, learns from deconvolution layers to upsample the feature map to the original resolution of the image. As this step consists of simply upsampling the score, superior performance is unlikely due to the limited amount of information.

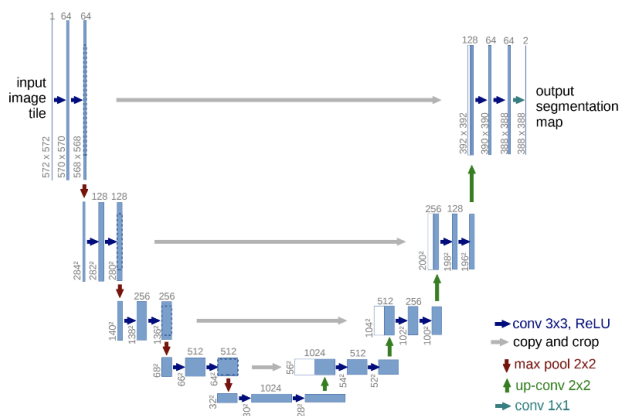


Figure 2.1. U-Net Architecture for Semantic Segmentation [12]

With regard to the structure of FCNs, [14] proposed U-Net, which has an encoder–decoder structure; the encoder part is utilized to capture the context of an image, and the decoder part is utilized to learn the precise localization of the results (Figure 2.1). U-Net also adds a skip connection to the encoder–decoder structure. Through this skip connection, which directly transfers

information from the contracting path to the expansive path, this model can improve localization performance by reducing information leakage through the networks. Moreover, this model is sensitive to detecting small objects and segmenting densely distributed objects.

2.1.2. Instance Segmentation

Instance segmentation refers to identifying individual objects and delineating each distinct object of interest. The main approach for this task is Mask R-CNN, whose architecture is shown in Figure 2.2.

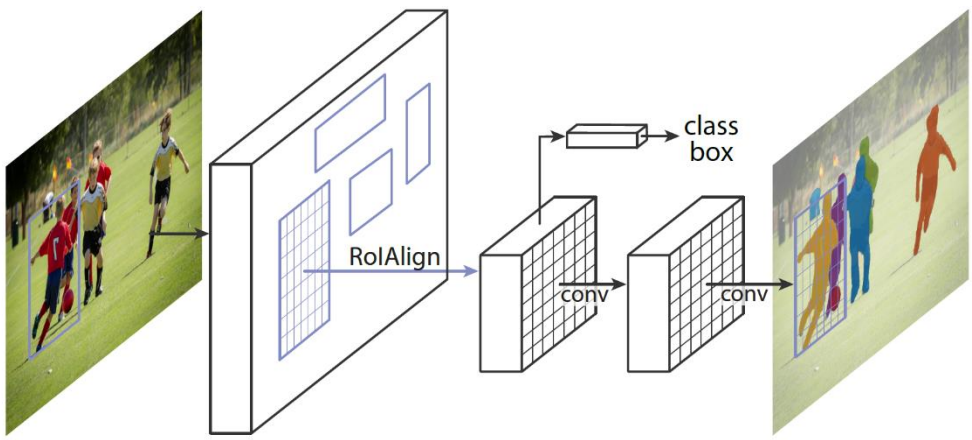


Figure 2.2. Mask R-CNN Framework for Instance Segmentation.

Mask R-CNN [15] performs detection and then segmentation. The detection process generates localized RoIs from the feature map from a feature extractor, such as residual network (ResNet) and AlexNet; this feature extractor is the same as the common two-stage object detector (Faster R-CNN) [16, 17]. After this step, the feature of each RoI is fed into simple convolutional layers and an FCN to obtain object masks for semantic segmentation. This approach can be considered a common form of the end-to-end instance segmentation model. Research has been conducted to improve each module in Mask R-CNN while maintaining the structure of the detection and segmentation modules after a feature extraction [18].

2.2. Building Footprint Extraction

Building footprint extraction is garnering considerable interest as an active field in remote sensing and computer vision research. Established building footprint maps are used in important applications, such as disaster risk management and urban monitoring.

Figure 2.3. Similar Tasks with Building Footprint Extraction: a) Building detection b) Building segmentation c) Building footprint extraction

Building detection and building segmentation are similar to building boundary extraction (Figure 2.3). First, building detection extracts the existence of buildings from a satellite image and performs geometric localization using bounding boxes. It is used in detecting and tracking the number of buildings in a time series image to observe the urban development and disaster response applications[1]. Next, building segmentation classifies the pixels occupied by buildings in an image, which include the roofs and visible walls of the buildings [19]. By contrast, building boundary extraction involves the ground boundaries of buildings. Therefore, inconsistencies may occur between the roof boundaries and building footprints from satellite and aerial images. In the case of orthogonal images, such as unmanned aerial vehicle (UAV) and drone images, roof and building boundaries are almost identical. However, for satellite images, the inclination of a building occurs

[1] https://spacenet.ai/sn7-challenge/

due to its view angle, and the roof and building boundaries could be inconsistent. Therefore, the above factors should be considered in selecting the training dataset, which will be explained in Section 2.4.

Table 2.1. Categories of Building Footprint Extraction

Conventional method	**Deep-learning based method**		
Morphological Operation	Semantic Segmentation	Graph Convolutional Neural Network	Instance Segmentation
• Extract morphological information such as edges from raw data • Limitation of the feature to identify building	• Classify each pixel with a corresponding class • Need additional post-processing to vectorize	• Calculated the loss between the nodes, coordinates rather than pixel values • Significant computational costs	• Combination of detection and segmentation • Identify each instance of each object • Can apply Polygonization Module
LiDAR with optical images	Optical Multispectral	Optical RGB	Optical RGB

As shown in Table 2.1, building footprint extraction could be performed through four approaches. Building footprints are often delineated with multistep, bottom-up approaches by using a combination of multispectral satellite and airborne LiDAR imagery [2]. LiDAR data help in localization by using point clouds, which can provide geometrically meaningful information. Building outlines can be obtained by merging convex polygons extracted from the point clouds using a binary space partitioning (BSP) tree. Satellite images are considered auxiliary data from the LiDAR points to remove trees that are confused with buildings by using the normalized difference vegetation index. Reference [20] generated building footprints by using DTM and DSM estimated from a morphological operator derived from

LiDAR data. This approach applies shape optimization algorithms, such as BSP, to extract building boundaries from LiDAR data. Additionally, point cloud data are used for automated 3D building reconstruction and extraction of height information.

2.2.1. Semantic Segmentation

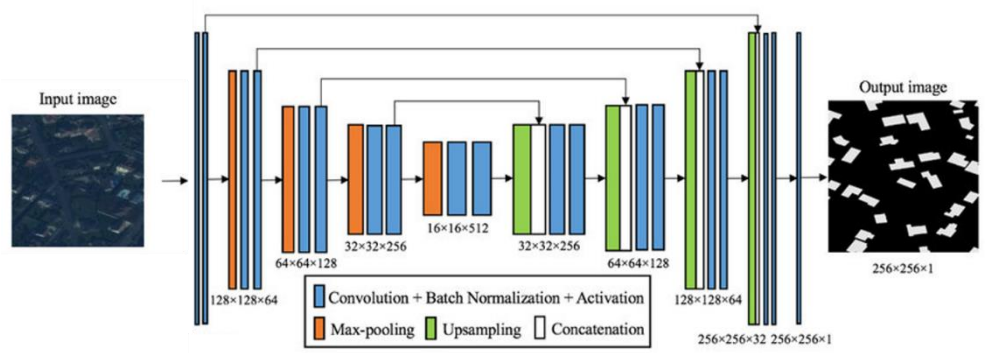


Figure 2.4. General Semantic Segmentation Network for Building Footprint Extraction [21]

A recent approach for building footprint extraction is semantic segmentation, which applies an FCN from a single optical image [5, 6, 21, 22]. Many winning networks in data competitions used multiple convolutional neural networks (CNNs) for semantic segmentation to extract building footprints (Figure 2.4). In particular, the U-Net model, which has an encoder–decoder structure, has been widely used for building extraction and proven effective for solving the binary segmentation problem. Most of the competitors from the SpaceNet 2 and Open Cities AI challenges performed semantic segmentation using ensembles of U-Net with state-of-the-art image classification encoders. The winning network structure of Open Cities AI is the same as that of SpaceNet 2, and the encoder of the model uses state-of-the-art structures, such as EfficientNet and InceptionNet [23, 24]. The use of new encoders helps increase accuracy by using a deep model design and mitigating the scaling problem.

To overcome this issue, probabilistic graph models, such as the conditional random field (CRF) [25], have been proposed to connect convolutional layers at the final layers as a post-processing step. The main concept behind this model is to transform pixel-wise classification into probabilistic inference. The final prediction is substantially improved to generate precise boundaries from the initial prediction of pixel-wise labels. However, this framework does not extract a sufficient number of features from the images to facilitate effective propagation of information. Reference [26] introduced an automatic building extraction method that integrates a graph convolutional network (GCN) and deep structured feature embedding into an end-to-end workflow.

To accurately describe edge information, [26] proposed GCN frameworks for building segmentation rather than using DCNNs. Similar to graph models, such as CRF, GCN can aggregate the information from neighbor nodes, which allows the model to learn about local structures. The grid-like data can be interpreted as a special type of graph data where the node is on the grid and the number of neighbors is fixed [26]. However, due to its significant computational cost, the GCN approach has been applied only to low-resolution images, such as Planet (5 m), rather than submeter-resolution satellite images [27].

Building footprint extraction performs beyond building segmentation by identifying each instance of each building. The segmentation result is a binary classification map that classifies only buildings and backgrounds, that is, without building objects. The boundary of each building mask is the result of building boundary extraction. Several challenges, such as SpaceNet 2 and SpaceNet 4, require instantiation in post-processing, as shown by using mean average precision (mAP) for evaluation.

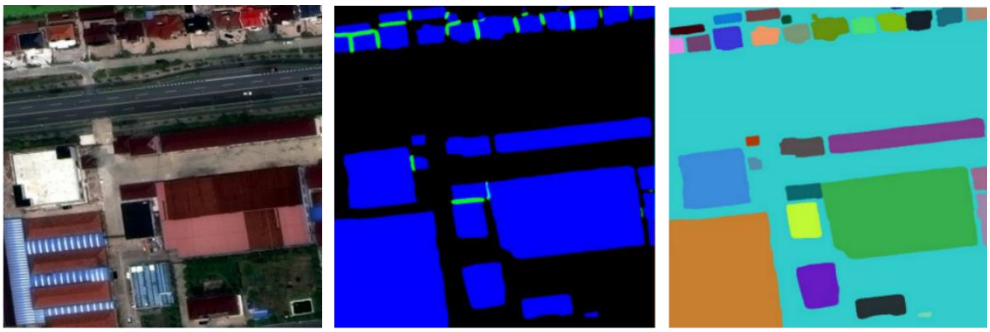


Figure 2.5. Workflow of the Winner's Algorithm for SpaceNet4 (cannab): a) Input image, b) Segmentation results, blue: building, green: contact points between juxtaposed buildings, c) Instantiation results

Since simple binary segmentation cannot lead to instantiation, SpaceNet 4 challenge winners introduced two additional networks for the division between adjacent buildings, namely, building outline labels and contact points between very closely juxtaposed buildings[2]. As shown in Figure 2.5.b, the proposed network yields three classes, namely, buildings, juxtaposed pixels, and background. The idea of instantiation using the additional adjacent pixels was widely used by various competitors for post-processing that combines the watershed algorithm and LightGBM.

Without post-processing via polygonization, semantic segmentation produces coarse results, such as blurred and rounded boundaries in the output caused by the use of convolutional layers with large receptive fields and pooling layers in DCNNs [28]. A DCNN also fails to use fine local details because it does not consider the interactions between pixels [26]. The lack of local details is significant for refining boundary information (even with the use of post-processing), especially for the detection of small objects which can be obtained as low-level features in satellite images.

[2] https://github.com/SpaceNetChallenge/SpaceNet_Off_Nadir_Solutions

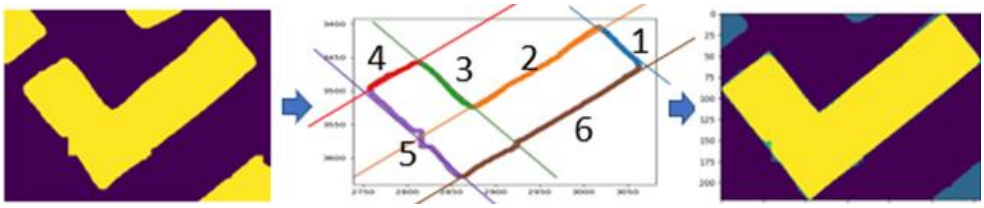


Figure 2.6. Polygonization Stage Proposed by Microsoft USBuildingFootprints Benchmarks

As shown in Figure 2.6, the benchmark produced by Microsoft's building dataset developed post-processing tools separately for polygonization to refine the predicted pixels from segmentation into polygons. Their research was motivated by the Douglas–Peucker algorithm [29], which was manually defined and automatically tuned to impose a priori building properties. For example, the interior angles of the generated polygons must be over 30 degrees so that they are not very sharp, and building angles likely consist of a few dominant angles[3].

[3] https://github.com/microsoft/USBuildingFootprints

2.2.2. Instance Segmentation

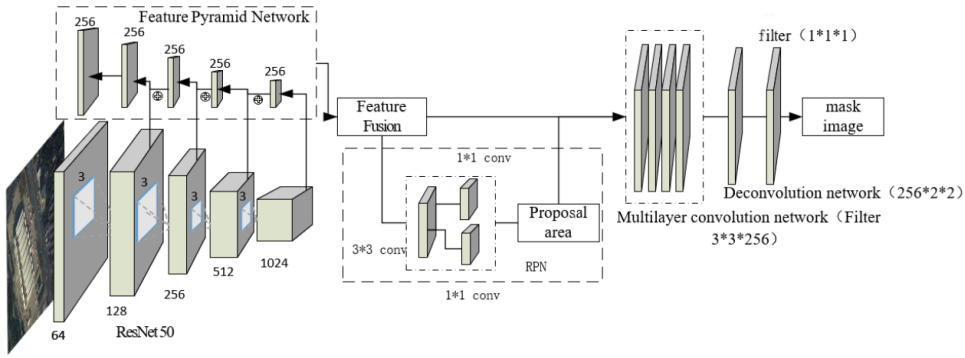


Figure 2.7. Additional Module Based on Single Class Mask R-CNN for Building Footprint Extraction [7]

Instead of generating pixel-wise segmentation maps, instance segmentation approaches were introduced to directly generate polygons in end-to-end networks. This approach is composed of two modules, namely, object detection and vectorization. Polygons are acquired by predicting the optimal locations of the polygon vertices and linking the outer vertices with straight lines, thereby creating formulaic polygons. PolygonRNN [30] and PolygonRNN++ [31] use fully convolutional layers to extract the bounding boxes of each instance; these layers are then fed into sequential recursive neural networks (RNNs), which predict a boundary mask, the locations of the polygon vertices, and the first vertex to start edge generation. In one sequence, the current boundary and vertex prediction are influenced by previous predictions. PolyMapper [9] additionally unifies building footprint extraction and road extraction into a dual pipeline and applies the same network on large-scale aerial images.

Since it is possible to distinguish independent objects, additional modules can be used to enhance the output results, such as reinforcement of edge information with attention modules for extracting effective features [7, 32] (Figure 2.7). Zhao et al. [3] introduced building boundary regularization,

where polygons are extracted via Mask R-CNN using the minimum description length framework to determine the hypothesis model. This method produces well-regularized polygons through a performance almost equivalent to that of Mask R-CNN. The authors developed the regularization idea by adopting GCNs to conduct polygonization from the extracted vertices.

Zhang et al. [7] directly predicted a point map using FCNs from each proposed RoI. To train the keypoint detector, they used a heatmap indicating the locations of the keypoints. Their study focused on detecting the vertices of an object, not from the segmentation result. Among the extracted keypoints of an object, the extreme keypoint, which is geometrically located on the far left or right, is first selected as the starting point. Then, the object is generated by establishing a connection between it and its nearest neighbor. Through an iterative process, a polygon is formed by uniting all of the edges.

2.3. Keypoints

Keypoint representation is a central component of image matching, retrieval, pose estimation, registration, and 3D reconstruction. Conventional feature detectors localize geometric structures through engineered algorithms, which are often called handcrafted detectors. These keypoint detectors have been extended to handle multiscale and affine transformations. Representatively, SIFT searches for blobs over multiple scale levels and selects stable points as keypoints.

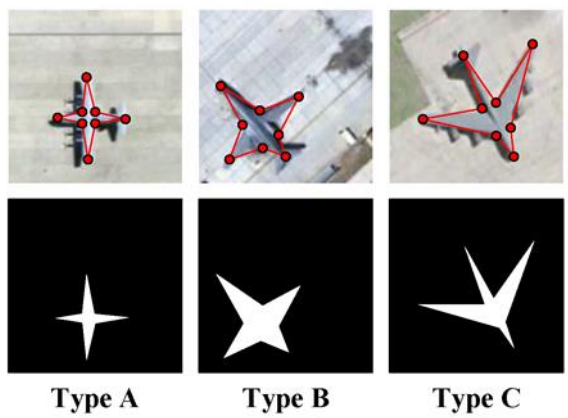


Figure 2.8. Example of Keypoint Detection Results for Classifying Aircraft Type. Keypoints' Predictions are Marked with Red Dots. [33]

The success of learned methods in general feature descriptors motivated the community to explorer similar ways for feature detectors [34]. FAST [35] was one of the first attempts to use machine learning to predict a corner keypoint detector. With the advance of approaches based on DNNs, global representations have been widely used to solve these problems, as they can be trained in a straightforward, end-to-end manner [12]. Despite the impractical inefficiency of the initial models, CNNs can significantly reduce the errors in local descriptors. Regarding predefined detector anchors, [36] introduced a new formulation for training neural networks.

Keypoint estimation, such as human pose estimation [37], chair corner point

estimation [38], and aircraft type detection, uses a fully convolutional encoder–decoder structure to predict a heatmap for each type of keypoints [39] (Figure 2.8). The network is trained through fully supervised learning with L2 loss for a rendered Gaussian heatmap. The training is guided by a multipeak Gaussian heatmap, applying a Gaussian kernel [37].

Keypoints in building extraction are points of interest based on the local image gradient direction, that is, the vertices of a building polygon. SIFT has been used to extract keypoint candidates from urban-area and building detection in satellite images [10]. The neighborhood between different vertices is summarized as edges, and several vertices are grouped together to identify an independent building. PolyMapper [9] deploys keypoint sequence prediction produced by recurrent neural networks for buildings. At each step, an RNN takes the current, revious and first vertex as inputs, and outputs a conditional probability distribution of the vertex next to the current one. PolyMapper provides compact representation for buildings, but it performs poorly for large buildings due to inaccuracies in its location information. Reference [40] combined the structure of instance segmentation and FCN-based keypoint estimation networks. However, the ability of the model when applied to satellite images could not be assessed easily because the model was assessed on the Aerial Imagery for Roof Segmentation (AIRS) dataset with a spatial resolution of 0.075 m. Moreover, since the grouping keypoint is adopted by establishing a connection between a vertex and its nearest neighbor, there is a limitation that closed polygons cannot be guaranteed for concave objects.

2.4. Open Datasets for Building Footprint Extraction

Various datasets for building footprint extraction are open to the public for evaluating models. These datasets contain aerial or satellite images with labels annotated as pixel-wise masks or object-wise labels, which refer to the coordinates of the building polygons at the object level. Some widely used datasets are discussed below.

- **xBD dataset** [41] aims to spur the creation of accurate machine-learning models that assess building damage from pre- and postdisaster satellite imagery. It contains over 5000 km^2 of RGB satellite images sized 512 × 512 pixels. The image spatial resolution is 50 cm, with satellite images provided from WorldView-2 and GeoEye-1. The annotations are rasterized into a binary mask image, and images are depicted based on a diversity of disasters.
- **Microsoft Building Footprints** was made by a collaboration between Microsoft Maps & Geospatial teams to provide building footprint extraction labels across the Unites States and Canada. The labels are automatically annotated using the Bing map, and over 136 million footprints were extracted. Unlike in other datasets, the backend of this dataset provides polygonization tools for improved segmentation results.
- **AIRS (Aerial Imagery for Roof Segmentation)** [42] consists of aerial images from the city of Christchurch, New Zealand, with 7.5 cm RGB bands. The annotations of building roof outlines are carefully refined and aligned in this database. The AIRS database includes over 220,000 buildings with object-wise labels.
- **SpaceNet 2 dataset** [43] contains 24,586 satellite images of WorldView-3 images for four cities, namely, Las Vegas, Paris, Shanghai, and Khartoum, which have different background complexities and contain a high diversity of building roof styles. The

images are provided in a variety of formats, including panchromatic, 8-band, and pansharpened RGB images. The RGB images are sized 650 × 650 pixels and have a 30 cm resolution. In this database, over 300,000 buildings are annotated with object-wise labels.

- **OpenCitiesAI dataset** features 7.5-cm-resolution drone imagery from African cities, which contain small and highly diverse building roof styles. The images have a size of 512 × 512 pixels with RGB bands. Building footprints are annotated with the local OSM data. The database covers more than 700,000 buildings in 12 African cities and regions.

Figure 2.9 Annotation Errors between Roof Outline and Building Boundary

Statistics related to the datasets are summarized in Table 2.2. The ideal building footprint label is an orthorectified image with manually annotations at the roof outlines, but no dataset perfectly meets this condition. Except AIRS, which was annotated manually for building roofs, datasets are automatically annotated using OSM building footprint data. As shown in Figure 2.9.b, there may be bias of misalignment between the roof contour observed in a satellite image and the building footprint, given that the label is only annotated to the building's ground footprint. This misalignment problem is highlighted in

datasets of satellite images; the larger the building height or view angle, the larger the actual bias. Therefore, in some competitions, such as SpaceNet 4, competitors try to overcome this problem by learning with additional one-hot-encoded information of the view angle or satellite position. However, typically, the larger the view angle, the less the accuracy.

Orthorectified images from UAVs or drones are acquired at nadir angle; thus, they are not affected by the misalignment problem. However, during geometric correction for image mosaicing, the texture of building boundaries may be distorted. In addition, some buildings are often not annotated due to omissions in OSM. Depending on the location where an image was acquired, each dataset has a different background and building roof style. As such, datasets for training should be selected by considering the advantages, disadvantages, and locations of image acquisition.

The SpaceNet 2 and Open Cities AI datasets are chosen in this study to evaluate the proposed model. SpaceNet 2 is the most widely used dataset, winning algorithms from the challenge are highly accessible, and different models for building footprint extraction have been trained on this dataset; thus, it is selected for direct comparison with state-of-the-art models. Since the dataset consists of images from four different continents, the generalization of the model algorithms must be assessed. However, several images, excluding Khartoum, are from rural landscapes and provided as satellite images, whose view angle may affect detection accuracy. The SpaceNet 2 dataset also does not include the building footprint characteristics of developing countries, where buildings are typically small and densely distributed.

In addition, SpaceNet 2, which is composed of satellite images, may exhibit misalignment between the roof contour and the building footprint. Hence, the Open Cities AI dataset, which is based on drone images, is used as auxiliary data for training. The Open Cities AI dataset can compensate for the limitation of SpaceNet 2 in our evaluation of the generalization of the proposed model,

since the dataset includes characteristics of developing countries in Africa. The dataset also contains orthorectified images with matched roof boundaries and building footprints. However, since the images included in the dataset were geometrically located, any sign of geometric distortion can easily be detected visually. For this reason, many of the teams participating in the challenge decreased the spatial resolution of the original images from 3 to 10 cm. Likewise, we lowered the resolution for the subsequent training phase, but to facilitate direct comparison with the SpaceNet 2 dataset, we downsampled the images to a matching spatial resolution of 30 cm.

Table 2.2. Statistics of the Datasets for Building Footprint Extraction

Dataset	xBD	Microsoft Building Footprints	AIRS	SpaceNet2	OpenCitiesAI
Year	2019	2019	2019	2018	2020
Type	Satellite	Aerial	Aerial	Satellite	Aerial
Location	Worldwide	US, Canada	New Zealand	Las Vegas, Paris, Shanghai, Khartoum	12 cities in Africa
Data Type	RGB	RGB	RGB	RGB + 8-band	RGB
Image Resolution (cm/pixel)	50	30	7.5	30	3
Image Size (pixels)	512×512	-	-	650×650	-
Annotations	550k	125m	220k	300k	790k
View angle	0~55	orthorectified	orthorectified	0~30	orthorectified

Chapter 3. Backbone Network

In this chapter, the backbone network for building footprint extraction is introduced. The backbone network follows the typical two-stage instance segmentation approach proposed from Mask R-CNN. This approach activates to predict segmentation mask, separating the involved tasks into detection and segmentation. This architecture outputs well-localized RoI features, which play a key role in the models in the next chapter.

The backbone network is derived for feature encoding, building detection, and localization. A combination of a ResNet and a feature pyramid network (ResNet-FPN) [44] is utilized to extract deep features at multiple scales. A two-stage object detection approach is employed through the RPN to detect and localize building objects [17], and a fully convolutional layer is used to regress the scale of bounding boxes. An RoIAlign [15] layer is applied to precisely crop the bounding boxes with the feature map, thereby obtaining a well-localized RoI for each object. Localized features are essential for predicting pixel-wise segmentation or keypoint detection. The structure of the backbone network is illustrated in Figure 3.1.

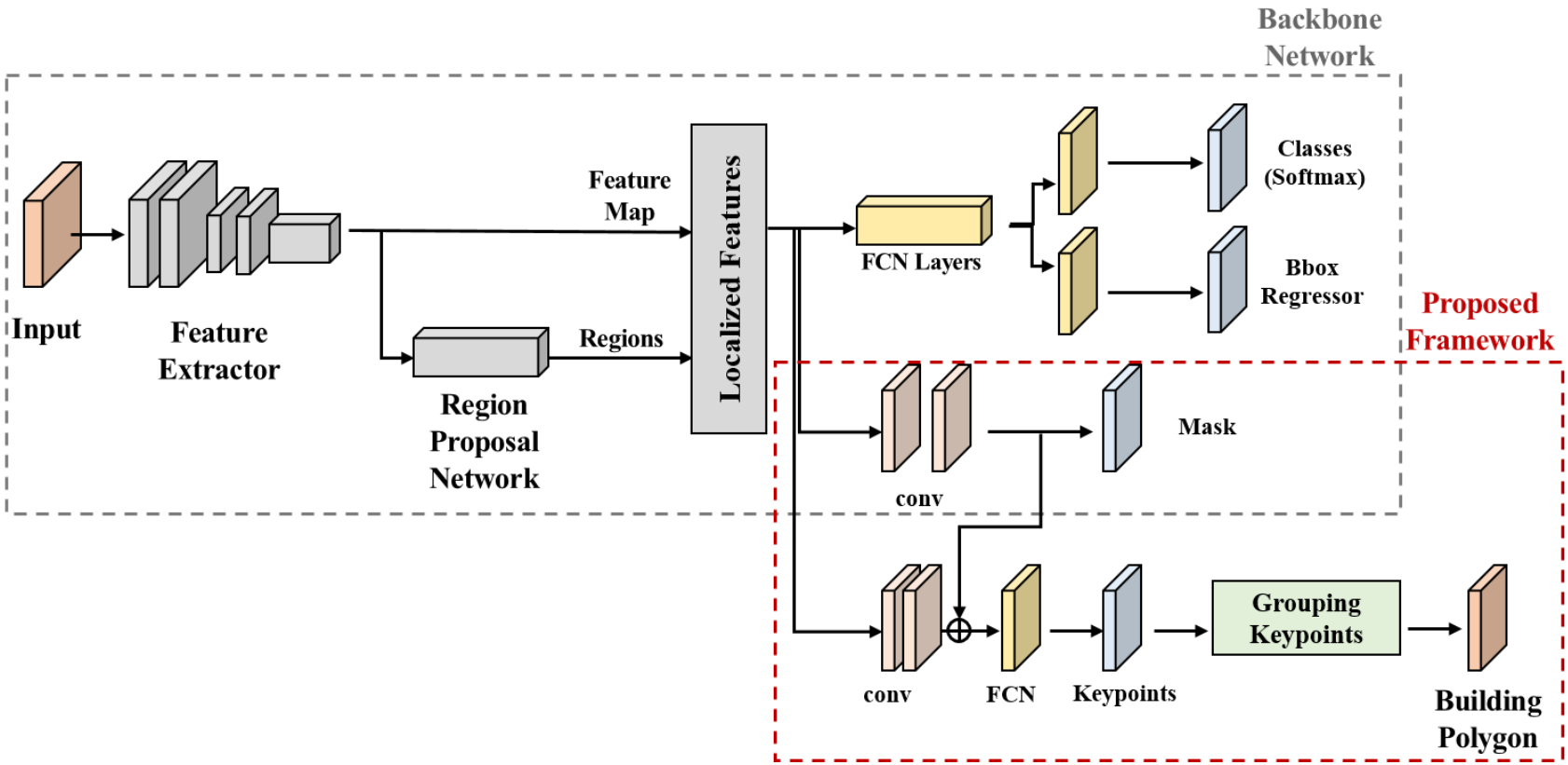


Figure 3.1. Overview of Backbone Network and Proposed Framework

3.1. Feature Extraction

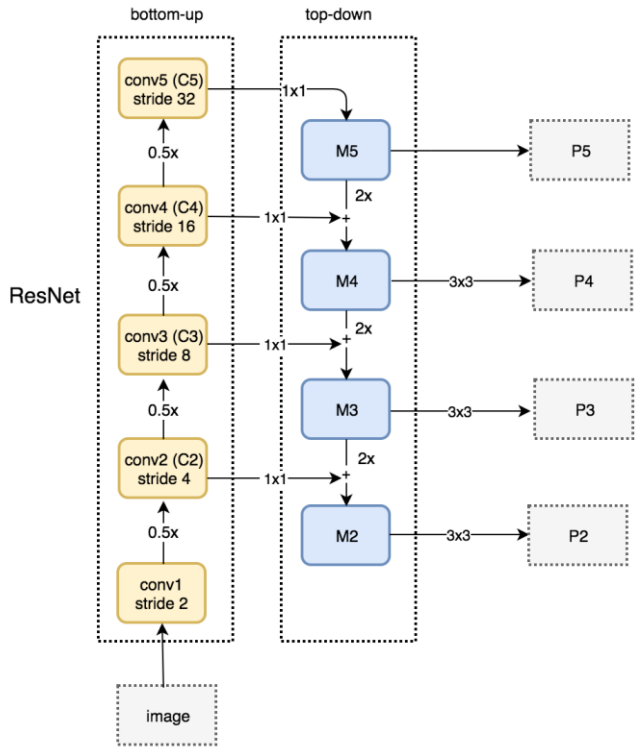


Figure 3.2. Feature Pyramid Networks for Object Detection

As shown in Figure 3.2, feature extraction is the conversion of a given input image into a set of features; this process is involves decreasing the quantity of assets needed to define a large set of information [45]. It is divided into bottom-up and top-bottom pathways. First, the image is input into a five-stage of ResNet (C1–C5); each stage of ResNet consists of several convolution layers and applies 2×2 pooling at the last layer to downsample the feature map.

The residual blocks substitute the convolutional blocks to deepen the networks. The top-down part of the network integrates features from different scales generated from the bottom-up part. It first applies a 1×1 convolution kernel to the current feature map and adds it element-wise with its upsampled previous feature map. It learns a 3×3 convolution to output the feature map, which has different scales (P2–P5).

Detecting objects at different scales is an essential task because the input satellite images cover larger areas and contain various sizes of building objects. The multiscale feature maps obtained from the FPN can detect building objects from different scales, unlike feature maps of only one scale (such as C5).

3.2. Localization

The RPN, which has predefined anchor boxes, generates the initial proposed bounding boxes, which are used to crop the feature maps to obtain the cropped features. RoI pooling is then adopted on the cropped features to obtain RoI features, which are input into the bounding-box regression and classification layer to produce the coordinate and class scores of the refined bounding boxes [15]. Lastly, the multiscale feature maps and the final bounding boxes are fed into the RoIAlign to generate precisely localized RoI features.

3.2.1. Region Proposal Network

The inputs of the RPN are the multiscale feature maps and predefined anchor boxes. These anchor boxes are generated with various sizes and width-to-height ratios for different strides on the input image (Figure 3.3). The RPN will predict bounding-box proposals for the corresponding anchor boxes through a box regression layer and class scores through classification on each entry of the feature maps. The classification layer is designed to distinguish positive and negative boxes, that is, boxes contain objects and empty boxes.

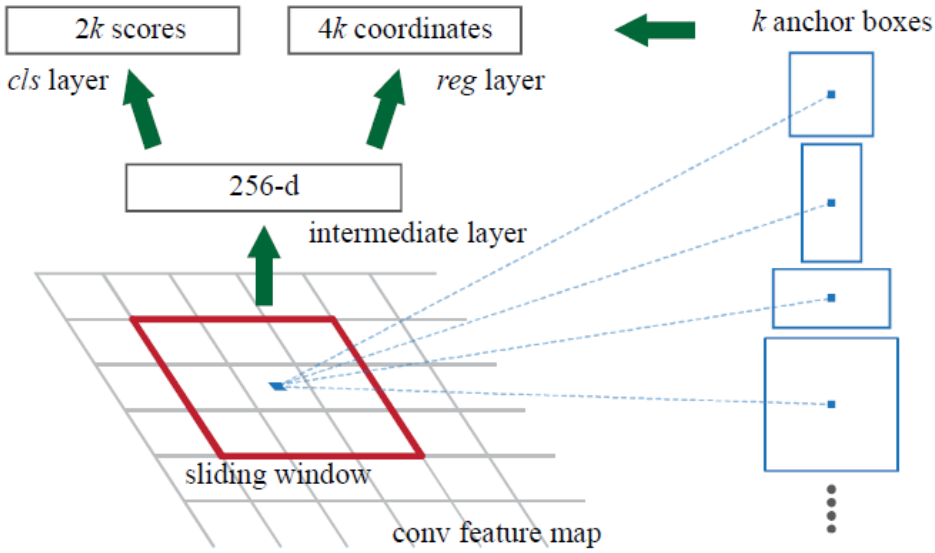


Figure 3.3. Region Proposal Network for Finding out the Possible Locations of the Targets in the Image [17]

The class score of each predicted bounding box is the objectness score. The bounding-box proposals are then filtered through Non Maximum Suppression (NMS) [46] based on a threshold of the objectness score to reduce its total number and to maintain the ratio of the positive and negative proposed boxes (usually 3:1). The box proposals are then closer to the object locations in the image compared with the input anchor.

3.2.2. Localization Layer

The multi-scale feature maps are cropped using the box proposals from the RPN. The feature maps are cropped based on the size of the box proposals according to the following equation:

$$k = [k_0 + \log_2(\sqrt{wh}/224)] \quad (3\text{-}1)$$

where w and h are the width and height, respectively, of the box proposal. $k_0 = 4$, and k is the level of scale for cropping the feature map. The cropped

features, which are of various scales, are fed into an RoI pooling layer to normalize them into RoI features with a fixed size of 14×14 [15]. The cropping operation performs feature scale selection, thus allowing the feature scales to match the size of the detected objects. Hence, feature maps with larger (smaller) resolutions correspond to smaller (bigger) objects. Identification of such correspondence takes advantage of the multiscale feature maps and can exploit richer and more accurate semantic information from different scales. As in the RPN, in the localization layer, the box regression and classification layer are applied to the RoI features to predict the coordinates and class scores of the bounding boxes. This second localization can further refine the bounding-box proposals generated from the initial bounding-box.

3.2.3. RoIAlign

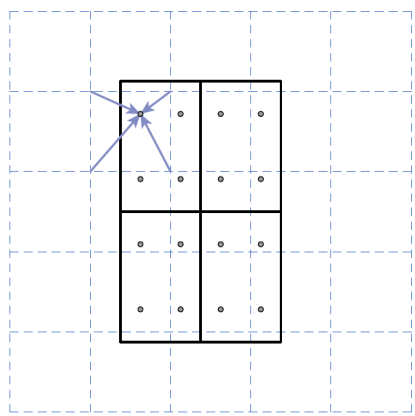


Figure 3.4. RoIAlign for Getting Precise Bounding Box Prediction[15]

The coordinates of the RoI features are usually floating numbers produced from the box regression layer. The cropping and RoI pooling will simply convert them into integer numbers in a quantization process, thus causing rounding errors and misalignment. As shown by a study on Mask R-CNN [13], simple cropping and RoI pooling are not sufficient for acquiring precisely localized feature maps for pixel-wise segmentation, which requires pixel-level accuracy (Figure 3.4). Therefore, it designs an RoIAlign layer to address

the misalignment problem and improve the accuracy of the RoI features. Instead of directly taking the integer of the floating RoI coordinates, which is affected by round-off errors in the cropping and RoI pooling, RoIAlign reserves the floating coordinates and uses differentiable bilinear interpolation to obtain the values of the floating points and the final localized RoI features.

3.3. Loss Function

For the training of the entire backbone network, losses should be calculated for the object detection of the two stages of the RPN, namely, box regression and classification. The loss function of the backbone network is the sum of the three types of losses, and it is defined as follows:

$$L = L_{cls} + L_{box} + L_{mask} \tag{3-2}$$

As for the classification score loss, a binary cross entropy loss of classification is computed:

$$L_{\text{cls}}\big(p(y)\big) = -(ylog\big(p(y)\big) + (1-y)\log\big(1-p(y)\big) \tag{3-3}$$

where *y* is the predicted class label (0 or 1) and p(y) is the probability score for the class label.

In accordance with Faster R-CNN [15], the box deltas between the predicted box and the ground truth box are calculated as inputs of the box regression loss function rather than the box coordinates. The deltas are defined as follows:

$$\begin{aligned} \mathrm{t_x} = (x - x_a)/w_a, \qquad \mathrm{t_y} = (y - y_a)/h_a, \\ t_w = \log(w/w_a), \qquad t_h = \log(h/h_a) \end{aligned} \tag{3-4}$$

$$\begin{aligned} \mathrm{t}_{\mathrm{x}}^{*} &= (x^{*} - x_a)/w_a, \qquad \mathrm{t}_{\mathrm{y}}^{*} = (y^{*} - y_a)/h_a, \\ t_w^{*} &= \log(w^{*}/w_a), \qquad t_h = \log(h^{*}/h_a) \end{aligned} \tag{3-5}$$

In the equation of the box regression, x and y are the coordinates of the center point of the bounding box, and w and h are its width and height, respectively. x, x_a, and x^{*} are for the predicted, anchor, and ground truth boxes, respectively.

The box deltas $(\mathrm{t}_{\mathrm{x}}, t_y, t_w, t_h)$ are predicted; they are equivalent to the regression from a predefined anchor box to a predicted box. To compare the predicted box deltas with the ground truth $(t_x^{*}, t_y^{*}, t_w^{*}, t_h^{*})$, which represent the regression values from an anchor box to its closest ground truth box, we adopt the smooth L1 loss [47].

$$L_{box}(t, t^{*}) = \sum smooth_{L1}(t - t^{*}) \tag{3-6}$$

The mask loss is defined as the average binary cross-entropy loss, which is a per-pixel sigmoid. The mask branch has an m^2-dimensional output for each RoI, which encodes binary masks of a resolution of m^2.

$$L_{mask} = -\frac{1}{\mathrm{m}^2} \sum_{1 \le i,j \le m} [y_{ij} log y_{ij}^{*} + (1 - y_{ij}) \log(1 - y_{ij}^{*})] \tag{3-7}$$

where y_{ij} is the ground truth of a pixel (i,j) in the true mask for a region of size $\mathrm{m} \times \mathrm{m}$. $\mathrm{y}_{\mathrm{ij}}^{*}$ is the predicted mask of the same cell in the mask.

Chapter 4. Proposed Network

The framework was designed to develop a segmentation model that can extract building footprints while producing low-complexity, geometry-preserving masks for each building. The proposed architecture combines polygon-based methods with Mask R-CNN (Figure 4.1). In this section, the backbone architecture for the building footprint extraction model is described first. Next, the proposed model is presented, which focuses on capturing the local geometry of the object.

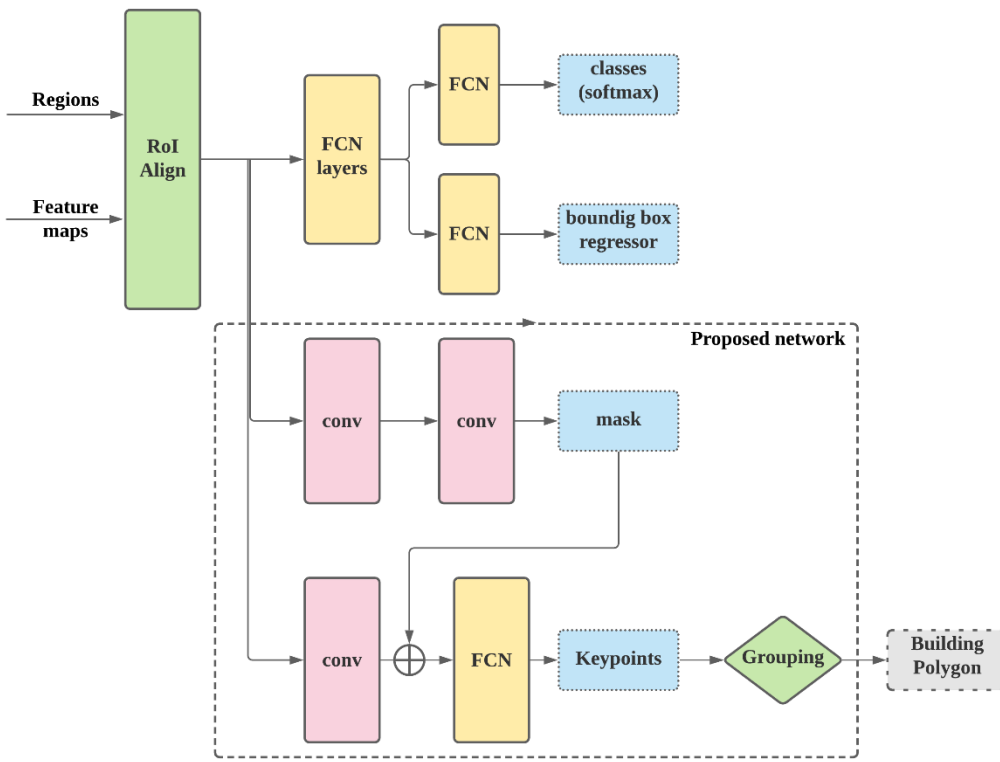


Figure 4.1. The Flowchart of the Proposed Approach for Building Footprint Extraction. FCN: Fully Convolutional Network, conv: convolutional block.

4.1. Instance Segmentation

The goal of instance segmentation is to provide a segmentation mask for polygon initialization for each individual object. The instance segmentation model is exploited to generate a segmentation mask for each instance in the scene, as in the typical Mask R-CNN. A bounding-box detection step is added to predict separate keypoints and partition the image into individual building instances. ResNet–FPN is integrated for supposing RoIs into the framework [44]. The FPN enhances the performance of the RPN by adding additional information through a multiscale pyramidal hierarchy of CNNs called feature pyramids.

Once bounding boxes of individual buildings are generated, the mask generation pipeline is processed. The pixel-wise mask of the target is predicted from the localized RoI extracted from the FPN. If the object detection model outputs proposal boxes, an FCN branch will be adopted on each RoI feature to predict the class probability of each instance, that is, to estimate a binary classification of buildings and backgrounds (Figure 4.2).

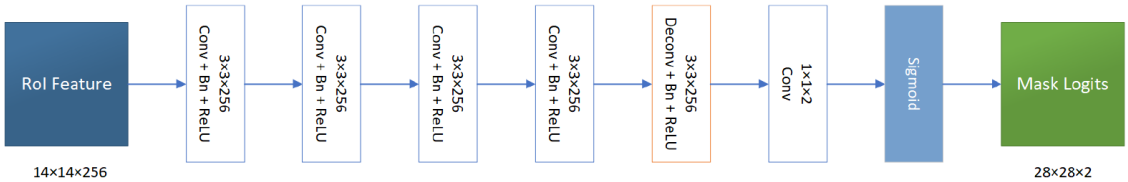


Figure 4.2. FCN Branch to Predict Mask Logits. The Input RoI Feature Fed into Sequential 3×3 Convolutional Layers, Batch Normalization and ReLU Layers [15].

The FCN branch consists of several sets of convolutions, batch normalization, and ReLU to learn the mask in addition to one deconvolutional layer to increase the resolution. At the end of the pipeline, a sigmoid activation layer is added to produce pixel-wise mask logits, which are used to compute the segmentation loss. The channel of the mask logits is 2, representing the probabilities for the foreground (building) and background

with respect to each pixel. A threshold of 0.5 is used to generated a binary mask from the foreground mask, which can be obtained from the pixel-wise segmentation of the building masks.

4.2. Keypoint Estimation

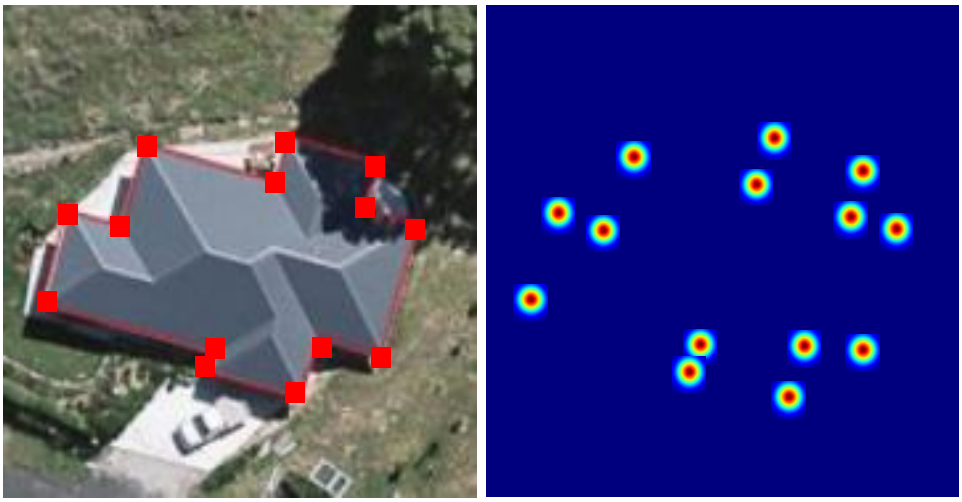


Figure 4.3. Input Image with Annotation and Its Keypoint Heatmap

The keypoint estimation step outputs a heatmap of keypoints for each object. The proposed keypoint prediction network is similar to that in [48]. Let $I \in \mathcal{R}^{H\times W\times 3}$ be an input image of height H and width W. For each image patch, there exists a corresponding ground truth heatmap such that $Y \in [0,1]^{H\times W}$. The aim of this process is to produce a corresponding heatmap of candidate keypoints, $\check{Y} \in [0,1]^{H\times W}$, which represents the vertices of each instance (Figure 4.3). A prediction $\check{Y} = 1$ corresponds to a detected keypoint, while $\check{Y} = 0$ denotes the background.

For each ground truth keypoint $p \in \mathcal{R}^2$, the ground truth keypoint map is guided by using the Gaussian kernel $\mathrm{Y}_{x,y} = \exp(-\frac{(x-p_x)^2+(y-p_y)^2}{2\sigma_p^2})$, where σ_p^2 is an object size-adaptive standard deviation [48]. In this regard, the penalty is reduced to negative locations within a radius of the positive location instead of applying equal penalization during training [7]. Therefore, the training object is set as a penalty-reduced pixel-wise logistic regression with modified focal loss to maintain a balance between positive and negative locations [49]. An example of the heatmap generation result is shown in

Figure 4.3.

$$L_{keypoint} = -\frac{1}{N}\sum_{xy}\begin{cases}\left(1-\check{Y}_{xy}\right)^{\alpha}\log\left(\check{Y}_{xy}\right) & if\ Y_{xy}=1\\ \left(1-\mathrm{Y}_{xy}\right)^{\beta}\left(\check{Y}_{xy}\right)^{\alpha}\log\left(1-\check{Y}_{xy}\right) & otherwise\end{cases} \quad (4\text{-}1)$$

where α and β are hyper-parameters for focal loss and N is the number of objects in a patch. For this study, the hyper-parameters are fixed as $\alpha = 2$ and $\beta = 4$, in accordance with [49].

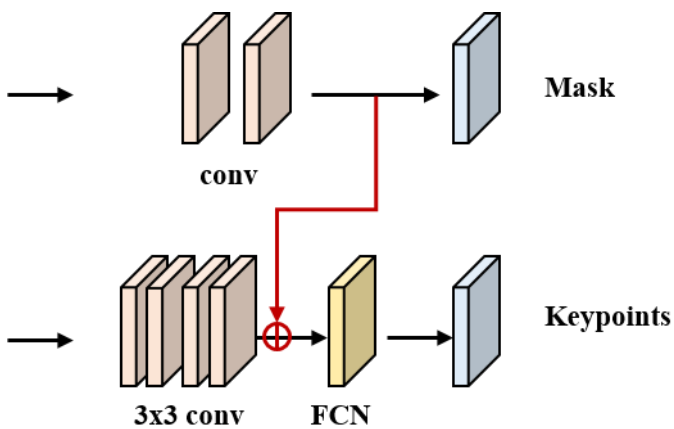


Figure 4.4. Mask and Keypoint Branches

Segmentation masks are regarded as a rough probability map that estimates the probability of each pixel belonging to the foreground or not [50, 51]. With the merging of the segmentation masks with the feature inside an RoI, the mask can be used to distinguish whether a keypoint is enclosed inside of an object or not (Figure 4.4). Then, the FCN is applied to the local features acquired by RoIAlign with the predicted mask to predict the heatmap of the keypoints.

Overall, the total loss function of the proposed network is a multitask loss function expressed as follows:

$$L = L_{cls} + L_{reg} + \lambda_{mask} L_{mask} + \lambda_{polygon} L_{polygon} \quad (4\text{-}2)$$

where L_{cls} is a cross-entropy loss for bounding-box classification and L_{reg} is a bounding-box loss for bounding-box regression, which is defined in [15]. For all experiments in this study, $\lambda_{mask} = 0.2$ and $\lambda_{polygon} = 1$ unless specified otherwise. The features of the backbone are passed through a separate 3×3 convolution, ReLU, and 1×1 convolution.

4.3. Grouping Keypoints

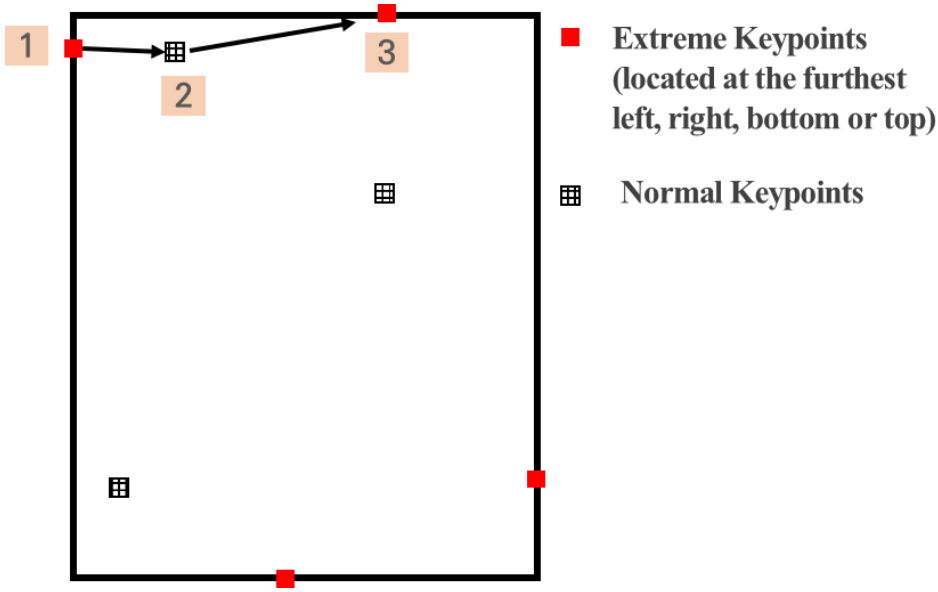


Figure 4.5. Strategy for Grouping Keypoints

To create a polygon where edges are sequentially connected with keypoints, a simple geometric method is adopted for the predicted keypoints (Figure 4.5). First, four extreme keypoints, located at the farthest left, right, bottom, or top of an instance, are selected as the group of the start point. Then, a Euclidean distance matrix of all keypoints is calculated to search for the distance connecting the extreme keypoints. The point with the shortest distance with respect to an extreme point is first selected as the start point, and the first edge is generated by establishing a connection between this initial point and its extreme point. Then, the generated point is considered the initial point for creating the next polygon; the next set of vertices is connected with those in its neighborhood. The grouping of keypoints is iterated until the final keypoint meets the initial keypoint. Finally, a polygon of an object is formed by integrating all of the generated edges. One limitation occurs when the shape of a ground truth object is concave; the grouping method fails to utilize all keypoints to create a complete polygon. This problem can be resolved by establishing a connection between the initial point and its next closest point.

Chapter 5. Experimental Design

In this chapter, the result of the proposed framework (introduced in Chapter 3) is presented. Section 4.1 introduces the characteristics of the data used to evaluate the models. Section 4.2 presents the experiments from data acquisition and pre-processing implementation of the network models to the training process and details.

5.1. Data Characteristics

SpaceNet 2 is one of the datasets utilized in this study for the evaluation of the proposed framework. The SpaceNet 2 dataset was distributed through the building footprint extraction challenge at the Conference on Computer Vision and Pattern Recognition (CVPR) 2018. The dataset contains high-resolution satellite images and ground truth for the building footprints. These images are from four cities (Las Vegas, Paris, Shanghai, and Khartoum) covering both urban and suburban regions. The dataset thus includes data with dissimilar characteristics. Each city is associated with four different continents, showing the high diversity of the dataset's characteristics, specifically with regards to the building roofs and landscape backgrounds. Sample images are shown in Figure 4.1. The images were captured by using the WorldView-3 satellite at a spatial resolution of 30 cm. Due to the non-nadir viewing angle, the roof boundaries and building footprints do not match accurately. The image size is 650×650, which is equal to an area of 200 m × 200 m. The total SpaceNet 2 dataset includes 24,587 satellite images with 302,701 building footprints.

The Open Cities AI dataset is also used in this study to evaluate the proposed framework. The dataset was provided by Open Cities AI Challenge: Segmenting Buildings for Disaster Resilience, which was hosted by the Global Facility for Disaster Reduction and Recovery. The images were acquired by using a drone with a spatial resolution of 7.5 cm to capture images

from Africa. The images contain small, highly diverse building roof styles. Unlike the SpaceNet 2 dataset, which consists of non-nadir satellite images, Open Cities AI contains orthorectified images. Hence, the roof boundaries and building footprints match correctly. However, since the labels were annotated automatically through OSM, the ground truth has omission errors. The entire dataset contains over 790,000 building footprint labels. Examples of images and annotations from both datasets are shown in Figure 5.1.

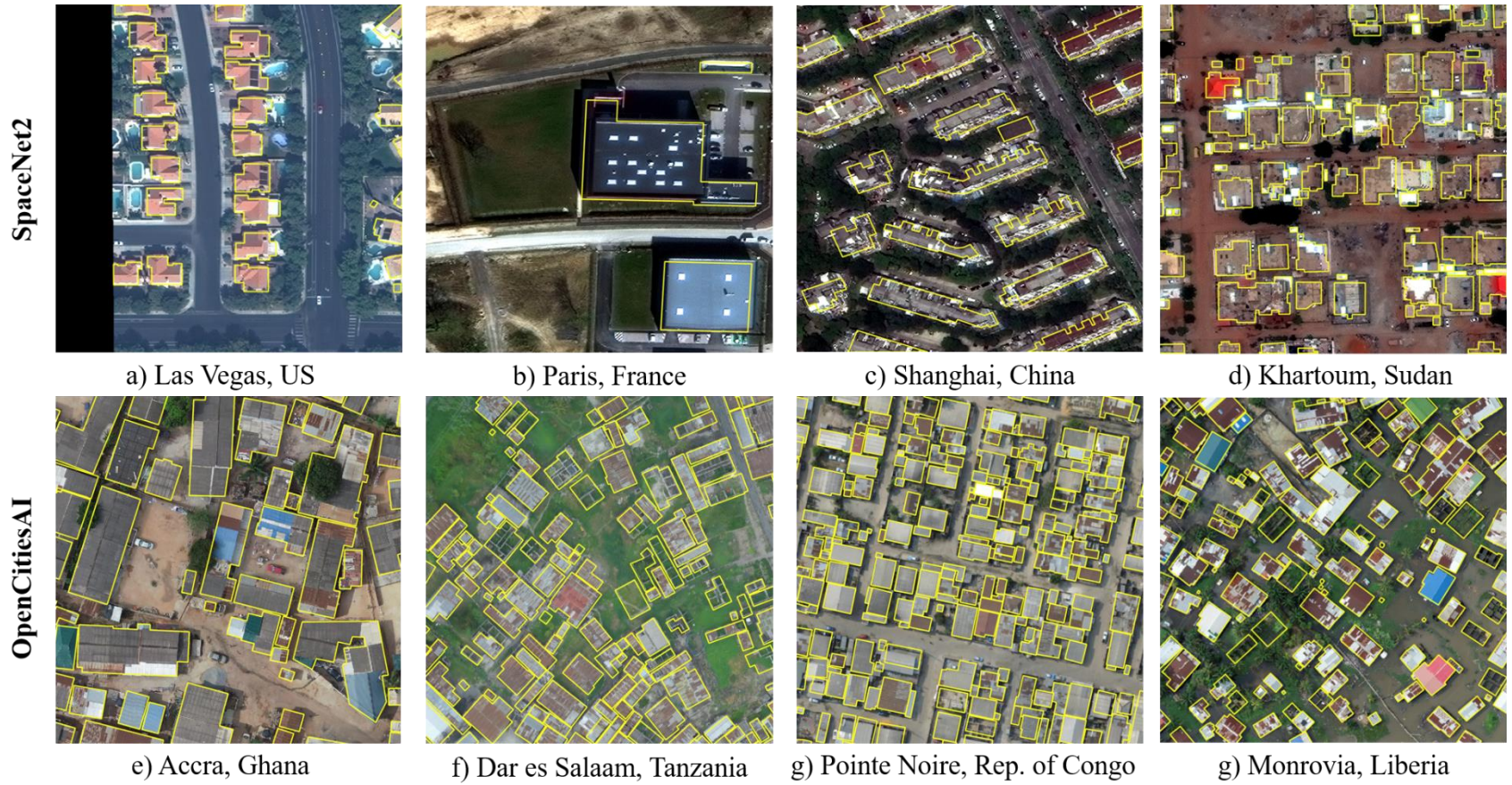


Figure 5.1. Sample Images of SpaceNet2 and OpenCitiesAI at the Four Different Sites.

5.2. Implementation Details

5.2.1. Data Pre-processing

SpaceNet 2 provides several types of satellite images, namely, multispectral, panchromatic, pansharpened multispectral, and pansharpened RGB images. The pansharpened RGB images were selected for this study's experiments, as the 3-band input enables the loading of pretrained weights for feature extractors, such as ResNet.

1) Each raw image has a radiometric resolution of 16 bits, so the resolution was converted to 8-bit, similar to that of a natural image. A two percent clip of each image's histogram was processed to improve the visual interpretation of the image.
2) The annotations of the building footprint labels are provided in GeoJSON format for each image. The annotations are specified with the following names: *ImageId*, *BuildingId*, *PolygonWKT-Pix*, and *PolygonWKT-Geo*. *ImageId* and *BuildingId* specify the unique identity of the images and building instances, respectively. *PolygonWKT-Pix* and *PolygonWKT-Geo* denote the coordinates of building polygon vertices in the image space (x,y) and the geographic space (latitude, longitude). The annotation *PolygonWKT-Geo* was deleted to convert the GeoJSON format to JSON..
3) The GeoJSON files were merged into one JSON file, in accordance with the COCO format [52]. For each object, the minimum and maximum coordinates $(x_{min}, y_{min}, x_{max}, y_{max})$ were identified and added to the collection of annotations.
4) The images were reshaped from 650×650 to 512×512 to modify the original images into a suitable size for common instance segmentation tasks. The original images were first upsampled using bicubic interpolation to produce 1024×1024 images. Then, these images were split into four 512×512 patches to minimize the

computational cost.

5) After pre-processing, the data for each city were divided into 70% training, 15% validation and 15% testing sample sets.

The Open Cities AI dataset features 7.5-cm-resolution drone images sized greater than $30{,}000 \times 30{,}000$ and in GeoTIFF data format. The annotation format of the GeoJSON files was converted to standard COCO format in a manner similar to pre-processing steps 2 and 3.

1) The resolution of each original image was reduced from 0.075 to 0.30 m by using nearest-neighbor interpolation. The original images include distortion created from geometrically correcting the mosaic images. As a result, multiple users downsampled the original resolution to 10 cm. A spatial resolution of 30 cm was selected for this study to compare the proposed network's performance with respect to SpaceNet 2.

Images that do not contain building footprints in the image scene were eliminated. Furthermore, since the shapes of the original images are irregular, a large number of nonuniform patches were created, thus adversely affecting the training phase.

5.2.2. Network Implementations and Configuration

The proposed framework was implemented using PyTorch 1.5.0. on Python 3.7. Instance segmentation models were implemented on top of Detectron2 [53]. Next, the hyper-parameter configuration of the framework will be presented.

The configuration of the backbone network is displayed in Table 5.1. The ResNet-50 architecture was used to control the feature extractor, and the anchor stride of the RPN layer was adjusted for object detection in the satellite images.

Table 5.1. Configurations of the Backbone Network

Items		Configurations
Feature Encoding	Input image size	(512, 512)
	ResNet layers	ResNet-50
	FPN Feature Size	(32,32), (64,64), (128,128)
Region Proposal Network	Anchor Stride	(8,16,32,64,128)
	Anchor Shape	(0.5, 1, 2)
	Anchor Scale	(32, 64, 128, 256, 512)
	NMS Threshold	0.5
	Max Box Number	256
Localization Layer	RoI Size	(28, 28)

5.2.3. Training and Testing Details

All models were trained on the SpaceNet 2 and Open Cities AI datasets. In the training of Mask R-CNN, the pretrained weights of ResNet-50 were adopted to initialize the backbone network. The batch size was set to 2, and the Adam optimizer was employed [54]. The learning rate was initialized as 10^{-4}, with a weight decay of 10^{-7} per 1000 epochs. Mask R-CNN was also trained and evaluated on the same dataset as the baseline with the same configuration using Detectron2. For performance comparison with the semantic segmentation approach, the winning algorithm of the Open Cities AI Challenge was also trained using the same conditions. This algorithm uses a U-Net encoder–decoder architecture with EfficientNet-D2 [23]. The network was trained on a single NVIDIA GeForce 2080 Ti with 12 GB memory.

5.3. Metrics for Quantitative Analysis

There are two measurements for comparing quantitative results in semantic segmentation: IoU and F1 score. In addition to these, the SSIM was used in this work to compare the shapes of the results.

IoU is defined as follows:

$$\text{IoU} = \frac{Area(A \cap B)}{Area(A \cup B)}$$

ere A and B denote the predicted and the ground truth, respectively, of a building polygon. IoU is equal to the intersection area of A and B divided by their union area, which is illustrated in Figure 5.2:

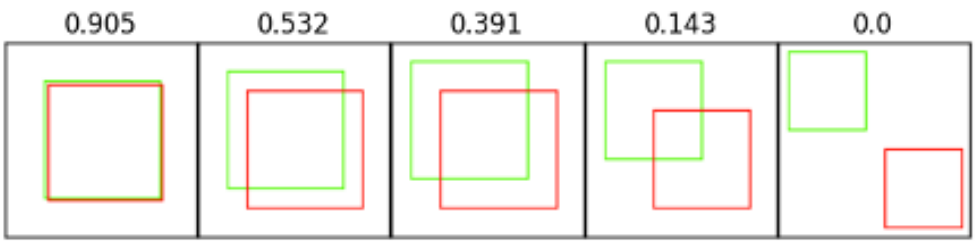


Figure 5.2. Sample Intersection over Union (IoU) Scores

The predicted building polygon was counted as a true positive (TP) if (1) it was the closest proposal to a label and (2) the IoU between the prediction and the label was beyond the prescribed threshold of 0.5. Otherwise, the proposed polygon was regarded as a false positive (FP). The labeled polygons that were not detected or missed in the predictions were denoted as false negatives (FNs). After the TP, FP, and FN polygons were counted, the F1 score was employed, which is the harmonic mean of precision and recall; that is, it combines the accuracy in the precision measure and the completeness in the recall measure. Suppose there are N polygon labels for the ground truth building footprints and M predicted polygons. The F1 score is calculated as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad Recall = \frac{TP}{TP + FN}$$

$$\text{F1} = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{M + N}$$

The additional metric, SSIM, was employed to evaluate the performance of the segmentation masks, namely, the predicted binary building mask and the ground truth. SSIM measures the similarity of two images from brightness, contrast, and structure information [55]. The SSIM is represented by the following:

$$\text{SSIM}(\text{x}, \text{y}) = \frac{\left(2\mu_x\mu_y + C_1\right) + (2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_1\right)}$$

where:

μ_x: $the\ average\ of\ x$;

μ_y: $the\ average\ of\ y$;

σ_x^2: $the\ variance\ of\ x$;

σ_y^2: $the\ variance\ of\ y$;

σ_{xy}: $the\ covariance\ of\ x\ and\ y$;

$\text{c}_1 = (k_1 L)^2,\ c_2 = (k_2 L)^2$:

$two\ variable\ to\ stabilize\ the\ division\ with\ weak\ denominator$;

$L = 2^{\#bite\ per\ pixel} - 1$: $the\ dynamic\ range\ of\ the\ pixel - values$;

$k_1 = 0.01\ and\ k_2 = 0.03\ by\ default.$

Chapter 6. Results and Discussion

In this section, the performance of the proposed framework is evaluated using three metrics. The mask accuracy is evaluated by the F1-score, IoU and SSIM.

6.1. Building Extraction Accuracy

The evaluation results for the two datasets and a total score were computed for the proposed model and control groups. The results are shown in Table 6.1. Compared with the typical instance segmentation model, in Mask R-CNN, the proposed method generates consistently higher segmentation accuracy over the other models. However, since the proposed network uses the same object detection structure as Mask R-CNN, there is a clear limitation in performance improvement.

Table 6.1. Building Extraction Accuracy

Dataset	Model	mIoU	F1-score	SSIM
SpaceNet2	Mask R-CNN	69.2	0.711	0.815
	EfficientNet-b1	78.1	0.814	0.851
	Proposed	70.3	0.721	0.824
OpenCitiesAI	Mask R-CNN	61.4	0.633	0.648
	EfficientNet-b1	71.1	0.730	0.716
	Proposed	62.6	0.636	0.663

Compared with the state-of-the-art semantic segmentation model that uses EfficientNet-B1, the proposed framework shows lower accuracy in all metrics. The winning algorithm of the Open Cities AI competition used a combination of EfficientNet-B1 and U-Net architecture. Since the two-stage detector has a limitation in that the bounding box of small objects may be omitted by the NMS algorithm, the detection accuracy for small objects tends to decrease more than others. The semantic segmentation inferences of the building mask

predictions are produced in a pixel-wise manner, thus helping improve the overall accuracy of the mask prediction. However, the semantic segmentation models only produce pixel-wise semantic labels to classify buildings; they cannot distinguish individual building objects. To address this problem, post-processing algorithms, such as the watershed algorithm, were employed by competitors to separate building regions or train the edges between adjacent buildings [13].

Similar trends are observed even when the same models are trained on different datasets, but they show lower performance on the Open Cities AI dataset. This result is due to the differences in the characteristics of the datasets; the Open Cities AI dataset consists of images from urban areas in developing countries in Africa, which have a large proportion of small and densely distributed buildings. This makes it difficult to distinguish between the background and the building in cases of adjacent buildings. The results above reveal that the selection of basic segmentation models and the incorporation of keypoint geometry marginally increase accuracy.

6.2. Impact of Detector

Since localized features are used for segmentation, the localization performance of the object detector significantly impacts performance. For comparison with the state-of-the-art object detector, FCOS [56] was substituted as the backbone network, while the same feature extractor, ResNet-50, was used. By performing deeper inference by using FCN, FCOS can perform better than RPN, which consists of multiple CNNs [18]. FCOS is also an anchor-free detector; hence, it is less affected by the NMS algorithm, which removes anchor candidates. The model performance was evaluated based on the integration of the proposed keypoint detection module.

Table 6.2. Accuracy Indices of Different Instance Segmentation Methods

Model	mIoU	F1-score	SSIM
Mask R-CNN	69.2	0.711	0.815
Mask R-CNN + Keypoints	70.3	0.721	0.824
CenterMask (FCOS+segm)	74.6	0.781	0.853
FCOS+ Keypoints	75.1	0.789	0.863

Table 6.3. Accuracy of F1-score on Four Cities in SpaceNet2

Model	LasVegas	Paris	Shanghai	Khartoum	Total
Mask R-CNN	0.882	0.742	0.625	0.516	0.711
Mask R-CNN + Keypoints	0.890	0.767	0.634	0.523	0.721

Table 6.2 shows that the building footprint extraction performance in the two-stage instance segmentation is closely related to the object detection performance since the segmentation tasks executed on the localized RoI features extracted by the detection results. Additional improvement in the detector performance can be expected through optimal hyper-parameter tuning.

Table 6.3 shows the effectiveness of the anchor-based detector, which is used in Mask R-CNN. The relatively low F1 scores for Shanghai and Khartoum result from the annotation of low-quality labels and the large number of buildings that were not orthorectified. Especially in Khartoum, buildings are densely distributed in irregular patterns. Consequently, an appropriate anchor cannot be easily detected to execute the segmentation branch.

6.3. Keypoint Detection

To detect the keypoints of an instance, the input feature not only uses the localized features but also concatenates them with the prediction mask. Experiments were conducted for three scenarios to verify this approach: (1) predicting the segmentation mask only with multiple convolutional networks, (2) detecting keypoints through a fully convolutional layer on the localized features directly, and (3) predicting keypoints through a fully convolutional layer by adding the localized features and the mask (Table 6.4).

Table 6.4. Accuracy of 3 different Segmentation Scenarios.

Model	mIoU	F1-score
Mask only	69.2	0.711
Keypoint only	68.1	0.671
Mask + Keypoint	70.3	0.721

If the keypoints are to be extracted from the localized features alone, the performance will be lower compared with the two other cases. This is because the keypoint of another instance is not distinguished when creating a heatmap for learning keypoints; thus, the keypoints of adjacent buildings are also detected. The recall is improved, but the IoU and F1 score tend to be lower. This problem can be resolved if the predicted mask is used as a feature together with the localized RoI features. The generated masks are regarded as possibly belonging to the same instance. Its inclusion in the prediction process enables the classification of instances in the RoI.

6.4. Qualitative Analysis

The result reveals that the proposed network produces visually superior results in extracting the footprints of the buildings detected from the extension of Mask R-CNN. However, since keypoint estimation is performed only on the object detected by the RPNs, many FN pixels are observed. For comparison with the semantic segmentation approach, U-Net with EfficientNet-B1 and the proposed framework were also compared. Since semantic segmentation omits the RPN process and performs pixel-based classification, it shows relatively better precision compared with the instance segmentation-based approach. However, without the polygonization process as post-processing, rough building polygons are extracted, and in the case of large buildings, FN pixels may be generated inside of the objects. Unlike U-Net and Mask R-CNN, the proposed framework can generate acceptable outlines in such a case.

Figure 6.1 shows the results of the proposed framework compared with those of Mask R-CNN. The instance segmentation model generates pixel-based masks after the instantiation of each object and considers each boundary as a building footprint. In the DCNNs of repeating pooling and downsampling layers, edge information is partially lost, and the generated mask is rough and round. By contrast, the proposed framework aims to directly predict the vertices of the object; the segmentation mask predicted in the model is considered additional information for keypoint estimation. By grouping keypoints into independent polygons, the proposed framework can predict more realistic building footprints from the satellite and aerial images.

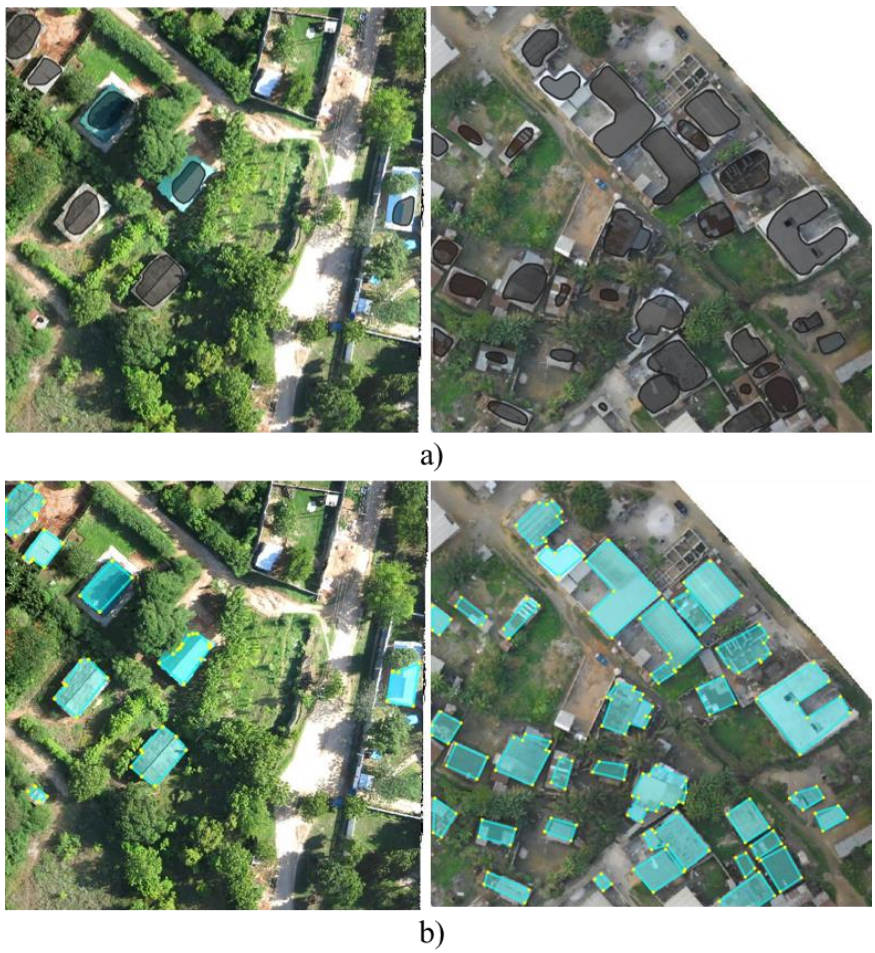


Figure 6.1. Comparison of the Results of Mask R-CNN and the Proposed Framework; a) Segmentation by Mask R-CNN, b) Results from proposed framework.

Chapter 7. Conclusion

In this study, a building footprint extraction framework using keypoint detection is presented. First, a keypoint detection module integrates the localized RoI features from mask prediction. Since the number of keypoints of a detected building is not fixed, keypoint detection is conducted using an FCN to predict the various numbers of keypoints. The proposed methodology involves deep learning models composed of a backbone network and the abovementioned keypoint detection module. It groups points to generate polygons for building footprint detection.

The proposed framework was evaluated using two datasets composed of satellite and aerial images that differed in spatial resolution and building distribution. The experiments demonstrated that the proposed framework successfully improves the visibility of the output mask's shape. Additional experiments were conducted to verify the validity of the keypoint detection module's branch design.

State-of-the-art one-stage models [48, 57] can directly detect bounding boxes instead of relying on anchors, and they are more suitable for building detection in urban areas with densely distributed buildings. Aside from the use of keypoints, the use of information extracted from kinetic polygonal partitioning, inspired by the superpixel algorithm, could improve the accuracy of building footprint extraction.

References

1. Pasquali, G., G.C. Iannelli, and F.J.R.S. Dell'Acqua, *Building Footprint Extraction from Multispectral, Spaceborne Earth Observation Datasets Using a Structurally Optimized U-Net Convolutional Neural Network.* 2019. **11**(23): p. 2803.
2. Sohn, G., I.J.I.J.o.P. Dowman, and R. Sensing, *Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction.* 2007. **62**(1): p. 43-63.
3. Zhao, K., et al. *Building Extraction From Satellite Images Using Mask R-CNN With Building Boundary Regularization*. in *CVPR Workshops*. 2018.
4. Fan, H., et al., *Quality assessment for building footprints data on OpenStreetMap.* 2014. **28**(4): p. 700-719.
5. Xu, Y., et al., *Building extraction in very high resolution remote sensing imagery using deep learning and guided filters.* 2018. **10**(1): p. 144.
6. Ji, S., et al., *Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set.* 2018. **57**(1): p. 574-586.
7. Zhang, L., et al., *An Efficient Building Extraction Method from High Spatial Resolution Remote Sensing Images Based on Improved Mask R-CNN.* 2020. **20**(5): p. 1465.
8. Iglovikov, V. and A.J.a.p.a. Shvets, *Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation.* 2018.
9. Li, Z., J.D. Wegner, and A. Lucchi. *Topological map extraction from overhead images*. in *Proceedings of the IEEE International Conference on Computer Vision*. 2019.
10. Sirmacek, B., C.J.I.T.o.G. Unsalan, and R. Sensing, *Urban-area and building detection using SIFT keypoints and graph theory.* 2009. **47**(4): p. 1156-1167.
11. Wei, F., et al. *Point-set anchors for object detection, instance segmentation and pose estimation*. in *European Conference on Computer Vision*. 2020. Springer.
12. Georgakis, G., et al. *End-to-end learning of keypoint detector and descriptor for pose invariant 3D matching*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
13. Long, J., E. Shelhamer, and T. Darrell. *Fully convolutional networks for semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
14. Ronneberger, O., P. Fischer, and T. Brox. *U-net: Convolutional networks for biomedical image segmentation*. in *International Conference on Medical image computing and computer-assisted intervention*. 2015. Springer.
15. He, K., et al. *Mask r-cnn*. in *Proceedings of the IEEE international conference on computer vision*. 2017.
16. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
17. Ren, S., et al. *Faster r-cnn: Towards real-time object detection with region proposal networks*. in *Advances in neural information processing systems*.

2015.
18. Lee, Y. and J. Park. *CenterMask: Real-time anchor-free instance segmentation*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
19. Dornaika, F., et al., *Building detection from orthophotos using a machine learning approach: An empirical study on image segmentation and descriptors.* 2016. **58**: p. 130-142.
20. Rottensteiner, F. and C. Briese. *A new method for building extraction in urban areas from high-resolution LIDAR data*. in *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*. 2002. Natural Resources Canada.
21. Ji, S., S. Wei, and M.J.I.J.o.R.S. Lu, *A scale robust convolutional neural network for automatic building extraction from aerial and satellite imagery.* 2019. **40**(9): p. 3308-3322.
22. Yuan, J.J.I.t.o.p.a. and m. intelligence, *Learning building extraction in aerial scenes with convolutional networks.* 2017. **40**(11): p. 2793-2798.
23. Tan, M., R. Pang, and Q.V. Le. *Efficientdet: Scalable and efficient object detection*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
24. Szegedy, C., et al., *Inception-v4, inception-resnet and the impact of residual connections on learning.* 2016.
25. Chen, L.-C., et al., *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.* 2017. **40**(4): p. 834-848.
26. Shi, Y., et al., *Building segmentation through a gated graph convolutional neural network with deep structured feature embedding.* 2020. **159**: p. 184-197.
27. Li, Q., et al., *Building Footprint Generation by Integrating Convolution Neural Network With Feature Pairwise Conditional Random Field (FPCRF).* 2020.
28. Liang, J., et al. *Polytransform: Deep polygon transformer for instance segmentation*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
29. Hershberger, J.E. and J. Snoeyink, *Speeding up the Douglas-Peucker line-simplification algorithm*. 1992: University of British Columbia, Department of Computer Science Vancouver, BC.
30. Castrejon, L., et al. *Annotating object instances with a polygon-rnn*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
31. Acuna, D., et al. *Efficient interactive annotation of segmentation datasets with polygon-rnn++*. in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018.
32. Yang, X., et al. *Scrdet: Towards more robust detection for small, cluttered and rotated objects*. in *Proceedings of the IEEE International Conference on Computer Vision*. 2019.
33. Zuo, J., et al., *Aircraft type recognition based on segmentation with deep convolutional neural networks.* 2018. **15**(2): p. 282-286.
34. Barroso-Laguna, A., et al. *Key. net: Keypoint detection by handcrafted and learned cnn filters*. in *Proceedings of the IEEE International Conference on*

Computer Vision. 2019.
35. Rosten, E. and T. Drummond. *Machine learning for high-speed corner detection*. in *European conference on computer vision*. 2006. Springer.
36. Zhang, X., et al. *Learning discriminative and transformation covariant local feature detectors*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
37. Alp Güler, R., N. Neverova, and I. Kokkinos. *Densepose: Dense human pose estimation in the wild*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
38. Pavlakos, G., et al. *6-dof object pose from semantic keypoints*. in *2017 IEEE international conference on robotics and automation (ICRA)*. 2017. IEEE.
39. Zhou, X., J. Zhuo, and P. Krahenbuhl. *Bottom-up object detection by grouping extreme and center points*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
40. Li, Q., et al., *Instance segmentation of buildings using keypoints.* 2020.
41. Gupta, R., et al., *xbd: A dataset for assessing building damage from satellite imagery.* 2019.
42. Chen, Q., et al., *Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings.* 2018.
43. Demir, I., et al. *Deepglobe 2018: A challenge to parse the earth through satellite images*. in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018. IEEE.
44. Lin, T.-Y., et al. *Feature pyramid networks for object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
45. Dara, S. and P. Tumma. *Feature extraction by using deep learning: a survey*. in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. 2018. IEEE.
46. Neubeck, A. and L. Van Gool. *Efficient non-maximum suppression*. in *18th International Conference on Pattern Recognition (ICPR'06)*. 2006. IEEE.
47. Liu, Y. and L. Jin. *Deep matching prior network: Toward tighter multi-oriented text detection*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
48. Law, H. and J. Deng. *Cornernet: Detecting objects as paired keypoints*. in *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.
49. Lin, T.-Y., et al. *Focal loss for dense object detection*. in *Proceedings of the IEEE international conference on computer vision*. 2017.
50. Li, M., F. Lafarge, and R. Marlet. *Approximating shapes in images with low-complexity polygons*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
51. Luo, C., X. Chu, and A.J.a.p.a. Yuille, *Orinet: A fully convolutional network for 3d human pose estimation.* 2018.
52. Lin, T.-Y., et al. *Microsoft coco: Common objects in context*. in *European conference on computer vision*. 2014. Springer.
53. Wu, Y., et al., *Detectron2*. 2019.
54. Kingma, D.P. and J.J.a.p.a. Ba, *Adam: A method for stochastic optimization.* 2014.
55. Wang, Z., et al., *Image quality assessment: from error visibility to structural similarity.* 2004. **13**(4): p. 600-612.

56. Tian, Z., et al. *Fcos: Fully convolutional one-stage object detection*. in *Proceedings of the IEEE international conference on computer vision*. 2019.
57. Li, K., et al., *Object detection in optical remote sensing images: A survey and a new benchmark.* 2020. **159**: p. 296-307.

국문 초록

고해상도 위성영상 및 항공영상에서의 건물경계추출을 위한 특징점 기반의 딥러닝 접근

서울대학교 대학원
공과대학 건설환경공학부
정 도 영

건물은 도심지역의 핵심적인 단위로서, 건물경계추출은 원격탐사 도메인에서 활발히 연구되는 주제이다. 건물경계란 건축물의 지상경계선을 말하는 것으로, 기본도 및 재난피해평가 등 도심지 분석을 위한 기본 단위이기에 LiDAR와 광학영상을 합성하는 접근의 연구가 수행되어왔다. 심층 합성곱 신경망의 도입 이후 광학영상 단독으로도 기존의 다중센서기반 건물경계추출과 비슷하거나 더 높은 성능을 보이고 있다. 그러나, 건물경계추출을 위한 주된 심층 합성곱 신경망 기법인 의미론적 분할 (semantic segmentation) 접근은 깊은 합성곱 레이어의 광의의 수용영역 (receptive field)과 풀링 레이어 (pooling layer)을 반복적으로 통과하며 경계정보가 누락되어 경계가 고르지 않은 건물 폴리곤을 생성하는 한계가 있다.

이러한 문제를 해결하기 위해 원본영상으로부터 벡터형태의 기하학적 형상을 직접 추출하기 위한 일련의 연구들이 제시되어왔다. 이 접근 방식의 연장선상에서, 본 논문은 분할 마스크 (segmentation mask)와 특징점 탐지 (keypoint) 기법을 결합한 딥러닝 프레임워크를 제안함으로써, 네트워크의 결과로 건물폴리곤의 모서리를 직접 예측 후 후처리를 통해 결합함으로써 정교한 건물경계 결과물을 도출하는 것을 목적으로 한다. 본 프레임워크의 타겟

특징점은 각 건물객체의 모서리로써, local gradient의 유의미한 차이를 보이는 곳을 의미한다. 제안한 프레임워크는 건물탐지와 탐지 후 각 객체의 특징점 탐지를 결합한 2단계 하향식 접근 방식을 제안한다. 일반적인 Instance Segmentation 네트워크인 Mask R-CNN을 백본 네트워크 (backbone network)로 사용하며, 동일한 방법으로 객체탐지를 수행하나, 분할 마스크 생성모듈을 새롭게 제안하는 특징점 탐지모듈로 대체한다.

특징점 탐지에 있어 인접건물의 영향을 축소하기위해 단순한 완전 합성곱 신경망으로부터 생성한 대략적인 건물 마스크와 관심영역의 국지적 feature을 병합 후 완전 합성곱 신경망을 적용함으로써 각 건물 개체의 특징점을 예측한다. 그 후, 간단한 기하학적 방법을 통해 특징점을 군집하여 벡터화된 건물 다각형을 생성한다.

제안한 프레임워크의 학습을 위하여 건물경계추출을 위한 위성영상 기반 SpaceNet2 데이터셋을 사용하였다. 단, SpaceNet은 건물경계와 지붕경계간의 이격이 발생하는 문제가 존재하며, 이 문제에서 자유로운 드론으로 촬영된 정사영상 기반의 OpenCitiesAI 데이터에서 추가적으로 학습을 수행하였다.

제안한 프레임워크를 검증하기 위해 일반적으로 사용되는 의미론적 분할모델인 U-net과 대표적인 인스턴스 단위 분할기법인 Mask R-CNN을 함께 사용하였다. 동일한 목적을 갖는 최신모델과의 직접적 비교를 위하여 OpenCitiesAI의 우승기법인 EfficientNe-U-Net 모델역시 비교대상에 포함하였다. 제안한 모델은 F1 score, IoU (Interest-of-Union) 및 SSIM (Structure Similarity Index Measure)의 세 가지 지표로 평가되었다.

제안된 프레임워크는 백본네트워크인 Mask R-CNN에 비교하여 정량적 평가에서 건물경계추출 성능을 개선했음을 보여준다. 하지만, 최신모델인 의미론적분할 기반의 EfficientNet–U-Net에 비해서는 여전히 낮은 지표를 보인다. 이는 제안한 네트워크가 건물탐지와 특징점탐지로 구분된 2단계 접근방식을 채택하였기 때문으로, 건물탐지성능에 모델의 퍼포먼스가 크게

의존한다. 그러나, 앞선 네트워크에서 탐지된 객체에 한정하여 제안된 프레임워크는 건물객체의 모서리를 직접 추출함으로써 딥러닝 네트워크의 결과만으로 벡터화된 객체를 도출하고, 실제 건물경계와 유사하게 모서리와 경계로 구성되어 시각적으로 향상된 건물경계추출이 가능하다는 점을 보였으며, 여기에 논문의 의의가 있다.

Keywords : 건물경계추출, 특징점 탐지, 인스턴스 단위 분할, 딥러닝, 위성영상
Student Number : 2019-20867