

2022-10-31

Automated Building Extraction from Remote Sensing Imagery Using Deep Learning

Yan, Hailun

Yan, H. (2022). Automated building extraction from remote sensing imagery using deep learning (Master's thesis, University of Calgary, Calgary, Canada). Retrieved from <https://prism.ucalgary.ca>.
<http://hdl.handle.net/1880/115403>

Downloaded from PRISM Repository, University of Calgary

UNIVERSITY OF CALGARY

Automated Building Extraction from Remote Sensing Imagery Using Deep Learning

by

Hailun Yan

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN GEOMATICS ENGINEERING

CALGARY, ALBERTA

OCTOBER, 2022

© Hailun Yan 2022

Abstract

Automatically extracting high-quality building-footprint polygons from satellite and aerial images is crucial for supporting various land use and land cover mapping applications. The conventional building polygon extraction process requires hand-crafted features and high human intervention, which is time-consuming and often has limited generalization capability. In recent years, deep learning-based methods have shed light on the problem with higher levels of automation, segmentation quality, and generalization capability. These methods often involve two stages: first, building segmentations are predicted from remote sensing images using deep neural networks (DNNs); next, the irregular-shaped building segmentations are regularized into straight-edged and right-angle-cornered building polygons using conventional or deep learning-based methods. As a result, the extraction performance is often highly affected by the quality of the segmentation predictions. However, from experiments, the current widely used segmentation DNNs show significant defects in their building segmentation results, especially for the buildings with rotated angles (i.e., neither parallel nor vertical) between the building edges and the image edges. Moreover, although DNN-based regularization methods have shown greater generalization potentials at regularizing buildings in various shapes compared to the conventional regularization methods, the qualities of the regularization results are generally dissatisfactory.

This thesis proposes an end-to-end deep learning-based building extraction method based on PolygonCNN. The proposed model consists of a segmentation module to predict building segmentations and a regularization module to regularize the building contours traced from the building segmentation results. First, an upgraded Mask R-CNN model, which is integrated with the rotatable bounding box technique, the Swin Transformer backbone network, and the FPN

module, is adopted as the segmentation module of the proposed model to segment buildings in vastly different scales and orientations. Moreover, the Feature Pooling module and the BRegNet of the original PolygonCNN are modified to exploit the multi-scale feature maps of the FPN module. As a result, the proposed model can effectively extract high-quality building polygons with various scales and orientations and has shown promising performance compared to several other popular end-to-end deep-learning-based building extraction models. In addition, the thesis provides supplemental architecture choices, which offer flexibility between the quality of the building extraction result and the memory consumption of the model.

Acknowledgements

First, I would like to express my most profound appreciation to my supervisor, Dr. Ruisheng Wang, for his involvement, insight, encouragement, and support during the course of my research, and for the help with my life outside school.

I would also like to thank several of my schoolmates, colleagues and good friends, Akshay Bharadwaj, Dr. Bo Guo, Hongxin Yang, and Perpetual Hope Akwensi, and Xianghe Xu, for helping me greatly with my research and life.

Last, I am deeply indebted to my parents for their love, encouragement, and inspiration. This endeavour would not have been possible without their generous support.

Table of Contents

Abstract	i
Acknowledgements	iii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
 CHAPTER 1: INTRODUCTION	 1
1.1 Background	1
1.2 Applications of Automated Building Footprint Extraction	2
1.3 Major Challenges for Automated Building Footprint Extraction	3
1.3.1 An Issue With The Leaning-Based Building Segmentation Methods	3
1.3.2 Issues With The Conventional Building Regularization Methods	4
1.3.3 Issues With Deep-Learning-Based Building Regularization Methods	5
1.4 Thesis Contributions	6
 CHAPTER 2: RELATED WORK	 8
2.1 Building Segmentation	8
2.1.1 Conventional Building Segmentation	9
2.1.2 Deep Learning-Based Building Segmentation	10
2.2 Building Boundary Regularization	12
2.2.1 Conventional Building Regularization	12
2.2.2 Deep Learning-Based Building Regularization	13
2.3 end-to-end building polygon prediction	14
2.4 Integrating Rotatable Bounding Boxes with Deep Learning Models	15
 CHAPTER 3: THE PROPOSED METHOD	 17
3.1 Model Reviews	17
3.1.1 Review of PolygonCNN	17
3.1.2 Review of Mask R-CNN	23
3.1.3 Review of The FPN module	29
3.1.4 Review of Swin Transformer	32
3.2 The Proposed Model	37
3.2.1 Integration of The Rotatable Bounding Box with Mask R-CNN	38

3.2.2 Integration of The FPN Module With The Swin-Transformer-Based Rotatable Mask R-CNN	49
3.2.3 Modified Feature Pooling	51
3.2.4 Modified BRegNet.....	52
 CHAPTER 4: IMPLEMENTATION OF THE PROPOSED METHOD AND EXPERIMENTAL STUDIES	54
4.1 Experimental Settings	54
4.1.1 Dataset.....	55
4.1.2 Implementation Details	58
4.1.3 Evaluation Methods	59
4.2 Experiments and Discussions of The Upgraded Mask R-CNN model.....	61
4.2.1 Selection of The Baseline Building Segmentation Model	62
4.2.2 Selection of The Backbone Network	65
4.2.3 Ablation Studies of The Upgraded Mask R-CNN Model.....	68
4.3 Experiments and Discussions of The Proposed Building Extraction Model.....	70
4.3.1 Ablation Studies of The Proposed Building Extraction Model	70
4.3.2 Comparison Study of The Proposed Building Extraction Model	72
4.3.3 Other Design Choices for The Proposed Building Extraction Model	75
4.4 Results and Discussions	80
 CHAPTER 5: CONCLUSION AND FUTURE WORK	84
5.1 Conclusions	84
5.2 Limitations and Future Work.....	86
 REFERENCES	88

List of Tables

Table 4.1. Properties of the SpaceNet 2 Building Detection Dataset.	55
Table 4.2. Performances of the chosen baseline image segmentation models.	63
Table 4.3. Performances of the chosen backbone networks on Mask R-CNN.	66
Table 4.4. Performances of Mask R-CNN models with different components.	68
Table 4.5. Performances of PolygonCNN models with different components.	71
Table 4.6. Performances of the chosen building extraction models.	72
Table 4.7. Performances of the two experimental model architectures and the final proposed model.	78
Table 4.8. Memory consumption and inference time of the two experimental model architectures and the final proposed model.	79

List of Figures

Figure 1.1. Regularizations of shapes with non-right-angled corners using a conventional method.	5
Figure 3.1. The architecture of the PolygonCNN model.	18
Figure 3.2. The architecture of PSPNet.	19
Figure 3.3. Vertices simplification and densification.	20
Figure 3.4. The feature pooling operation.	21
Figure 3.5. The architecture of the BRegNet.	22
Figure 3.6. The architecture of the Mask R-CNN model.	23
Figure 3.7. An example of the traditional bounding box.	24
Figure 3.8. An example of anchor boxes with three scales and three ratios.	25
Figure 3.9. The IoU between two bounding boxes.	26
Figure 3.10. The processes of RoI Align.	27
Figure 3.11. A feature pyramid that is built upon an image pyramid for multi-scale object detection.	30
Figure 3.12. Multi-scale feature maps generated from the forward pass of a feature extractor for multi-scale predictions.	30
Figure 3.13. The Architecture of the FPN module.	31
Figure 3.14. The architecture of Swin Transformer.	33
Figure 3.15. An example of the patch merging process.	34
Figure 3.16. The architecture of the Swin Transformer Block.	35
Figure 3.17. A comparison between the standard MSA and the window-based MSA.	36
Figure 3.18. The window shifting process of the SW-MSA.	37

Figure 3.19. Architecture of the proposed model.	38
Figure 3.20. Examples of the proposed representation for rotatable bounding box.	41
Figure 3.21. An example of rotatable anchor boxes with one scale, three ratios, and two rotations.	42
Figure 3.22. Examples of the possible shapes of the positive overlapping region between two rotatable bounding boxes.	44
Figure 3.23. Example of cropping the two types of bounding boxes with partially excluded regions.	45
Figure 3.24. Symmetric padding of the RoI Align for the rotatable bounding box.....	46
Figure 3.25. An example of pasting the mask prediction to an empty image.....	47
Figure 3.26. The proposed solution for pasting a rotated mask onto an empty image.	48
Figure 3.27. Integration of FPN with Swin-Transformer-based Rotatable Mask R-CNN.	49
Figure 3.28. Architecture of the modified Feature Pooling module.	51
Figure 3.29. The architecture of the modified BRegNet.	53
Figure 4.1. Examples of the images and the corresponding building polygon and mask labels of the SpaceNet 2 Building Detection Dataset.	57
Figure 4.2. Building segmentation results of the chosen baseline image segmentation models. .	64
Figure 4.3. Comparison of the building segmentation results of the Mask R-CNN models with incrementally added components.....	69
Figure 4.4. Building extraction results of the chosen building extraction models.	74
Figure 4.5. Architecture #1.	75
Figure 4.6. Architecture #2.	77
Figure 4.7. Building detection and extraction results from the proposed methods.	81

Figure 4.8. Examples of mispredicted buildings.	83
--	----

CHAPTER 1: INTRODUCTION

1.1 Background

The satellite sensor technologies in the recent decade have enabled massive amounts of data to be acquired daily and opened possibilities for many new land use and land cover (LULC) mapping applications. These applications often require digitization and interpretation of objects within the raw image data so that further analyses can be done with geographic information systems. Building, as one of the most frequently appearing objects in the urban scene, the extraction of its footprint from satellite images has played an essential role in supporting LULC applications such as urban planning, change detection, and disaster management. Traditionally, the building footprint extraction process requires human image interpreters, which is extremely labour-intensive and time-consuming. Thus, researchers focused on developing automated extraction methods. In the early years, these methods often consist of handcrafted features (e.g., spectral, spatial, textural) followed by traditional machine learning classification methods. However, manual feature extractions usually require experienced experts and high human intervention, which may not always be feasible. Moreover, since the extracted features are often designed to model specific building types, the generalization capabilities of these methods are highly limited. As a result, much valuable information in this enormous amount of new data is unavailable.

In recent years, deep learning techniques have shed light on this problem due to the fast-paced development of the hardware computational power and the availability of vast training data. Many image segmentation methods based on convolutional neural networks (CNN) and fully convolutional networks (FCN) have shown dominance over conventional methods in terms of automation, generalization capability, and segmentation quality. However, a problem with such

techniques is that the generated building segmentations tend to have irregular shapes that differ significantly from the straight-edged and right-angle-cornered real-world building footprints. As a result, the vector-format building contours directly traced from such irregular-shaped segmentations are undesired for most cartographic and engineering applications. Thus, a method is required to regularize the irregular-shaped building contours. The conventional building regularization methods often involve many manually defined rules and thresholds and much human post-editing, resulting in a lack of automation and generalization capability; on the other hand, the DNN-based regularization methods can potentially be used to automatically extract buildings in arbitrary shapes. Despite the advantage and popularity of deep learning technologies these days, most of the widely used building regularization methods continue to rely on conventional processes due to the difficulties in developing contour-regularization DNNs and the general superiority of the conventional methods over the DNN-based methods in terms of the simplicity and regularity of the resulted building vectors. Thus, there is still much work to be done on developing DNN-based regularization methods. Moreover, since the building extraction result is highly dependent on the qualities of the building segmentations, developing methods to obtain more accurate building segmentation is the key to extracting higher-quality building polygons.

1.2 Applications of Automated Building Footprint Extraction

In the recent decade, the unprecedentedly fast-paced developments of satellite sensor technologies and urban constructions around the world have led to a massive demand for high-quality vector data on building footprints in various industrial applications.

Building footprint data is largely used in risk assessment tasks. For example, natural hazards such as pluvial floods (Löwe and Arnbjerg-Nielsen, 2020), earthquakes (Sahar et al., 2010), and landslides (Hasan et al., 2018) in urban regions cause enormous damage to people's assets and

lives every year. While many of these events are unavoidable, a comprehensive risk assessment is crucial for minimizing the risks and damages. In addition, urban planning is a significant sector involving building footprint data. Examples of its applications include: the development and planning of city infrastructures (Lee et al., 2008) such as public works infrastructures (e.g., water supply, sewage, electricity, and telecommunications), community infrastructures (e.g., schools, hospitals, and parks), and safety and transportation infrastructures (e.g., roads, police, and fire facilities); the planning of different types of land-use of the city (e.g., residential, commercial, industrial, municipal) (Dai et al., 2001); conducting real estate evaluations remotely with the help of building footprint data and other related situational datasets (Kodors et al., 2017); applications related to geo-marketing such as market planning and development, site selection and forecasting, supply channels analysis and modernization, etc. (Cliquet and Baray, 2020); telecommunications companies utilize building footprint data to select placements of signal towers to achieve optimal signal transmissions and maximum amount of coverage with a minimum number of towers. Moreover, building footprint data are used extensively to create navigation maps and produce highly accurate street annotations and road geometry (Royan et al., 2003).

1.3 Major Challenges for Automated Building Footprint Extraction

While significant efforts have been made in remote sensing literature to pursue a solution to automatically extracting building polygons from remote sensing images, some challenging problems remain in the subject. The major obstacles lie as follows.

1.3.1 An Issue With The Learning-Based Building Segmentation Methods

Most current image segmentation DNNs tend to produce irregular-shaped building segmentations, especially for rotated buildings. Experiments (i.e., Section 5.2.1) show that the segmentation predictions of some widely used DNNs such as PSPNet, UPerNet, DeepLabV3+, FastFCN, HRNet,

and Mask R-CNN do not look like real-life building footprints with straight edges and right-angled corners; instead, mispredicted pixels are frequently presented along the building edges. Specifically, for the encoder-decoder networks, mispredicted pixels tend to occur on all building types. In contrast, for the two-stage detector Mask R-CNN where segmentations are made within the 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), whereas accurate segmentations tend to occur on non-rotated buildings (i.e., building edges are either vertical or parallel to the image edges). Such results show that the bounding-box-based method may provide regularization effects to those non-rotated buildings, resulting in higher-quality segmentation results. However, despite the help of the bounding boxes, a significant fraction of buildings in satellite images have randomly rotated orientations, which leaves sizeable empty background spaces between the buildings and their bounding boxes, causing mispredicted pixels to occur.

1.3.2 Issues With The Conventional Building Regularization Methods

To regularize the building segmentations into regular-shaped polygons, most of the currently popular methods (Shu, 2014; Zhao et al., 2018; Zhang et al., 2018; Wei et al., 2019) still involve a large number of manually specified parameters and rules, which rely heavily on high human interventions and are strictly designed for buildings with straight edges and 90-degree corners; therefore, they often fail to generalize to buildings with more complexed shapes such as triangles or polygons with more than four edges. For example, figure 1.1 shows a triangular-shaped and a pentagonal-shaped building polygon and their corresponding regularized polygons using a conventional regularization algorithm described in Shu (2014). In these cases, the regularized polygons fail to restore the original building shapes due to the non-right-angled corners.

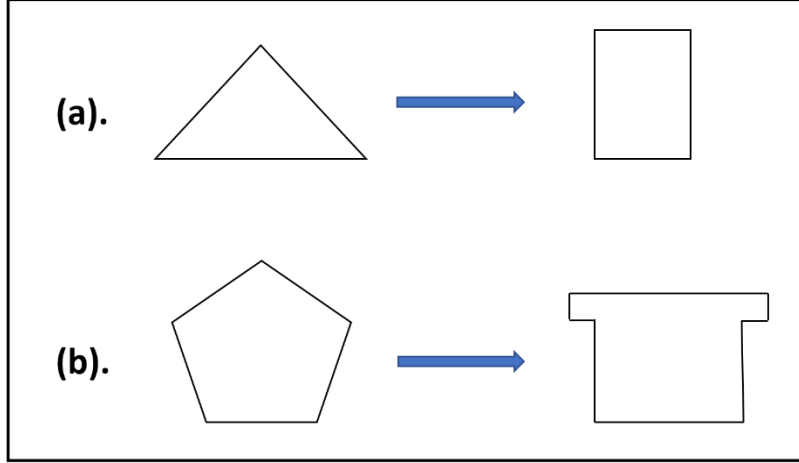


Figure 1.1. Regularizations of shapes with non-right-angled corners using a conventional method. (a) Regularization of a triangular polygon; (b) regularization of a pentagonal polygon.

1.3.3 Issues With Deep-Learning-Based Building Regularization Methods

Due to the lack of generalization capability and automation of conventional methods, developing learning-based regularization methods have become a new trend in recent years. Learning-based methods are theoretically highly generalizable to various types of shapes (Li et al., 2019). However, designing learning-based regularization methods is challenging, and only a few methods have been proposed. The difficulty mainly lies in the random number of output vertices since most CNN architectures accept a fixed size of input and a fixed size of output.

Although FCNs (Long et al., 2015) can accept inputs of arbitrary sizes, the output sizes are always proportional to the ones of the inputs. Nevertheless, for a building regularization task, the input and output are sets of building vertices consisting of the x and y coordinates. The input size largely varies depending on the number of vertices traced from the building segmentation prediction; the output size depends on the number of vertices in the regularized polygon, which is often disproportional to the input size. These random and disproportional sized inputs and outputs make the problem unsolvable by directly using the traditional CNN-based architectures. In addition,

despite the excellent generalization capabilities and high level of automation, the current learning-based regularization methods tend to show significantly worse performances than the conventional methods in terms of the general building extraction qualities.

1.4 Thesis Contributions

An end-to-end deep-learning-based building extraction method based on PolygonCNN (Chen et al., 2020) is introduced in this thesis. The proposed model mainly comprises a segmentation module and a regularization module. The segmentation module consists of a Swin Transformer-based rotatable Mask R-CNN that is integrated with the Feature Pyramid Network (FPN) (Lin et al., 2017), which is designed to alleviate the irregular-shaped building segmentation issue of the current segmentation DNNs; the regularization module consists of a modified BRegNet in combination with a modified feature pooling module, which is aimed to boost the regularization quality of the current deep-learning-based building regularization methods. The major contributions of the thesis lie as follows:

- (1) A rotatable bounding box technique is adopted on the Mask R-CNN model to specifically target the irregular-shaped building segmentation problem of the rotated buildings.
- (2) The FPN module and the recent popular Swin Transformer (Liu et al., 2021) backbone network are integrated with the rotatable Mask R-CNN to improve the model's ability in segmenting buildings in vastly different sizes, especially the small ones.
- (3) The Feature Pooling module and the BRegNet of the original PolygonCNN are modified to exploit the multi-scale, pyramidal hierarchy feature maps generated from the FPN module of the improved Mask R-CNN.

(4) Additional architecture design choices are provided for the modified Feature Pooling module and BRegNet, which offer flexibilities between the quality of the building extraction result and the memory consumption.

CHAPTER 2: RELATED WORK

Throughout the past decades, significant efforts have been made in remote sensing literature to pursue a solution to automatically extracting building polygons from remote sensing images. The existing methods can generally be categorized into two-stage and end-to-end methods. The two-stage methods often consist of a building segmentation step and a boundary regularization step. On the other hand, the end-to-end methods aim to take aerial images as input and directly output the building vectors with one deep neural network. Therefore, this section will review the related works in the following three sections: building segmentation in Section 2.1, building boundary regularization in Section 2.2, and end-to-end building polygon prediction in Section 2.3. Moreover, some past attempts at utilizing rotatable bounding boxes in deep learning models are also reviewed in Section 2.4.

2.1 Building Segmentation

Most image segmentation algorithms proposed over the past decades can be roughly classified into four categories: threshold-based, edge-based, region-based, and classification-based methods. Segmenting objects from images using threshold values is a simple and commonly-used method where pixels with different ranges of values are classified into different classes according to manually or automatically selected thresholds (Glasbey, 1993). However, the most significant drawback of image thresholding is that it cannot differentiate among different regions with similar grayscale values. Edge-based methods adopt edge-detection filters such as Laplacian of Gaussian (Chen et al., 1987), Sobel (Kanopoulos et al., 1988) and Canny (Canny, 1988), to detect the sharp transitions among neighbouring pixels, thus, to generate boundaries for segmentation. Region-based methods segment different parts of an image through clustering (Wu and Leahy, 1993; Chuang et al., 2006; Zhen et al., 1997; Pappas, 1992), region-growing (Tremeau and Borel, 1997),

shape analysis (Ok, 2013; Karantzas and Paragios, 2008), or graph-based methods (Felzenszwalb and Huttenlocher, 2004). 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 methods treat image segmentation as a process of classifying the category of every pixel (Li et al., 2014). Compared with the other three categories of methods, it tends to produce more precise segmentations when proper feature extractors and classifiers are utilized.

Since the studies on building segmentation have transitioned significantly from the early days' conventional methods into the nowadays' popular deep learning-based methods, they are reviewed chronologically in the following sections 2.1.1 and 2.1.2.

2.1.1 Conventional Building Segmentation

The conventional methods usually involve a two-step procedure of feature extraction and classification. First, features based on the spatial (i.e., key points, corner points, edges), textual, or spectral characteristics of the image are extracted through mathematical feature descriptors such as haar-like (Viola and Jones, 2001), scale-invariant feature transform (Lowe, 1999), local binary pattern (Ojala et al., 2002), and histogram of oriented gradients (HOG) (Dalal and Triggs, 2005). Next, the segmentation prediction of the building footprint is made based on the extracted features through methods such as template matching (Sirmacek and Unsalan, 2009), graph cut (Manno-Kovacs and Ok, 2015), adaptive boosting (AdaBoost) (Aytekin et al., 2012), random forests (Pelizari et al., 2018), support vector machines (Turker and Koc-San, 2015) or conditional random fields (CRF) (Li et al., 2015).

For instance, Wei et al. (2004) proposed an unsupervised clustering by histogram peak selection to classify the image into some classes, extracting the building shadows which are used to verify the presence of buildings, then extracting the candidate building objects from the clustering classes; Belgiu and Drăguț (2014) proposed and compared supervised and unsupervised multi-resolution segmentation methods that are integrated with the random forest classifier for building segmentation from high-resolution satellite images; Huang and Zhang (2012) proposed the morphological shadow index (MSI) to detect shadows which are used as a spatial constraint of buildings. Moreover, they proposed a dual-threshold filtering method to integrate the information from the morphological building index with the one from MSI; Ok et al. (2013) proposed a fuzzy landscape generation method that models the directional spatial relationship of the building and its shadow for automatic building segmentation; Qin and Fang (2014) proposed a hierarchical building detection method for very high resolution remotely sensed images using graph cut optimization. Although many significant achievements have been made, these studies were based on traditional methods, which are often focused on relatively small study regions. As a result, they tend to lack performance when evaluated in complex regions with a high diversity of buildings. In addition, the hand-crafted features in many of them rely heavily on the intervention of experienced experts, which causes a lack of automation and may not always be practical or feasible.

2.1.2 Deep Learning-Based Building Segmentation

In recent years, deep-learning-based building segmentation methods have become widely popular mainly due to the breakthrough of convolutional neural networks (CNN) in the ImageNet classification contest in 2012 (Krizhevsky et al., 2012). The CNN-based method can be considered a one-step method that combines feature extraction and classification within a single model. Moreover, they overcome the explicit feature design problem of the conventional methods by

allowing adaptive feature learning from labelled training data; therefore, they often show significantly improved generalization capabilities.

Alshehhi et al. (2017) propose a single patch-based CNN architecture for the segmentation of roads and buildings from high-resolution remote sensing data, and the low-level features of roads and buildings (e.g., asymmetry and compactness) of adjacent regions are integrated with CNN features during the post-processing stage to improve the performance. Guo et al. (2017) also proposed a patch-based CNN to identify village buildings based on multiscale feature learning by assembling the feature extractor parts of a few state-of-the-art CNN models into a robust model. Despite the promising results, these methods are limited to overly redundant computations due to the heavy overlaps between patches. For example, when a patch-based method processes an image in patches of 32×32 pixels, its memory cost is increased by 32×32 times compared to the conventional methods. For larger areas or patch sizes, such approaches encounter dramatically increased memory costs and significantly reduced processing efficiency.

To overcome the problem, FCNs have been proposed to achieve efficient pixels-to-pixels classifications by replacing the fully connected layer with up-sampling operations. For instance, Long et al. (2015) first proposed an FCN, which was trained end-to-end, pixels-to-pixels, and had exceeded the state-of-the-art in semantic segmentation; other similar convolutional encoder-decoder models include the SegNet (Badrinarayanan et al., 2015) and DeconvNet (Noh et al., 2015); however, they tend to produce segmentations with low edge accuracies due to the loss of information in the lower layers. To overcome the limitation, Ronneberger et al. (2015) proposed the U-Net, which utilizes the skip connection technique to combine the lower and higher layer features to generate the final output.

A significant drawback of such semantic segmentation methods is that their prediction is instance-indistinguishable (e.g., when several objects of the same class are clustered together, the number and the boundary of the objects cannot be retrieved). Therefore, models that can produce instance-level segmentation have been proposed. For example, Deepmask (Pedro et al., 2015) learns first to propose segmentation candidates, which are later classified with an object detection algorithm. Iglovikov et al. (2018) proposed a method for instance segmentation with U-Net. An extra output channel is added to predict borders between closely positioned or touching objects. In these cases, segmentation is performed before detection. In addition, He et al. (2017) proposed an instance segmentation model named Mask R-CNN, which achieved state-of-the-art when published. The model extends the Faster R-CNN network (Ren et al., 2015) with an additional binary mask prediction branch, allowing segmentations to be performed within each bounding box

2.2 Building Boundary Regularization

Despite the great segmentation performances of the CCNs, slightly mispredicted pixels often exist along the predicted building boundaries, resulting in irregular shapes of the directly traced building polygons. Therefore, many studies focused on regularizing the building segmentations into accurate, simple, and regular polygons.

2.2.1 Conventional Building Regularization

Typically, building regularization requires additional data sources, such as airborne LiDAR scanning or public GIS data, to assist precise regularizations (Boehm, 2019; Li et al., 2019). When only image data is available, pre-specified constraints such as 90-degree corners and principal orientations are usually applied to regularize and simplify the building boundaries. For instance, Shu (2014) proposed a building regularization method involving cropping the original image using the segmentation prediction; detecting line segments in the cropped building image; computing

the two perpendicular principal directions and rotating the line segments to the principal directions; merging and simplifying the line segments such that the final polygon has 90-degree corners. Maggiori et al. (2017) proposed a method which formulates the polygonization problem into a mesh-based approximation of the input binary segmentation image. Zhao et al. (2018) proposed a hypothesis generation method to generate a set of points along the irregular polygon, then optimize the candidate points with the Minimum Description Length optimization to produce the right-angle-cornered polygon.

Despite the significant regularization effects, such conventional methods usually rely heavily on high human interventions and are strictly designed for buildings with right-angled cornered; therefore, they often fail to generalize to buildings with more complex shapes, such as triangles or polygons with more than four edges.

2.2.2 Deep Learning-Based Building Regularization

Due to the low generalization capabilities of the conventional methods, seeking deep-learning-based building regularization methods has become a new trend.

Zorzi and Fraundorfer (2019) proposed a method for building boundary regularization in satellite images using a fully convolutional neural network trained with a combination of adversarial and regularized losses. Zhao et al. (2020) proposed a building boundary extraction network using graph models to learn geometric information about polygon shapes. Zorzi et al. (2021) proposed a generative adversarial network to perform building boundary regularization. The network consists of a generator which tries to generate a regularized version of the segmentation mask, and a discriminator that examines the generated footprints. Besides regularizing the irregular polygons, Jung et al. (2021) proposed a method to predict regularized segmentations directly. A boundary

enhancement module was proposed, which adopts the Holistically-Nested Edge Detection to extract edge features at an encoder of a given architecture. The extracted edge is combined with the segmentation mask to share mutual information, allowing direct prediction of regularized building segmentations.

Although the deep learning-based regularization methods have successfully improved the generalization capability and level of automation compared to the conventional building regularization methods, the regularized building vectors often show lacked simplicity and regularity compared to the conventional methods.

2.3 end-to-end building polygon prediction

Typically, the building segmentation results are vectorized by employing a conventional or deep learning-based regularization method as a post-processing step. However, some recent approaches attempted to integrate the polygon regularization procedure into an end-to-end deep learning-based model, which aims to take remotely sensed images as input and directly output the regularized building polygons.

To the best of my knowledge, the earliest attempt at such an end-to-end deep learning-based model was made by Girard and Tarabalka (2018). The authors proposed a CNN to directly predict vertices of building polygons. However, the architecture suffers significantly from the fixed prediction size (i.e., the method can only predict 4-sided polygons). Cheng et al. (2019) proposed an end-to-end deep neural network named deep active ray network (DARNet) for building polygon extraction. They use a backbone CNN to predict energy maps, which are utilized to construct an energy function. Then, the building polygons are derived by minimizing the energy function. Despite the end-to-end structure, the network fails to take into consideration the simplicity and regularity of

building polygons. Castrejon et al. (2017) developed a deep neural network named PolygonRNN to predict the object contour points sequentially. The network consists of a CNN feature extractor and a recurrent neural network (RNN) which decodes one polygon vertex at a time. Despite the end-to-end structure, the network is limited by its high memory requirement. Motivated by PolygonRNN, Li et al. (2019) developed PolyMapper, which focuses on delineating the road and building vector boundaries from a given image. Manual bounding box labels are no longer required due to the integration of the Feature Pyramid Network (FPN) module on top of PolygonRNN. However, PolyMapper lacks the ability to predict objects with complex shapes; moreover, its convolutional Long Short-Term Memory module is computationally expensive. To overcome the expensive memory issue of these RNN-based networks, Chen et al. (2020) proposed an end-to-end network based on a segmentation CNN and a boundary regularization CNN. The semantic segmentation network PSPNet (Zhao et al., 2017) is used to generate the initial building contour, while a modified PointNet predicts the coordinate offsets of the polygon vertices to generate the regularized buildings.

2.4 Integrating Rotatable Bounding Boxes with Deep Learning Models

The idea of utilizing rotatable bounding boxes in deep learning models to detect rotated remote sensing targets have emerged in recent years. Liu et al. (2017) proposed a DRBox algorithm which combines the rotatable bounding box with deep learning methods to detect airplane, vehicle, and ship targets in remote sensing images. Although the method could output the precise location and orientation information of the target, the approximate Intersection over Union (IoU) calculation in between the rotated bounding boxes makes the performance unstable. Xia et al. (2018) proposed an improved Faster R-CNN algorithm based on orientated bounding boxes. They represented the oriented bounding boxes using eight parameters (i.e., the coordinates of the four corner points:

$x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$). The algorithm can effectively detect the oriented targets; however, the predicted bounding box sometimes becomes deformed since there is no constraint to maintain the rectangular shape of the box. Li et al. (2018) proposed a rotated bounding box-based deep learning method using five parameters (i.e., x, y, w, h, θ . x and y are the center point coordinate; h and w are the long side and short side of the box; θ is the angle between the long side and the horizontal right direction and is in the range of $[0, 180)$ degrees) for detecting ship targets in remote sensing images and using a range control strategy to reduce the amount of calculation when calculating the matching between the prior rotated boxes and ground-truth boxes. However, the range control strategy by manually thresholding can cause missing/false matches, which results in target missing and false alarm problems.

CHAPTER 3: THE PROPOSED METHOD

To automatically extract high-quality building polygons in various scales and orientations from satellite images, an end-to-end deep-learning-based building extraction method based on PolygonCNN is proposed and presented in this chapter. The proposed model consists of a segmentation module to predict building segmentations, and a regularization module to regularize the building contours traced from the building segmentation results. First, an upgraded Mask R-CNN model, which is integrated with the rotatable bounding box technique, the Swin Transformer backbone network, and the FPN module, is adopted as the segmentation module of the proposed model. Next, for the regularization module, the Feature Pooling module and the BRegNet of the original PolygonCNN are modified to exploit the multi-scale, pyramidal hierarchy feature maps generated from the FPN module of the improved Mask R-CNN.

This chapter first reviews the PolygonCNN model, the Mask R-CNN model, the FPN, and the Swin Transformer backbone in section 3.1. Next, the architecture of the proposed model is described in section 3.2.

3.1 Model Reviews

3.1.1 Review of PolygonCNN

Most of the current building regularization methods involve conventional processes which rely heavily on human interventions and are non-generalizable for buildings with non-right-angle-cornered shapes such as triangles or polygons with more than four edges. In recent years, deep-learning-based regularization methods have been proposed to improve the automation and generalization capability of the conventional regularization methods. Among the deep-learning-based methods, PolygonCNN has shown state-of-the-art performance in extracting regularized

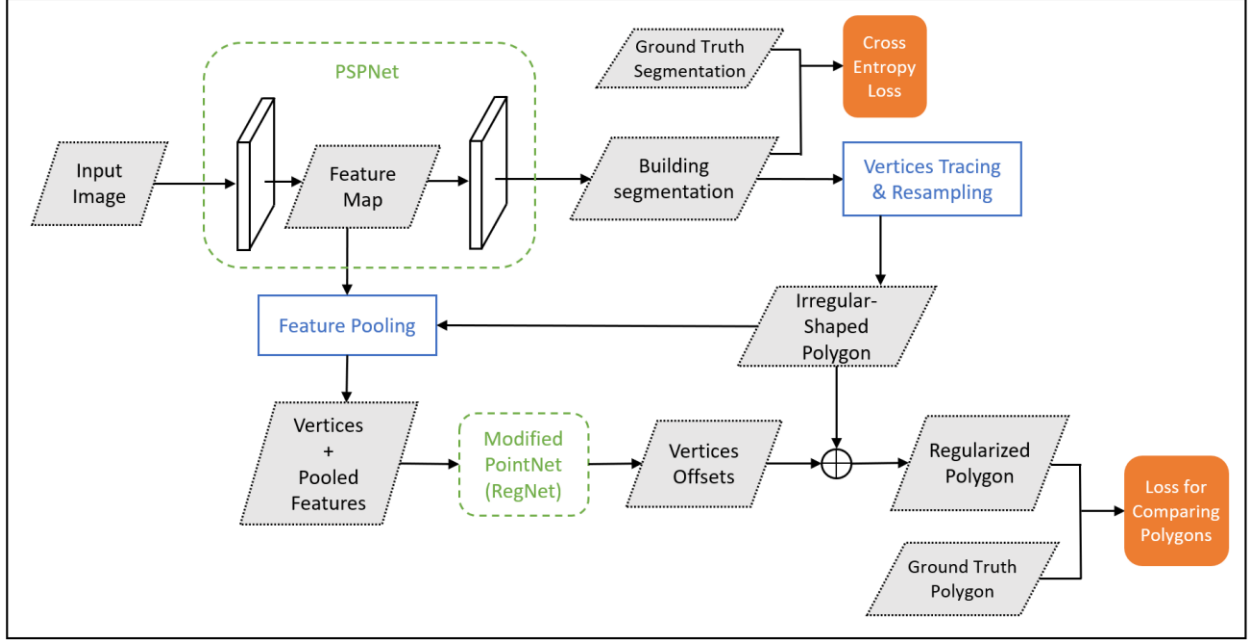


Figure 3.1. The architecture of the PolygonCNN model.

building polygons from aerial images. PolygonCNN is composed of a pyramid scene parsing network (PSPNet) (Zhao et al., 2017) for predicting building segmentations and a modified PointNet (BRegNet) to regularize the irregular-shaped segmentations. The two networks are combined into one to enable end-to-end training and inference. As shown in Figure 3.1, the major components of PolygonCNN are as follows: the PSPNet for predicting building segmentations; the vertices tracing and resampling components; the feature pooling module for extracting the feature vector (i.e., from the feature map of PSPNet) corresponding to every resampled vertex; the BRegNet for predicting the offsets of the regularized polygon vertices. The detailed components of PolygonCNN are reviewed as follows.

3.1.1.1 PSPNet

As shown in Figure 3.1, the input image of PolygonCNN is first passed through the PSPNet. PSPNet is a widely-used FCN-like model for semantic image segmentation, and has proven to be

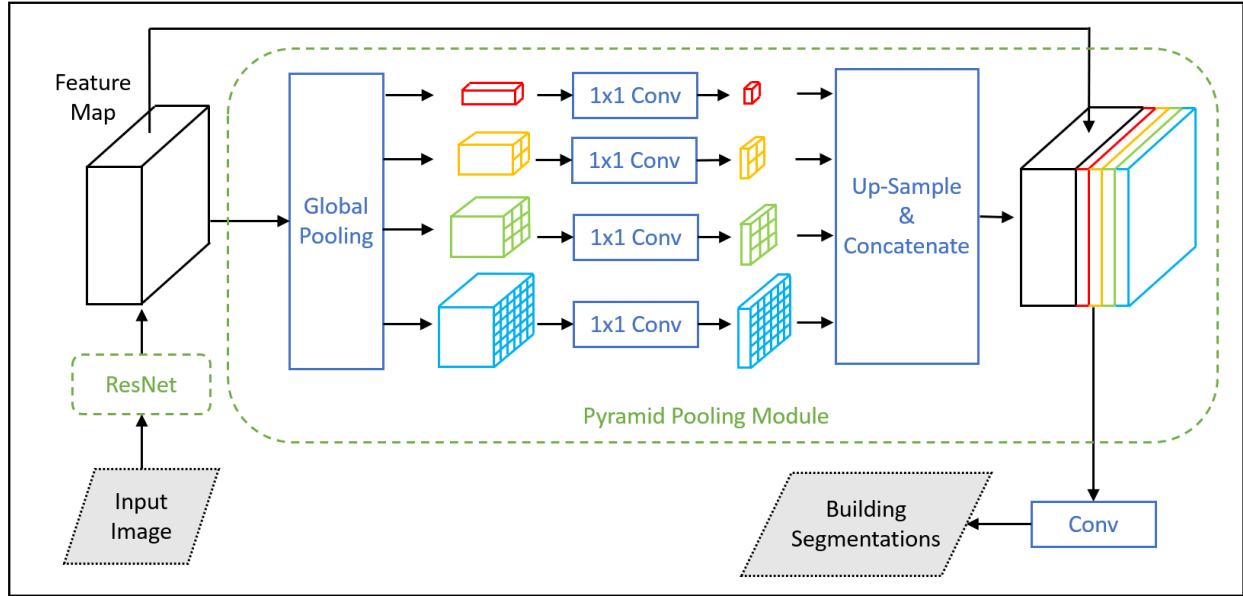


Figure 3.2. The architecture of PSPNet.

effective in building segmentation from aerial images (Chen et al., 2019). The architecture of PSPNet is illustrated in Figure 3.2. First, an input image is passed through a ResNet (He et al., 2016) backbone to generate the feature map. Next, a pyramid pooling module is applied to the feature map to produce an improved feature map which contains multi-scale global information. Lastly, the improved feature map is fed to a convolution layer to obtain the final per-pixel prediction map.

The key highlight of PSPNet is the pyramid pooling module, which provides an effective global contextual feature map for pixel-level scene parsing. It utilizes global pooling to generate four levels of feature maps with varied sizes, each containing the global information. Next, the dimension of each level of the feature map is reduced with a convolution layer. Then, all levels of feature maps are up-sampled to match the size of the original feature map. Finally, the up-sampled feature maps and the original feature map are concatenated to obtain the enhanced feature map.

3.1.1.2 Vertices Tracing and Resampling

The probability map generated by PSPNet is first binarized. Then, irregular-shaped building polygons can be extracted from the binary map by directly tracing the contours (Suzuki and Be, 1985) with closed areas. The traced contour is essentially a sequence of dense coordinates of the building outline pixels, as shown in Figure 3.3.(a), which contains a large number of redundant vertices and is inefficient for further processes.

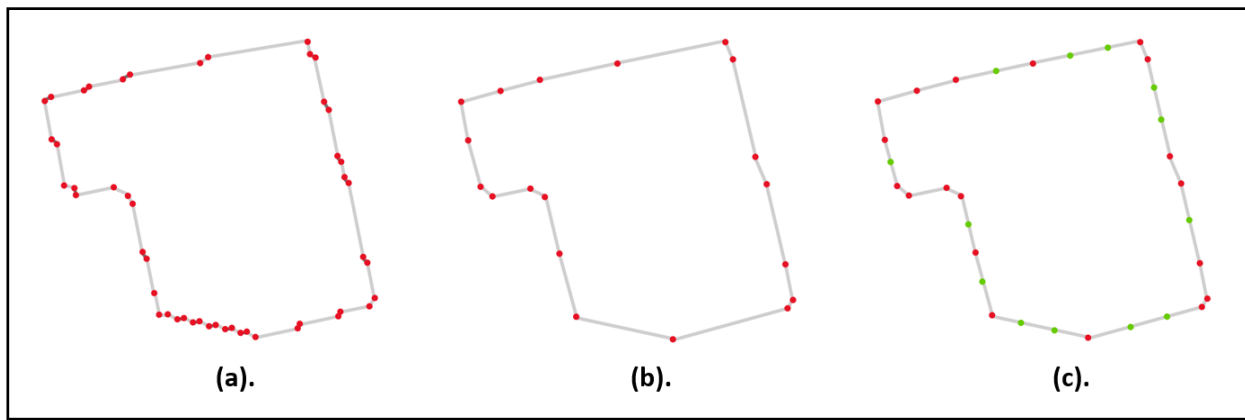


Figure 3.3. Vertices simplification and densification. (a) The directly traced contour; (b) the simplified contour using the DP algorithm; (c) the densified contour.

Therefore, the Douglas–Peucker (DP) algorithm (Douglas and Peucker, 1973) is adopted to remove the intermediate points on straight line segments and the points that lead to slight changes in directions, as shown in Figure 3.3.(b). However, despite the significant reduction of the redundant vertices and the well-preserved major geometric signals of the contour, the distances between every two consecutive vertices could broadly vary due to structural changes or slight segmentation errors. In this way, when a fixed-size convolution operation from the BRegNet in the later stage (i.e., Figure 3.1) is applied to such a polygon, the receptive fields of the convolutions at different locations may also broadly vary, which could lead to biases for those learned features.

Considering the problem, a densify operation is applied to the simplified polygon, as shown in Figure 3.3.(c), where additional points are sampled evenly along the edges with lengths larger than a predefined threshold. This way, a sliding convolution kernel can have similar receptive fields at different polygon locations.

3.1.1.3 Feature Pooling

As illustrated in Figure 3.1, the output polygon from the Vertices Tracing and Resampling module is fed to the Feature Pooling module along with the feature map generated from the backbone of PSPNet to extract a feature vector for each polygon vertex. The extracted feature vectors are then concatenated with the coordinates of the vertices to form the final output.

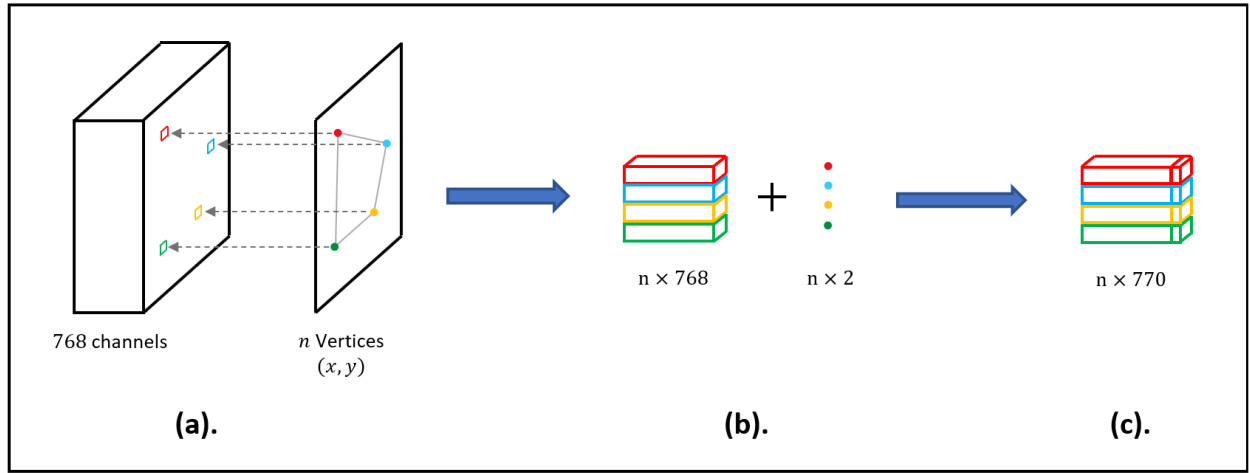


Figure 3.4. The feature pooling operation.

For example, as illustrated in Figure 3.4.(a), assuming a feature map with 768 channels and a vector map with n vertices of (x, y) coordinates are fed to the Feature Pooling module. For each vertex, at its corresponding location on the feature map, a feature vector is extracted. Therefore, as shown in Figure 3.4.(b), for n vertices, the pooled feature vectors have a size of $n \times 768$, and the vertices have a size of $n \times 2$. The pooled features are then concatenated with the coordinates

of the vertices to obtain the final output, which has a size of $n \times (768 + 2) = n \times 770$, as shown in Figure 3.4.(c).

3.1.1.4 BRegNet

As illustrated in Figure 3.1, the output of the Feature Pooling module, which consists of the coordinates of the irregular polygon vertices and their corresponding feature vectors, is passed through the BRegNet to obtain a set of coordinate offsets for the regularized polygon vertices.

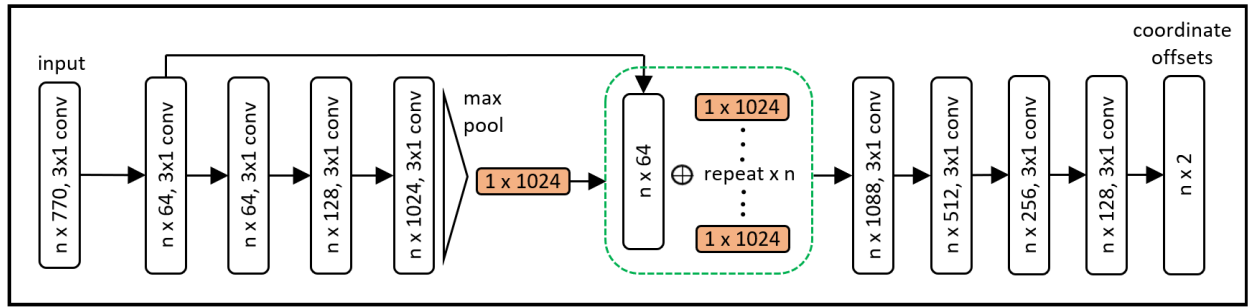


Figure 3.5. The architecture of the BRegNet.

As shown in Figure 3.5, the architecture of the BRegNet is similar to the one of PointNet (Qi et al., 2017) with two major differences. First, PointNet uses a 1×1 convolution layer to encode the coordinates of every point because the points in a point cloud are often unordered. However, since the vertices of an enclosed building polygon form an ordered sequence, the receptive fields of the convolutions can be expended by replacing the 1×1 convolution layers with 3×1 convolution layers, thus allowing local context between neighbouring vertices to be learned. Second, when applying a 3×1 convolution to a set of polygon vertices, the output data size is reduced. To avoid the loss of data size, recursive padding is adopted before every convolution operation, where the vertex at each end is padded to the other end such that the point sequence forms a closed loop. The recursive padding ensures that the features of every neighbouring point in the sequence contribute equally to the learning procedure, and the network is invariant of the starting point selection. The

output coordinate offsets of the BRegNet are concatenated with the vertices of the irregular shaped polygon, as shown in Figure 3.1, to obtain the final regularized polygon.

3.1.2 Review of Mask R-CNN

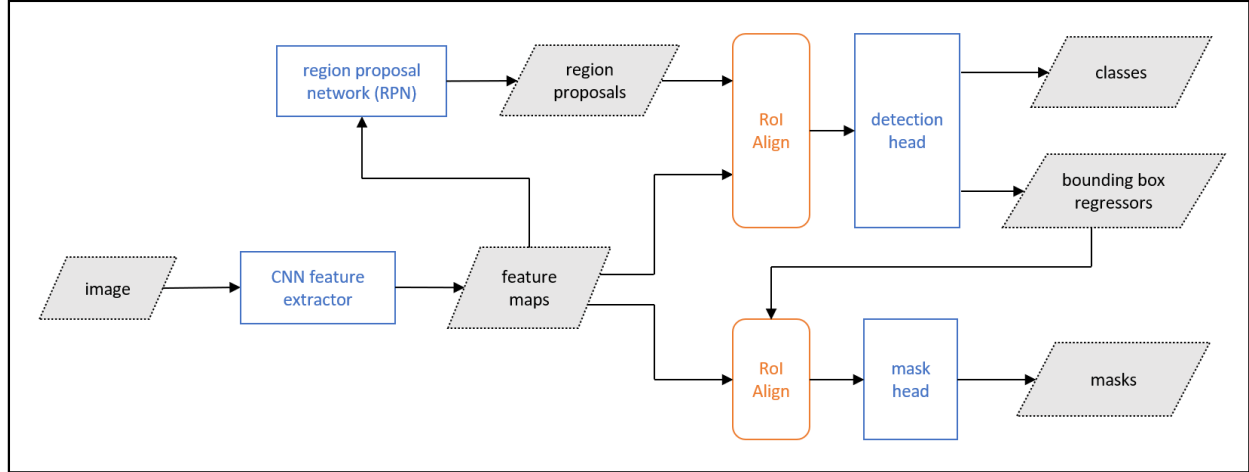


Figure 3.6. The architecture of the Mask R-CNN model.

Mask R-CNN is a state-of-the-art image segmentation CNN, proposed by Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick in 2017. It is developed on top of Faster R-CNN (Ren et al., 2015) by adding an additional mask branch to generate a high-quality segmentation mask for each instance. As shown in Figure 3.6, the major components of the Mask R-CNN network are as follows: a feature extraction CNN, a region proposal network, an RoI Align module, a detection prediction head, and a mask prediction head. Each of these components is reviewed as follows.

3.1.2.1 Feature Extraction CNN

Feature extraction CNNs are commonly used as backbones in larger networks to solve various computer vision tasks such as pattern recognition, vocal recognition, natural language processing, and video analysis (Koushik, 2016); their role is often to extract features from raw data for further analyses. For building extractions, the input satellite images are first fed into the feature extractor

to obtain feature maps that are used to analyze building locations and building segmentations. Examples of some widely used backbone networks are ResNet (He et al., 2016), DenseNet (Huang et al., 2017), VGGNet (Simonyan et al., 2014), HRNet (Wang et al., 2020), MobileNet (Howard et al., 2017), Swin Transformer (Liu et al., 2021). A comparison of the performances of the Mask R-CNN models with several popular backbone networks is shown in section 5.1.

3.1.2.2 Region Proposal Network

The feature map extracted by the backbone CNN is fed into a lightweight neural network named Region Proposal Network (RPN), which proposes regions on the feature map that may contain objects. To bind features to their raw image locations, a method named ‘anchor’ is used. Anchors are a set of boxes with predefined locations, ratios, and scales relative to the images. Both the anchors and the bounding box predictions are represented by the coordinates of their top left and bottom right corners (i.e., $x_{min}, y_{min}, x_{max}, y_{max}$). For example, a bounding box which has a top left corner coordinate of (2, 1) and a bottom right corner coordinate of (7, 6) is shown in Figure

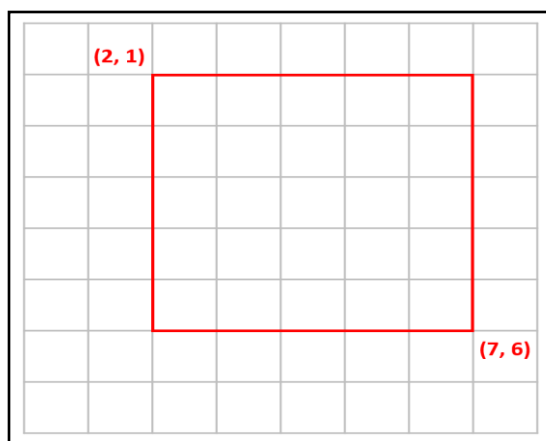


Figure 3.7. An example of the traditional bounding box.

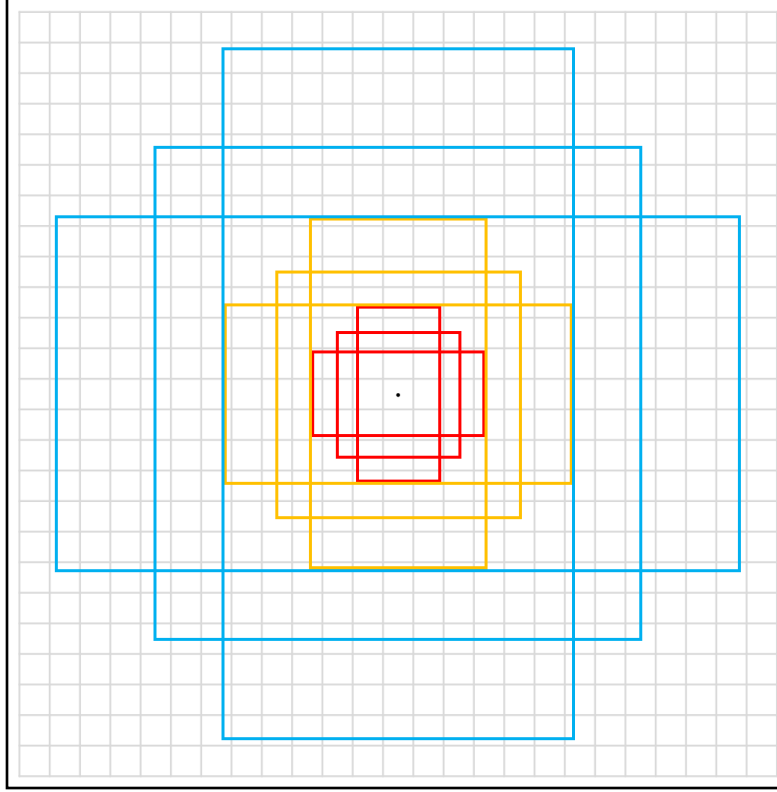


Figure 3.8. An example of anchor boxes with three scales and three ratios. The three scales are 4×4 , 8×8 , 16×16 , shown in red, yellow, and blue colors, respectively; the three ratios are 1:1, 1:2 and 2:1.

3.7. It can be represented by (2, 1, 7, 6). In addition, Figure 3.8 illustrates a set of anchor boxes with three scales (i.e., 4×4 , 8×8 , 16×16) and three ratios (i.e., 1:1, 1:2 and 2:1). The three scales are shown in three colors; for each scale, anchors of three ratios are generated. Therefore, nine anchors are generated at one location on the image. For an image with 625 anchor locations, a total of $625 \times 9 = 5625$ anchors will be generated. The ground-truth bounding boxes are assigned to individual anchors according to a predefined intersection over union (IoU) threshold. The IoU between two bounding boxes is calculated by:

$$IoU = \frac{Area\ of\ OverLap}{Area\ of\ Union}$$

(3.1)

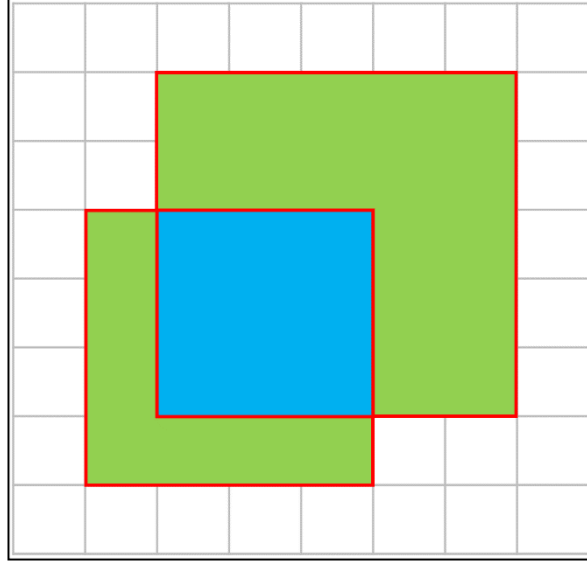


Figure 3.9. The IoU between two bounding boxes. The intersection between the two boxes is represented in blue color; and union between them is represented in both the green and blue colors.

For example, as shown in Figure 3.9, the area of overlap between the two boxes is represented in blue, and the union area between the two boxes is represented in green and blue. Therefore, the IoU between the two boxes of is calculated by $\frac{Area\ of\ Blue}{Area\ of\ Green + Area\ of\ Blue}$. Anchors with high overlaps with the ground-truth boxes are considered foreground and have an objectness score of 1; those with lower overlaps are considered background and have an objectness score of 0. For each anchor, the RPN predicts an objectness value (i.e., ranged from 0 to 1, where 0 represents non-object and 1 represents object) and four regression values (i.e., the offsets of the anchor box) to indicate the possibility of an anchor being background or foreground, and the refined location and shape of the anchor box.

3.1.2.3 RoI Align

The final bounding boxes predicted by the RPN are used to crop their corresponding regions of feature maps on the feature maps extracted by the backbone CNN. Those cropped feature map

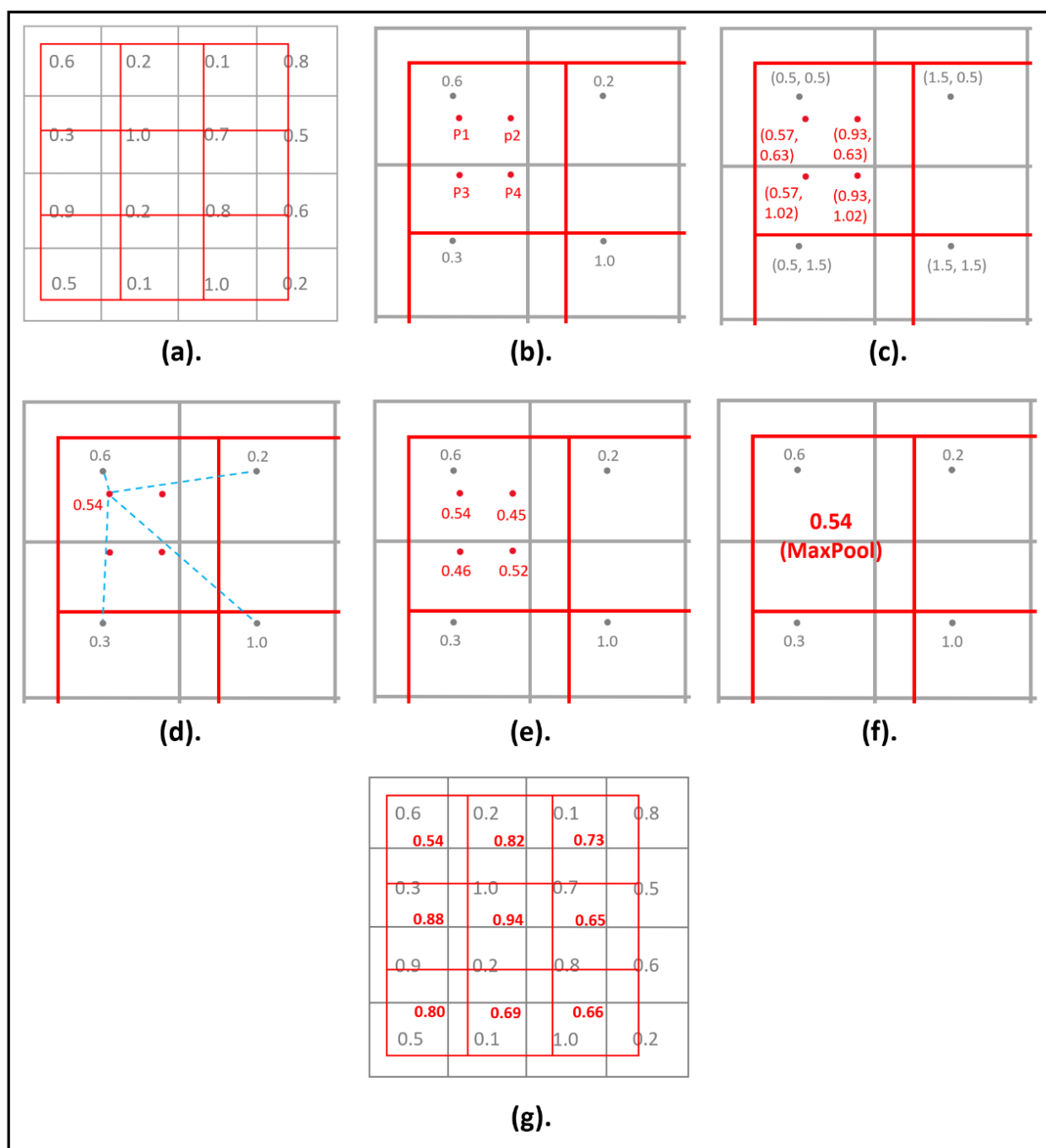


Figure 3.10. The processes of RoI Align.

regions are named Region of Interests (RoIs). The RoIs are passed through the detection head, which consists of a sequence of fully connected layers to obtain their corresponding object classes (i.e., dog, bird, cat, etc.) and further refined bounding boxes. However, these RoIs cannot be fed into the detection head because they vary in size, while the fully connected layers only accept uniform-sized inputs. Thus, instead of directly cropping the feature maps with the bounding boxes, a method is required to produce uniform-sized RoIs before feeding them to the detection head. This is done by RoI Align.

The RoI Align method is demonstrated in Figure 3.10. As shown in Figure 3.10.(a), assuming the gray-colour grid is a feature map extracted by the backbone CNN, and the feature map values are also labelled in gray; the red box is a bounding box predicted by the RPN, which is evenly divided into nine cells assuming all RoIs have a fixed shape of 3×3 . The goal of RoI Align is to interpolate the nine cells of the RoI with the feature map values while minimizing data loss. For illustration purposes, only the computation of one cell is demonstrated below. First, as shown in Figure 3.10.(b), four sampling points are evenly generated within a cell. Since the corner coordinates of the RoI are known, the coordinates of the four points could be easily obtained (i.e., shown in Figure 3.10.(c) in red, while the center coordinates of the feature map pixels are labelled in gray). To extract values for each of the four sampling points, RoI Align adopts bilinear interpolation (Wang and Yang, 2008), which has the following formula:

$$P = \frac{y_2 - y}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} Q_{11} - \frac{x - x_1}{x_2 - x_1} Q_{21} \right) + \frac{y - y_1}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} Q_{12} - \frac{x - x_1}{x_2 - x_1} Q_{22} \right) \quad (3.2)$$

Where for a sampling point p , x and y are the coordinate of p , and P is the interpolated value of p . For example, as shown in Figure 3.10.(d), the four feature map pixels with the closest center

distances to p_1 are used for data sampling. The top left pixel has a center coordinate of $(x_1, y_1) = (0.5, 0.5)$ and a value of $Q_{11} = 0.6$; the top right pixel has a center coordinate of $(x_2, y_1) = (1.5, 0.5)$ and a value of $Q_{21} = 0.2$; the bottom left pixel has a center coordinate of $(x_1, y_2) = (0.5, 1.5)$ and a value of $Q_{12} = 0.3$; the bottom right pixel has a center coordinate of $(x_2, y_2) = (1.5, 1.5)$ and a value of $Q_{22} = 1.0$. After substituting the variables for the formula 3.2, $p_1 = 0.54$ is obtained. The same calculation is repeated for each of the four sampling points, and the resulting values are shown in Figure 3.10.(e). Finally, a max pooling operation is applied to the four sampled values to obtain the final sampled value of the cell, as shown in Figure 3.10.(f). The exact process is repeated for all nine cells, and the final interpolated RoI is shown in Figure 3.10.(g).

3.1.2.4 Detection Head and Mask Head

With the adoption of RoI Align, all RoIs have uniform shapes and can be passed through the detection head, which consists of a sequence of fully connected layers following two sibling branches of fully connected layers. One branch is responsible for outputting object classes (e.g., dog, bird, cat), and the other is responsible for outputting further refined bounding boxes. Those further refined bounding boxes are used to generate higher quality RoIs, again, with the help of RoI Align. Those new RoIs are fed to the mask head, which is an FCN that is responsible for predicting a segmentation mask for each RoI in a pixel-to-pixel manner relative to the input image.

3.1.3 Review of The FPN module

To improve the Mask R-CNN's ability to segment buildings in vastly different sizes, especially the small ones, the FPN module is adopted. This section presents a review of the FPN module.

Detecting objects in vastly different scales has been challenging in computer vision. In the early years, the problem was tackled by utilizing engineered feature pyramids that were built upon image

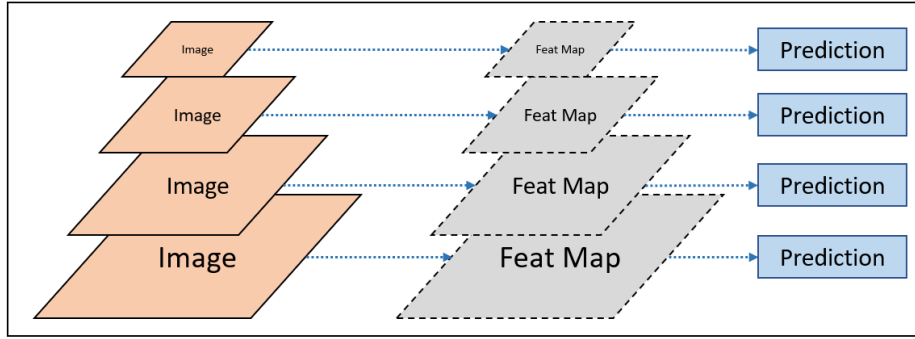


Figure 3.11. A feature pyramid that is built upon an image pyramid for multi-scale object detection.

pyramids (Adelson et al., 1984). The process is illustrated in Figure 3.11, where an image is enlarged/shrunk into multiple different scales. Then, multi-scale features that correspond to the multi-scale images are generated for multi-scale object detection. However, such methods require considerably increased processing time (i.e., by n times corresponding to the n scales of images); moreover, when replacing the hand-crafted features with CNN-generated features, the memory demand of training deep networks end-to-end on an image pyramid is too high (Ren et al., 2015).

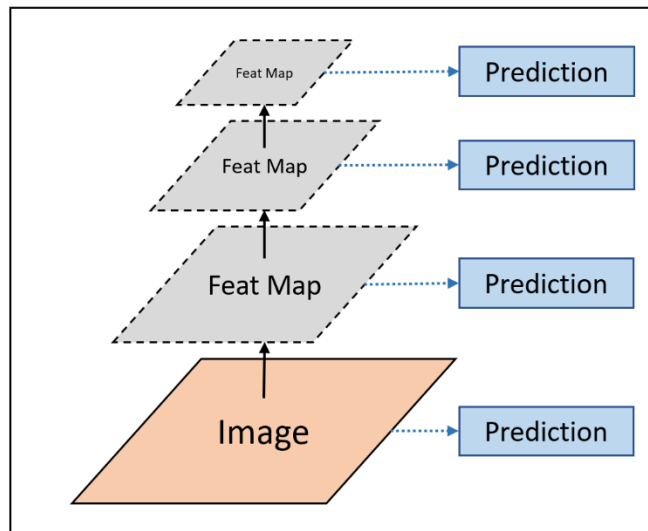


Figure 3.12. Multi-scale feature maps generated from the forward pass of a feature extractor for multi-scale predictions.

In the recent decade, feature extraction using CNNs has become mainstream. CNNs can represent high-level semantics and are robust to variance in scale, which facilitates recognition from features computed on a single input scale. As shown in Figure 3.12, for an input image, multi-scale feature maps are generated along the forward pass of the CNN; however, the feature layers closer to the image layer are composed of low-level structures, which are ineffective for accurate object detection. Despite the promising detection capability of the final semantically strong feature layer, more accurate results can be obtained using feature pyramids with strong semantics at all scales.

To achieve this goal, FPN is proposed. As shown in Figure 3.13, the architecture of FPN mainly consists of a bottom-up pathway and a corresponding top-down pathway. The bottom-up pathway is constructed using the multi-scale feature maps generated along the forward pass of a backbone CNN. All feature maps in the bottom-up pathway have different scales. Starting from the shallowest feature layer (i.e., the one closest to the image layer), each deeper layer has a scaling

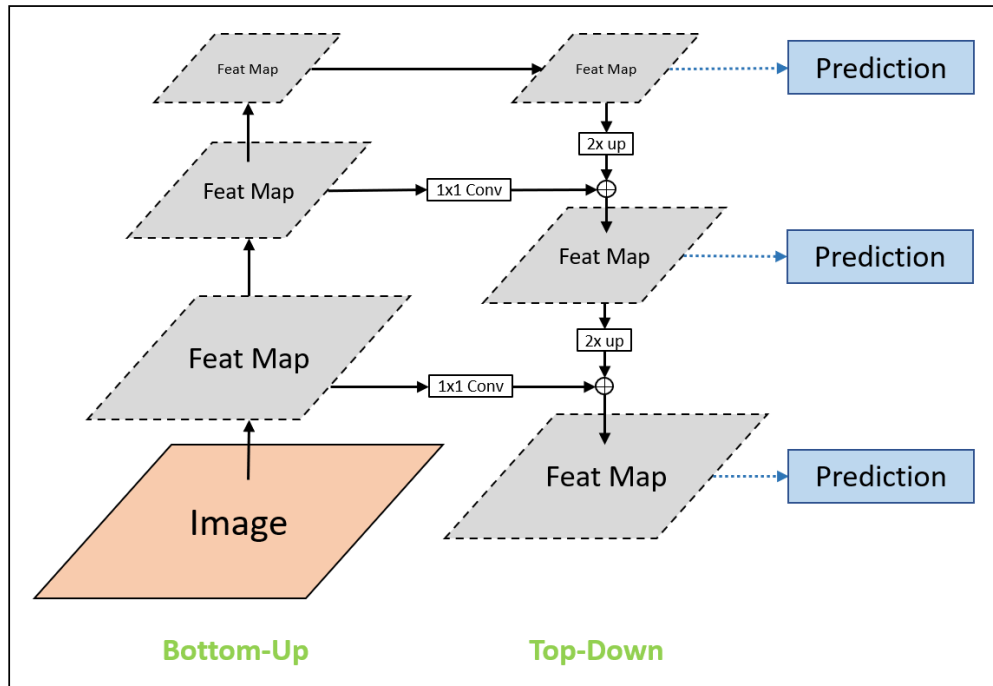


Figure 3.13. The Architecture of the FPN module.

factor of 0.5 (i.e., $\frac{1}{4}$ in size) compared to the preceding layer. To build a top-down pathway, the deepest feature layer in the bottom-up pathway with the coarsest resolution and strongest semantic features is up-sampled by a factor of 2. Then, the up-sampled map is merged with the feature layer in the bottom-up pathway at one lower level via a lateral connection. The dimension of the lower-level feature layer is reduced through a 1×1 convolution before the merge, and two feature layers are merged using element-wise addition. Again, the merged feature map is up-sampled by a factor of 2 and merged with the feature layer in the bottom-up pathway at one lower level via a lateral connection. The process is repeated until the number of feature layers in both bottom-up and top-down pathways are equal.

The FPN module naturally leverages the pyramidal feature hierarchy of a CNN while creating a feature pyramid with strong semantics at all scales. Moreover, compared to the methods that generate feature pyramids using multi-scale images (i.e., shown in Figure 4.6), FPN builds feature pyramids quickly from a single input image scale without sacrificing representational power, speed, or memory. Experiments show that integrating the FPN module does not significantly impact the model's training and inference speed (Lin et al., 2017).

3.1.4 Review of Swin Transformer

In recent years, transformers (Vaswani et al., 2017) have shown dominancy in natural language processing (NLP) tasks, which has motivated many research efforts to adapt transformers for vision tasks. In 2020, the Vision Transformer (ViT) (Vaswani et al., 2020) attracted much attention from the computer vision community due to its pure transformer architecture with promising results in vision tasks. However, despite the achievement, ViT mainly suffers from overly expensive computations for high-resolution images; its fixed scale tokens do not allow inputs of

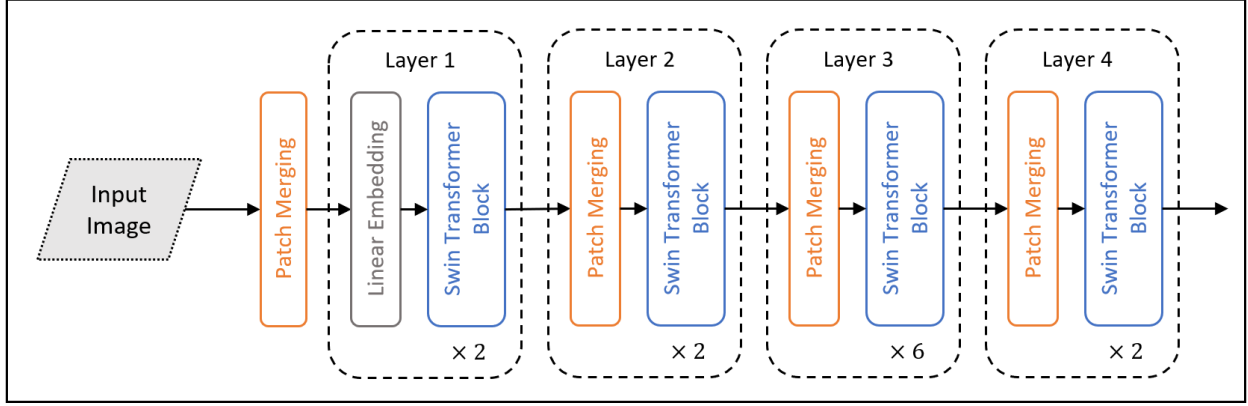


Figure 3.14. The architecture of Swin Transformer.

variable scales, which is unsuitable for many vision tasks. To overcome the problems, in 2021, Microsoft researchers published a breakthrough work following ViT, named Swin Transformer (Liu et al., 2021). Swin Transformer resolves the issues that plagued the original ViT with hierarchical feature maps and shifted window MSA, and has shown to be the state-of-the-art backbone in many vision tasks. As shown in Figure 3.14, the architecture of the Swin Transformer consists of two major components—the Patch Merging module and the Swin Transformer Block. These components are reviewed as follows.

3.1.4.1 Patch Merging

A major achievement of Swin Transformer is that it produces hierarchical feature maps just like a CNN. CNNs generate hierarchical feature maps by down-sampling feature maps with convolutional operations. However, down-sampling becomes a challenge in a pure transformer network without convolution operations. Swin Transformer achieves the goal using a patch merging technique, as illustrated in Figure 3.15. For example, assuming the patch merging technique is applied to a 4×4 feature map (i.e., Figure 3.15.(a)) to down-sample it by a factor of $n = 2$. First, the feature map is split into $\frac{4}{n} \times \frac{4}{n}$ patches, each consists of $n \times n = 2 \times 2$ pixels.

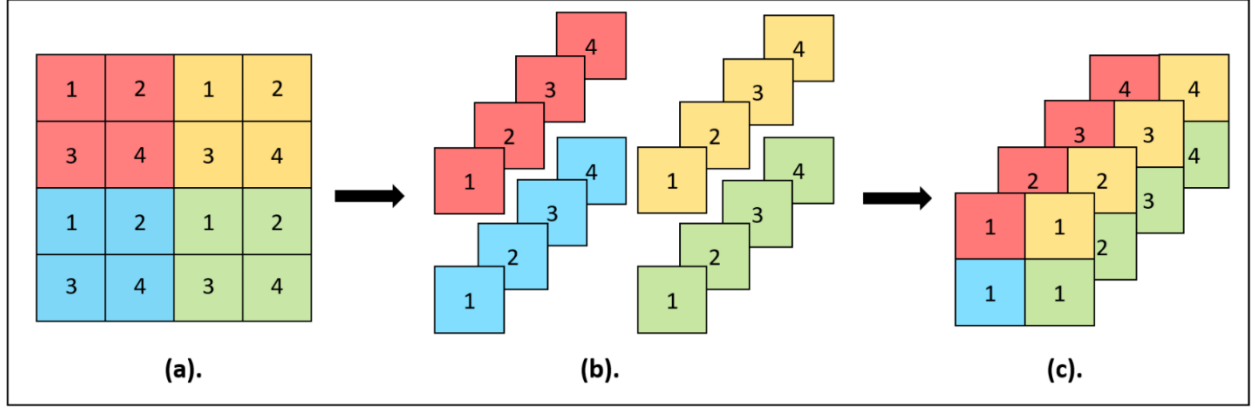


Figure 3.15. An example of the patch merging process. (a) Splitting a feature map into patches. Each patch is represented using a unique color; (b) each patch is stacked depth-wise to form a feature vector; (c) concatenating the feature vectors following the positions of the patches on the feature map.

Next, each patch is stacked depth-wise in the order shown in Figure 3.15.(b) to form a feature vector of depth 4 (i.e., the channel number of the down-sampled feature map). Finally, all the feature vectors are concatenated based on the positions of the patches on the feature map, forming a down-sampled feature map with size $\frac{4}{n} \times \frac{4}{n} \times 4 = 2 \times 2 \times 4$ as shown in Figure 3.15.(c). The patch merging technique effectively down-samples the input by a factor of n , transforming the input from a shape of $H \times W \times C$ to $(\frac{H}{n}) \times (\frac{W}{n}) \times (n \times n \times C)$, where H , W and C refers to the height, width, and channel depth of the input, respectively.

3.1.4.2 Swin Transformer Block

The merged patches are fed to the Swin Transformer block to generate feature representations. The architecture of the Swin Transformer block is illustrated in Figure 3.16, which mainly consists of two sub-units. Each sub-unit consists of a normalization layer, followed by a modified MSA module, followed by another normalization layer, and an MLP layer. The modified MSA in the

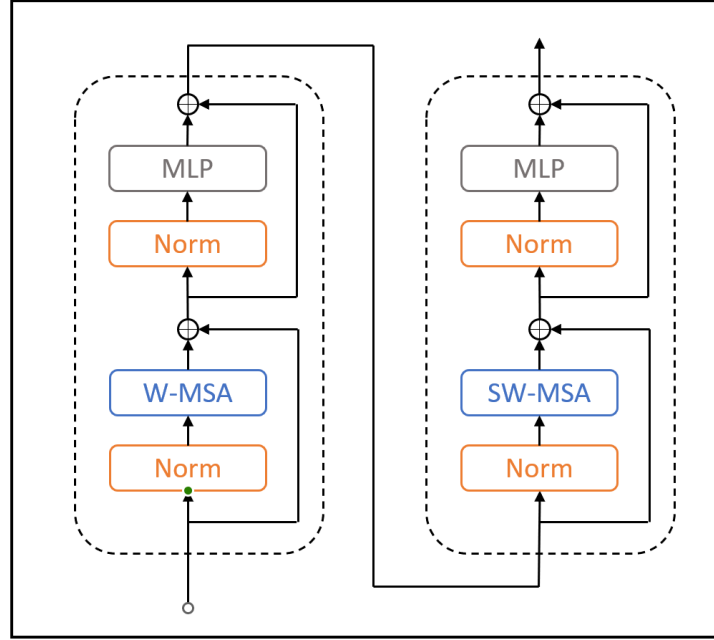


Figure 3.16. The architecture of the Swin Transformer Block.

first sub-unit is named Window MSA (W-MSA), and the one in the second sub-unit is named Shifted Window MSA (SW-MSA). Each of the modified MSAs is explained as follows.

3.1.4.2.1 Window MSA

Besides producing hierarchical feature maps, another major achievement of the Swin Transformer is the significantly reduced computational complexity compared to the ViT. As shown in Figure 3.17.(a), the standard MSA used in ViT performs global self-attention, which computes the relationships between each patch and all other patches. Such a computation results in quadratic complexity with respect to the number of patches, making it unsuitable for high-resolution images. To address the issue, the window-based MSA, as shown in Figure 3.17.(b), evenly divides the patches into fix-sized non-overlapping windows, and the same self-attentions are performed within

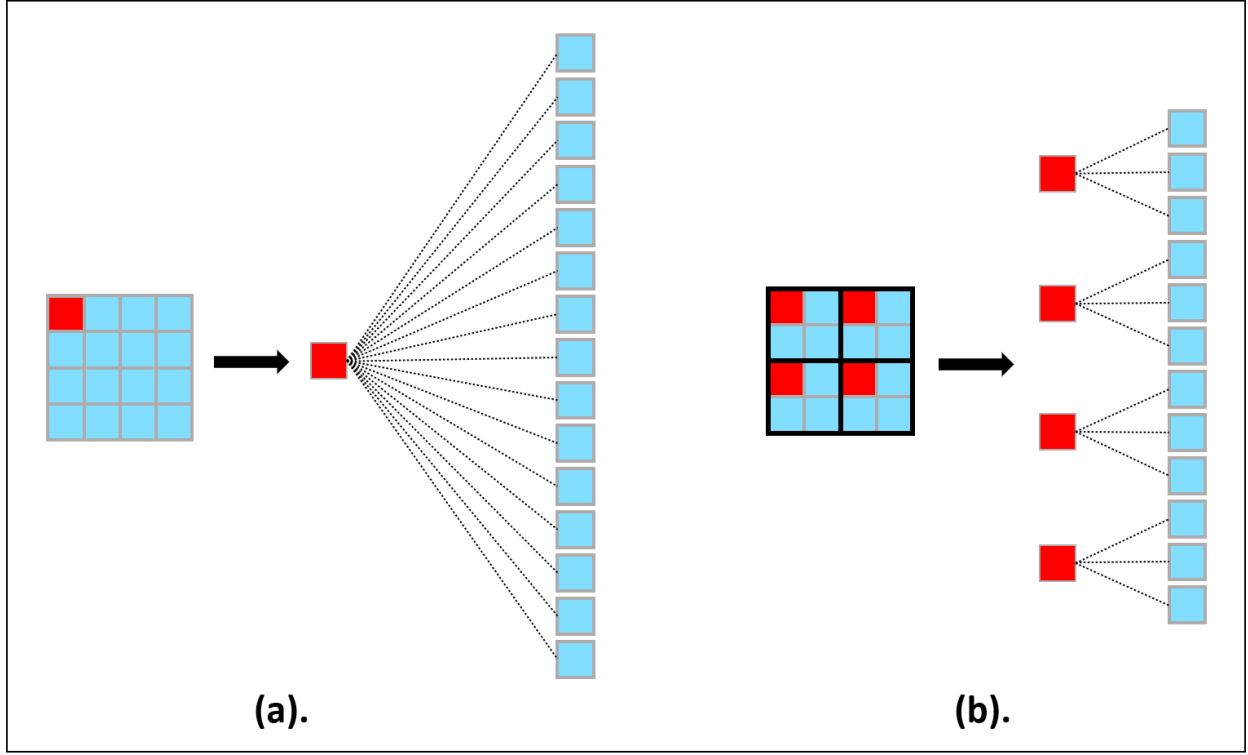


Figure 3.17. A comparison between the standard MSA and the window-based MSA. (a) the standard MSA with a quadratic complexity (with respect to the number of patches). Each small square represents a patch.; (b) the window-based MSA with a linear complexity.

each window. This way, the computation complexity becomes linear, which is significantly more efficient than the one of the standard MSA.

3.1.4.2.2 Shifted Window MSA

Despite the greatly improved computation efficiency, a significant drawback of the window-based MSA is the lack of connections across windows, which results in limited modelling power. To overcome the issue, a Shifted Window MSA module, which introduces cross-window connections, is added after the W-MSA module. As illustrated in Figure 3.18.(a) and (b), the SW-MSA shifts the windows used in W-MSA towards the bottom direction by $\frac{h}{2}$ patches and the right direction by $\frac{w}{2}$ patches, where h and w are the height and width of a window, respectively. However, such a

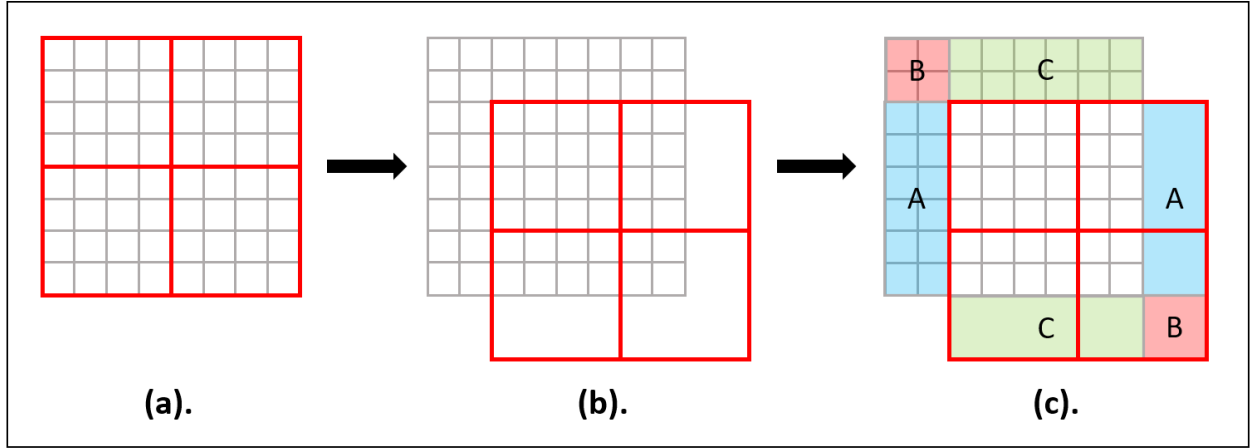


Figure 3.18. The window shifting process of the SW-MSA.

shift introduces extra patches outside the windows and empty spots within the windows. To perform self-attentions within each shifted window, a ‘cyclic shift’ technique is introduced, as shown in Figure 3.18.(c), which shifts the extra patches into the empty spots. This way, connections are introduced across windows while the computation efficiency is maintained at a linear rate.

3.2 The Proposed Model

The PolygonCNN model is modified to more precisely extract building footprint in vastly different sizes and orientations. The architecture of the improved model is shown in Figure 3.19. First, the PSPNet of PolygonCNN has been replaced by the Swin-Transformer-based Mask R-CNN, which is integrated with the rotatable bounding box technique and the FPN module to produce significantly enhanced building segmentations for buildings in vastly different sizes and orientations. Second, to adapt the multi-scale feature maps of the FPN module, the Feature Pooling module of PolygonCNN is modified such that it accepts a feature pyramid as an input instead of a single feature map, and outputs multi-scale feature representations; furthermore, the BRegNet is widened and deepened to exploit the pooled multi-scale features.

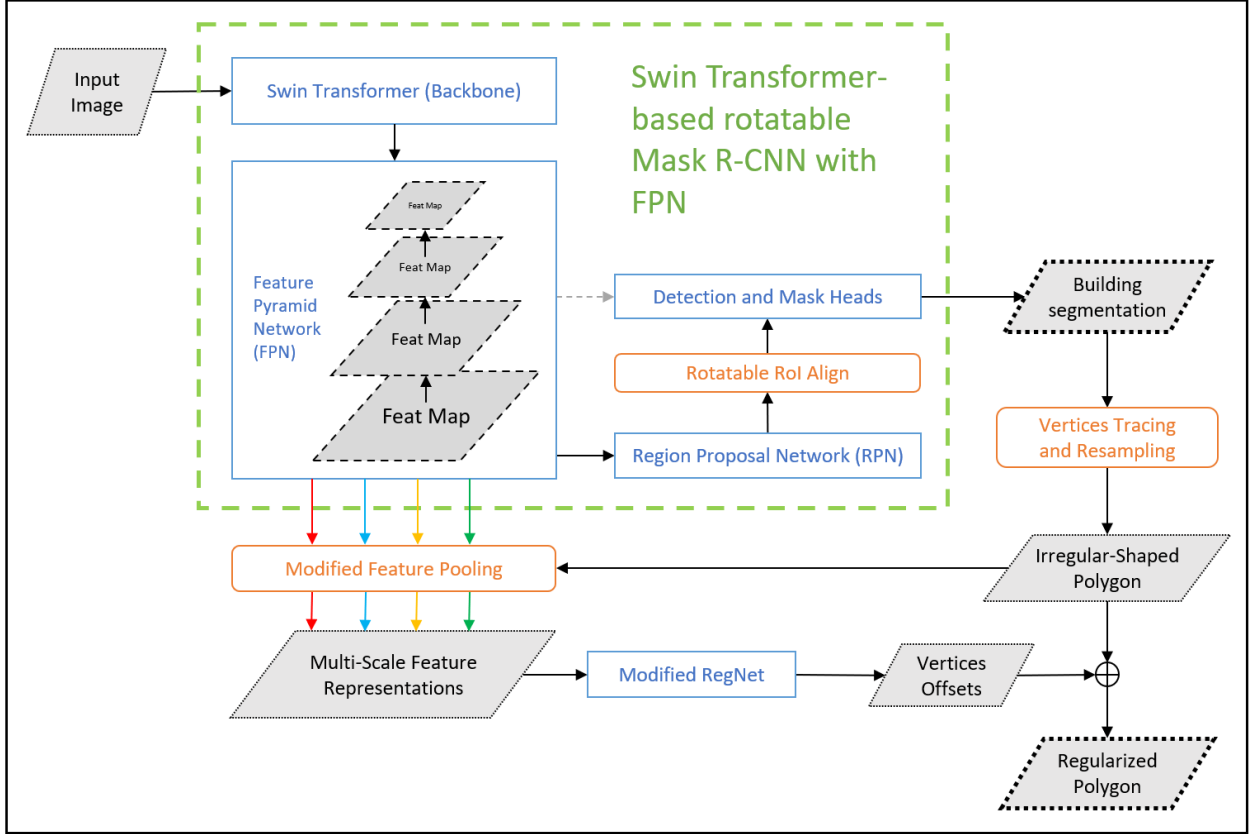


Figure 3.19. Architecture of the proposed model.

This section is outlined as follows. First, the integration of Mask R-CNN with the rotatable bounding box is explained in Section 3.2.1. Then, the integration of the FPN module with the Swin-Transformer-based rotatable Mask R-CNN is described in Section 3.2.2. Next, the architecture of the modified Feature Pooling module is explained in section 3.2.3, while the modified BRegNet is presented in section 3.2.4.

3.2.1 Integration of The Rotatable Bounding Box with Mask R-CNN

As shown in section 4.2.1, the current Mask R-CNN model tends to produce significantly defected building segmentations, especially for the buildings with rotated orientations (i.e., building edges are neither vertical nor parallel to the image edges). This is largely due to the useless empty background between the rotated building and the traditional non-rotatable bounding box. To

alleviate this issue, a rotatable bounding box technique is integrated with the Mask R-CNN model, which aims to improve the segmentation quality by minimizing the background spaces. The integration mainly consists of the following components: the representation of the rotatable bounding box, the generation of the rotatable anchor, the IoU calculation of the rotatable bounding box, the RoI Align of the rotatable bounding box, and finally, pasting the rotatable mask prediction to the empty image to obtain the final mask. Each of these components is revealed as follows.

3.2.1.1 Rotatable Bounding Box Representation

One major challenge of the rotatable bounding box scheme is the representation method. Traditionally, the non-rotatable boxes are represented by the coordinates of their top left and bottom right corners which consist of only four values (i.e., $x_{min}, y_{min}, x_{max}, y_{max}$). To represent rotatable bounding boxes, the eight-parameter (i.e., the coordinates of the four corner points: $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$) (Xia et al., 2018) and the five-parameter (i.e., the center coordinate, width, height, and rotation angle of the box: x, y, w, h, θ) (Li et al., 2018) methods have been proposed. This thesis adopts the five-parameter method similar to the one used in (Li et al., 2018) due to its advantages in simplicity and the stableness of rectangular shape. However, their h and w parameters are restricted to be the longer and shorter sides of the box while θ is fixed in the range of $[0^\circ, 180^\circ)$; this thesis uses a simplified representation where θ is represented in a smaller range of $[0^\circ, 90^\circ)$ (i.e., $[0, \frac{\pi}{2})$ in radians) while the lengths of h and w are not restricted. More specifically, as shown in Figure 3.14, for any rotatable bounding box R centered at (c_x, c_y) , let a Polar Coordinate System to be generated at (c_x, c_y) ; for the two perpendicular principal directions associated with the sides of R (i.e. denoted as two blue dotted lines passing through (c_x, c_y)), the one that has a polar angle (i.e., the smallest positive angle between the polar axis and the principal

direction) within $[0, \frac{\pi}{2})$ is denoted as θ . The length of the opposite sides of R that have a direction associated with θ is denoted as w , while the length of the other opposite sides is denoted as h . Therefore, R can be represented as (c_x, c_y, w, h, θ) . In addition, for a given R with a polar angle α outside of the $[0, \frac{\pi}{2})$ range, θ can be conducted by:

$$\theta = p \bmod \frac{\pi}{2} \quad (3.3)$$

where mod represents the modulo operator which finds the remainder when p is divided by $\frac{\pi}{2}$. For example, as shown in Figure 3.20.(a), let R be the red bounding box, and (c_x, c_y) be the center point of R . Since the principal directions of R are aligned with the polar axis, $\theta = 0$; as the opposite sides of R that are associated with θ have a length of 3, $w = 3$; the other opposite sides have a length of 1, thus $h = 1$. In this case, R can be represented as which is $(c_x, c_y, 3, 1, 0)$. If R is rotated counter-clockwise for $\frac{\pi}{4}$ around (c_x, c_y) as shown in Figure 3.20.(b), R can be represented as $(c_x, c_y, 3, 1, \frac{\pi}{4})$. However, as shown in Figure 3.20.(c), if R is rotated counter-clockwise for $\frac{\pi}{2}$, θ can no longer be $\frac{\pi}{2}$ since $\frac{\pi}{2}$ is out of the $[0, \frac{\pi}{2})$ range. In this case, θ can be calculated using Equation (3.3), where $p = \frac{\pi}{2}$. Therefore, $\theta = p \bmod \frac{\pi}{2} = \frac{\pi}{2} \bmod \frac{\pi}{2} = 0$. In addition, since the opposite sides of R that have a polar angle of 0 have a length of 1, $w = 1$ and $h = 3$. Thus, R can be represented as $(c_x, c_y, 1, 3, 0)$. Similarly, as shown in as shown in Figure 3.20.(d), if R is rotated counter clockwise for $\frac{3\pi}{4}$, θ can be calculated as $\theta = p \bmod \frac{\pi}{2} =$

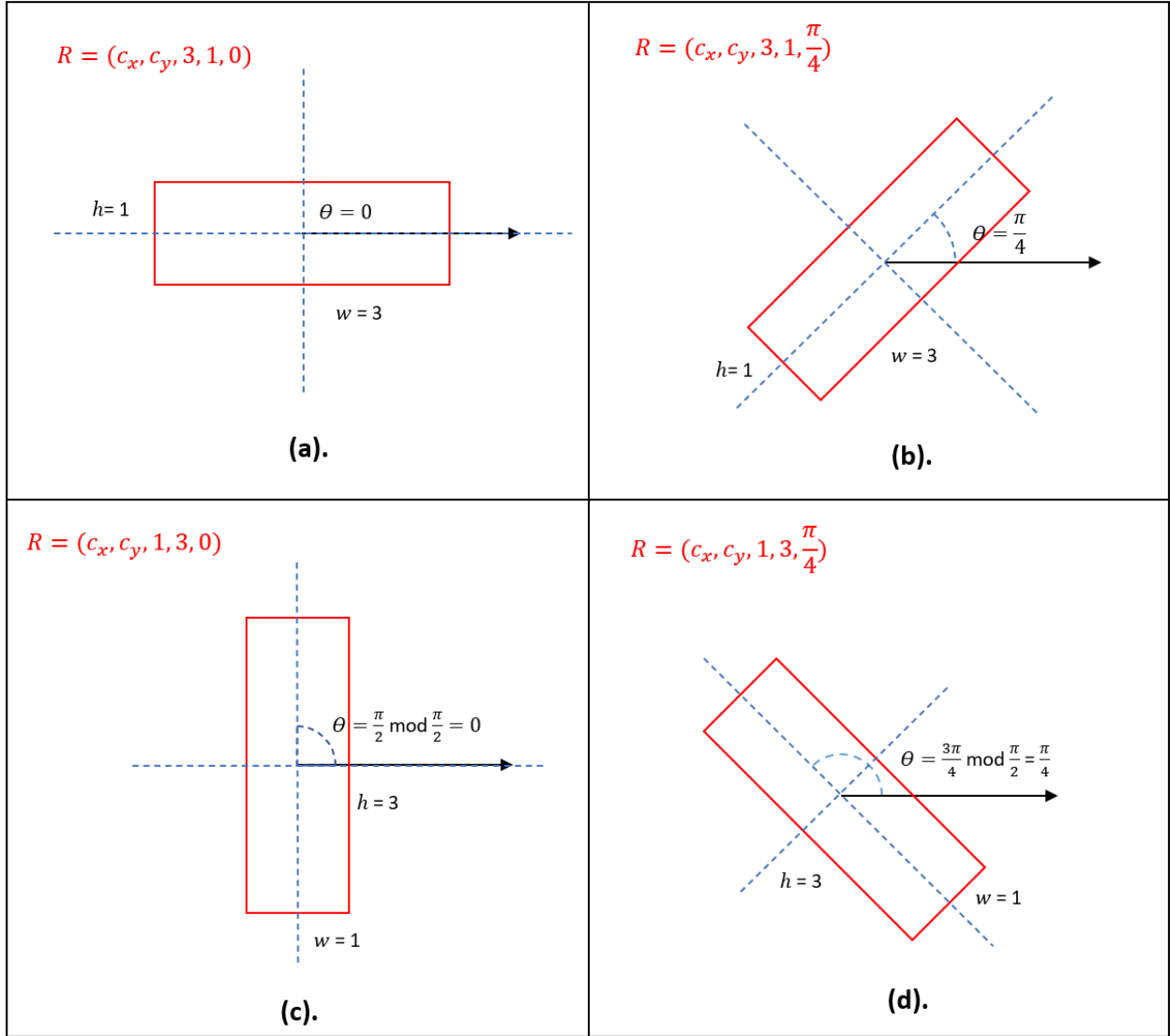


Figure 3.20. Examples of the proposed representation for rotatable bounding box.

$\frac{3\pi}{4} \bmod \frac{\pi}{2} = \frac{\pi}{4}$. Therefore, $w = 1$, $h = 3$, and $R = (c_x, c_y, 1, 3, \frac{\pi}{4})$. The same representation can be applied to any R with an arbitrary c_x, c_y, w, h , and rotation angle.

3.2.1.2 Rotatable Anchor Generation

The traditional non-rotatable anchors are generated at the RPN stage of Mask R-CNN, as described in section 3.1.2.2. To predict rotatable bounding boxes, rotatable anchors are integrated with the anchor generation. Similar to the traditional anchor generation described in section 3.1.2.2, the rotatable anchors also have predefined scales and ratios. However, in addition, a set of rotation values is defined. For example, Figure 3.21 illustrates a set of rotatable anchor boxes with one scale (i.e., 16×16), three ratios (i.e., 1:1, 1:2 and 2:1), and two rotations (i.e., 0 and $\frac{\pi}{4}$, shown in red and blue colors, respectively). It is noted that in actual practice, multiple scales are often generated; however, only one scale is illustrated here for the simplicity of visualization; however, only one scale is illustrated here for the simplicity of visualization. In this case, for each scale,

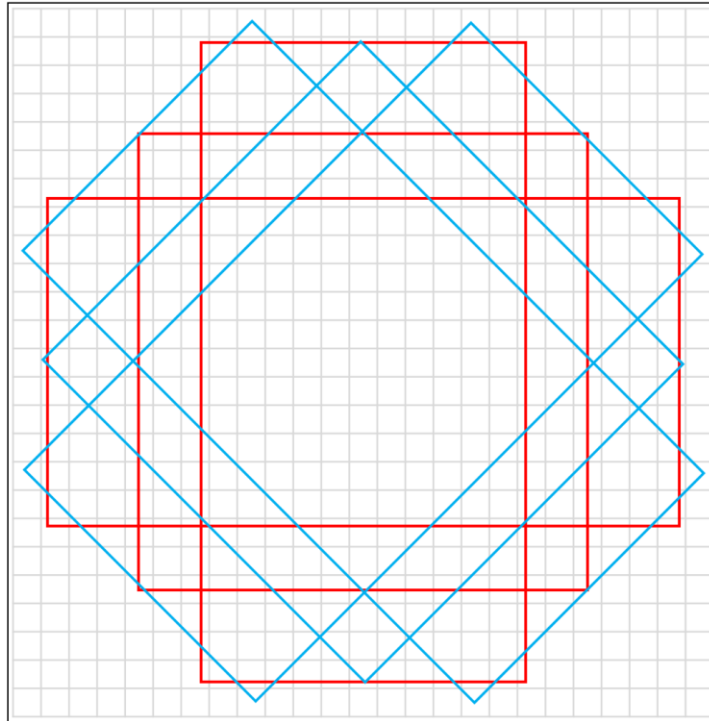


Figure 3.21. An example of rotatable anchor boxes with one scale, three ratios, and two rotations. The one scale is 16×16 ; the three ratios are 1:1, 1:2, 2:1; the two rotations are 0 and $\frac{\pi}{4}$, shown in red and blue colors, respectively.

anchors with three different ratios are generated; for each ratio, anchors two different rotations are generated. Thus, each predefined location on the image generates a total of $1 \times 3 \times 2 = 6$ anchor boxes. For each anchor, the RPN predicts an objectness value (i.e., ranged from 0 to 1, where 0 represents non-object and 1 represents object) and five regression values (i.e., the offsets of the anchor box) to indicate the possibility of an anchor being background or foreground, and the refined location, shape, and rotation of the anchor box.

3.2.1.3 IoU Calculation of The Rotatable Bounding Box

The IoU calculation between the bounding boxes plays an essential role in Mask R-CNN. It is used in choosing positive samples during training and in the non-maximum suppression and evaluation processes during testing. The IoU calculation between two non-rotatable bounding boxes is described in Figure 3.9, where a positive intersection is always a rectangular region which is trivial to compute. However, the IoU calculation between two rotatable bounding boxes becomes sophisticated and computationally expensive.

As shown in Figure 3.22, the possible shapes of the positive overlapping region between two rotatable bounding boxes consist of a triangle, quadrangle, pentagon, hexagon, heptagon, and octagon. To precisely calculate the IoU between two rotatable bounding boxes, the plane sweep (Berg et al., 1997) algorithm is used for finding the vertices of the intersection polygon. First, the coordinates of the four vertices of the two boxes are calculated and sorted in counter-clockwise order. Each box can be seen as an intersection of half-planes defined by the edges. Moreover, one box is defined as the candidate intersection polygon, and the other is defined as the cutting polygon. Next, iterate through every edge of the cutting polygon in order; for each edge, all regions from the candidate intersection polygon on the outer half plane of the edge are removed. When finished,

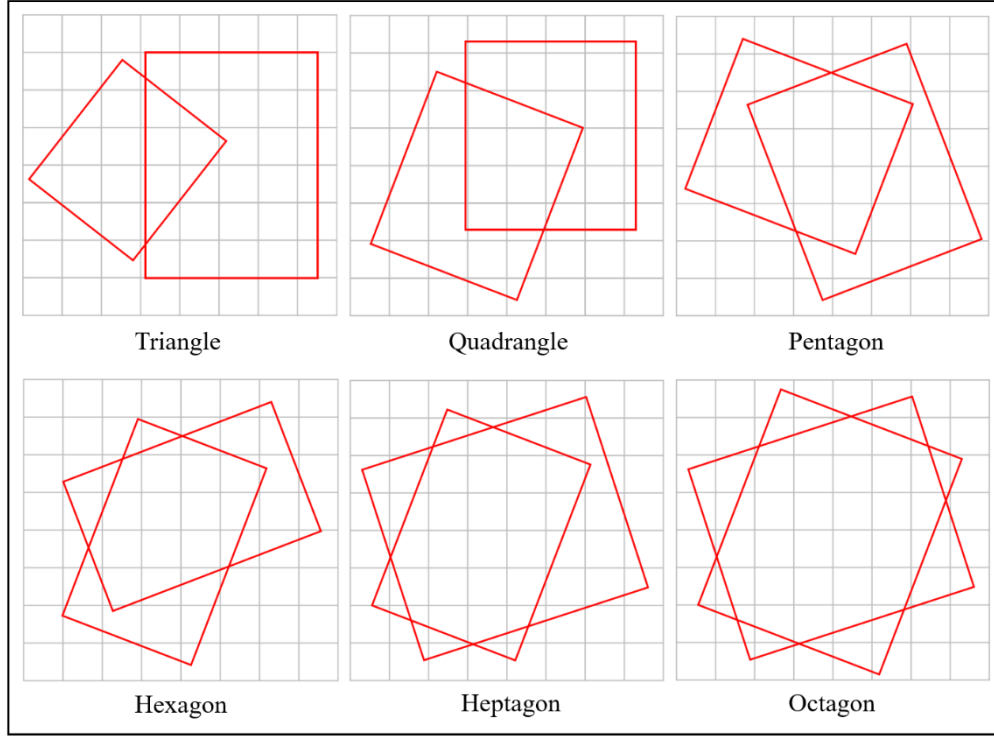


Figure 3.22. Examples of the possible shapes of the positive overlapping region between two rotatable bounding boxes.

the candidate intersection polygon becomes the intersection polygon of the two rotatable boxes, and again, its vertices are sorted in counter-clockwise order. Finally, the area A_{inter} of the intersection polygon can be calculated with the formula:

$$A_{inter} = \left| \frac{(x_1y_2 - y_1x_2) + (x_2y_3 - y_2x_3) \cdots + (x_1y_n - y_1x_n)}{2} \right| \quad (3.4)$$

where (x_n, y_n) is the coordinate of the n_{th} vertex of the polygon. When A_{inter} is obtained, the IoU can be calculated with Equation (3.1), where *Area of Overlap* = A_{inter} , and *Area of Union* = $A_{b1} + A_{b2} - A_{inter}$ where A_{b1} and A_{b2} are the areas of the two bounding boxes.

3.2.1.4 RoI Align for The Rotatable Bounding Box

The RPN's prediction of rotatable bounding box is fed to the RoI Align module along with the feature maps to generate uniform-shaped rotatable RoIs. The RoI Align for the rotatable bounding box involves similar processes as the original RoI Align as described in section 3.1.2.3, where the non-rotatable bounding box is replaced with the rotatable bounding box; however, a major difference is that the rotated boxes can have regions outside the feature maps, which needs to be handled separately.

The non-rotatable bounding box predictions of the RPN are post-processed by cropping with the feature map edges such that all their corners are located within (or on edge) of the feature maps. As a result, the cropped boxes maintain their rectangular shapes, as shown in Figure 3.23.(a). On the other hand, if a rotatable bounding box is partially located outside the feature map, directly cropping the box can result in a non-rectangular shape (e.g., Figure 3.23.(b)). A non-rectangular bounding box cannot be used to perform RoI Align since the algorithm requires the box to be

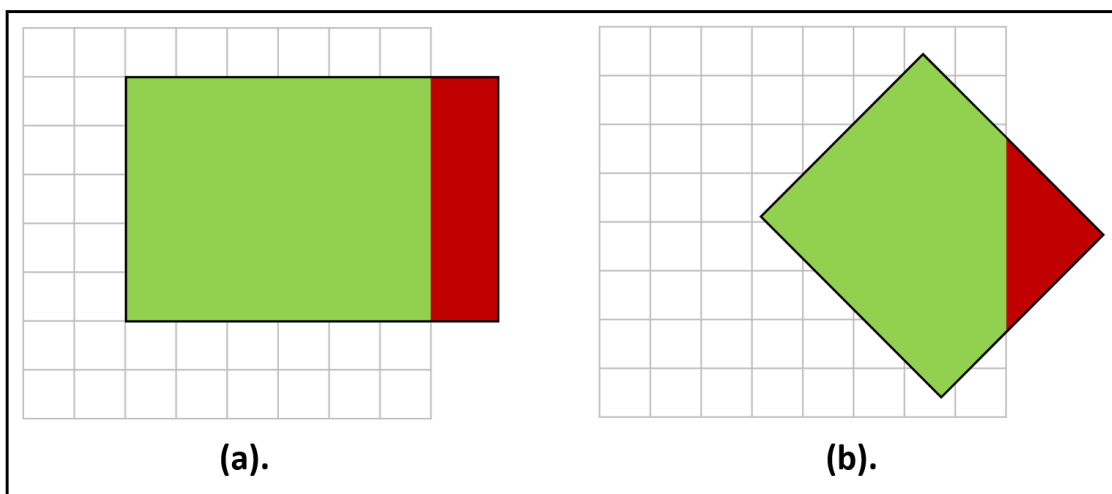


Figure 3.23. Example of cropping the two types of bounding boxes with partially excluded regions. (a) cropping a non-rotatable bounding box with a feature map; (b) cropping a rotatable bounding box with a feature map. Green represents the cropped region; red represents the eliminated region.

divided even into $n \times n$ cells. Moreover, RoI Align cannot be performed on a box with excluded regions since the sampling points on the excluded regions are not overlaying on any feature map pixels (e.g., Figure 3.24.(a)). To maintain the rectangular shape of the rotatable box while allowing all sampling points to bilinear interpolate the feature map, a symmetric padding strategy is used to symmetrically copy the feature map values to the excluded regions, as shown in Figure 3.24.(b).

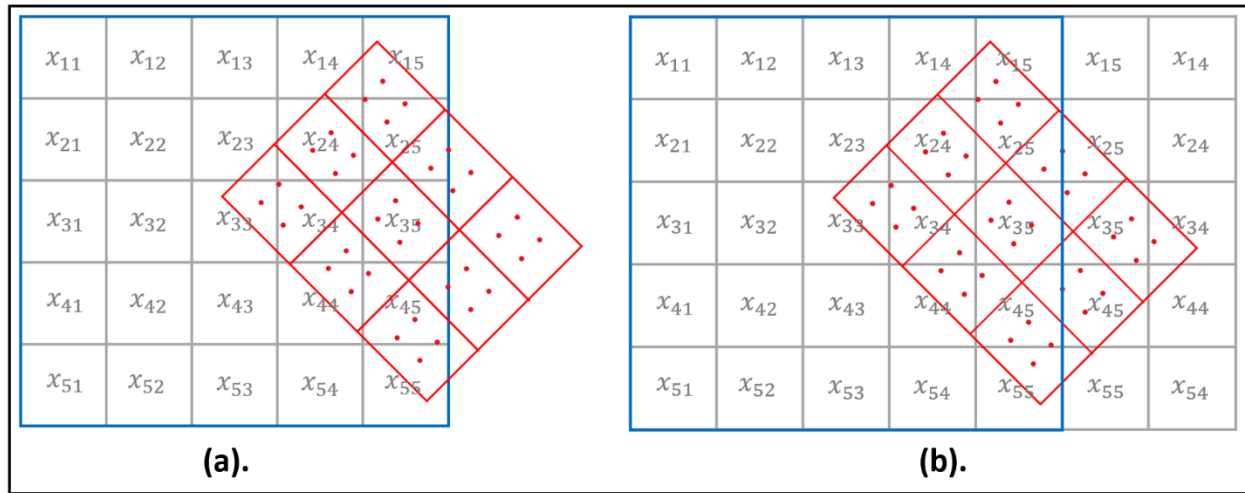


Figure 3.24. Symmetric padding of the RoI Align for the rotatable bounding box. (a) a rotated bounding box with regions excluded from the feature map. To perform RoI Align, the box is evenly divided into 9 cells and sampling points are generated within each cell; (b) Symmetrically copying the feature map pixels to cover the excluded regions such that bilinear interpolation can be performed for every sampling points.

3.2.1.5 Pasting The Rotatable Mask Prediction to The Empty Image

In general, for both the rotatable and non-rotatable bounding box cases, the detection head of Mask R-CNN predicts a refined bounding box for each RoI generated from RoI Align; within each refined bounding box, the mask head predicts a binary mask (i.e., a pixel of 0 represents background and 1 represents object) for an object. An example of a predicted bounding box for a building object in a satellite image is shown in Figure 3.25.(a), while the mask prediction for the building object within the bounding box is shown in Figure 3.25.(b). Such mask predictions are

non-uniform sized (i.e., based on the sizes of the object) and unacceptable since the standard mask outputs are required to be at image size while each mask reveals the object's location on the image. To produce such image-sized outputs, a method is required to paste the tightly bounded mask onto an empty image (i.e., pixels of 0's) using the location information of the bounding box, as shown in Figure 3.25.(c).

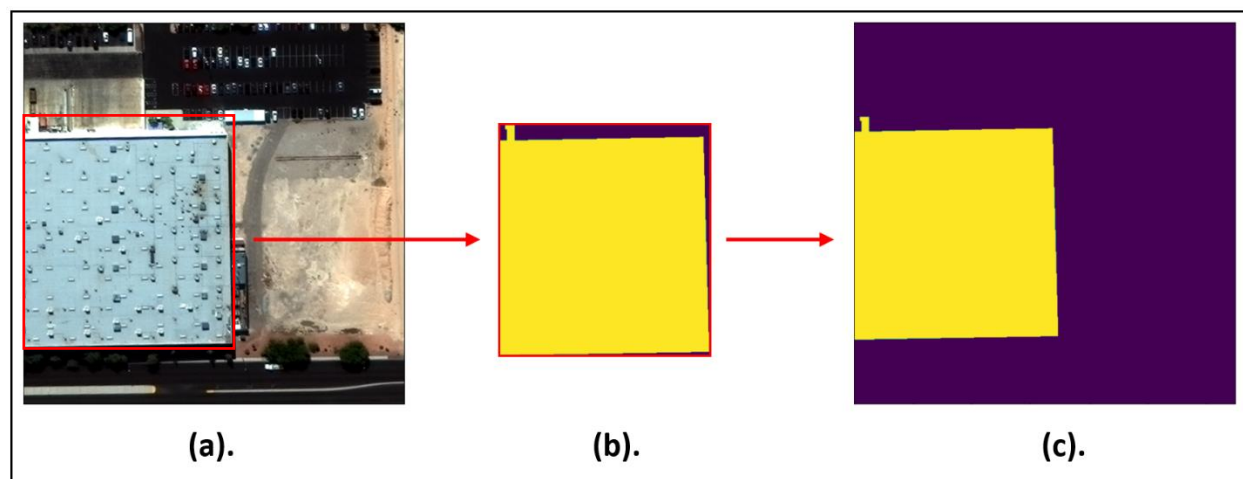


Figure 3.25. An example of pasting the mask prediction to an empty image. (a) the input image; (b) the bounding box and mask predictions of a building in the image; (c) pasting the tight-bounded mask to an empty image using the location information of the bounding box. In (b) and (c), purple-colored pixels represent background and yellow-colored pixels represent foreground.

Pasting non-rotatable mask predictions onto an empty image is a simple task because the pixels of the mask predictions are axis-aligned with the ones of the image such that they can be directly pasted onto the image without using any interpolations. However, it is much more sophisticated to paste a rotated mask onto an empty image. As described in section 3.2.1.4, the rotated RoI generated from the modified RoI Align may contain useless features that are excluded from the feature map. Therefore, the final mask predictions of Mask R-CNN may also contain pixels excluded from the input image, which cannot be concatenated with the empty image, as all final masks must have the same size as the input image. Moreover, since the pixels of the rotated mask

prediction are not axis-aligned with the pixels of the input image, interpolation is required when pasting the mask to the empty image. To overcome these issues, Figure 3.26 illustrates the proposed method. First, for a rotated mask with regions excluded from the empty image (e.g., Figure 3.26.(a)), a pixel-aligned non-rotatable bounding box is generated within the empty image, which completely covers all the mask pixels that lie within the empty image (e.g., Figure 3.26.(b),

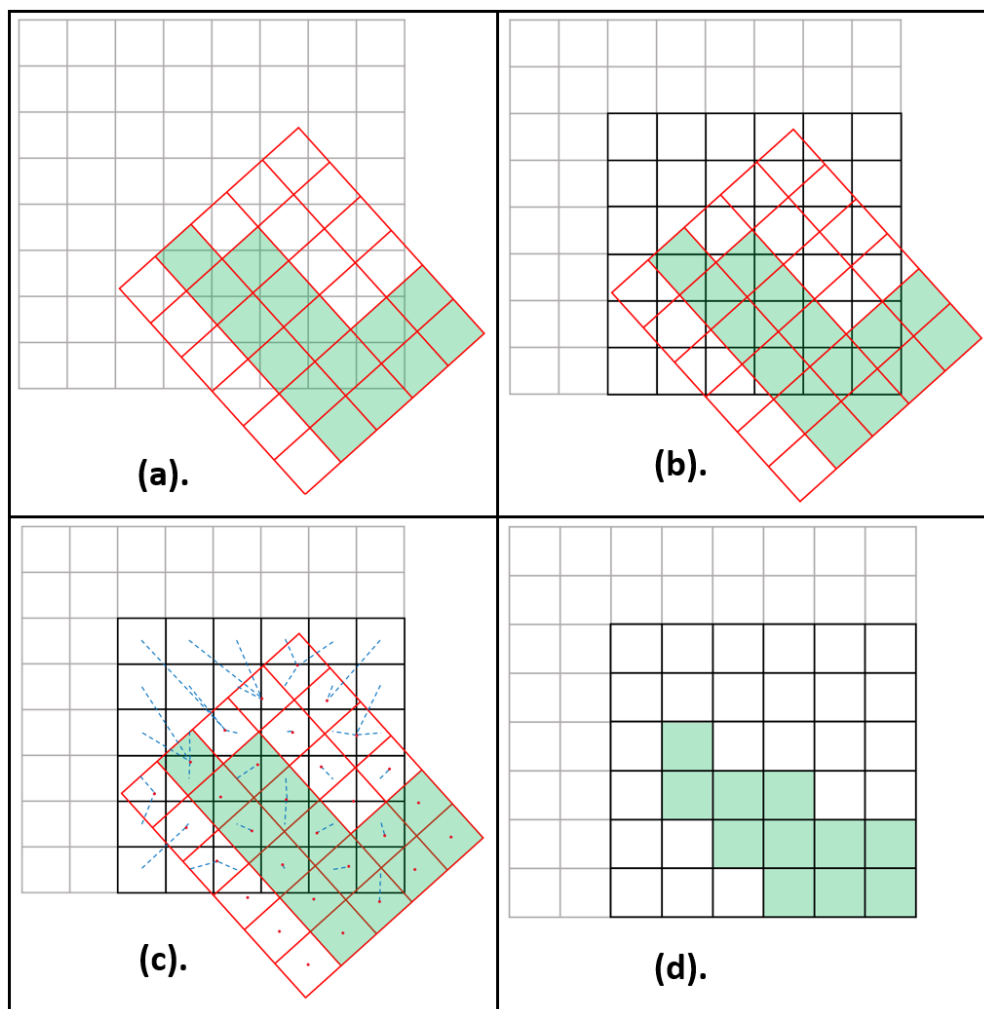


Figure 3.26. The proposed solution for pasting a rotated mask onto an empty image. (a) a rotated mask with regions excluded from the empty image; (b) generating a pixel-aligned non-rotatable bounding box that completely covers all the mask pixels that lie within the empty image, shown in black; (c) replacing the 0 value of each pixel in the black box with the value of its nearest mask pixel if their distance is less than $\sqrt{0.5}$; (d) the interpolated final mask. Green-colored pixels represents foreground.

shown in black). Next, the background value (i.e., 0) of each black box pixel is replaced with the value of its nearest mask pixel if their distance is less than $\sqrt{0.5}$, as illustrated in Figure 3.26.(c). Finally, as shown in Figure 3.26.(d), the interpolated black box pixels can be directly pasted onto the empty image to obtain the final mask.

3.2.2 Integration of The FPN Module With The Swin-Transformer-Based Rotatable Mask R-CNN

Besides integrating the rotatable bounding box with Mask R-CNN, the FPN module is also integrated into the Mask R-CNN model to improve its ability to segment building footprints in various sizes. Figure 3.27 shows the detailed integration of FPN with the Swin-Transformer-based rotatable Mask R-CNN.

As described in Section 3.2.1, the RPN in the rotatable Mask R-CNN is applied to the last feature layer of a backbone network to generate rotatable bounding box proposals; then, this single-scale

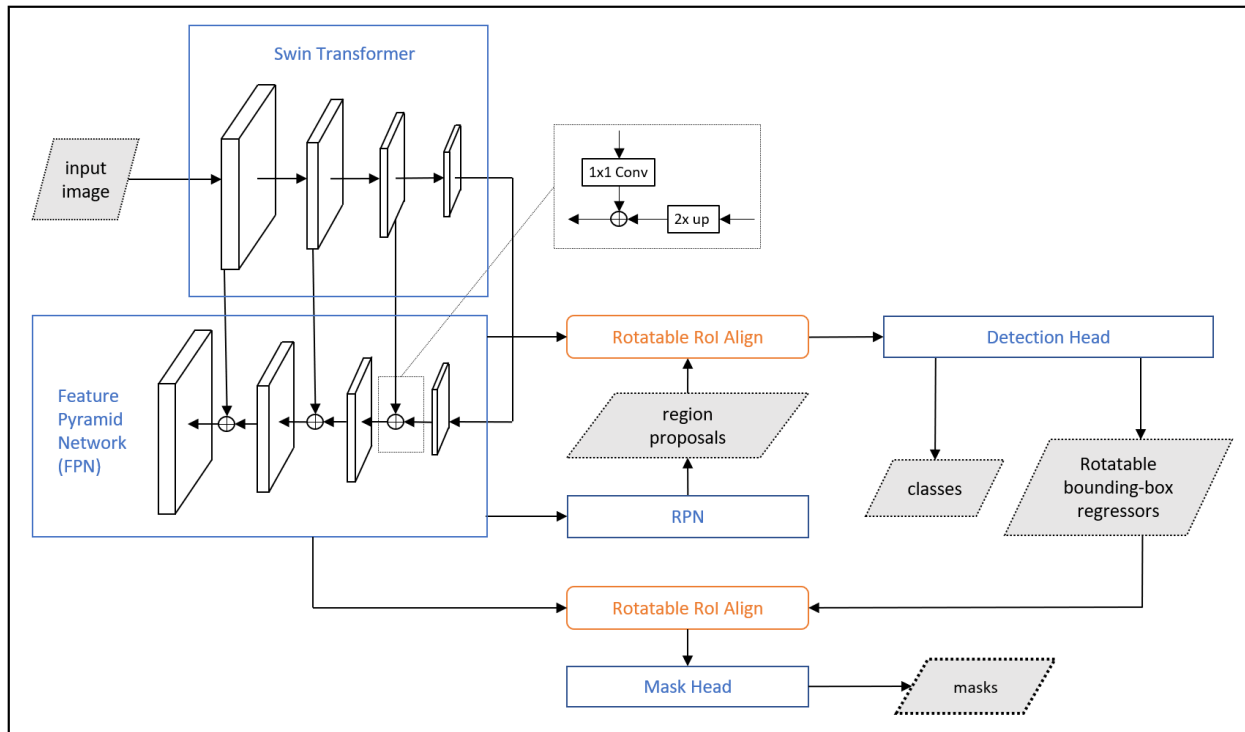


Figure 3.27. Integration of FPN with Swin-Transformer-based Rotatable Mask R-CNN.

feature map is cropped by the rotatable bounding boxes. Using the Rotated RoI Align module, uniformly sized RoIs are produced for the detection head. In addition, the feature map is later cropped again by the refined rotated bounding boxes (i.e., produced by the detection head), and RoIs are produced using the Rotated RoI Align module for the mask head to predict building masks. However, with the integration of FPN, the rotatable Mask R-CNN is modified in the following ways. First, an RPN with shared weights is applied to each scale of the feature maps in the FPN to predict bounding box proposals. Based on the width and height of the rotatable bounding box, a corresponding-scale feature map is used for generating the RoI, which is fed to the detection head. The following equation is used to select the appropriate level of feature map in the feature pyramid:

$$k = \left\lfloor k_0 + \log_2\left(\frac{\sqrt{wh}}{650}\right) \right\rfloor \quad (3.5)$$

Where w and h are the width and height of the rotatable bounding box; k_0 is the level of the last layer of the feature map. In this case, $k_0 = 4$. For Equation (3.5), a larger-scale bounding box corresponds to a coarser-resolution feature map (i.e., larger k); a smaller-scale bounding box corresponds to a finer-resolution feature map (i.e., smaller k). Next, again, based on the width and height of the refined rotatable bounding box produced by the detection head, a corresponding-scale feature map is used for generating the RoI, which is fed to the mask head for the building mask prediction. In addition, for the rotatable anchor box generation, instead of generating multiple scales of anchor boxes at a single feature map (i.e., described in section 3.2.1.2), each scale of anchor boxes (i.e., with multiple aspect ratios and rotations) is generated on the corresponding scale of the feature map in the FPN. With the FPN module integrated in such a way, it is hoped

that the performance of the rotated Mask R-CNN in segmentation building footprints in broadly varied sizes will be significantly enhanced.

3.2.3 Modified Feature Pooling

To exploit the FPN module of the improved Mask R-CNN, the feature pooling module of PolygonCNN is modified to accept the multi-scale feature maps of the FPN.

As described in section 3.1.1.3, the Feature Pooling module of the original PolygonCNN pools features vectors from a single-scale feature map, and outputs a concatenation of the feature vectors and their corresponding polygon vertices. However, with the availability of multi-scale feature maps from the FPN, the Feature Pooling module is modified to pool multiple scales of feature vectors from the feature pyramid, and outputs multi-scale feature representations, which are concatenations of the multi-scale feature vectors and their corresponding polygon vertices. For example, Figure 3.28 illustrates the architecture of the modified Feature Pooling module.

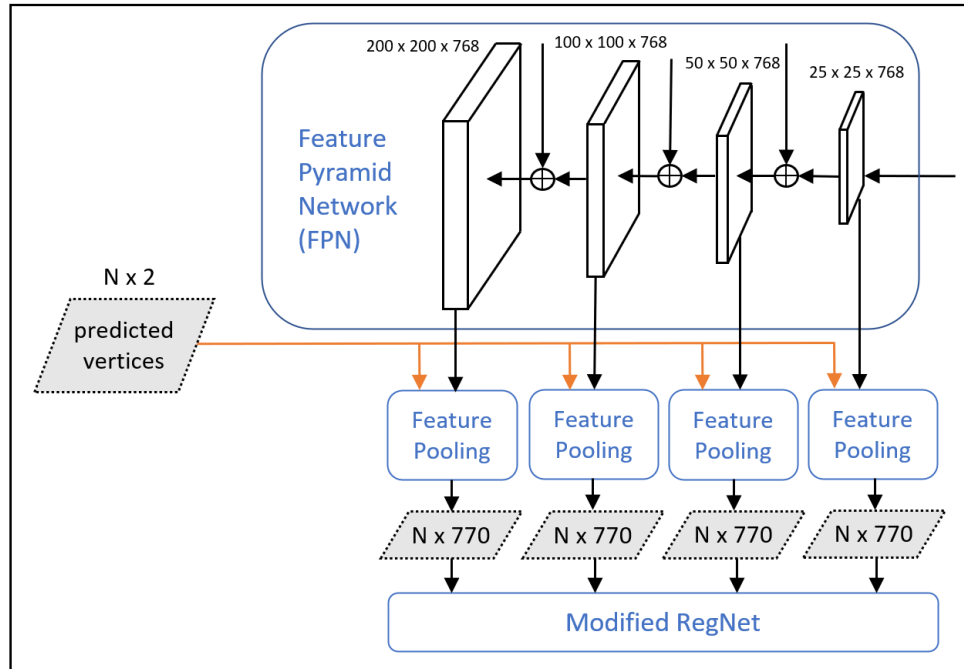


Figure 3.28. Architecture of the modified Feature Pooling module.

Assuming there are N predicted polygon vertices (i.e., a tensor of size $N \times 2$), and four levels of feature maps in the FPN with dimensions of 768. At each level of the feature maps, the modified Feature Pooling module extracts a set of feature vectors at the positions of the set of polygon vertices. Then, the polygon vertices (i.e., size $N \times 2$) are concatenated with their corresponding feature vectors at each level (i.e., size $N \times 768$), resulting in four feature representations of size $N \times (768 + 2) = N \times 770$. The four feature representations consist of semantically strong features at four different scales, which are fed concurrently into the modified BRegNet (i.e., section 3.2.4), resulting in an improved building polygon optimization.

3.2.4 Modified BRegNet

The BRegNet is widened and deepened to effectively utilize the multi-scale feature representations outputted from the modified Feature Pooling module. The BRegNet, as described in section 4.1.1.4, takes a single input which is a concatenation of the predicted polygon vertices with their corresponding feature vectors extracted from a single-scaled feature map. However, the improved BRegNet takes four inputs consisting of concatenations of predicted polygon vertices with their corresponding feature vectors extracted from the four levels of the FPN.

As shown in Figure 3.29, to enable multi-scale inputs, the BRegNet is first widened by adding three more duplicated paths of feature transformation and global pooling layers (i.e., as shown in the blue-dotted box). The outputs of the four duplicated paths (i.e., size $N \times 1088$) consist of both local and global information of the multi-scale feature representation inputs. These outputs are concatenated into one tensor of size $n \times 4352$. Next, to reduce the large dimension size of the aggregated feature, the BRegNet is deepened by adding two more feature transformation layers with output sizes of $n \times 2048$ and $n \times 1024$. In this way, the dimension of the feature is reduced more gradually with less information losses. The modified BRegNet utilizes the semantically

strong multi-scale feature representations to generate offsets of more precisely regularized building polygon vertices.

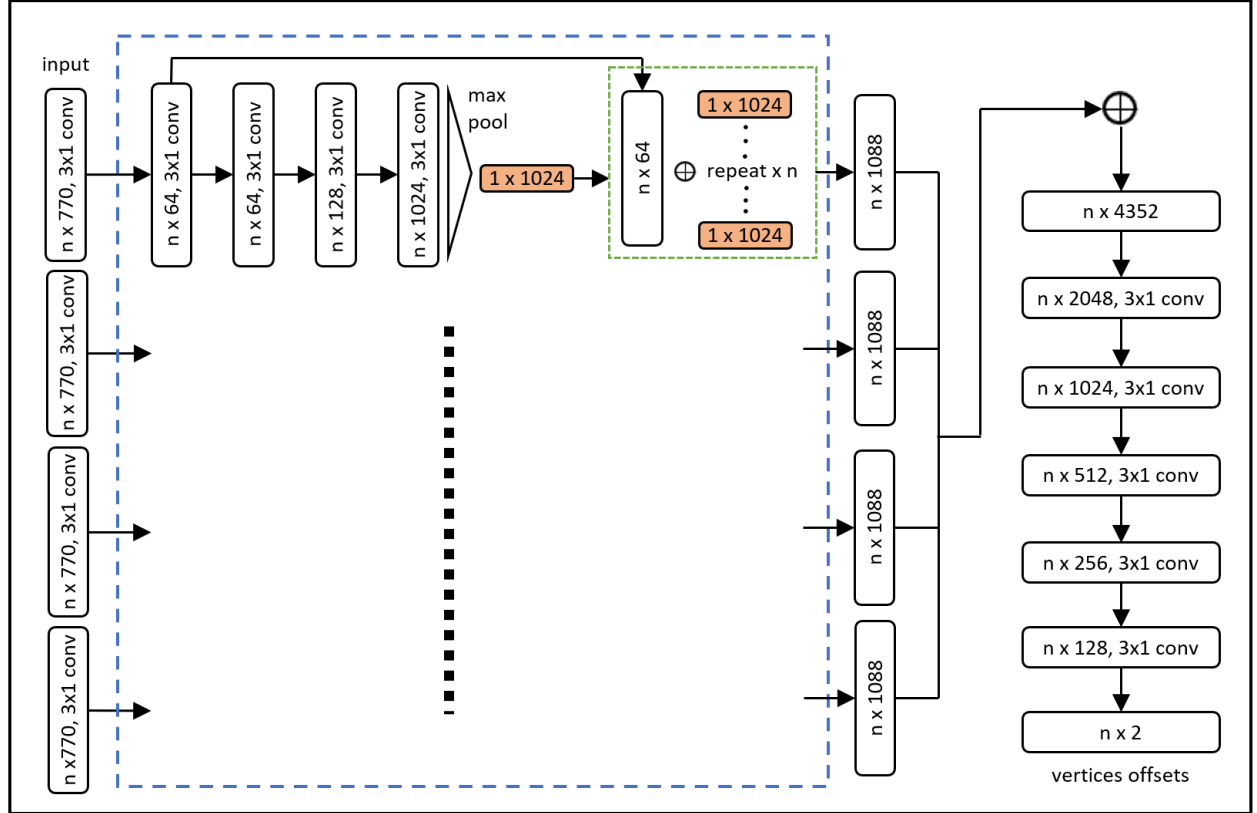


Figure 3.29. The architecture of the modified BRegNet.

CHAPTER 4: IMPLEMENTATION OF THE PROPOSED METHOD AND EXPERIMENTAL STUDIES

To determine the effectiveness of the proposed building segmentation and extraction methods, as well as the reasons behind the model selections, in this chapter, a series of comparison studies are conducted, and complete test results are presented and discussed. First, section 4.1 presents the general settings for the experiments conducted in the rest of the chapter. Next, section 4.2 presents comparison experiments, which explain the selection reasons behind the baseline segmentation and backbone models, and ablation experiments, which show the advantage of the upgraded Mask R-CNN model. Then, in Section 4.3, the proposed model is compared with other popular end-to-end deep-learning-based building extraction models; ablation studies are done to demonstrate the effectiveness of the modified model components; in addition, other design choices of the proposed model are presented to offer flexibility between the quality of the building extraction result and the memory consumption of the model. Last, Section 4.4 visualizes and discusses the test results of the improved Mask R-CNN and PolygonCNN models.

4.1 Experimental Settings

The various settings of the experiments are revealed in this section, including the dataset used, the model, hardware and software settings, and the methods used to evaluate the performances of the models.

4.1.1 Dataset

Name	Pan-Sharpended	Spatial Resolution	Spectral Resolution
SpaceNet 2 Building Detection Dataset	True	$0.3m \times 0.3m$	RGB channels
Image Size	Training Images	Validation Images	Testing Images
650×650 pixels	3500	500	2500

Table 4.1. Properties of the SpaceNet 2 Building Detection Dataset.

This thesis uses the Round 2 dataset of the SpaceNet Building Detection Challenge to evaluate the performances of both the building segmentation and extraction methods. The dataset provides satellite images of four urban cities, including Las Vegas, Paris, Shanghai, and Khartoum, containing over 302,701 building footprints. Table 4.1 shows a list of the dataset properties. All images are pan-sharpened with a spatial resolution of $0.3m \times 0.3m$ per pixel; the spectral resolution consists of the RGB (i.e., red, green, blue) true-color channels, and the image size is 650×650 pixels. In addition, a large portion of the images of each city provide manually labeled building footprint outlines in vector formats contained in GeoJSON files. To train the networks with a reasonable sized dataset, 4000 labelled images are chosen among all the images provided in the SpaceNet2 Building Detection dataset. They are split into 3500 training images for training the deep-learning models and 500 validation images for validating the models during training. In addition, 2500 non-labelled images are chosen for testing the trained models. To ensure the robustness of the trained model at segmenting buildings in various challenging conditions, the chosen training and validation images contain the following characteristics:

- Buildings are in vastly different scales and orientations.
- Buildings are highly diversified in shapes.
- Containing building with similar hues to the background.
- Containing building-like objects such as trucks and large containers.

Figure 4.1.(a) shows examples of the dataset images. First, the vector building labels used to train and validate the building extraction models are visualized in Figure 4.1.(b). Moreover, the vector-format building labels are transformed into raster-format building masks to train and validate the building segmentation models, as illustrated in Figure 4.1.(c). It is noted that for the training of Mask R-CNN model, each mask corresponds to only one building, containing 0's and 1's representing the background and foreground. Thus, for an image containing multiple buildings, multiple masks are created. However, for visualization purposes, the masks that belong to one image are stacked into one segmentation image and shown in different colours.

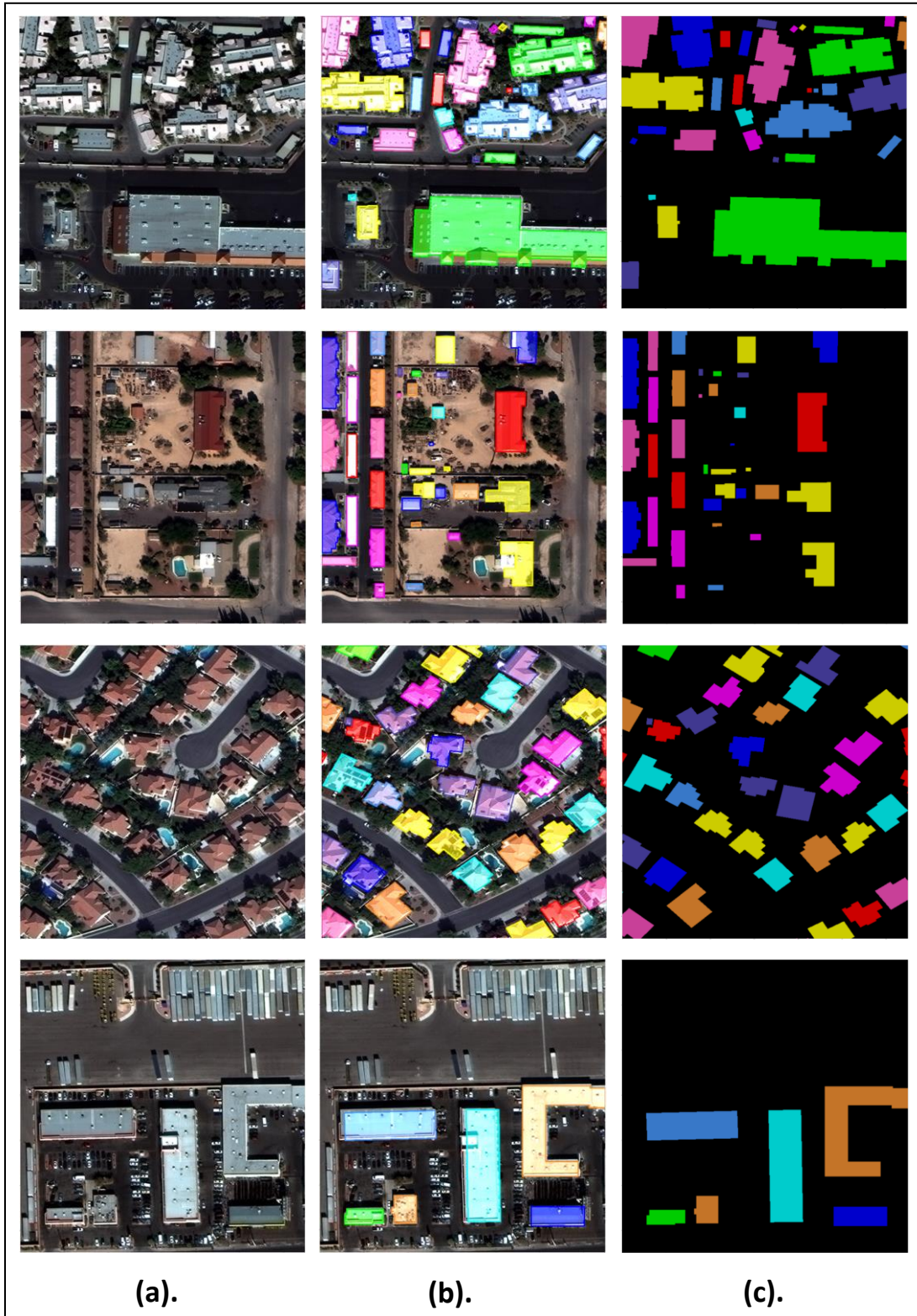


Figure 4.1. Examples of the images and the corresponding building polygon and mask labels of the SpaceNet 2 Building Detection Dataset. (a) The dataset images; (b) the vector-format building outlines, overlaid on the images; (c) the pixel-format building masks, consisting of 0's and 1's, converted from the vector labels.

4.1.2 Implementation Details

To ensure experimental fairness, the consistency of the hardware, software, and model parameter settings are maintained throughout all experiments, and are described as follows.

First, all building segmentation and extraction models in the experiments are implemented and tested in Pytorch (Steiner et al., 2019) using the Python language on a 64-bit Ubuntu system equipped with three NVIDIA TITAN X GPUs. Moreover, the following network settings are shared among all models in the experiments:

- **Learning rate:** the learning rate scheduler named ‘ReduceLROnplateau’ in the Pytorch library is adopted with a starting learning rate of 0.1, reduce factor of 0.2, and patience of 10. This scheduler reads the validation accuracy of the model after each epoch of training, and if no improvement is seen for 10 epochs, the learning rate is reduced by a factor of 0.2.
- **Optimizer:** the stochastic gradient descent (i.e., SGD) (Paszke et al., 1991) optimizer is used with a weight decay of 0.05 and momentum of 0.9 for the optimization of the models during training.
- **Batch size:** set to 4 for the segmentation networks; set to 1 for the extraction networks.
- **Backbone:** unless otherwise stated, the backbone network of all building segmentation and extraction models, if replaceable, are replaced by ResNet-50 to most fairly comparing the model performances considering the amount of training data.
- **Data Augmentation:** all images are resized from 650×650 pixels to 800×800 pixels to fit the special input requirement of Swin Transformer, which only takes input image with width and height that are divisible by 32. Moreover, all images and the building vector labels are normalized to have a mean of 0 and a standard deviation of 1 to facilitate the

training of models. In addition, all training images and labels are augmented to have a random horizontal and vertical flip chance of 0.5, and a random rotation chance of 0.25 among the rotation angles of 0° , 90° , 180° , and 270° . Such a data augmentation strategy enhances the robustness of the trained models by greatly expanding the diversity of the training dataset.

Besides the shareable settings, all the building polygons generated by the building extraction networks are processed by the Douglas-Peucker (DP) algorithm with a ε of 1 pixel for fair comparisons of the building extraction models. In addition, the scales and ratios of the anchor boxes in all Mask R-CNN models (i.e., including the rotatable Mask R-CNN) are respectively set to $\{16, 32, 64, 128, 256\}$ and $\{0.5, 1, 2\}$ instead of the default values considering the general sizes and shapes of the building footprints in the dataset; the rotation angles of the rotatable anchor boxes in the rotatable Mask R-CNN is set to $\{0, \frac{\pi}{4}\}$. Other non-specified parameters of all models are left with the default values as that are described in the corresponding referenced papers.

4.1.3 Evaluation Methods

4.1.3.1 Evaluation of Building Segmentation Network

To evaluate the performances of the instance segmentation networks (i.e., the Mask R-CNN models), the mean Average Precision (AP) and mean Average Recall (AR) are calculated based on the IoU (Jaccard, 1912) metric. The IoU is calculated between the building segmentations, similar to the one calculated between the bounding boxes described in Section 3.1.2. Specifically, AP and AR are averaged over ten IoU values with thresholds from .50 to 0.95 with steps of 0.05, as shown in Equations (4.1) and (4.2), respectively.

$$AP = \frac{AP_{0.50} + AP_{0.55} + \dots + AP_{0.95}}{10}$$

(4.1)

where $AP_n = \frac{True\ Positive}{True\ Positive + False\ Positive}$ for an IoU threshold of n . It measures the fraction between the number of the correctly detected objects (i.e., predictions with $IoU > n$) and the number of all the detected objects.

$$AR = \frac{AR_{0.50} + AR_{0.55} + \dots + AR_{0.95}}{10}$$

(4.2)

where $AR_n = \frac{True\ Positive}{True\ Positive + False\ Negative}$ for an IoU threshold of n . It measures the fraction between the number of the correctly detected objects and the total number of the ground truth objects.

In addition, to measure the performance of the Mask R-CNN models in segmenting buildings in vastly different sizes, the AP and AR of building masks in small, medium, and large sizes are calculated (i.e., $AP_{(small, medium, large)}$, $AR_{(small, medium, large)}$). Small represents a mask smaller than 32^2 pixels, medium represents a mask between 32^2 and 96^2 pixels, and large represents a mask larger than 96^2 pixels.

However, an issue with the above-mentioned object-based APs and ARs is that they cannot apply to the semantic building segmentation networks such as U-Net and PSPNet. This is because the segmentation results of such networks are single binary maps, which only show each pixel's class

information (i.e., building or background) while the object information is missing. On the other hand, the region-based CNNs (e.g., Mask R-CNN) directly predict a mask for each building object in the image. Although some boundary tracing algorithms can be applied to the binary map to trace the building outlines, individual building instances in those tightly clustered buildings cannot be determined. Therefore, a pixel-based IoU (i.e., IoU between the predicted binary maps and the ground truth binary maps) is often used as the metric for those semantic segmentation models. To fairly compare the performances of the semantic segmentation networks and the Mask R-CNN, the instance mask predictions of Mask R-CNN are stacked into binary maps (i.e., as shown in Figure 5.1.(c)) in order to apply the pixel-based IoU metric.

4.1.3.2 Evaluation of Building Extraction Network

To evaluate the performances of the building extraction networks, the same AP , AR , $AP_{(small, medium, large)}$, $AR_{(small, medium, large)}$ in the previous section are applied to measure the quality of the building polygons. In addition, the F1-score metric used by the SpaceNet 2 Building Detection Challenge is also adopted. F1-score is a harmonic average of the polygon-based AP and AR, which gives equal weights to both, and is defined as:

$$F1 = \frac{2 \times AP \times AR}{AP + AR} \quad (4.3)$$

where the F1 is between 0 and 1; larger numbers correspond to better scores.

4.2 Experiments and Discussions of The Upgraded Mask R-CNN model

This section presents various experiments to demonstrate the selection reasons behind the baseline segmentation and backbone models and the advantage of the upgraded Mask R-CNN model.

First, the Mask R-CNN model is compared with several famous image segmentation networks from different application domains to demonstrate its superiority in segmenting building footprints from satellite images. Moreover, to find the suitable backbone network for the building segmentation task, the performances of some popular backbone models have been compared on the baseline Mask R-CNN model. Finally, ablation studies are performed on the Mask R-CNN model by integrating the Swin Transformer backbone, the rotatable bounding box scheme, and the FPN module.

4.2.1 Selection of The Baseline Building Segmentation Model

Since the quality of the extracted building footprint polygon is largely dependent on the quality of the building segmentation, seeking an optimal building segmentation method is the key to developing a state-of-the-art building extraction method. Therefore, several famous baseline image segmentation deep neural networks from different application domains are selected, shown as follows:

- PSPNet (Zhao et al., 2017)
- DeepLabV3+ (Chen et al., 2018)
- UPerNet (Xiao et al., 2018)
- FastFCN (Wu et al., 2019)
- HRNet (Wang et al., 2020)
- Mask R-CNN (He et al., 2017)

The ResNet-50 backbone is equipped on all the models for a fair comparison. The models are trained on the SpaceNet 2 Building Detection dataset and evaluated using the pixel-based IoU metric. As shown in the evaluation results in table 4.2, the bounding-box-based detector Mask R-

CNN significantly outperforms all encoder-decoder segmentation networks with an IoU of 0.873; DeepLabV3+ and HRNet have lower IoUs of 0.858 and 0.859, respectively; both Fast FCN and PSPNet have the second lowest IoU of 0.844, while UPerNet has the lowest IoU of 0.842.

Model	IoU
UPerNet	0.842
PSPNet	0.844
FastFCN	0.844
DeepLabV3+	0.858
HRNet	0.859
Mask R-CNN	0.883

Table 4.2. Performances of the chosen baseline image segmentation models.








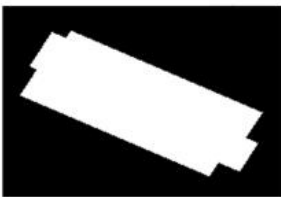





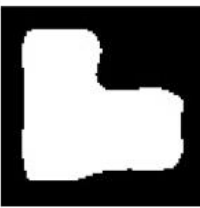

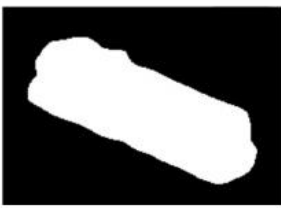



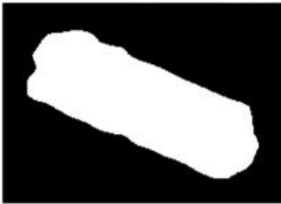











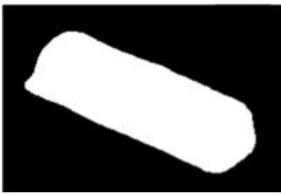
Image				
Ground Truth				
UPerNet				
PSPNet				
FastFCN				
DeepLabV3				
HRNet				
Mask R-CNN				

Figure 4.2. Building segmentation results of the chosen baseline image segmentation models.

Figure 4.2 shows examples of the building segmentation results of the models. A significant finding from the results is that for the encoder-decoder networks, mispredicted pixels tend to occur on buildings with all orientations; in contrast, for the two-stage detector Mask R-CNN, the segmentations on the non-rotated buildings (i.e., building edges are either vertical or parallel to the image edges) tend to have significantly more regularized shapes, whereas the ones on those rotated buildings (i.e., building edges are neither vertical nor parallel to the image edges) are similar to the results of the encoder-decoder networks.

The result indicates that the non-rotatable bounding boxes of Mask R-CNN likely provide regularization effects to the segmentations of non-rotated buildings, as such buildings usually have a large fraction of edges that lean tightly against the bounding boxes. Such a regularization effect contributes significantly to the overall high segmentation accuracy of Mask R-CNN, making the model specifically suitable for segmenting building footprints from satellite images. However, despite the bounding boxes' help, many buildings in satellite images have randomly rotated orientations, which leaves sizeable empty background spaces between the buildings and their bounding boxes, causing mispredicted pixels.

4.2.2 Selection of The Backbone Network

The ResNet-50 backbone in the baseline Mask R-CNN model can potentially be replaced for improved performance. To determine the optimal backbone network for building segmentation, the following popular backbone networks are selected and evaluated on the Mask R-CNN model:

- ResNet-50 and ResNet-101 (He et al., 2016)
- DenseNet-121 and DenseNet-161 (Huang et al., 2017)
- VGG-16 (Simonyan et al., 2014)

- HRNet (Wang et al., 2020)
- MobileNetV3 (Koonce, 2021)
- Swin Transformer (Liu et al., 2021)

Since all the backbones are mounted and evaluated on the Mask R-CNN model, the AP and AR metrics as described in Section 4.1.3.1 are adopted. A comparison of the evaluation results is shown in table 4.3.

Backbone	<u>AP</u>	AP_S	AP_M	AP_L	<u>AR</u>	AR_S	AR_M	AR_L
MobileNetV3	0.437	0.222	0.521	0.524	0.459	0.276	0.545	0.536
VGG-16	0.461	0.245	0.554	0.569	0.498	0.293	0.585	0.580
DenseNet121	0.519	0.258	0.624	0.636	0.565	0.327	0.669	0.650
DenseNet161	0.532	0.263	0.635	0.632	0.578	0.326	0.672	0.661
ResNet50	0.554	0.277	0.651	0.624	0.602	0.354	0.703	0.679
ResNet101	0.555	0.271	0.653	0.649	0.602	0.349	0.704	0.694
HRNet	0.560	0.282	0.658	0.665	0.609	0.358	0.711	0.707
Swin Transformer	0.569	0.279	0.666	0.676	0.614	0.353	0.718	0.729

Table 4.3. Performances of the chosen backbone networks on Mask R-CNN.

As shown in Table 4.3, unsurprisingly, the baseline Mask R-CNN model with the MobileNetV3 backbone has the lowest AP and AR (i.e., 0.437 and 0.459, respectively) among all models since MobileNetV3 is a lightweight model designed explicitly for fast inference speed, which trades off for lower and accuracies and recalls. In addition, the VGG-16 backbone model shows an improved

performance compared to the MobileNetV3 backbone model with an AP and AR of 0.461 and 0.498, respectively. Furthermore, the DenseNet121 backbone model has an AP and AR of 0.519 and 0.565, respectively, while its heavier version model (i.e., DenseNet121) shows slightly improved AP and AR of 0.532 and 0.578, respectively. The improvements are likely due to the model's more considerable depth and the sacrifice to more expensive computation. Compared to the DenseNet models, the two ResNet models have shown considerably better performances. Nevertheless, the heavier version model (i.e., ResNet101) does not show a significantly improved performance compared to the lighter version model (i.e., ResNet50)—the ResNet50 backbone model has an AP and AR of 0.554 and 0.602, respectively, while the ResNet101 backbone model has an AP and AR of 0.555 and 0.602, respectively. Such a result is likely due to the small size of the training dataset, which fails to leverage the full power of the large backbone model. Surprisingly, the HRNet backbone model outperforms all the ResNet and DenseNet models with an AP and AR of 0.506 and 0.551, respectively; also, it has the highest AP_s and AR_s among all models, which indicates that the rich, high-resolution features and the multi-scale fusion of HRNet are specifically effective for the feature extraction of small-sized building objects from satellite images. Lastly, the Swin Transformer model obtains the highest AP, AR, AP_M , AR_M , AP_L , AR_L among all models while maintaining promising AP_s and AR_s that are very close to the ones of the HRNet model. Such results indicate that the novel patch-merging and shifted-window-based self-attention operations of Swin Transformer are specifically suitable for multi-scale building footprint segmentation. Due to its impressive overall performance, the Swin Transformer-based Mask R-CNN is chosen as the baseline segmentation network to replace the PSPNet in the original PolygonCNN.

4.2.3 Ablation Studies of The Upgraded Mask R-CNN Model

As discussed in Section 4.2.1, the non-rotatable bounding box of Mask R-CNN provides regularization effects to the segmentations of the non-rotated buildings; however, it misses the opportunity to exploit the rotated buildings. Thus, this thesis enhances the Mask R-CNN by integrating the rotatable bounding box scheme. In addition, to further improve the model's ability to segment building footprints in vastly different scales, the FPN module is integrated. Table 4.4 compares the performances of different Mask R-CNN models with incrementally added components.

Backbone Network		Rotatable Bounding Box	FPN	\underline{AP}	AP_S	AP_M	AP_L	\underline{AR}	AR_S	AR_M	AR_L
ResNet-50 (Baseline)	Swin Transformer										
✗				0.554	0.277	0.651	0.624	0.602	0.354	0.703	0.679
	✗			0.569	0.279	0.666	0.676	0.614	0.353	0.718	0.729
	✗	✗		0.619	0.460	0.677	0.722	0.664	0.502	0.747	0.788
	✗	✗	✗	0.670	0.498	0.733	0.782	0.699	0.528	0.786	0.837

Table 4.4. Performances of Mask R-CNN models with different components.

As shown in Table 4.4, the baseline Mask R-CNN model with the ResNet-50 backbone has an AP and AR of 0.554 and 0.602, respectively. When replacing the ResNet-50 backbone with the Swin Transformer backbone, the AP and AR improve slightly to 0.569 and 0.614, respectively. Furthermore, when integrating the rotatable bounding box with the Swin-Transformer-based Mask R-CNN, the AP and AR increase significantly to 0.619 and 0.664, respectively. It is noted that most of the increases are contributed by AP_S and AR_S , meaning that integrating the rotatable bounding box drastically improves the segmentation of small-sized buildings. Lastly, further

integrating the FPN module largely improves the model's ability to segment buildings in vastly different sizes, while the AP and AR further improve to 0.670 and 0.699, respectively.

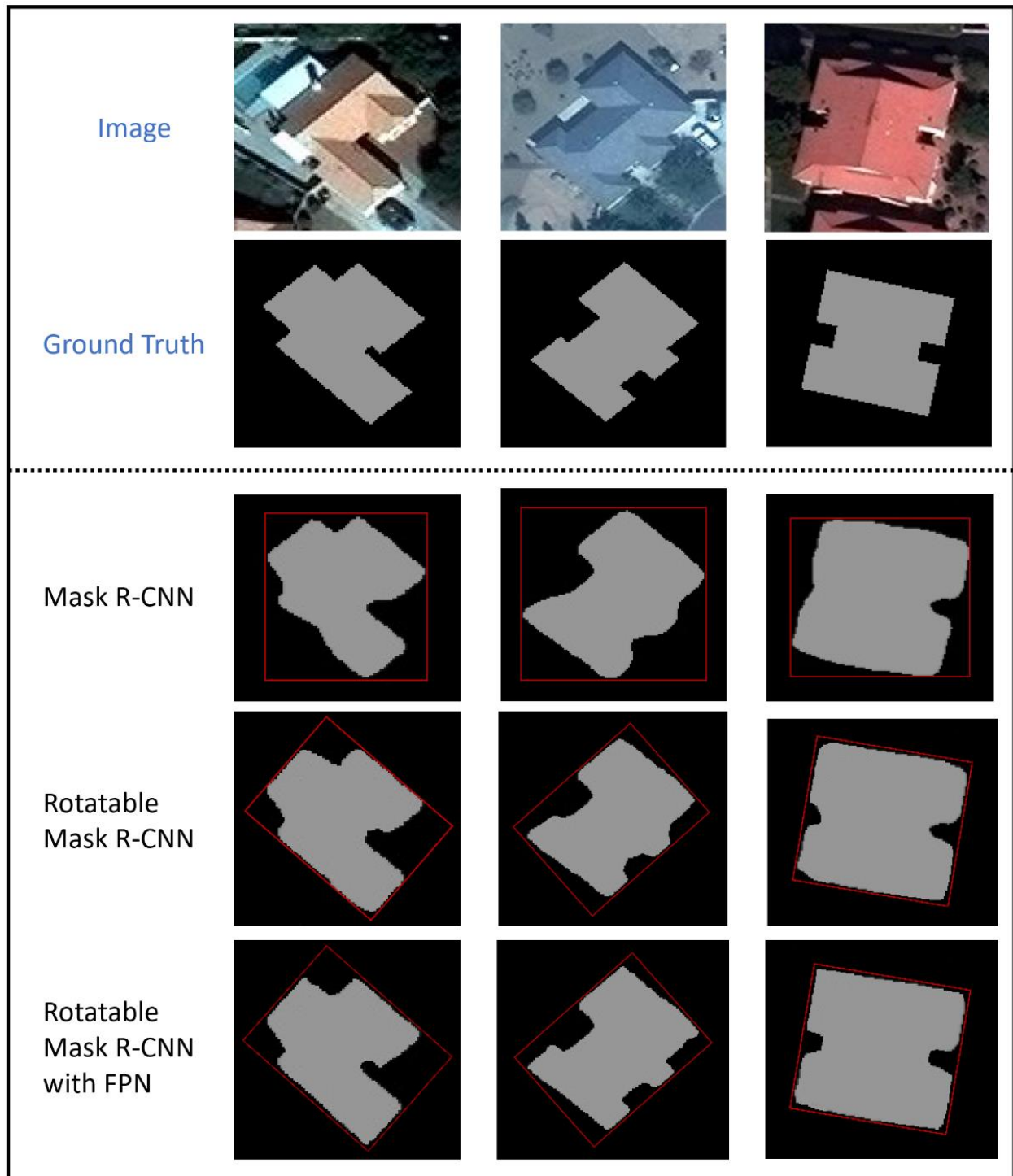


Figure 4.3. Comparison of the building segmentation results of the Mask R-CNN models with incrementally added components.

Figure 4.3 illustrates examples of the buildings segmentation results of the three models (i.e., Swin-Transformer-based Mask R-CNN, rotatable Swin-Transformer-based Mask R-CNN, and rotatable Swin-Transformer-based Mask R-CNN with FPN). As shown in Figure 5.3.(a), large areas of empty background exist between those rotated buildings and their bounding boxes, which result in some significantly irregular-shaped segmentations. With the adoption of rotatable bounding boxes, as shown in Figure 5.3.(b), the building edges that lean tightly against the rotatable bounding boxes are effectively being regularized. However, empty background spaces within the rotatable bounding boxes still exist for those complex-shaped buildings, which results in mispredicted pixels. To improve such cases, as shown in Figure 5.3.(c), the FPN module is adopted, reducing mispredicted pixels along the building edges.

4.3 Experiments and Discussions of The Proposed Building Extraction Model

First, an ablation experiment is performed on the proposed building extraction model by gradually integrating the improved Mask R-CNN model and the modified Feature Pooling module and BregNet to show the effectiveness of these modules. Second, to demonstrate the advantage of the proposed building extraction model, it is compared with several state-of-the-art end-to-end deep-learning-based methods. Third, other design choices of the proposed building extraction model are presented, and the performances of the different designs are compared and discussed. Last, the test results of the proposed building extraction model are visualized and discussed.

4.3.1 Ablation Studies of The Proposed Building Extraction Model

To enable the precise extraction of building polygons in vastly different scales and orientations from satellite images, this thesis replaces the segmentation module in the PolygonCNN with the proposed improved Mask R-CNN model. In addition, to adapt the multi-scale feature maps outputted from the FPN module of the improved Mask R-CNN, the Feature Pooling module and

the BRegNet in the original PolygonCNN are modified such that they accept multi-scale feature maps instead a single-scale feature map. To assess the effectiveness of each upgraded component, an ablation study is done by incrementally integrating the improved Mask R-CNN and the modified feature pooling and BRegNet with the original PolygonCNN. Besides the polygon-based APs and ARs metrics, the F1-score (i.e., as described in Section 4.1.3.2) is adopted to measure the overall quality of the polygon predictions. Table 4.5 shows a comparison of the evaluation results.

Method	Segmentation Network		Modified Feature Pooling and RegNet	<u>F1</u>	<u>AP</u>	<u>AP_S</u>	<u>AP_M</u>	<u>AP_L</u>	<u>AR</u>	<u>AR_S</u>	<u>AR_M</u>	<u>AR_L</u>
	PSPNet	Improved Mask R-CNN										
PolygonCNN (Chen et al., 2020)				0.474	0.455	0.226	0.533	0.508	0.495	0.289	0.575	0.553
				0.511	0.501	0.372	0.546	0.581	0.521	0.394	0.585	0.622
Improved PolygonCNN				0.545	0.529	0.392	0.584	0.601	0.563	0.416	0.631	0.671

Table 4.5. Performances of PolygonCNN models with different components.

As shown in Table 4.5, the baseline PolygonCNN model with the PSPNet segmentation network has a 0.474 F1-score, 0.455 AP, and 0.495 AR. When replacing the PSPNet with the improved Mask R-CNN (i.e., Swin-Transformer-based rotatable Mask R-CNN integrated with the FPN module), the F1-score, AP and AR improve significantly to 0.511, 0.501, and 0.521, respectively. It is noted that the adoption of the improved Mask R-CNN has dramatically improved AP_S and AR_S (i.e., the extraction quality of small-sized buildings). Furthermore, after adopting the modified Feature Pooling module and BRegNet, the F1-score, AP and AR are further improved to 0.545, 0.529, and 0.563, respectively.

The evaluation results show that both the adoption of the improved Mask R-CNN and the modified Feature Pooling module and BRegNet have made drastic improvements on the baseline

PolygonCNN’s ability to extract building polygons in vastly different sizes, especially for the small-sized ones.

4.3.2 Comparison Study of The Proposed Building Extraction Model

Next, to demonstrate the advantage of the proposed building extraction model, it is compared with three other popular end-to-end deep-learning-based methods, listed as follows:

- Polygon-RNN (Castrejon et al., 2017)
- DARNet (Cheng et al., 2019)
- PolyMapper (Li et al., 2019)

Table 4.6 shows a comparison of the evaluation results. The results show that PolyMapper has the highest F1-score of 0.559, while the proposed model has a similar F1-score of 0.545; DARNet has the third lowest F1-score of 0.512; the original PolygonCNN has a significantly lower F1-score of 0.474, while Polygon-RNN has the worst F1-score of 0.465.

Method	<u>F1</u>	<u>AP</u>	AP_S	AP_M	AP_L	<u>AR</u>	AR_S	AR_M	AR_L
Polygon-RNN	0.465	0.458	0.175	0.544	0.587	0.473	0.271	0.592	0.576
DARNet	0.512	0.504	0.203	0.613	0.599	0.521	0.298	0.635	0.652
PolyMapper	0.559	0.541	0.237	0.632	0.647	0.578	0.320	0.671	0.689
PolygonCNN	0.474	0.455	0.226	0.533	0.508	0.495	0.289	0.575	0.553
Proposed Model	0.545	0.529	0.392	0.584	0.601	0.563	0.416	0.631	0.671

Table 4.6. Performances of the chosen building extraction models.

To better understand this ranking, Figure 4.4 visually compares the building polygons extracted by the selected building extraction models and the proposed model. As shown in the figure,

PolyMapper tends to make accurate predictions on the building parts with simple structures; however, it fails to model buildings with more complex shapes, which may be caused by the insufficient capability of its segmentation module. In addition, DARNet is capable of capturing the main structure of various types of buildings; nevertheless, the predicted polygons show a lack of regularization and simplification (i.e., they fail to capture sharp corners and always contain redundant vertices). Furthermore, Polygon-RNN can model the rough structure of simple building shapes but fails to capture details of more complicated parts. This is likely also caused by the lack of segmentation capability of its segmentation module. Lastly, the proposed model, benefited from its Swin Transformer-based rotatable Mask R-CNN, the multiscale feature maps of the FPN, and the enhanced Feature Pooling module and BRegNet, is capable of generating well-regularized and simplified polygons for various types of buildings in different orientations and sizes.

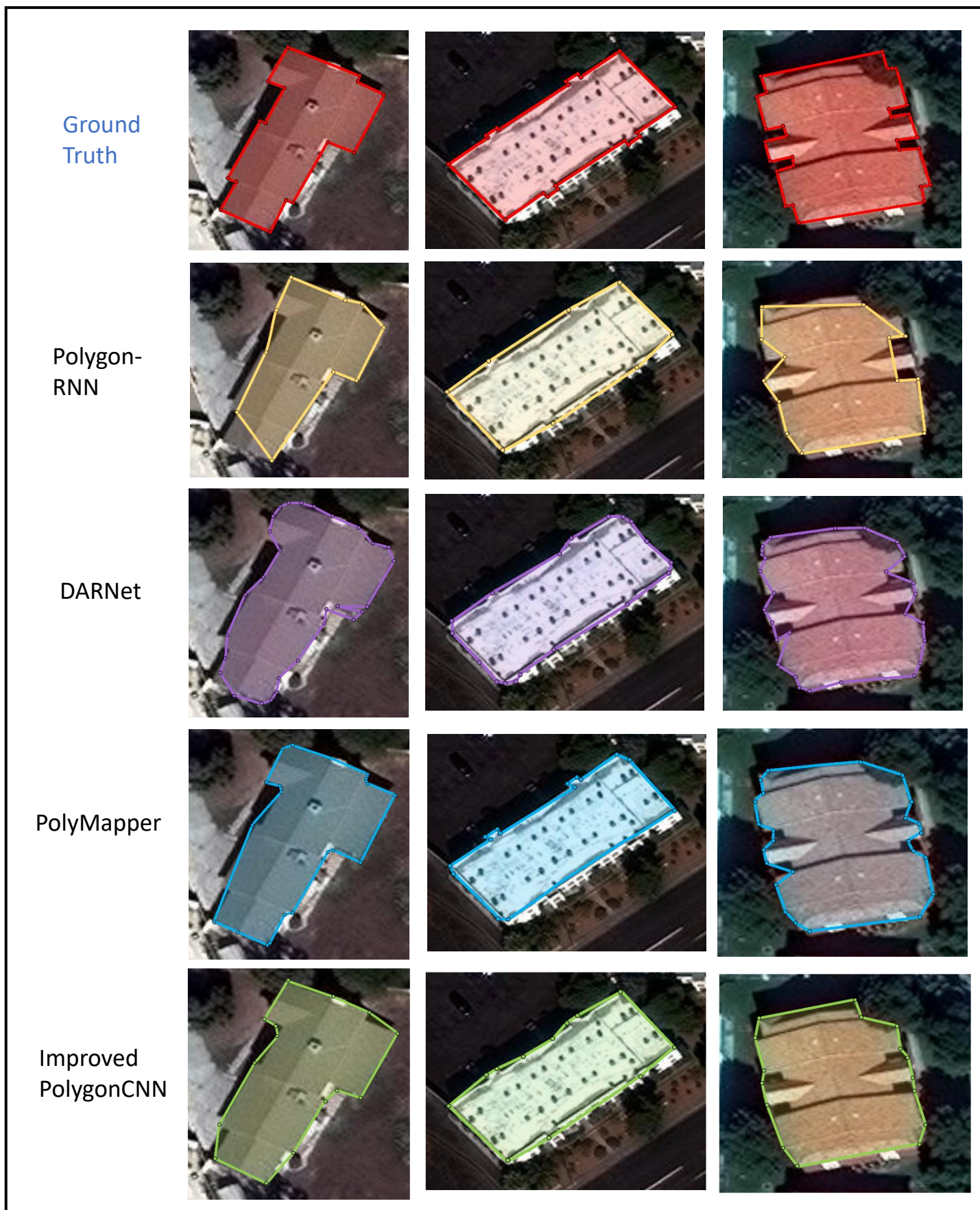


Figure 4.4. Building extraction results of the chosen building extraction models.

4.3.3 Other Design Choices for The Proposed Building Extraction Model

Along the way of developing the final version of the proposed building extraction model, other architectural designs have been experimented, while two of them have shown promising performances. This sub-section presents the architectures of the two other models, then compares and discusses the performances of all three versions of models.

4.3.3.1 Architecture #1

The Mask R-CNN model exploits the multi-scale feature maps of the FPN module by selecting a feature map of appropriate scale from the feature pyramid based on the width and height of the bounding box proposals predicted by the RPN. Similarly, in order to take advantage of the FPN module of the improved Mask R-CNN to optimize the performance of the PolygonCNN model, my first attempt was also to select a feature map of appropriate scale from the feature pyramid

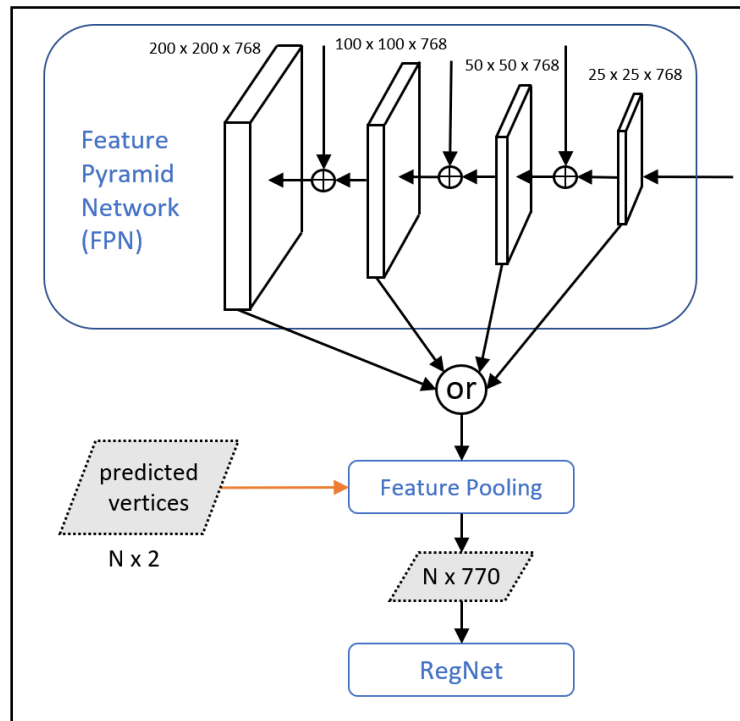


Figure 4.5. Architecture #1. Adaptive feature map selection for the Feature Pooling.

based on the width and height of the building segmentations outputted from the improved Mask R-CNN, then use this single-scale feature map for the feature pooling.

Figure 4.5 illustrates the main components of the first architecture. The “predicted vertices” represent the vertices that are traced and resampled from the building mask predicted by the Mask R-CNN model (i.e., as described in the architecture of the proposed model, Figure 3.19). Based on the scale of the bound box of the building mask, a corresponding scale (i.e., calculated using Equation (3.5)) of the feature map is selected from the feature pyramid. Then, the original feature pooling module and BRegNet used in the original PolygonCNN are applied to the selected single-scale feature map.

4.3.3.2 Architecture #2

Instead of selecting a single feature map of appropriate scale from the feature pyramid for the feature pooling, the second architecture involves pooling all scales of feature maps from the feature pyramid. As shown in Figure 4.6, first, using the coordinates of the traced and resampled vertices, feature vectors are pooled from each level of the feature maps. Next, instead of feeding the multi-scale feature vectors directly into the modified version of BRegNet (i.e., see Section 3.2.4), all feature vectors are concatenated into one large feature vector, which is then fed into the original BRegNet for the prediction of regularized building vertices.

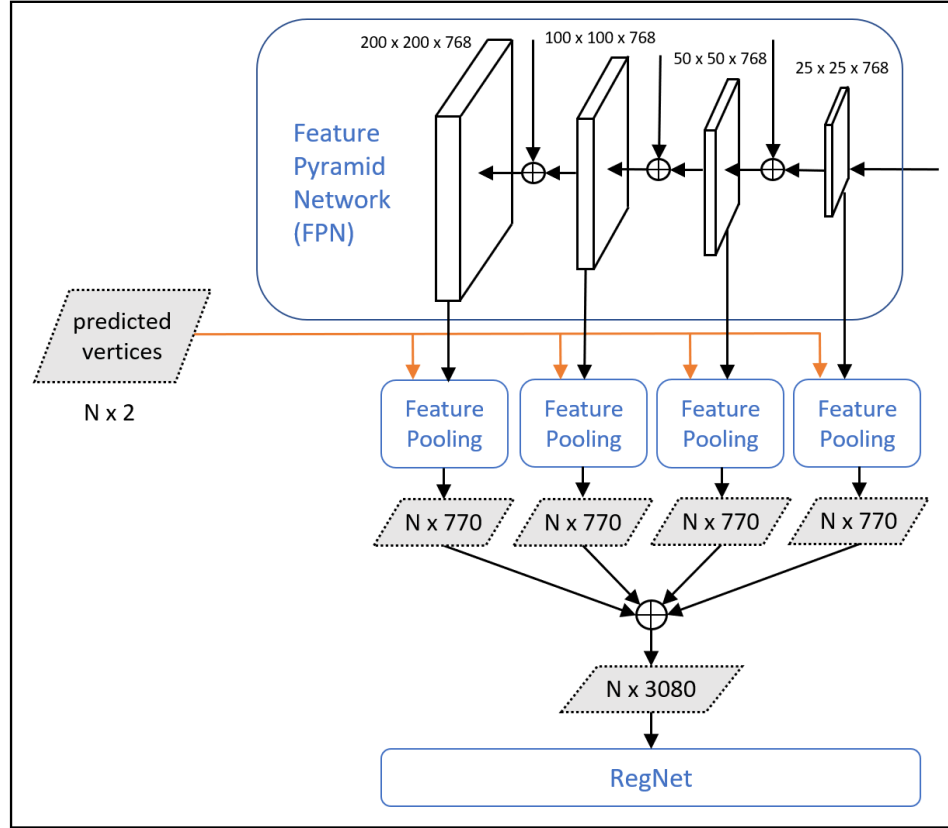


Figure 4.6. Architecture #2. Multi-scale feature pooling and feature vector concatenation.

4.3.3.3 Comparison Study of The Different Design Choices

The above-mentioned two experimental model architectures (i.e., the adaptive feature map selection, and multi-scale feature pooling with concatenated feature vectors), as well as the final version of the proposed model (i.e., multi-scale feature pooling with modified BRegNet), are examined, and the evaluation results are compared in Table 4.7.

As shown in Table 4.7, the first model architecture with the improved Mask R-CNN and the adaptive feature map selection technique achieves an F1-score of 0.522; when replacing the adaptive feature map selection with multi-scale feature pooling and feature vector concatenation, the F1 score of the model increases slightly to 0.530. Such a modification has shown considerable

Version of Improved PolygonCNN	F1	AP	AP _S	AP _M	AP _L	AR	AR _S	AR _M	AR _L
Adaptive Feature Map Selection (#1)	0.522	0.507	0.375	0.559	0.576	0.539	0.398	0.604	0.643
Multi-Scale Feature Pooling + Concatenated Feature Vectors (#2)	0.530	0.514	0.381	0.568	0.584	0.548	0.405	0.614	0.653
Multi-Scale Feature Pooling + Modified RegNet (Final)	0.545	0.529	0.392	0.584	0.601	0.563	0.416	0.631	0.671

Table 4.7. Performances of the two experimental model architectures and the final proposed model.

improvements in the extraction quality of small-sized buildings (i.e., AP_S and AR_S), while maintaining performance in extracting medium and large-sized buildings. Lastly, based on architecture #2, when adopting the modified BRegNet, which directly accepts multi-scale feature vectors, the F1-score is increased significantly to 0.545. The modified BRegNet, compared to the original BRegNet, has effectively improved the extraction qualities of buildings in various sizes.

Despite the higher building extraction quality of the final model architecture, there are significant trade-offs between the extraction quality and memory consumption of the three versions of the models. As shown in Table 4.8, architecture #1 yields a memory consumption of 3.65GB during inference and an inference time of 262.73ms/image; the multi-scale feature pooling and feature vector concatenation of architecture #2 result in slight increases in memory consumption (i.e., 4.22GB) and inference time (i.e., 294.86ms/image), which are generally insignificant during actual

practices; moreover, compared to architecture #2, the modified BRegNet of the final architecture has a dramatic increase in the memory consumption (i.e., 11.31GB) and a considerable increase in the inference time (i.e., 369.44ms/image) mainly due to the widened and deepened layers of the modified BRegNet. As illustrated in Figure 4.12, the Modified BRegNet consists of three additional feature extraction paths that are distributed parallelly with the original path (i.e., marked in blue dotted lines), an extra step to concatenate the extracted features, and two feature down-sampling layers compared to the original BRegNet (i.e., see Figure 3.1.1.4). The additional feature extraction paths have significantly contributed to the overall increase in memory consumption; however, they theoretically do not contribute to the increase in the inference time as they are parallelly distributed. The feature concatenation and the two feature down-sampling layers have

Version of Improved PolygonCNN	Memory (GB)	Inference Time (ms/image)
Adaptive Feature Map Selection (#1)	3.65	262.73
Multi-Scale Feature Pooling + Concatenated Feature Vectors (#2)	4.22	294.86
Multi-Scale Feature Pooling + Modified RegNet (Final)	11.31	369.44

Table 4.8. Memory consumption and inference time of the two experimental model architectures and the final proposed model.

made a minor contribution to memory consumption and a major contribution to inference time as they add depths to the original network.

Since the building extraction quality is the focus of this thesis while the memory consumption and inference time are less considered, the final architecture adopts the model with the multi-scale feature pooling and the modified BRegNet. For practical cases where memory consumption or inference time needs to be considered, architectures #1 and #2 offer flexibility.

4.4 Results and Discussions

Figure 4.7 shows two exemplar building segmentation and extraction results on full-sized images obtained by the proposed improved Mask R-CNN and PolygonCNN. As shown in Figure 4.7.(a), the chosen images are overlaid with ground truth building polygons in various complexity levels and vastly different scales and orientations. Figure 4.7.(b) shows the rotatable bounding boxes and building segmentations obtained by the improved Mask R-CNN. With the rotatable bounding box technique and the FPN module, the improved Mask R-CNN model has been shown to not only grasp the general structure of most building footprints, but also present the small concaves and convexes on most complex-shaped footprints. In addition, as shown in the final vector map obtained by the proposed model in Figure 4.7.(c), the proposed model can effectively re-organize the building vertices that are traced from the irregular-shaped building segmentations into regularized polygons. Surprisingly, such regularization effects are prominent on buildings with detailed concaves and convexes, which is likely due to the use of the multi-scale feature vectors and the modified BRegNet.

(a).



(b).



(c).

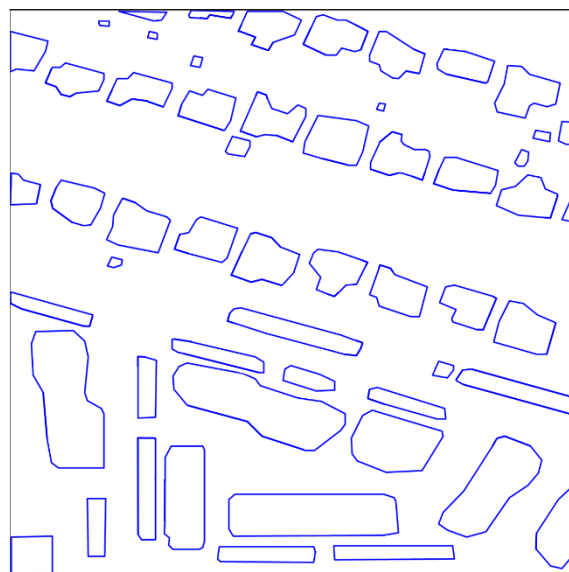
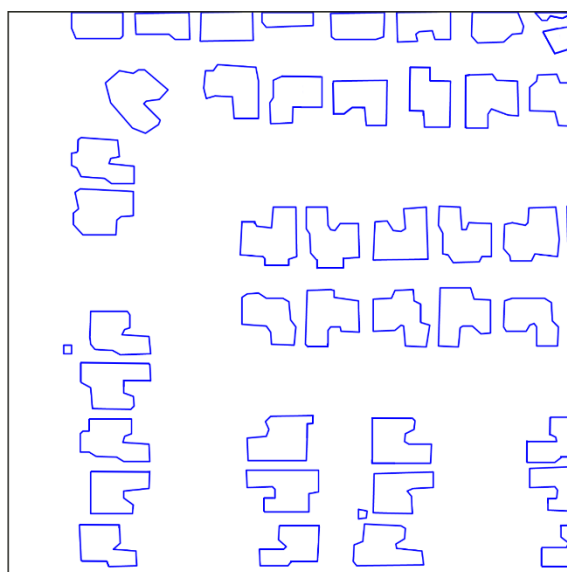


Figure 4.7. Building detection and extraction results from the proposed methods. (a) Input images overlaid with ground truth building polygons; (b) the segmentation results obtained from the improved Mask R-CNN model; (c) the vector building polygons extracted using the proposed model.

Despite the promising extraction quality, issues still exist in the proposed model. A major factor that negatively impacts the extraction quality is the very-complexed building shapes. Examples of mis-extracted buildings are presented in Figure 4.8. As shown in the figure, both buildings (a) and (b) have complex shapes with large numbers of concaves and convexes, varied in size and angle, while building (c) contains a hole at its central location. The extracted polygons for buildings (a) and (b) show that many concaves and convexes are mis-extracted, likely due to the inaccurate segmentations obtained by the improved Mask R-CNN. Although the rotatable bounding box technique of the improved Mask R-CNN can effectively minimize background areas within bounding boxes to regularize the building edges that lean tightly against the bounding boxes, large empty spaces are created by the large concaves and convexes of buildings (a) and (b), causing mispredictions. In addition, as shown in the extracted polygon of building (c), the hole in the building center is missing, which is also caused by the misprediction of the segmentation. Nevertheless, in this case, the false segmentation is likely caused by the small-sized training samples of buildings with holes contained in the training dataset.

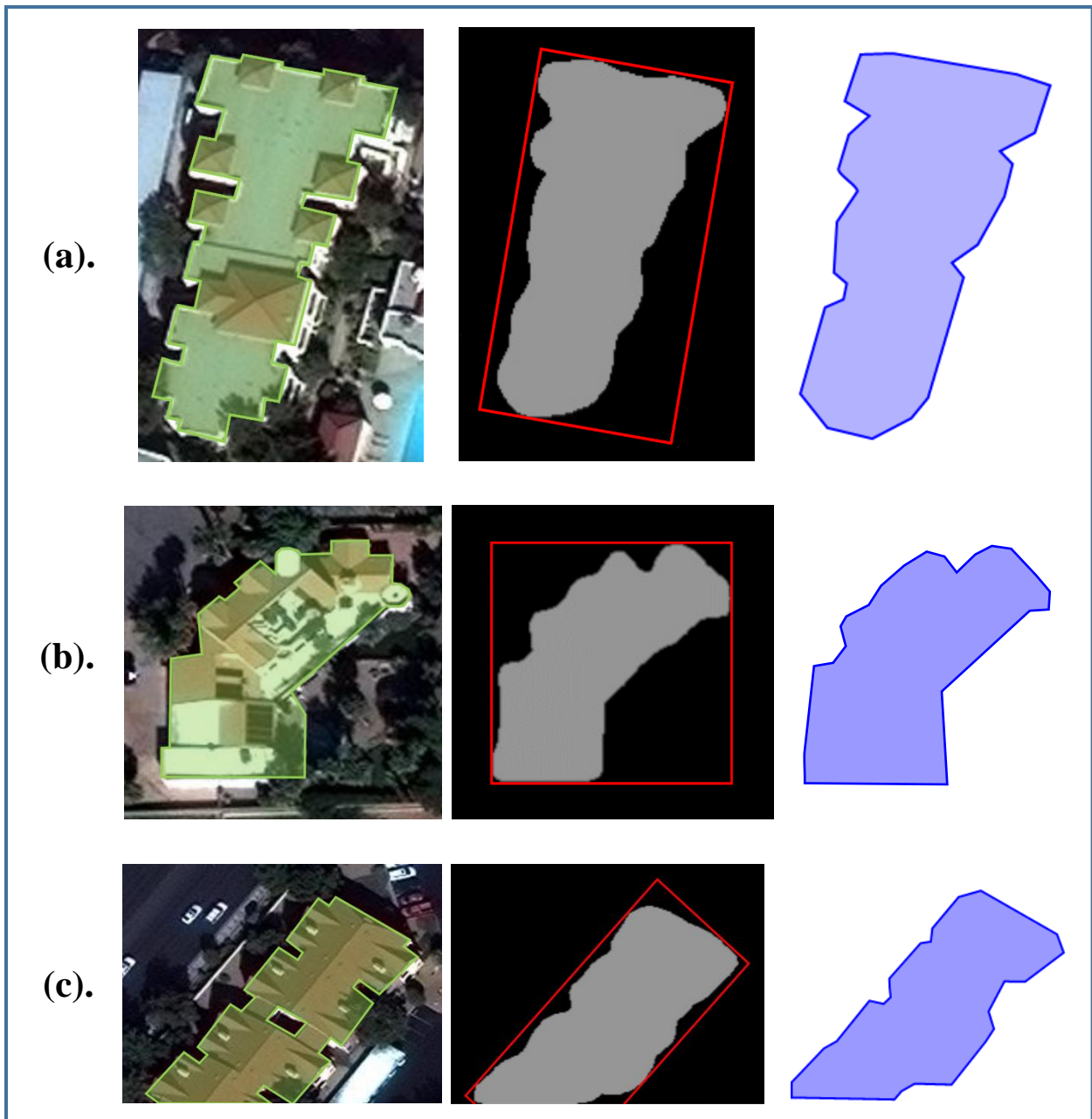


Figure 4.8. Examples of mispredicted buildings. The first column shows ground truth building polygons; the second column shows the corresponding segmentation results obtained from the improved Mask R-CNN model; the third column shows the corresponding vector building polygons extracted using the proposed model.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Conclusions

Automatically extracting high-quality building polygons from satellite and aerial images is crucial for supporting various land use and land cover mapping applications. This thesis introduces an improved end-to-end deep-learning-based building extraction method based on PolygonCNN to accurately extract building footprints in vastly different scales and orientations.

Chapter 1 reviews the background and applications related to the development of automated methods for extracting building footprints from satellite images. In addition, three major issues regarding the subject have been identified: first, current DNNs tend to produce irregular-shaped building segmentations, especially for those rotated buildings; second, the conventional building regularization methods often fail to correctly regularize buildings with non-right-angle-cornered shapes such as triangles or polygons with more than four edges; third, despite that learning-based building regularization methods are highly generalizable for buildings in complexed shapes, they generally show dissatisfactory quality in the regularized building polygons.

Chapter 2 reviews the various conventional and deep-learning-based building segmentation and extraction methods developed over the past decades and the past attempts at utilizing rotatable bounding boxes in DNNs.

Chapter 3 presents the proposed building extraction method. First, encouraged by the irregular building segmentation issue of the popular segmentation DNNs, an upgraded Mask R-CNN model, which is integrated with the rotatable bounding box technique, the Swin Transformer backbone network, and the FPN module, is adopted as the segmentation module of the proposed model. Next, to improve the dissatisfactory building regularization quality of the current DNNs, a

modified Feature Pooling module and the BRegNet are adopted as the regularization module of the proposed model to exploit the multi-scale, pyramidal hierarchy feature maps generated from the FPN module of the improved Mask R-CNN.

Chapter 4 presents various experimental studies to show the effectiveness of the improved Mask R-CNN and PolygonCNN models. Experimental results show that each of the rotatable bounding box, FPN module, and Swin Transformer backbone of the improved Mask R-CNN have significantly contributed to its dramatic increase in the building segmentation quality compared to the baseline Mask R-CNN model. The improved Mask R-CNN model has achieved an AP and AR of 0.670 and 0.699, respectively, on the SpaceNet 2 Building Detection Dataset. The results significantly surpass the building segmentation performances of several other famous image segmentation networks. In addition, experimental results show that the improved Mask R-CNN and the modified Feature Pooling module and BRegNet have significantly increased the building extraction performance of the original PolygonCNN from an F1-score of 0.474 to 0.545. The results are on par with the state-of-the-art end-to-end deep-learning-based building extraction methods. Furthermore, additional architecture design choices for the modified Feature Pooling module and BRegNet are provided, which offer flexibility between the quality of the building extraction result and the memory consumption of the model. Lastly, visualized results show that the improved Mask R-CNN and PolygonCNN models can effectively segment and extract building footprints in vastly different scales and orientations. However, defects are seen in the extraction of very complex buildings, such as the ones with large numbers of concaves and convexes varied in sizes and angle, and the ones that contain holes.

5.2 Limitations and Future Work

Based on the study of this thesis, three areas have been identified as recommendations for future research.

First, despite the promising extraction quality, the memory consumption of the proposed final version of the extraction model is considerably high. Inferencing a single 650×650 image takes about 11.31GB RAM, almost occupying a 12 RAM GPU; during training, the model is distributed onto two GPUs due to the higher memory requirement for the backpropagation process. Such an overly expensive requirement of memory can rise significantly with an increase in the spatial resolution or training/inferencing batch size of the input image, which limits the usages of the model in practices that require very high spatial resolution images, efficient large-scale training/testing, or that have limited computational powers in the hardware. Although lighter version models are presented in Section 4.3.3.3, the reduced memory consumptions are traded off for significantly decreased extraction qualities. Therefore, a major future research direction is to develop lighter-weighted building extraction networks that can be efficiently trained and tested on large-scale, very high spatial resolution images without sacrificing the extraction qualities.

Second, as shown in the visualized results in Section 4.2, a significant factor that negatively impacts the extraction quality of the proposed method is the very complex building type. The rotatable bounding box technique of the improved Mask R-CNN can effectively regularize the building edges that tightly lean against the bounding boxes by minimizing the background areas within the bounding boxes; however, for complex building shapes, many edges are diverged from their bounding boxes, resulting in worse segmentation and extraction qualities. Therefore, another future research direction is to develop methods to precisely extract very complex-shaped building footprints from satellite images.

Lastly, the proposed method in this thesis explicitly targets the extraction of building objects from satellite images; extending the method into other fields would be desired. For example, due to the excellent generalization capability of the deep-learning-based regularization network, the proposed method can potentially be extended to extract many objects with similar geometries to buildings, such as automobiles, traffic signs, or machine parts, from aerial, street-view, or close-range images, respectively. Moreover, the precise extractions of irregular-shaped objects such as water bodies, roads, and vegetation from satellite images are valuable future research directions.

REFERENCES

- Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J. and Ogden, J.M., 1984. Pyramid methods in image processing. *RCA engineer*, 29(6), pp.33-41.
- Alshehhi, R., Marpu, P.R., Woon, W.L. and Dalla Mura, M., 2017. Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130, pp.139-149.
- Aytekin, Ö., Zöngür, U. and Halici, U., 2012. Texture-based airport runway detection. *IEEE Geoscience and Remote Sensing Letters*, 10(3), pp.471-475.
- Badrinarayanan, V., Kendall, A. and Cipolla, R., 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12), pp.2481-2495.
- Belgiu, M. and Drăguț, L., 2014. Comparing supervised and unsupervised multiresolution segmentation approaches for extracting buildings from very high resolution imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 96, pp.67-75.
- Berg, M.D., Kreveld, M.V., Overmars, M. and Schwarzkopf, O., 1997. Computational geometry. In *Computational geometry* (pp. 1-17). Springer, Berlin, Heidelberg.
- Bottou, L., 1991. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nimes*, 91(8), p.12.
- Canny, J., 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), pp.679-698.

- Castrejon, L., Kundu, K., Urtasun, R. and Fidler, S., 2017. Annotating object instances with a polygon-rnn. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5230-5238).
- Chen, J.S., Huertas, A. and Medioni, G., 1987. Fast convolution with Laplacian-of-Gaussian masks. IEEE Transactions on Pattern Analysis and Machine Intelligence, (4), pp.584-590.
- Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV) (pp. 801-818).
- Chen, Q., Wang, L., Waslander, S.L. and Liu, X., 2020. An end-to-end shape modeling framework for vectorized building outline generation from aerial images. ISPRS Journal of Photogrammetry and Remote Sensing, 170, pp.114-126.
- Chen, Q., Wang, L., Wu, Y., Wu, G., Guo, Z. and Waslander, S.L., 2019. TEMPORARY REMOVAL: Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings. ISPRS journal of photogrammetry and remote sensing, 147, pp.42-55.
- Cheng, D., Liao, R., Fidler, S. and Urtasun, R., 2019. Darnet: Deep active ray network for building segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7431-7439).
- Chuang, K.S., Tzeng, H.L., Chen, S., Wu, J. and Chen, T.J., 2006. Fuzzy c-means clustering with spatial information for image segmentation. computerized medical imaging and graphics, 30(1), pp.9-15.

- Cliquet, G. and Baray, J., 2020. Location-based marketing: geomarketing and geolocation. John Wiley & Sons.
- Dai, F.C., Lee, C.F. and Zhang, X.H., 2001. GIS-based geo-environmental evaluation for urban land-use planning: a case study. *Engineering geology*, 61(4), pp.257-271.
- Dalal, N. and Triggs, B., 2005, June. Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). Ieee.
- Douglas, D.H. and Peucker, T.K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2), pp.112-122.
- Felzenszwalb, P.F. and Huttenlocher, D.P., 2004. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2), pp.167-181.
- Girard, N. and Tarabalka, Y., 2018, July. End-to-end learning of polygons for remote sensing image classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 2083-2086). IEEE.
- Glasbey, C.A., 1993. An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical models and image processing*, 55(6), pp.532-537.
- Manno-Kovács, A. and Ok, A.O., 2015. Building detection from monocular VHR images by integrated urban area knowledge. *IEEE Geoscience and Remote Sensing Letters*, 12(10), pp.2140-2144.

- Griffiths, D. and Boehm, J., 2019. Improving public data for building segmentation from Convolutional Neural Networks (CNNs) for fused airborne LiDAR and image data using active contours. *ISPRS journal of photogrammetry and remote sensing*, 154, pp.70-83.
- Guo, Z., Chen, Q., Wu, G., Xu, Y., Shibasaki, R. and Shao, X., 2017. Village building identification based on ensemble convolutional neural networks. *Sensors*, 17(11), p.2487.
- Hasan, R.C., Rosle, Q.A.Z., Asmadi, M.A. and Kamal, N.A.M., 2018. Extraction of element at risk for landslides using remote sensing method. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42(4/W9).
- He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).
- He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

- Huang, X. and Zhang, L., 2011. Morphological building/shadow index for building extraction from high-resolution imagery over urban areas. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(1), pp.161-172.
- Iglovikov, V., Seferbekov, S., Buslaev, A. and Shvets, A., 2018. Terausnetv2: Fully convolutional network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 233-237).
- Jung, H., Choi, H.S. and Kang, M., 2021. Boundary enhancement semantic segmentation for building extraction from remote sensed image. *IEEE Transactions on Geoscience and Remote Sensing*, 60, pp.1-12.
- Kanopoulos, N., Vasanthavada, N. and Baker, R.L., 1988. Design of an image edge detection filter using the Sobel operator. *IEEE Journal of solid-state circuits*, 23(2), pp.358-367.
- Karantza, K. and Paragios, N., 2008. Recognition-driven two-dimensional competing priors toward automatic and accurate building detection. *IEEE Transactions on Geoscience and Remote Sensing*, 47(1), pp.133-144.
- Kodors, S., Rausis, A., Ratkevics, A., Zvirgzds, J., Teilans, A. and Ansone, I., 2017. Real estate monitoring system based on remote sensing and image recognition technologies. *Procedia Computer Science*, 104, pp.460-467.
- Koonce, B., 2021. MobileNetV3. In *Convolutional Neural Networks with Swift for Tensorflow* (pp. 125-144). Apress, Berkeley, CA.

- Koushik, J., 2016. Understanding convolutional neural networks. arXiv preprint arXiv:1605.09081.
- Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Lee, S.H., Yigitcanlar, T., Han, J.H. and Leem, Y.T., 2008. Ubiquitous urban infrastructure: Infrastructure planning and development in Korea. *Innovation*, 10(2-3), pp.282-292.
- Li, E., Femiani, J., Xu, S., Zhang, X. and Wonka, P., 2015. Robust rooftop extraction from visible band images using higher order CRF. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8), pp.4483-4495.
- Li, M., Zang, S., Zhang, B., Li, S. and Wu, C., 2014. A review of remote sensing image classification techniques: The role of spatio-contextual information. *European Journal of Remote Sensing*, 47(1), pp.389-411.
- Li, S., Zhang, Z., Li, B. and Li, C., 2018. Multiscale rotated bounding box-based deep learning method for detecting ship targets in remote sensing images. *Sensors*, 18(8), p.2702.
- Li, W., He, C., Fang, J., Zheng, J., Fu, H. and Yu, L., 2019. Semantic segmentation-based building footprint extraction using very high-resolution satellite images and multi-source GIS data. *Remote Sensing*, 11(4), p.403.
- Li, Z., Wegner, J.D. and Lucchi, A., 2019. Topological map extraction from overhead images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1715-1724).

- Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S., 2017. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).
- Ling, F., Li, X., Xiao, F., Fang, S. and Du, Y., 2012. Object-based sub-pixel mapping of buildings incorporating the prior shape information from remotely sensed imagery. *International Journal of Applied Earth Observation and Geoinformation*, 18, pp.283-292.
- Liu, L., Pan, Z. and Lei, B., 2017. Learning a rotation invariant detector with rotatable bounding box. arXiv preprint arXiv:1711.09405.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 10012-10022).
- Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).
- Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W., 2005. Geographic information systems and science. John Wiley & Sons.
- Lowe, D.G., 1999, September. Object recognition from local scale-invariant features. In Proceedings of the seventh IEEE international conference on computer vision (Vol. 2, pp. 1150-1157). Ieee.

- Löwe, R. and Arnbjerg-Nielsen, K., 2020. Urban pluvial flood risk assessment—data resolution and spatial scale when developing screening approaches on the microscale. *Natural Hazards and Earth System Sciences*, 20(4), pp.981-997.
- Maggiori, E., Tarabalka, Y., Charpiat, G. and Alliez, P., 2017, September. Polygonization of remote sensing classification maps by mesh approximation. In *2017 IEEE International Conference on Image Processing (ICIP)* (pp. 560-564). IEEE.
- Maggiori, E., Tarabalka, Y., Charpiat, G. and Alliez, P., 2017. High-resolution aerial image labeling with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12), pp.7092-7103.
- Noh, H., Hong, S. and Han, B., 2015. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1520-1528).
- Ojala, T., Pietikainen, M. and Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7), pp.971-987.
- Ok, A.O., 2013. Automated detection of buildings from single VHR multispectral images using shadow information and graph cuts. *ISPRS journal of photogrammetry and remote sensing*, 86, pp.21-40.
- Ok, A.O., Senaras, C. and Yuksel, B., 2012. Automated detection of arbitrarily shaped buildings in complex environments from monocular VHR optical satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 51(3), pp.1701-1717.

- Pappas, T.N. An adaptive clustering algorithm for image segmentation. *IEEE Trans. Signal Process.* 1992, 40, 901–914.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. and Desmaison, A., 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Pelizari, P.A., Spröhnle, K., Geiß, C., Schoepfer, E., Plank, S. and Taubenböck, H., 2018. Multi-sensor feature fusion for very high spatial resolution built-up area extraction in temporary settlements. *Remote Sensing of Environment*, 209, pp.793-807.
- Qi, C.R., Su, H., Mo, K. and Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 652-660).
- Qin, R. and Fang, W., 2014. A hierarchical building detection method for very high resolution remotely sensed images combined with DSM using graph cut optimization. *Photogrammetric Engineering & Remote Sensing*, 80(9), pp.873-883.
- Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.
- Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

- Royan, J., Bouville, C. and Gioia, P., 2003, November. PBTree-A New Progressive and Hierarchical Representation for Network-Based Navigation in Urban Environments. In VMV (Vol. 3, pp. 299-307).
- Sahar, L., Muthukumar, S. and French, S.P., 2010. Using aerial imagery and GIS in automated building footprint extraction and shape recognition for earthquake risk assessment of urban inventories. *IEEE Transactions on Geoscience and Remote Sensing*, 48(9), pp.3511-3520.
- Shu, Y., 2014. Deep Convolutional Neural Networks for Object Extraction from High Spatial Resolution Remotely Sensed Imagery. A thesis presented to the University of Waterloo in fulfillment of the thesis requirement for the degree of Doctor of Philosophy in Geography.
- Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Sirmacek, B. and Unsalan, C., 2009. Urban-area and building detection using SIFT keypoints and
- Suzuki, S., 1985. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1), pp.32-46.
- Tremeau, A. and Borel, N., 1997. A region growing and merging algorithm to color segmentation. *Pattern recognition*, 30(7), pp.1191-1203.
- Turker, M. and Koc-San, D., 2015. Building extraction from high-resolution optical spaceborne images using the integration of support vector machine (SVM) classification, Hough transformation and perceptual grouping. *International Journal of Applied Earth Observation and Geoinformation*, 34, pp.58-69.

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Viola, P. and Jones, M., 2001, December. 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, pp. I-I)*. Ieee.
- Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X. and Liu, W., 2020. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10), pp.3349-3364.
- Wang, S. and Yang, K.J., 2008. An image scaling algorithm based on bilinear interpolation with VC++. *Techniques of Automation and Applications*, 27(7), pp.44-45.
- Wei, S., Ji, S. and Lu, M., 2019. Toward automatic building footprint delineation from aerial images using CNN and regularization. *IEEE Transactions on Geoscience and Remote Sensing*, 58(3), pp.2178-2189.
- Wei, Y., Zhao, Z. and Song, J., 2004, September. Urban building extraction from high-resolution satellite panchromatic image using clustering and edge detection. In *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium (Vol. 3, pp. 2008-2010)*. Ieee.
- Wu, H., Zhang, J., Huang, K., Liang, K. and Yu, Y., 2019. Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation. *arXiv preprint arXiv:1903.11816*.

- Wu, Z. and Leahy, R., 1993. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 15(11), pp.1101-1113.
- Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M. and Zhang, L., 2018. DOTA: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3974-3983).
- Xiao, T., Liu, Y., Zhou, B., Jiang, Y. and Sun, J., 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 418-434).
- Zhang, C., Hu, Y. and Cui, W., 2018. Semiautomatic right-angle building extraction from very high-resolution aerial images using graph cuts with star shape constraint and regularization. *Journal of Applied Remote Sensing*, 12(2), p.026005.
- Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J., 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2881-2890).
- Zhao, K., Kamran, M. and Sohn, G., 2020. Boundary regularized building footprint extraction from satellite images using deep neural network. *arXiv preprint arXiv:2006.13176*.
- Zhao, K., Kang, J., Jung, J. and Sohn, G., 2018. Building extraction from satellite images using mask R-CNN with building boundary regularization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 247-251).

Zhen, D.; Zhongshan, H.; Jingyu, Y.; Zhenming, T. FCM Algorithm for the Research of Intensity Image Segmentation. *Acta Electron. Sin.* 1997, 5, 39–43.

Zorzi, S. and Fraundorfer, F., 2019, July. Regularization of building boundaries in satellite images using adversarial and regularized losses. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium* (pp. 5140-5143). IEEE.

Zorzi, S., Bittner, K. and Fraundorfer, F., 2021, January. Machine-learned regularization and polygonization of building segmentation masks. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 3098-3105). IEEE.