Deep Convolutional Neural Networks for Object Extraction from High Spatial Resolution Remotely Sensed Imagery

by

Yuanming Shu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Geography

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.
I understand that my thesis may be made electronically available to the public.

Abstract

Developing methods to automatically extract objects from high spatial resolution (HSR) remotely sensed imagery on a large scale is crucial for supporting land user and land cover (LULC) mapping with HSR imagery. However, this task is notoriously challenging. Deep learning, a recent breakthrough in machine learning, has shed light on this problem. The goal of this thesis is to develop a deep insight into the use of deep learning to develop reliable automated object extraction methods for applications with HSR imagery.

The thesis starts by re-examining the knowledge the remote sensing community has achieved on the problem, but in the context of deep learning. Attention is given to object-based image analysis (OBIA) methods, which are currently considered to be the prevailing framework for this problem and have had a far-reaching impact on the history of remote sensing. In contrast to common beliefs, experiments show that object-based methods suffer seriously from ill-defined image segmentation. They are less effective at leveraging the power of the features learned by deep convolutional neural networks (CNNs) than conventionally patch-based methods.

This thesis then studies ways to further improve the accuracy of object extraction with deep CNNs. Given that vector maps are required as the final format in many applications, the focus is on addressing the issues of generating high-quality vector maps with deep CNNs. A method combining bottom-up deep CNN prediction with top-down object modeling is proposed for building extraction. This method also exhibits the potential to

extend to other objects of interest. Experiments show that implementing the proposed method on a single GPU results in the capability of processing 756 km^2 of 12 cm aerial images in about 30 hours. By post-editing on top of the resulting automated extraction, high-quality building vector maps can be produced about 4-times faster than conventional manual digitization methods.

Acknowledgements

First of all, I want to express great gratitude and appreciation to my supervisor, Professor Dr. Jonathan Li, for accepting me to work with his GeoSTARS group, for his involvement, insight, encouragement, and support during the course of my research, and for the help with my life outside school. I am also very grateful for my thesis committee member, Professor Dr. Richard Kelly, Professor Dr. Michael Chapman, Professor Dr. Alexander Wong, and Professor Dr. Yun Zhang, and former thesis committee member, Professor Dr. Phil Howarth and Professor Dr. Guangzhe Fan for their critical comments and valuable suggestions on my thesis.

My gratitude also goes to the GeoSTARS group members Dr. Yu Li, Anqi Fu, Haocheng Zhang for their collaboration and discussion in my research. I would also like to thank all staff members in the Department of Geography and Environmental Management, particularly, Ms. Susie Castela, for helping me greatly in various ways.

My thanks further goes to the GIS Department of the Region of Waterloo, Ontario, Canada for providing the aerial imagery for this research. Further, I would like to give thanks for the financial support of the University of Waterloo and the Natural Science and Engineering Research Council of Canada (NSERC).

Finally, and most importantly, I am deeply indebted to my parents for their love, encouragement, and inspiration. Without them, I would not have been able to succeed in this Ph.D. study during the past five years in Canada.

Table of Content

List of Figures

List of Tables

Chapter 1

Introduction

High spatial resolution (HSR) remotely sensed imagery is becoming increasingly available nowadays. New companies are moving fast to exploit technological developments; Skybox Imaging and Planet Labs will each soon have a constellation of micro satellites in orbit with goal of collecting 1 – 3 m optical imagery on a daily basis, and various Unmanned Aerial Vehicle (UAV) companies are introducing affordable options for individuals to collect sub-meter aerial imagery directly (Colomina & Molina, 2014). Established players are also innovating; DigitalGlobe is adding to its collection of satellites with Worldview-3 to offer more frequent image data at 31 cm spatial resolution (DigitalGlobe, 2014), and various aerial imaging companies are introducing new offerings such as 10 cm oblique imagery. Lastly, governments are easing regulations; the US government recently lifted restrictions on selling 25 cm satellite imagery (SpaceNews, 2014), the US Federal Aviation Administration (FAA) approved the first commercial use of UAVs (FAA, 2014), and the Open Data movement in North America is drastically increasing accessibility to previously restricted datasets (Tauberer, 2014).

These new developments will allow for common access to massive amounts of HSR imagery[1] at affordable rates over the coming decade. HSR imagery introduces the capability of mapping land use and land cover (LULC) of the earth surface with a high level of detail that has led to many new applications. However, these new applications require transferring raw image data into

[1] "Imagery" in this thesis is referred to "remotely sensed imagery".

tangible information so that quantification or analysis can be further done with geographic information systems (GISs). The core of this transformation is the digitization and interpretation of objects within each image. This object extraction process is currently very labor intensive and time consuming – requiring human image interpreters. As a result, most of information contained in this enormous amount of new data is not available to those who need it most.

Developing methods to automatically extract objects from HSR imagery is critical to support LULC mapping with HSR imagery. The goal of this thesis is to investigate issues surrounding the development of automated object extraction methods for HSR imagery and to design methods that work reliably for large-scale real-world applications. This first chapter serves as an overview of the entire thesis, and begins by describing common applications associated with HSR imagery in Section 1.1. The major challenges for developing automated object extraction methods for HSR imagery are then discussed in Section 1.2. Following this discussion, the contributions of the thesis are summarized in Section 1.3 and, lastly, the outline of the thesis is presented in Section 1.4.

1.1 Applications of HSR Imagery

With recent significant improvements in spatial resolution, HSR imagery is capable of mapping LULC with a high level of thematic detail. This has led to a large number of new applications.

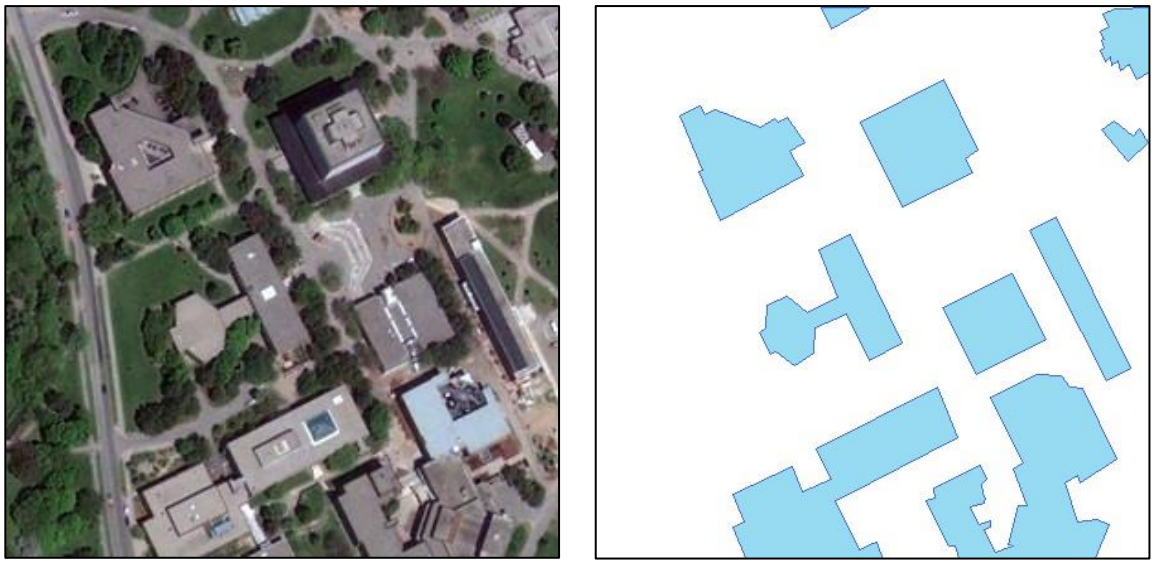
The most common examples are applying HSR imagery for local urban development and management, e.g., road network mapping and updating (Mena, 2003; Miao et al., 2013), traffic

flow monitoring (Jin & Davis, 2007; Eikivl et al., 2009), and building extraction and change analysis (Katartzis & Sahli, 2008; Sirmacek & Unsalan, 2011; Doxani et al., 2012) for tax assessment. Other applications include using HSR imagery for forestry and agriculture such as identifying tree species (Key et al., 2001; Carleer & Wolf, 2004; Agarwal et al., 2013), monitoring forest health (Wulder et al., 2006; Coops et al., 2006; Garrity et al., 2013), mapping crop types (Senay et al., 2000), and estimating crop yield (Seelan et al., 2003). In addition, HSR imagery can be used by public administrations for natural hazard management such as earthquake loss estimation (Stramondo et al., 2006; Brunner et al., 2010; Ehrlich et al., 2013), forest wildfire monitoring (Leblon, 2001; Mitri & Gitas, 2013), and flood risk and flood damage assessment (van der Sande et al., 2003; Thomas et al. 2014). Furthermore, with the support of HSR imagery, it is now possible to carry out LULC change studies on a much finer scale than was previously possible with low spatial resolution (LSR) imagery (e.g., 30m Landsat or 250m MODIS). Related applications include using HSR imagery for sea level rise monitoring (Liu & Jezek, 2004; Li et al., 2008), coral reef habitat mapping (Andrefoueet et al., 2003; Mumby et al., 2004), and urban growth analysis (Park et al., 1999; Moeller & Blaschke, 2006; Xu, 2013).

1.2 Major Challenges for Automated Object Extraction

Despite the potential of the above applications, all require raw HSR imagery to be first transformed into tangible information so that quantitatively analysis can then be performed using modern GISs. The core of this transformation is the digitization and interpretation of objects within each image. For example, to calculate the number and size of buildings in the image of Figure 1.1(a), one must first recognize them from the image and then digitize them into the

format such as the vector map in Figure 1.1(b). This object extraction process is currently very labor intensive and time consuming. For example, digitizing all the buildings for a small county with an area of 150 km^2 can take one GIS analyst thousands of hours. As a result, most of information contained in the massive amounts of new HSR imagery is not being tapped. Developing effective methods to automatically extract objects from HSR imagery is the key to unlock the value hidden in this big geospatial data. However, this task is very challenging. Major obstacles lie in the following four aspects:



(a) (b)

Figure 1.1. Building extraction from HSR Imagery. (a) Original image at spatial resolution of 0.5 m (courtesy of Google Earth). (b) Digitized building vector map.

Figure 1.2. Demonstration of major obstacles to object extraction from HSR imagery (courtesy of Google Earth) (a) and (b) Large intra-class variation. (c) Effect of shadows. (d) Effect of occlusions.

(1) Large intra-class variation. As the resolution of HSR imagery increases, the internal structure of objects becomes discernible. As a result, large variations may be observed within the same object category. For example, buildings may appear in an HSR image with different colors, shapes, locations and orientations, as shown in Figure 1.2(a) and (b). This intra-class variation

greatly increases the challenge of designing sophisticated methods for discriminating between different objects.

(2) Effect of shadows and occlusions. Appearances of objects can be affected by the presence of shadows and geometrical occlusions in a HSR image. For example, roads in Figure 1.2(c) are partially covered by shadows caused by tall buildings along the roads, and buildings in Figure 1.2(d) are partially blocked by surrounding trees. In such cases, performing object extraction automatically becomes increasingly difficult, requiring not only the delineation of boundaries of non-affected parts, but also the reconstruction of affected parts.

(3) Chicken and egg. Object extraction from HSR imagery is by nature a "chicken and egg" problem: given the outline of an object, recognition becomes easier. But in order to get the object's proper outline, recognition is first needed to determine the type of the object. Such a dilemma creates considerable barriers to developing effective methods for object extraction from HSR imagery.

(4) Large-scale datasets. Many applications require large amounts of image data to be processed. For example, one needs to process 30 GB data in order to extract all of the buildings from a city of 150 km^2. Methods of processing this sort of data need to be efficient enough so that processing time for real-world applications is feasible.

1.3 Thesis Contributions

Significant efforts have been made in remote sensing literature on pursuing a solution. However, existing studies are often limited to a small set of test samples in some relatively simple scenes. None of them, to the best of author's knowledge, has proven to work reliably for complex real-world scenarios on a large scale. As a result, much of this extraction is still heavily reliant on labor-intensive and time-consuming manual digitization. In practice, this has become the main hurdle for applications with HSR imagery.

Deep learning, a recent groundbreaking development in the machine learning community, has shed light on this problem. Rather than hand-crafting features, deep learning has shown how hierarchies of features can be learned from the data itself (Hinton & Salakhutdinov, 2006; Hinton et al., 2006; Bengio et al., 2007). These learned features have shown the ability to be generalized to a variety of visual recognition tasks, and have exhibited significant accuracy improvements in recent studies (Krizhevsky et al., 2012; Mnih, 2013; Girshick et al., 2014). In particular, Mnih (2013) has shown using features extracted from a large local image patch via deep convolutional neural networks (CNNs) can largely enhance the accuracy of automated road and building extractions from 1 m aerial images in complex urban scenes. These new developments suggest that automated object extraction systems for HSR imagery that work reliably for real-world applications may be within reach. This thesis is built on top of these new developments, with the goal of further deepening the insight on using deep learning to develop automated object extraction methods for HSR imagery. The main contributions of this thesis lie on the following two aspects:

(1) The thesis re-examines the knowledge the remote sensing community has developed on the problem of object extraction from HSR imagery, but in the context of deep learning. Attention is given to object-based image analysis (OBIA) methods whose developments have largely shaped the minds of the remote sensing community on this problem. In contrast to common beliefs, experiments show that object-based methods suffer seriously from ill-defined image segmentation. Further, they are less effective at leveraging the power of the features learned by deep CNNs than conventionally patch-based methods. This new discovery suggests the necessity of re-investigating the knowledge system that has been built on the foundation of OBIA.

(2) This thesis explores the ways of further improving the accuracy of object extraction with deep CNNs. Given that vector maps are required as the final format in many applications, the focus is on addressing the issues of generating high-quality vector maps with deep CNNs. A method that combines bottom-up deep CNN prediction with top-down object modeling is proposed for building extraction, and the extension of this method to other objects of interest is also discussed. Given that conventional criteria are ineffective at evaluating the accuracy of extracted vector maps, measuring post-editing time is further proposed as a simple yet effective substitute. Experiments show that implementing the proposed method on a modern GPU allows for speed and accuracy improvements that are promising for practical use.

1.4 Thesis Outlines

The rest of the thesis is organized as follows:

In Chapter 2, methods for automated object extraction from HSR imagery developed in remote sensing literature are reviewed under four categories: patch-based methods, object-based methods, current trends, and deep learning.

In Chapter 3, given its far-reaching impact on the remote sensing community, the concept of OBIA is re-examined in the context of deep learning. A comparison between patch-based CNNs, object-based CNNs, and standard OBIA is made, focusing on their effectiveness in extracting a variety of objects from HSR imagery. The issues of using deep CNNs under the framework of OBIA, and the necessity of re-investigating the knowledge system built on the foundation of OBIA is also discussed.

In Chapter 4, ways of further improving the accuracy of object extraction using deep CNNs are studied. The focus is placed on tackling the issues surrounding the generation of high-quality vector maps using deep CNNs. For this purpose, a method combining bottom-up deep CNN prediction with top-down object modeling is proposed. Inherent issues of the methods used to evaluate the accuracy of extracted vector maps are also discussed. Experiments are developed to verify the effectiveness of the proposed method.

In Chapter 5, the findings of this thesis are summarized and avenues for future research are recommended.

Chapter 2

Related Work on Automated Object Extraction

Throughout the past decade, numerous methods have been proposed in pursuit of a solution to the problem of automated object extraction from HSR imagery. In this chapter, these methods are reviewed roughly in chronological order in order to demonstrate how they have evolved throughout the history of remote sensing. This chapter starts by describing patch-based methods in Section 2.1, which are adapted from classic pixel-based methods used mainly for object extraction from LSR imagery. The object-based methods, which largely shaped the knowledge of the remote sensing community, are then reviewed in Section 2.2. In Section 2.3, current trends for developing object extraction methods are presented. Each of their limitations is also discussed. In the last section, special attention is given to deep learning, whose development has cast new light on this challenging problem.

2.1 Patch-based Methods

Patch-based methods have roots in classic pixel-based methods (Jensen, 2005), which are mainly used for classification[2] of LULC using LSR imagery. In pixel-based methods, a trained classifier employs spectral features at each individual pixel to determine the object label of respective pixel. These spectral features work sufficiently well to discriminate between broad objects of interest on LSR imagery with reasonably high accuracy, such as forest, rivers, and urban areas (Hansen et al., 2013; McNairn et al., 2009; Sexton et al., 2013). However, they have been

[2] The terminology "object extraction" may be used as "classification" or "image labeling" in some other context. In this thesis, these terminologies will be used interchangeably unless otherwise stated.

considered to be notoriously inappropriate for characterizing finer object classes from HSR imagery, such as roads, buildings, and trees (Blaschke & Strobl, 2001). At resolutions higher than 1 pixel/m^2, spectral values of different object types (e.g., roads and buildings all made of cement) can be very similar. This makes it nearly impossible to extract objects from HSR imagery while using spectral features alone.

Therefore, in order to adapt these methods to HSR imagery, it is important to develop methods to retrieve texture, geometric, and contextual features from neighboring pixels and use these features to improve object extraction. Patch-based methods are proposed for the purpose, which, at a high level, derive rich features from a local window centered at each pixel. Varieties of local window feature descriptors have been developed in remote sensing literature. For example, Benediktsson et al. (2003) employed mathematical morphological operations to characterize pixels' local features in their study of urban LULC classification using Indian Remote Sensing 1C and IKONOS images. Other work also using morphological operations includes Chanussot et al. (2006) and Tuia et al. (2009). In addition, Zhang (2001), Puissant et al. (2005), and Pacifici et al. (2009) used a gray-level co-occurrence matrix (GLCM) to describe the pixels' texture features and showed accuracy improvements for the classification of HSR satellite imagery in urban scenes. Wulder et al. (2000) used a local maximum filter for extracting tree locations and basal areas from HSR aerial imagery. Ouma et al. (2006) utilized wavelets to delineate urban-trees from QuickBird imagery. Lastly, Sirmacek & Unsalan (2009) used Gabor filters to detect buildings from IKONOS images.

These studies have consistently shown that employing rich features derived from local windows

helps to increase the accuracy of object extraction from HSR imagery. However, these features are still very local, as the window size they used is typically less than 9 pixels. Within such a small window, it is difficult to capture long-range geometric and contextual information that is required to distinguish between different objects and overcome the adverse effects caused by shadows and occlusions. For example, the width of a road can be as long as 30 pixels on a 0.5 m image. Given a 9-pixel window, it is hard to distinguish between a road pixel and a parking-lot pixel. This challenge is compounded when the road is occluded by trees or shadows. Without enough geometric and contextual information, it is nearly impossible to tell one from the other. Ideally, the window size should be large enough to contain as much context as possible. However, the complexity of the context increases with the size of the window. Effectively and efficiently describing context and retrieving discriminative features has become a problem in itself. Therefore, the success of the above patch-based methods is quite limited, and none of the above methods has been proven to work effectively for extracting objects from HSR imagery in complex scenes.

2.2 Object-based Methods

Considering the disadvantages of patch-based methods, OBIA methods were proposed for object extraction from HSR imagery (Baatz & Schape, 2000; Blaschke & Strobl, 2001; Burnett & Blaschke, 2003; Benz et al., 2004). In object-based methods, individual pixels of an image are first grouped into several homogenous regions based on their similarities in terms of spectra and texture features. In literature, this step is usually referred to as image segmentation. A set of spectral, texture, geometric, and contextual features are then extracted from these regions to

characterize their attributes. Lastly, a classifier is utilized to label each region with a unique object class based on the extracted features of the region. As can be seen, the main difference between patch-based methods and object-based methods is that the latter first aggregates individual pixels into homogenous regions and then applies feature extraction and classification to these regions rather than operating on individual pixels as in patch-based methods. These regions are considered to be potential objects, and therefore this type of methods is referred to as "object-based image analysis" in remote sensing literature. It was renamed "geographic object-based image analysis (GEOBIA)" to distinguish between concepts of the "object-based" in other communities (Hay & Castilla, 2008).

Object-based methods are currently the most widely used methods for the task of automatically extracting objects from HSR imagery. The development of these methods was considered as a breakthrough in remote sensing literature and has largely shaped the knowledge of the remote sensing community surrounding this task. Since the advent of the first commercial software "eCognition" implementing object-based methods (Baatz & Schape, 2000; Flanders et al., 2003; Benz et al., 2004), many researchers have studied the use of these methods to extract various objects from HSR images in different scenes. For example, Yu et al. (2006) applied an object-based method for detailed vegetation extraction with an HSR airborne image and empirically demonstrated that object-based methods outperformed the conventional pixel-based methods in terms of the accuracy of the extraction. Mallinis et al. (2008) carried out a multi-scale object-based analysis of a QuickBird imagery to delineate forest polygons and illustrated that the adoption of objects instead of pixels as primary units could take advantages of a rich amount of spatial information for the extraction. Fernandes et al. (2014) developed and tested an object-

based method to map giant reeds in riparian habitats with HSR airborne imagery and WorldView-2 satellite imagery and suggested that giant reeds can be extracted with reasonable accuracy. Powers et al. (2015) assessed an object-based method to map industrial disturbance using SPOT 5 imagery in the oil sands regions of Alberta and showed this method is able to effectively delineate fine-spatial resolution industrial disturbance. van der Sande et al. (2003) applied an object-based method to produce land cover maps from IKONOS imagery for flood risk and flood damage assessment in the southern part of the Netherlands and showed such maps could be useful for decision makers and insurance companies. A more detailed review is referred to Blaschke (2010) and Blaschke et al. (2014). Given the success of object-based methods, some scholars advocated treating GEOBIA as a new sub-discipline in recent studies (Hay & Castilla, 2008; Blaschke, 2010; Blaschke et al, 2014).

Compared to patch-based methods, object-based methods using image segmentation techniques are considered much more effective at the task of deriving varieties of long-range geometric and contextual features. It is generally agreed that this advantage largely enhances the capabilities of object-based methods in dealing with issues surrounding the large intra-class variation as compared to patch-based methods (Blaschke & Strobl, 2001; Hay & Castilla, 2008; Blaschke, 2010; Blaschke et al., 2014). However, object-based methods suffer from their own problems. First of all, the accuracy of object-based methods heavily relies on the quality of the image segmentation. But, when pixels are grouped into regions, only low-level features (i.e., spectra and texture features) are used to measure the homogeneity without including any high-level features (i.e., geometry and context features). There is no guarantee that regions generated through such a process correspond to real objects or object parts due to the ambiguity of low-

level features, even with state-of-art image segmentation algorithms (Kolmogorov & Zabih, 2004; Arbelaez et al., 2011; Arbelaez et al., 2014). For example, roads and entrances of parking lots might be grouped into one region due to their similarity in terms of spectra features. Further, features extracted from mis-segmentation may not represent properties of real objects and could lead to classification errors (Moller et al., 2007; Kampouraki et al., 2008; Liu & Xia, 2010). These issues may become even severe when shadows and geometric occlusions are presented in the image. Secondly, even if the generated region is perfectly lined up with the boundary of an object, extracting features in order to distinguish it from other objects is still an unsolved problem. For example, commonly used feature descriptors such as spectral mean, spectral standard deviation, texture mean, texture entropy, region size, elongation, Hu's moment (Jensen, 2005), have been proven to work reasonably well for characterizing the features of natural scenes such as grass, trees, rivers. However, they are too primitive to discriminate between complicated man-made objects like buildings, parking lots, or roads in HSR imagery. Little success has been reported on extracting complex man-made objects with these features.

2.3 Current Trends

To overcome the issues discussed in Section 2.1 and 2.2, numerous studies have been conducted recently in the remote sensing community. Notable trends include:

- The use of more discriminative features,
- The switch to more powerful classifiers, and
- The change to more sophisticated frameworks.

The details of the three trends are discussed in the following subsections.

2.3.1 Discriminative Features

Since it is of great difficulty to achieve satisfactory classification results by using HSR imagery alone, it is natural to think of adding more discriminative features through auxiliary data. One typical example is to use digital surface model (DSM) that is either extracted from stereo photogrammetry or directly from airborne light detection and ranging (LiDAR). A large amount of studies have shown that incorporating elevation data can significantly increase accuracy of object extraction, even with simple pixel-based methods (Kosaka et al., 2005; Sohn & Dowman, 2007; Gong et al., 2011; Kim & Kim 2014). However, collecting HSR evaluation data is very expensive. For example, collecting DSM for a city as large as 150 km^2 using airborne LiDAR may cost up to $12,000, which is four times more expensive than collecting aerial images alone. It is therefore cost-prohibitive to rely on such auxiliary data for object extraction on a large scale. Also, given that a human image analyst can perform the object extraction task very well by using an HSR image alone, features extracted from the auxiliary data are actually redundant. Hence, from a research point of view, it would be interesting to omit the use of auxiliary features.

Several efforts are made in this direction, which attempt to extract more discriminative features from the HSR image source alone. For example, Bruzzone et al. (2006) proposed to extract spectral, texture and geometric features from multi-scale segmentation and demonstrated that these features helped improve the accuracy of urban LULC mapping. Huang & Zhang (2012) developed a morphological building/shadow index and suggested this feature was useful for

extracting buildings from HSR imagery. Zhang et al. (2013) invented a novel spatial feature called object correlative index to enhance the accuracy of object extraction from HSR imagery. Zhang et al. (2014) presented a novel method to extract pixel shape features and indicated that these features were more discriminative than traditionally used spectral and GLCM features. Other sophisticated features that are widely-used in computer vision community such as scale-invariant feature transform (SIFT) (Lowe 2004), histogram of oriented gradients (HOG) (Dalal & Triggs 2005), spatial pyramid (Lazebnik et al., 2006) may also be applicable to enhance object extraction from HSR imagery, but historically have not been widely used by the remote sensing community.

2.3.2 Powerful Classifiers

Distinguishing between different objects with similar spectra and texture features such as roads, parking lots, or buildings requires knowledge of objects' geometry and context. This leads to the needs of learning nonlinear decision boundaries, which require more powerful classifiers. Several studies have been conducted to explore the potential of improving the object extraction accuracy through the use of advanced classifiers. For example, Huang et al. (2002) assessed the results achieved by a support vector machine (SVM), maximum likelihood, artificial neural network (ANN), and decision tree for LULC mapping using HSR satellite images. It was demonstrated that accuracy achieved by SVM was relatively higher than other methods. Mountrakis et al. (2011) also gave a thorough review on using SVM for object extraction from remotely sensed imagery. Gong et al. (2011) compared the capabilities of an optimized artificial immune network, ANN, decision tree, and typical immune network in LULC classification using QuickBird and LiDAR data and suggested that the optimized artificial immune network was

more effective than the rest of classifiers. Pal (2005) gave a summary of the advantages of using random forest for object extraction from remotely sensed imagery. Zhong et al. (2014) proposed a novel classification framework based on conditional random field and showed this classification framework had a competitive performance compared to other state-of-art classifiers. Lastly, Tokarczyk et al. (2015) demonstrated effectiveness of using adboost for object extraction of HSR images.

2.3.3 Sophisticated Frameworks

As discussed in Section 1.2, object extraction from HSR imagery is by nature a "chicken and egg" problem: given the outline of an object, recognition becomes easier. But in order to get the object's proper outline, recognition is first needed to determine the type of the object. One commonly used strategy to resolve this dilemma is to infer the object's outline and category through a bottom-up process. Patch-based and object-based methods both apply this technique. These methods start with individual pixels (or a group of pixels), which indicate possible locations of objects. They then extract a set of features from each location to determine the object class for each pixel (or a group of pixels) with a classifier. The decision boundary of the classifier is learned from a number of training samples using discriminative methods such as logistic regression or support vector machine. Bottom-up methods are essentially data-driven methods; they are widely used for object extraction from HSR imagery due to their computational efficiency. However, they do not use any object prior knowledge – everything is learned from the data in a discriminative way. As a result, the bottom-up methods "cannot say what signals were expected, only what distinguished typical signals in each category" (Mumford

& Desolneux (2010)). This makes bottom-up methods more sensitive to unexpected scenes in the testing data, e.g., the adverse effects of shadows and occlusions.

One possible way to incorporate object prior knowledge is through a top-down process. In contrast to bottom-up methods, top-down methods are model-driven, encoding object prior knowledge into a set of object models and localizing objects by matching these models to the image. One well-known example of this routine is the marked point process. Stoica et al. (2004) first applied the marked point process to road network extraction from HSR imagery. In their method, road segments were modeled as a set of marks parameterized by their orientation, length, and width in a Gibbs filed, which favors to form connected line-networks. A reversible jump Markov Chain Monte Carlo (RJMCMC) algorithm was used to find the optimal match between road prior knowledge models and the image. This method was later extended to extract buildings, tree crowns, and marine oil spills from remotely sensed imagery in subsequent studies (Lacoste et al., 2005; Perrin et al., 2005; Ortner et al., 2008; Lacoste et al., 2010; Lafarge et al., 2010; Li & Li, 2010; Benedek et al., 2012; Verdie & Lafarge, 2014). These studies have consistently shown imposing object prior knowledge help address issues caused by the adverse effects of shadows and occlusions. However, this benefit comes with a large computational cost. Since objects may appear on multiple scales and multiple orientations in an image, these methods have to go through an enormous number of locations to find the best matches between the object models and the image. This could be very time-consuming compared to bottom-up methods. This computational burden has seriously affected the broad applications of top-down methods for object extraction from HSR imagery.

As can be seen, bottom-up methods and top-down methods are very supplementary to each other: bottom-up methods provide the possible locations which top-down methods need to avoid exhaustive searching; top-down methods offer the prior knowledge that bottom-up methods desire to cope with unexpected scenes. Given the disadvantages of using them alone, it would be ideal to combine them. Porway et al. (2010) did an explorative study in this direction. They proposed a hierarchical and contextual grammar model for aerial image parsing in complex urban scenes. In this model, they used a range of object detectors to propose possible object locations in a bottom-up manner, and used the hierarchical grammar model to verify the detected objects and predict missing objects in a top-down manner using RJMCMC algorithm. Their experiments showed that the bottom-up and top-down processes could indeed contribute collaboratively towards providing more effective object extraction from HSR image data.

2.4 Deep Learning

Although these recent studies reviewed above have shown using discriminative features, powerful classifiers, and sophisticated frameworks can improve the accuracy of object extraction from HSR imagery to some degree, none of these proposed methods has proven to work efficiently and effectively for large-scale real world applications (to the best of author's knowledge). The test dataset of these studies is relative small, for example; three to four exemplar images with a few thousand by a few thousands pixels in size. The test scenes are comparatively simple compared to challenging real-world scenarios. One fundamental reason that prevents these methods from attaining reliable performance on challenging datasets is a lack of effective maneuvers for retrieving powerful features from the HSR imagery in order to effectively discriminate between different objects. Without discriminative features, it is very

difficult to address the issues aroused by the large intra-class variation presented in HSR imagery.

Deep learning, a recent groundbreaking development in machine learning, has shed new light on this problem. Rather than having engineers spend years handcrafting features, deep learning has shown how hierarchies of discriminative features can be learned from data with a deep neural network (Hinton & Salakhutdinov, 2006; Hinton et al., 2006; Bengio et al., 2007). These learned features have been shown to be adaptable to a variety of recognition tasks and have exhibited significantly improved accuracies in recent studies. These examples include speech recognition in Dahl et al. (2012), video activity recognition in Le et al. (2011), natural language processing in Collobert & Weston (2008), image classification in Krizhevsky et al. (2012), and object detection and semantic segmentation in Girshick et al. (2014). In particular, Mnih (2013) has shown that with deep learning, a simple patch-based method can achieve astonishing results for large-scale road and building extraction from 1 m aerial images in complex urban scenes. The trick of this method is nothing but using deep CNNs to extract features from a much larger local window and using a GPU to accelerate the extraction process. These results, for the first time in remote sensing literature, suggest that automated object extraction systems, which are reliable for practical use, may be within reach.

Chapter 3

Re-examining OBIA

The exciting results achieved by deep CNNs have motivated the author to re-examine the knowledge the remote sensing community has built on the problem of automated object extraction from HSR imagery. Since simple patch-based methods can work very effectively with features learned by deep CNNs, it is natural to think that other more sophisticated methods can also leverage the power of deep CNNs to further improve the accuracy of the extraction. Attention is given to object-based image analysis (OBIA), whose development was considered as a breakthrough in remote sensing literature and is currently the most popular solution for this problem. This chapter re-investigates the concept of OBIA in the context of deep learning. It starts by giving the motivation of conducting the investigation in Section 3.1. It then reviews the way patch-based methods use deep CNNs for object extraction in Section 3.2, and proposes an approach for fitting deep CNNs into the object-based framework in Section 3.3. Furthermore, it conducts a comparison between patch-based and object-based methods on their effectiveness of leveraging the power of deep CNN features for automatically extracting objects from HSR imagery in Section 3.4. Following the comparison, the issues of OBIA for object extraction in the context of deep learning are discussed in Section 3.5. The findings of this chapter are summarized in Section 3.6.

3.1 Motivations

Mnih (2013) has shown that, by using features extracted by deep CNNs, a simple local patch-based method can achieve promising results for extracting roads and buildings from 1 m aerial images in complex urban scenes on a large scale. The trick of this method is indeed nothing by using deep CNNs to extract features from a much larger local window, i.e., 64×64 pixels. As discussed in Section 2.1, it is conventionally believed that patch-based methods are suitable for deriving local texture features, though not for retrieving long-range geometric and contextual features. However, as can be seen from Mnih (2013), what's really missing from previous patch-based methods is a powerful feature extractor and a window large enough to include the desired spatial context. Since simple patch-based methods can work very effectively with deep CNN features, it is natural to think that using deep CNNs under a more sophisticated framework would lead to further improvements in the accuracy of object extraction. Attention is given to OBIA, which is currently the most prevailing framework for this problem. As discussed in Section 2.2, the development of object-based methods has been considered a breakthrough in remote sensing literature, and has largely shaped the knowledge of the remote sensing community on this problem. OBIA techniques have been widely adopted by a variety of remote sensing software packages such as eCognition, ENVI, and ERDAS, which are currently the industry standards for object extraction from HSR imagery. Given the success and important status of OBIA in remote sensing literature, it is desired to see whether OBIA can also leverage the power of deep learning to further improve the accuracy of object extraction.

One distinguishing characteristic of OBIA is the use of image segmentation at its fundamental level to bridge the gap between pixels and objects. As discussed in Section 2.2, it is generally

believed that image segmentation is the cornerstone for deriving the rich texture, geometric, and contextual features that are considered crucial for distinguishing between different objects within HSR imagery. However, as shown in Mnih (2013), sufficiently rich geometric and contextual features can be extracted from a large local window using deep CNNs without conducting image segmentation. If this is the case, what kind of role does image segmentation really play for object extraction from HSR imagery? Given the far-reaching impact of this concept on the development of automated object extraction methods in remote sensing literature, it is desirable to re-inspect the role image segmentation plays in object extraction.

Motivated by the above two thoughts, this chapter re-examines the concept of OBIA in the context of deep learning. At a high level, the re-examination starts by reviewing basic concepts of deep CNNs and presenting how deep CNNs are used in the patch-based framework. It then presents a way of using deep CNNs under the framework of OBIA. Furthermore, it compares the effectiveness of patch-based CNNs and object-based CNNs on a number of object extraction tasks with HSR imagery and discusses the performance of OBIA in the context of deep learning. Conclusions are drawn from these comparisons and discussions. The details of the re-examination are presented in each of the following sections.

3.2 Patch-based CNNs

As shown in Figure 3.1, procedures of patch-based CNNs at test time can be summarized into three main steps:

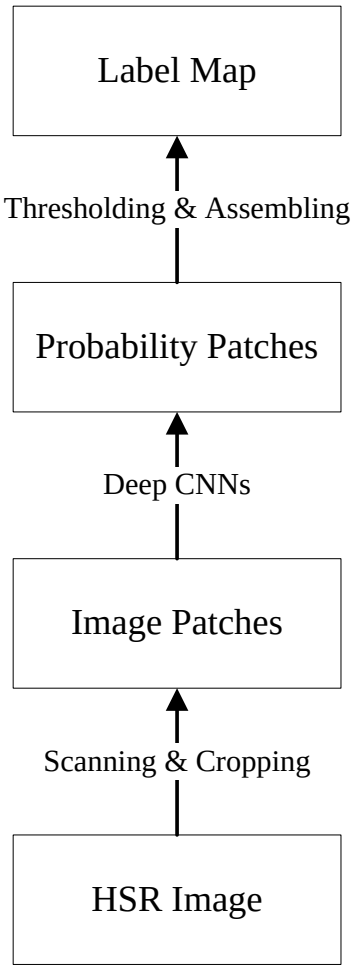


Figure 3.1. Pipeline of patch-based CNNs. The method scans through an image and crops a patch from the image at each location of the scanning. For each patch, it uses deep CNNs to predict the probabilities of pixels in a small window at the center of the patch being a certain object. It then assembles the predicted probabilities to form the probability map of the entire image and applies a threshold to the probability map to obtain the label map for the entire image**.**

(1) Image-patch generation. A $s_n \times s_n$ window scans through the entire image at stride of *t*. At each location *i*, the image within the window is cropped to form a $s_n \times s_n$ image patch *N*. Following Mnih (2013), s_n is set to 64 pixels for all the experiments in this thesis. Setting s_n too small would not capture enough spatial context for object extraction, while setting s_n too large would result in the retrieved context being too complicated for the extraction and increase the

amount of computation required in the step of deep CNN prediction. The stride *t* is set to be 8 pixels. Although a smaller *t* could provide enhanced extraction accuracy, it would also increase the amount of overall computation. Through extensive experiments, 8 pixels are found to achieve a good balance on both sides.

(2) Deep CNN prediction. Deep CNNs are applied to each image patch *N* to predict probabilities of pixels of a label patch *M* being a certain object. The label patch *M* is centered at the current location *i* and has the size $s_m \times s_m$. s_m is typically set smaller than s_n, because some context is needed to predict object labels of the patch *M*. While s_m is usually set as one pixel to predict one label at a time in many previous studies (referred to the discussion of Section 2.1), it is much more efficient to predict a small window of labels from the same context. Following Mnih (2013), s_m is set as 16 pixels in all the experiments of this thesis. The details of the deep CNN prediction are presented in Section 3.2.1.

(3) Label-map generation. The predicted probabilities of all the individual label patches are assembled to form the probability map of the entire image. Probabilities of pixels in the overlapped area of two consecutive label patches are determined by the maximum value of the overlapped pixels. To further obtain the label map of the entire image, a threshold T_p is applied to the probability map. T_p is typically set as 0.5 representing a 50% chance of being a certain object for a binary classification. Tuning the threshold T_p allows a controlled trade-off between false positives and false negatives.

3.2.1 Deep CNNs on Patches

As shown in Figure 3.2, deep CNNs take image patch *N* as an input, apply a sequence of linear/non-linear transformations to extract features from the patch in a layer-by-layer fashion, and map the extracted features to probabilities of pixels of label patch *M* being a certain object as the output.

The architecture of deep CNNs is inspired by the biological visual system (LeCun, 1989). The convolution and pooling operations it possesses enable the networks to learn features that are shift-invariant and robust to small image distortions. These features have been proven to be extremely useful for image applications (LeCun et al., 1998; Krizhevsky et al., 2012; Mnih 2013; Girshick et al. 2014), as compared to other networks such as Restricted Boltzmann Machine (Hinton & Salakhutdinov, 2006).

The details of the mathematical model of using patch-based CNNs for object extraction, the architecture of deep CNNs, and the approach of the learning parameters of deep CNNs, are presented from Section 3.2.1.1 to Section 3.2.1.3, respectively.

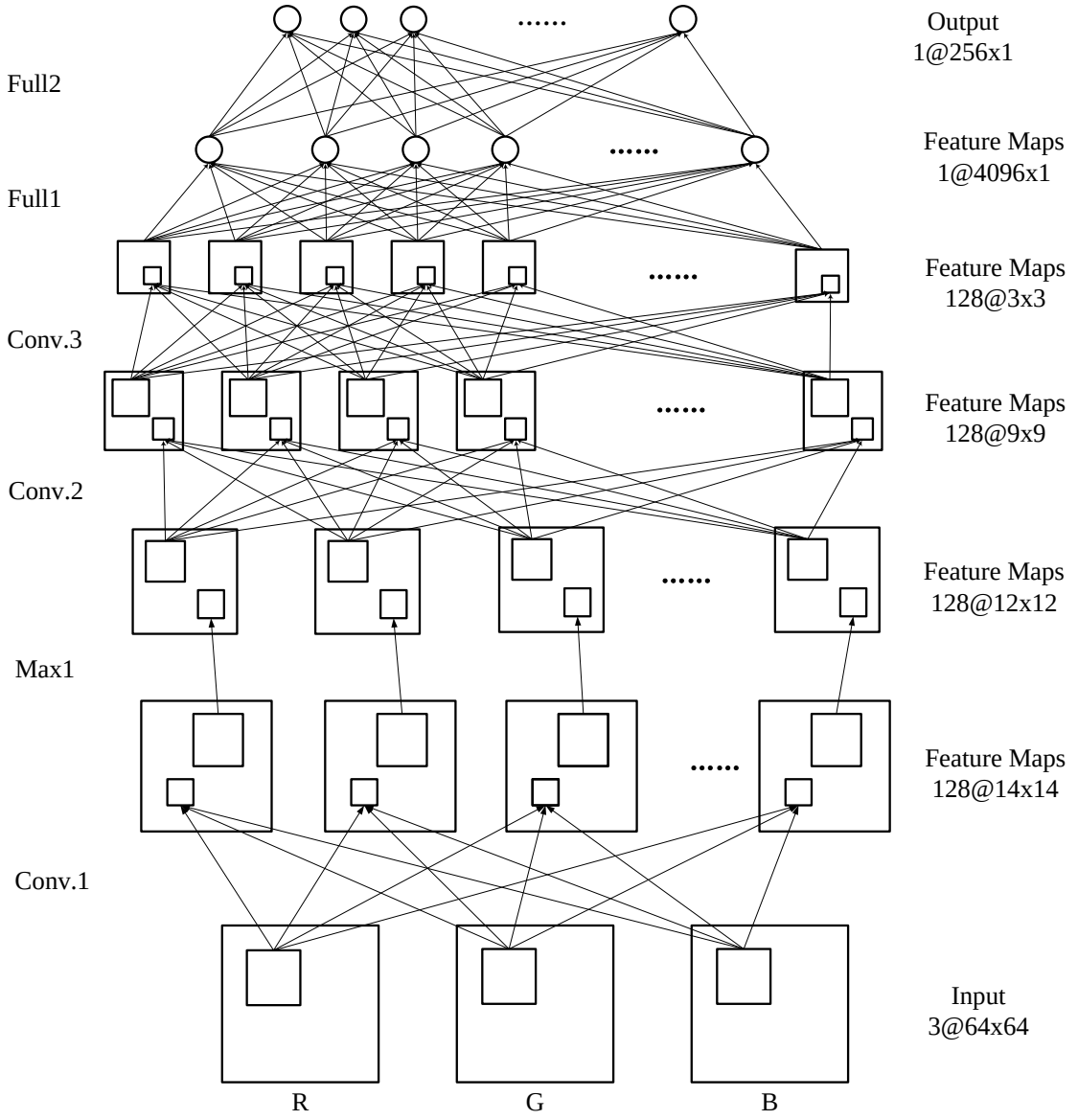


Figure 3.2. Hierarchical feature extraction and prediction via deep CNNs. Given an input image patch, deep CNNs extract hierarchies of features from the patch in a layer-by-layer fashion. The output of the bottom layer is used as the input of the top layer. The output of the last layer is fed to a logistic regression to predict the probabilities of pixels being a certain object.

Layer	1	2	3	4	Output
Stage	conv. + max	conv.	conv.	full	full
Number of channels	128	128	128	4096	256
Filter size	16×16	4×4	3×3	–	–
Convolution stride	4	1	1	–	–
Pooling size	2×2	–	–	–	–
Pooling stride	1	–	–	–	–
Spatial input size	64×64	12×12	9×9	3 × 3	–
Activation function	relu	relu	relu	relu	logistic

Table 3.1. Architecture specified for deep CNN model. "conv." stands for convolutional layer; "max" denotes max pooling operation; "full" indicates fully-connected layer. "relu" denotes the rectified linear transformation. These layers and operations are explained with details in Section 3.2.1.1 to Section 3.2.1.2.

3.2.1.1 Problem Formulation

The problem of predicting a label patch *M* from an image patch *N* is defined as one of learning a model of the conditional probability (Mnih, 2013):

$$P(M|N) \tag{3.1}$$

where *N* denotes a three-dimensional array of size $s_n \times s_n \times c$ with s_n being spatial dimension and c being channel dimension. Image patch *N* can be either a single channel or multiple channels, representing a patch from different types of images such as gray-scale, RGB, multispectral, or hyper-spectral imagery. *M* denotes a two-dimensional array of size $s_m \times s_m$, which takes the values of a set of possible object labels $\{0, 1, \ldots, K\}$.

Assume the label of each pixel *i* in the label patch *M* is independent given the image patch *N*. Equation 3.1 can be re-written as:

$$P(M|N) = \prod_{i=1}^{s_m^2} P(M_i|N) \quad (3.2)$$

Deep CNNs are used to model the distribution in Equation 3.2. Let f denote the functional form of deep CNNs, which maps the input image patch *N* to a distribution over the label patch *M*. The input of f is always fixed for each entry of image patch *N*. However, the output of f may vary, and is determined by the number of object classes. For a binary classification task, a logistic output unit is used to represent the probability of pixel *i* being label 1. More formally,

$$f_i = \sigma\big(a_i(N)\big) = P(M_i = 1|N) \quad (3.3)$$

where $\sigma(x)$ is a logistic activation function and written as

$$\sigma(x) = 1/(1 + \exp(-x)) \quad (3.4)$$

where f_i is the value of the *i*-th output unit; a_i is the total input to *i*-th output unit. The mathematical details of mapping an image patch *N* into a probability f_i via deep CNNs are presented in Section 3.2.1.2.

For multi-class classification tasks, a softmax output unit can be used. However, limited by the availability of high-quality multi-class training samples, the focus of this thesis is on discussing binary classification problems. Readers interested in softmax are referred to (Bishop, 2006).

3.2.1.2 The Architecture

The architecture of deep CNNs used in this thesis is similar to Mnih (2013) with slight differences; the number of filters is increased to 128 at each of the first three convolutional layers. Extensive experiments show that such modification helps improve the accuracy of object extraction across multiple tasks, without dramatically increasing the amount of computation required. Tuning the parameters or modifying the architecture may lead to further improvements in the accuracy of object extraction. However this is beyond the scope of this thesis. Readers interested in this topic are referred to Mnih (2013), Simonyan et al. (2013), Szegedy et al. (2013), and Zeiler & Fergus (2013).

Table 3.1 and Figure 3.2 present the details of the architecture. It contains five layers: the first three are convolutional, and the remaining two are fully connected. These five layers are connected in a hierarchical manner, where the output of the bottom layer is the input of the top layer. The input of first layer is an image patch[3]. The output of each intermediate layer is made of a set of two-dimensional array called feature maps. In the last layer, the features map is fed to a logistic regression to predict probabilities of pixels being label 1. Each layer may further contain multiple stages, with each stage applying different operations. For example, the first convolutional layer contains three stages; it first convolves the input image patch with a set of linear filters; it then applies a non-linear transformation to the result of convolution, followed by a max pooling in its last stage. The details of operations applied in different layers are presented as follows:

[3] Figure 3.2 only gives an example of input with RGB channels, because the thesis focuses on discussing object extraction from HSR imagery with RGB channels. However, the architecture of deep CNNs presented in this thesis is not limited to RGB imagery. It can be trivially extended to gray-scale, multi-spectral, or hyper-spectral imagery by replacing the three-channel input with a single-channel or multi-channel input.

a. Convolutional Layer

The "Conv.1" in Figure 3.2 gives an example of a convolutional layer. A typical convolutional layer contains three stages, convolution, non-linear transformation, and spatial pooling, although spatial pooling may not be used in some cases (e.g., the second and third convolutional layer of the deep CNNs used in this thesis). Let X denote the input of the convolutional layer, which is represented in a three-dimensional array of size $s_x \times s_x \times c_x$ with s_x being spatial dimension and c_x being channel dimension. Let Y denote the output of the convolutional layer, which is a three-dimensional array of size $s_y \times s_y \times c_y$ with s_y being dimension and c_y being channel dimension. Let W denote the weights of linear filters. It is represented in a four-dimensional tensor of size $s_w \times s_w \times c_x \times c_y$, which contains weights of a set of two-dimensional filters of size $s_w \times s_w$ connecting the input X with the output Y. For a typical three-stage convolutional layer, the output of the convolution layer can be expressed as,

$$Y_j = pool(g(b_j + \sum_{i=1}^{c_x} W_{ij} * X_i)) \tag{3.5}$$

where Y_j denotes the array in the j-th channel of output Y; X_i denotes the array in the i-th channel of input X; W_{ij} denotes the weights of the filter connecting input X_i to output Y_j; b_j denotes a vector of biases; $*$ denotes a two-dimensional convolution operator. The use of convolution operator leads to weight sharing across the input filed (due to the fact that weights W_{ij} of each filter are replicated across the input filed). Such weight sharing enables CNNs to learn shift-invariant features, which have proven to be very useful for solving visual problems. $g(x)$ is a point-wise non-linear activation function which can be defined in different forms; in case logistic activation function ("logistic") is used, it has the form of Equation 3.4; in case where rectified linear activation function ("relu") is used, it can be written as (Nair & Hinton, 2010),

$$g(x) = \max(x, 0) \quad (3.6)$$

$pool$ is a function that applies spatial pooling to activations (the result of the activation function). In the case where a max-pooling operator is used, it considers a neighbourhood of activations and produces one pooling per neighbourhood by taking the maximum activation within the neighbourhood. Pooling over a small neighbourhood provides additional variance resistance to small input shift and distortions, which help CNNs achieve better generalization on object recognition.

b. Fully Connected Layer

The "Full2" in Figure 3.2 shows an example of a fully connected layer. Let *X* denote the input of a fully connected layer, which is represented in a vector of size s_x. Let *W* denote a weight matrix of size $s_y \times s_x$. The output of the layer *Y* can be expressed as,

$$Y = g(b + WX) \quad (3.7)$$

where b denotes a vector of biases; $g(x)$ is a nonlinear activation function (e.g., "logistic" or "relu").

c. Full Model

With the architecture and operations of each layer fully defined, the whole process of deep CNN prediction is shown in Figure 3.2. The input of deep CNNs is a 3-channel 64×64-pixel image patch *N*, representing the spectra values of R, G, B channels within the patch. In the first layer, the input image patch is convolved with 3×128 filters of spatial dimension of 16×16 pixels at stride of 4 pixels, followed by a rectified linear transformation. The result is 128 feature maps of

spatial dimension of 13×13 pixels. A max pooling operation with pooling size of 2×2 pixels is then applied to the result at stride of 1 pixel. It produces the output of the first layer, which is 128 feature maps with spatial dimensions of 12×12 pixels. In the second layer, the input of the first layer is convolved with 128×128 filters with spatial dimensions 4×4 pixels at stride of 1 pixel, followed by a rectified linear transformation. The result is 128 feature maps with spatial dimensions of 9×9 pixels, which is fed to the third layer as input. In the third layer, the input is convolved with 128×128 filters with spatial dimensions of 2×2 pixels at a stride of 1 pixel, followed by a rectified linear transformation. The result is 128 feature maps with spatial dimensions of 3×3 pixels. These feature maps are concatenated into an 1152-dimensional feature vector, which is fed to the fourth layer as the input. In the fourth layer, the input is mapped into a 4096-dimensional vector by multiplying a weight matrix of size 1152×4096, followed by a rectified linear transformation. The resulting vector is then fed to the last layer as input. In the last layer, the input is mapped into a 256-dimensional vector by multiplying a 4096×256 weight matrix, followed by a logistic regression. The result of the last layer represents the probabilities of pixels of label map M (16×16 pixels) being label 1.

3.2.1.3 Training

There are about total of 6 million parameters in the above deep CNNs, including weights W and biases b in each layer. These parameters are learned by minimizing the total cross entropy between ground truth and predicted labels, which is given by,

$$L(W,b) = -\sum_{j=1}^{s_n^2}\sum_{i=1}^{s_m^2}(M_i ln(f_i(N;(W,b))) + (1-M_i)ln(1-f_i(N;(W,b)))) \quad (3.8)$$

The outer sum of the objective function (3.8) is over all possible image and label patch pairs in the training data. These pairs are generated at randomly by selecting and cropping patches from original size images and their corresponding label maps. 90% of the generated pairs are used for the training data and the rest are used for the validation data. Since there are a very large number of patch pairs in the training data, stochastic gradient descent with mini-batches is used as the optimizer (Bishop, 2006). There are a number of hyper parameters that need to be set for stochastic gradient descent. These hyper parameters are set to the ones used by Mnih (2013), which have shown to perform well in experiments of this thesis[4]; the mini-batch size is set to 128; the momentum w_m is set to 0.9; the weight decay w_d is set to 0.0002; the weights in each layer are initialized from a zero-mean Gaussian distribution with deviation 0.01; the biases in each layer are initialized with the constant 0; an equal learning rate is used for all layers throughout the training, which is initialized at 10^{-5}. Let *x* denote a variable that represents either the weights *W* or the biases *b*. The update rule for the weights *W* or the biases *b* can be written as:

$$v_{i+1} := w_m v_i - w_d \lambda x_i - \lambda \langle \frac{\partial L}{\partial x} | x_i \rangle_{D_i} \tag{3.9}$$

$$x_{i+1} := x_{i+1} + v_i \tag{3.10}$$

where *i* is the iteration index; *v* is the momentum variable; λ is the learning rate; $\langle \frac{\partial L}{\partial x} | x_i \rangle_{D_i}$ is the average over the *i*-th mini-batch D_i of the derivative of the objective function *L* with respect to variable *x*, evaluated at x_i. The deep CNNs are trained for a number of cycles through the training set until the validation error stops dropping.

[4] In the few cases, the author tried to tune values of these parameters for individual object extraction and saw slight improvements in accuracy. However, the order of different methods in terms of performance remained the same as for fixed hyper-parameters. For this reason, this thesis uses the same values of hyper-parameters for all the experiments of the comparison.

3.3 Object-based CNNs

As shown in Figure 3.1, the procedures of object-based CNNs at test time can be summarized into three main steps:

(1) Image segmentation. Pixels in an image are aggregated into several homogenous segments according to their similarities in terms of spectra or texture features by a region-merging algorithm. The details of image segmentation are presented in Section 3.3.1.

(2) Deep CNN prediction. Deep CNNs are applied to each segment to predict its probability of being a certain object. The architecture of deep CNNs used for object-based CNNs is the same as the one used for patch-based CNNs. This architecture requires an input of a fixed 64 × 64 pixel size. To use deep CNN under the object-based framework, imagery data in each segment must be converted into a form that is compatible with the deep CNNs. The details of the conversion are presented in Section 3.3.2.

(3) Label map generation. The predicted probabilities of all the individual segments are assembled to form the probability map of the entire image. To further obtain the label map of the image, a threshold T_p is applied to the probability map. Similar to patch-based CNNs, T_p is typically set as 0.5 representing 50% chance of being a certain object.

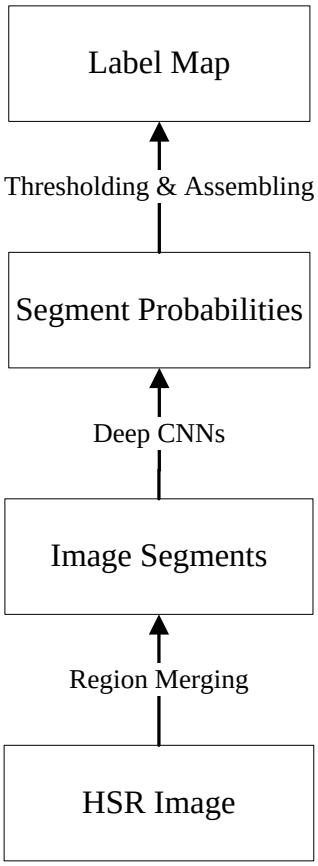


Figure 3.3. Pipeline of object-based CNNs. The whole image is first partitioned into a set of homogenous segments by using the region-merging algorithm. For each segment, deep CNNs are used to extract features, and predict its probability of being a certain object. The predicted probabilities of all the individual segments are assembled to form the probability map of the entire image. A threshold is further applied to the probability map to determine the label map of the image.

3.3.1 Image Segmentation

Many algorithms can potentially be used for image segmentation. Given the fact that many applications with HSR imagery require to process hundreds of gigabytes of data within a few weeks, algorithms that exhibit low speeds or are difficult to parallelize are in inappropriate for automated object extraction from HSR imagery. The region-merging algorithm developed by

Robinson et al. (2002) is used in this thesis, which has been integrated into the OBIA module of ENVI (one very popular remote sensing software package).

In Robinson et al. (2002), before region merging is conducted, watershed transformation is applied to partition the input image into a number of regions that widely over-segment the image. These regions are referred to as "superpixels" in remote sensing literature. The number of superpixels is usually 2 orders of magnitude less than the number of pixels in the image, and therefore it is much more efficient to work with superpixels than pixels. Superpixels are then iteratively aggregated into large segments in a greedy fashion. At each iteration, the similarity between every two adjacent segments is computed, which is defined as:

$$S_{ij} = \frac{\frac{|R_i|\cdot|R_j|}{|R_i|+|R_j|}\cdot\|u_i-u_j\|}{length(\partial(R_i,R_j))} \tag{3.11}$$

where $|R_i|$ is the area of segment *i*; u_i is the average spectra value in segment *i*; $\| u_i - u_j \|$ is the Euclidean distance between the average spectral values of segment *i* and *j*; $length(\partial(R_i, R_j))$ is the length of the common boundary between segment *i* and *j*. Adjacent segments with the highest similarity (i.e., smallest value of S_{ij}) are merged during this round of the iteration if their similarity is also smaller than a pre-defined threshold. The iteration carries on until the similarities between all adjacent segments are larger than a threshold T_s.

Tuning the threshold T_s allows for generating different levels of segmentation. For example, a large T_s would encourage the region merging, resulting in a coarse segmentation. On the contrary, a small T_s would inhibit the region merging, leading to a fine segmentation. Since objects can

appear on different scales of HSR imagery, extracting different objects would require different levels of segmentation. For example, extracting small object like cars desires a fine segmentation, while extracting large objects like commercial buildings prefers relatively coarse segmentation. To determine a suitable threshold of image segmentation for a specific object extraction, this thesis tests different values of thresholds and selects the one that achieves the best performance on the extraction. In the experiments of this chapter, only the result corresponding to the threshold that gives the best performance is selected for the comparison with patch-based CNNs' result.

3.3.2 Deep CNNs on Segments

The same architecture of deep CNNs presented in Table 3.1 is used for extracting features from each segment and predicting its probability of being a certain object. This architecture requires inputs of a fixed 64 × 64-pixel size. In order to use deep CNN features under the object-based framework, imagery data in each segment must be converted into a format that is compatible with the deep CNNs. Among many possible methods of transferring an arbitrary-shaped segment into a fixed size window, a simple one is adopted: as shown in Figure 3.4, all pixels of the candidate segment A are warped in a tight bounding box. A few pixels (denoted by the shaded area) along the shorter side of bounding box are added to make it square. Prior to warping, the square box is dilated by 25 pixels so that at the warped size there are exactly 16 pixels of warped image context surrounding the original square box. The dilated square box of 100 × 100 pixels shown in (b) is then warped into the square box of 64 × 64 pixels shown in (c). Extensive experiments show that other transformation methods don't bring any significant benefit.

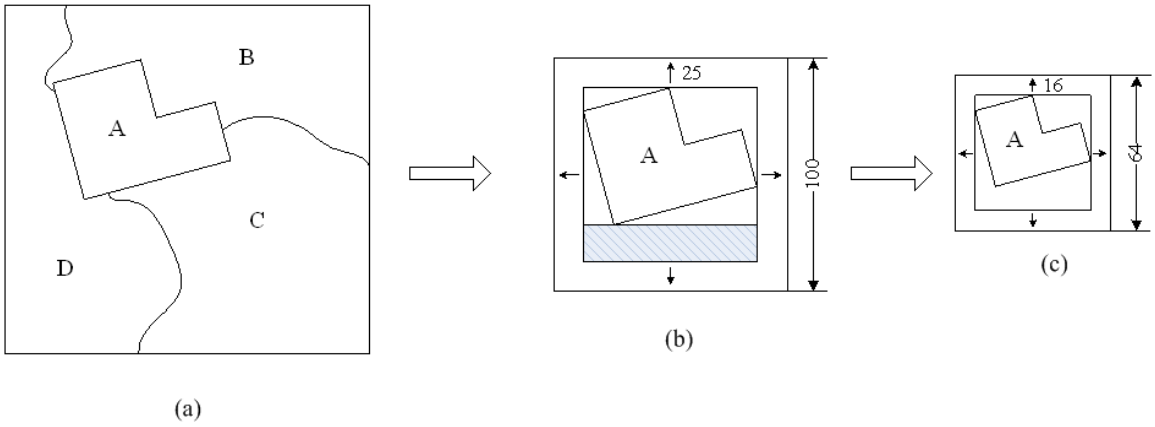


Figure 3.4. Transferring a segment to a square box. (a) Segmented image. (b) Square box prior to warping. (c) Square box after warping.

Once the segment is converted into a square box, two different strategies can be used to extract deep CNN features from the segment. The first strategy ignores the segment's foreground mask and computes deep CNN features directly from the warped window. The second strategy computes deep CNN features only on the segment's foreground mask by replacing the background with zero. As can be seen, the first strategy could potentially include more spatial contextual features from the surrounding of the segment, while the second strategy focuses on features from the region of the segment itself. However, extensive experiments show there is no significant difference between the performances of the two strategies, which is also confirmed by the study of Girshick et al. (2013). For this reason, this thesis only presents the results obtained by using the first strategy when comparing with patch-based CNNs' results.

3.3.2.1 Training

In general, the process of training deep CNNs under the framework of object-based methods is very similar to the one under patch-based methods (expect that image segments need to be converted into square boxes). More specifically, a number of HSR images and their

corresponding label maps are first collected as training samples. These images are partitioned into a set of homogenous segments using the region merging algorithm presented in Section 3.2.1. The label of each segment is determined by labels of the majority pixels within the respective segment. These generated segments are then transferred into square boxes, which are further assembled into mini-batches with their corresponding labels. These mini-batches are used to train the deep CNNs following the same method used by patch-based CNNs (see Section 3.2.1.3). The same values of hyper parameters used for the stochastic gradient descent process in patch-based CNNs are used here.

a. Pre-training and Fine-tuning

For certain type of object extraction (e.g., waterbody), the number of available training samples may be too small to learn the 6 million parameters of deep CNNs due to the issues inherent to image segmentation (as presented in Section 3.4.4.2). The "pre-training and fine-tuning" strategy is used to train deep CNN model in such a case (Hinton & Salakhutdinov, 2006; Hinton et al., 2006; Bengio et al., 2007). The core idea of the strategy is to use auxiliary data to pre-train the model and then use the available training data to "fine-tune" the pre-trained model. In case of training object-based CNNs here, the model learned by patch-based CNNs is used as the base model, which is then fine-tuned with the training data available for object-based CNNs. More specifically, the parameters learned by patch-based CNNs for the same task of object extraction are used to replace the randomly initialized parameters of object-based CNNs. The stochastic gradient descent then starts at a learning rate of 10^{-6} (1/10 of the standard training rate), which allows fine-tuning to make progress while not drowning out the initialization. The remaining

hyper parameters and mini-batches used for the stochastic gradient descent stay the same as for the standard training process presented above.

3.4 Comparison

The effectiveness of patch-based CNNs and object-based CNNs on a number of object extraction tasks with HSR imagery are compared in this section, with the goal of addressing the two questions disscussed in Section 3.1. The parameters of these two methods used in the comparsion are by default the ones described through Section 3.2 to Section 3.3 unless otherwise stated. The results of standard OBIA methods on these tasks are also presented in this section, which serve as the baseline performance for the comparsion. For the segmentation, standard OBIA uses the same region merging described in Section 3.3.1. For the feature extraction, it uses spectra mean, spectral standard deviation, texture mean, texture entropy, region area, elongation and Hu's moment. The details of these feature descriptors are referred to Jensen (2005). For the classification, it uses the k-nearest neighbor (KNN) with *k* set to be 9. These settings are very standard settings for OBIA, and have been employed by many previous studies.

3.4.1 Dataset

The dataset used for the comparison was collected from the Region of Waterloo, Ontario, Canada (courtesy of GIS department, Region of Waterloo). It contains 753 orthorectified aerial images with each being 8350 × 8350 pixels at spatial resolution (RS) of 12 cm in bands R, G, and B. These images were acquired on April 5th to April 7th, 2009 in a substantially cloud free condition, with the Vexcel UltramCam-D digital camera system flying at an elevation of approximately 4300 ft. The camera has 55° field of view (FOV) across track and 38° FOV along

track. The sun angle when images are acquired was greater than 30°; the ground condition was snow/ice free and free of leaf cover.

Three object extraction tasks are designed for the comparison, namely, extraction of buildings, roads, and water bodies. For all three tasks, 100 images are selected for training; 10 images are used for validation; and 18 images are chosen for testing (due to limit of ground truth data). Their corresponding label maps[5] are generated by rasterizing the vector maps which are created by manual digitation. For buildings and water bodies, vector polygons are available. Therefore, their label maps can be created by simply rasterizing the polygons. For roads, only road centerlines are available. To convert the centerlines into label maps, a 7-pixel buffer polygon is created around each centerline. The buffer polygons are then rasterized into road label maps. In addition, the data used for the building extraction task is resized to 0.48 m, while data used for road and waterbody extraction tasks is resized to 1.2 m in this comparison. The issues and corresponding solutions for performing object extraction on at original resolution of 12 cm will be discussed in Chapter 4.

3.4.2 Evaluation Methods

Traditional evaluation criteria used for measuring classification accuaracy of LSR imagery (Jensen, 2005), such as such as kappa coifficient, user accuracy, producer accuaracy, are not suitable for evaluting performance of object extraction from HSR imagery. In real-world applications with HSR imagery, the position of object outlines is important and has to be created precisely. Outlines that are off by a certain distance are considered as inappropriate (or even

[5]All label maps are in raster format in this thesis unless otherwise stated.

useless in some cases). The traditional evaluation criteria derived from confusion matrix cannot truly reflect such properties. Therefore, this thesis follows Wiedemann et al. (1998) to use correctness and completeness to measure the accuracy of the extracted object outlines as compared to true object outlines. Correctness and completeness may be referred to as precision and recall in other contexts (e.g., Mnih (2013)). However, it should be noted that correctness and completeness are more commonly used words in the remote sensing community. Therefore, they are used in this thesis. 3 decimal places are used to represent the precision of correctness and completeness, which are also used in many previous studies to compare different methods (e.g., Widemann et al. (1998)). Presenting results with more than 3 decimal places may not have practical meaning for the purpose of this comparison, considering the bias of data and parameter setting used for the comparison. However, it should be noted that this precision discussed in the thesis is not equivalent to the precision of actual mapping. The precision of mapping is determined by the precision of both data acquisition and digitization. This thesis, however, only focuses studying the precision of digitization when using automated extraction methods, without diving into any errors caused by data acquisition. Figure 3.5 shows an example of measuring correctness. Given a l-pixel buffer around true object outlines, correctness is defined as the fraction of predicted object outlines that are within the buffer. Similarly, given a l-pixel buffer around the predicted object outlines, completeness is defined as the fraction of true object outlines that are within the buffer. For the size of buffer, l is set to be 4 pixels for all the experiments presented in this thesis. In addition, this thesis evaluates the consistency of performance of a method on different object extraction tasks; it analyses whether a method works effectively for extraction of one type of objects can be trivially applied for extraction of another type of objects and achieve satisfactory results.

Since that the patch-based and object-based CNNs presented in this chapter predicts the probability of each pixel being an object, it is possible to trade off correctness for completeness by tuning the threshold to alter the results of the object extraction. To fully assess the performance of these two methods, two criteria are used for the comparison: the first one is to present and compare their entire correctness-completeness curves, which show the trade-off between correctness and completeness for all thresholds. The second one is to report *F*-measure on correctness and completeness or harmonic mean of precision and recall at threshold of 0.5. Although correctness and completeness at this single point don't convey full information as compared to the entire curves, this measurement is still used in the thesis because one has chosen a threshold to obtain concrete object outlines in real-world applications. A threshold of 0.5 is commonly used in many previous studies, indicating the probability of being a given object or not as 50/50. Also, since the standard of OBIA is to output an object label rather than a probability (due to use of KNN classifier), it is convenient to using this measurement to compare the three methods. Several others summary statistics for correctness-completeness curves may also be used, such as correctness at a fixed completeness level. However, the thesis doesn't choose to present any of them as this comparison leads to the same conclusions.

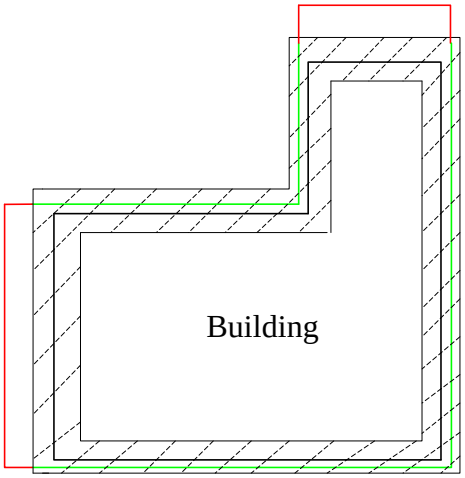


Figure 3.5. Example of measuring correctness. Given the true building outline (the dark real line), a buffer (the dashed area) is created around the outline. Correctness is measured by the ratio between the length of the predicted outline within the buffer (green line) and the total length of predicted outline (green line + red line).

3.4.3 Results

Figure 3.6 shows the correctness-completeness curves of patch-based CNNs and object-based CNNs on the tasks of extracting buildings, roads, and waterbodies from HSR aerial imagery. First of all, it can be seen from the results of patch-based CNNs that the features learned by deep CNNs are very powerful and can be generalized to perform three very different tasks with high accuracy. This observation is also supported by the visual examination of the sample results in Figure 3.7 to 3.9. Although these images contain very different objects with large intra-class variations, patch-based CNNs are able to constantly produce promising results. Close visual examination of these sample results in Figure 3.10 further demonstrates that patch-based CNNs are, to some degree, resistant to the adverse effects of shadows on the building rooftops (Figure 3.10(e)), and occlusions caused by trees along the roads (Figure 3.10(e)). Lastly, patch-based CNNs are very efficient: implementing the method on a single GPU (GeForce GTX Titan) with

CUDA and C++[6] results in the capability of processing a 2087×2087-pixel image (at spatial resolution of 0.48 m) in about half minute. These results have confirmed many previous studies regarding the power of deep learning (see Section 2.4. for the details).

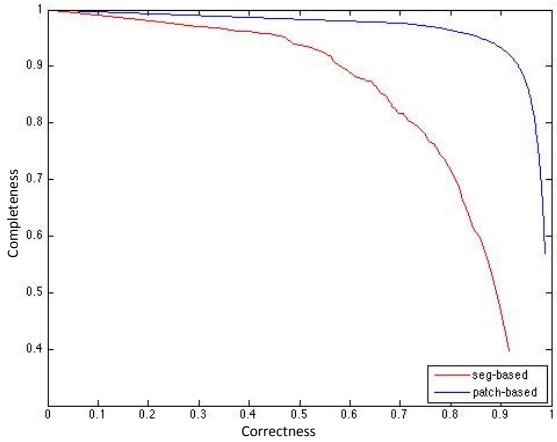
However, it is surprisingly noticed from Figure 3.6 that patch-based CNNs significantly outperform object-based CNNs on all the three tasks: the correctness of patch-based CNNs is constantly higher than object-based CNNs at any fixed recall as denoted by the curves, and vice-versa. Table 3.2 shows the comparison of the two methods at the threshold of 0.5, which further affirms the observation. As can be seen, patch-based CNNs achieve a 0.109 improvement over object-based CNNs in *F*-measure of building extraction, a 0.283 improvement of road extraction, and a 0.387 improvement of waterbody extraction. Visual examination of the sample results on buildings, roads, and waterbodies as shown in Fig. 3.7 to 3.9 also suggests that the performance of patch-based CNNs is much better than that of object-based CNNs. As can be seen from these figures, the results of patch-based CNNs contain far fewer false positives (marked by red pixels) and false negatives (marked by blue pixels) compared to the results of object-based CNNs.

[6] The deep CNN component is implemented by modifying the convnet code (Krizhevsky et al, 2012) which is available at http://www.cs.toronto.edu/~kriz/

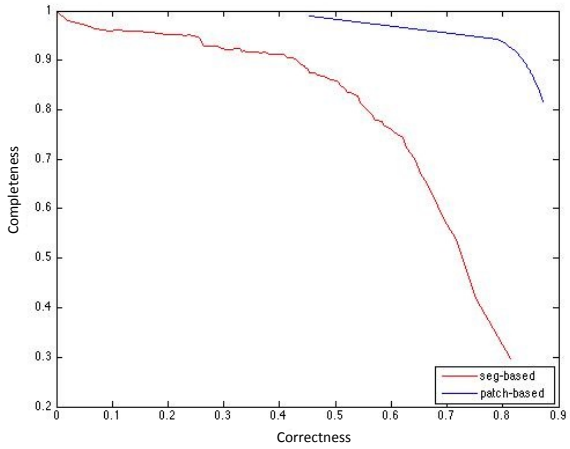
	Patch-based CNNs			Object-based CNNs			Standard OBIA		
	Corr	Comp	*F*	Corr	Comp	*F*	Corr	Comp	*F*
Building	0.901	0.930	0.915	0.823	0.789	0.806	0.707	0.530	0.606
Road	0.900	0.836	0.867	0.897	0.433	0.584	0.689	0.557	0.616
Waterbody	0.973	0.836	0.899	0.970	0.348	0.512	0.992	0.515	0.678

Table 3.2. Comparison of correctness, completeness, and corresponding *F*-measure at threshold of 0.5 between different methods. Corr. denotes the correctness; Com. denotes the completeness; *F* denotes the *F*-measure.

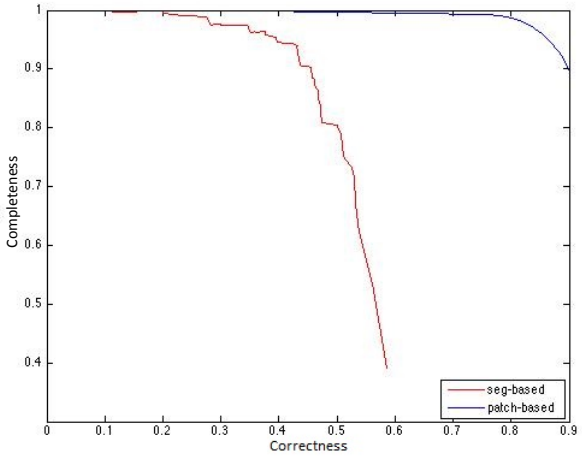
What is even more surprising is the fact that object-based methods with deep CNN features have not shown to consistently outperform standard OBIA, although these features have shown to help patch-based methods largely exceed standard OBIA in all the three object extraction tasks. For example, as can be seen from Table 3.2, using deep CNN features helped increase the *F*-measure of object-based methods on building extraction by 0.200 compared to using the standard features. However, such accuracy gains are not constant for road and waterbody extraction. In these tasks, standard OBIA outperforms object-based CNNs in terms of *F*-measure by 0.032 for road extraction and 0.166 for waterbody extraction. Meanwhile, deep CNN features have shown to help patch-based methods consistently outperform standard OBIA by 0.309, 0.251, and 0.221 for building, road, and waterbody extraction respectively. Visual inspection of sample results achieved by the three methods on different tasks as shown in Fig. 3.7 to 3.9 also strongly supports such a conclusion.



(a) Building extraction



(b) Road extraction

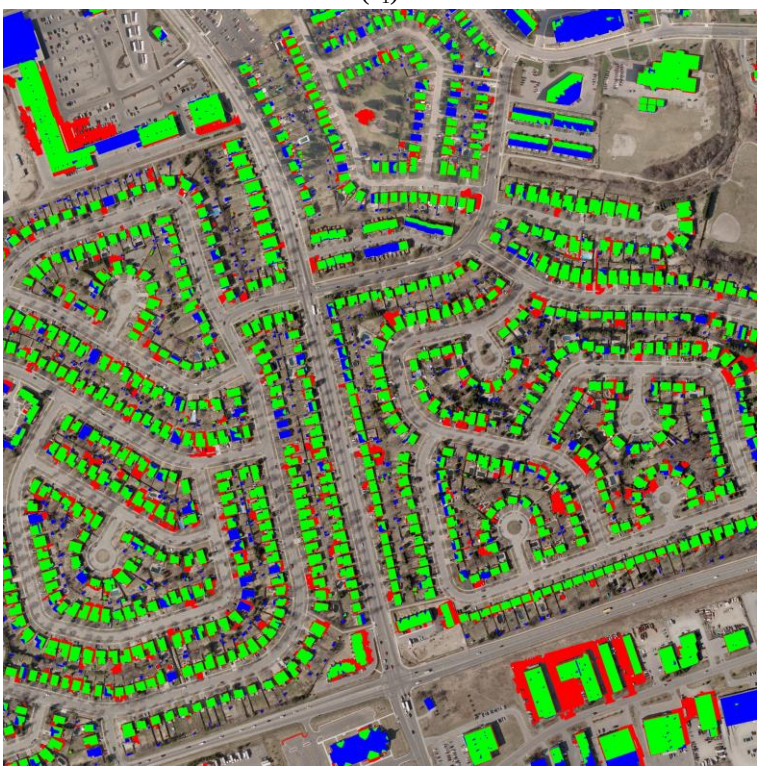


(c) Waterbody extraction

Figure 3.6. Correctness-completeness curves of patch-based and object-based CNNs on three object extraction tasks.

(a_1)

(b_1)



(c_1)

(d_1)

(a_2)

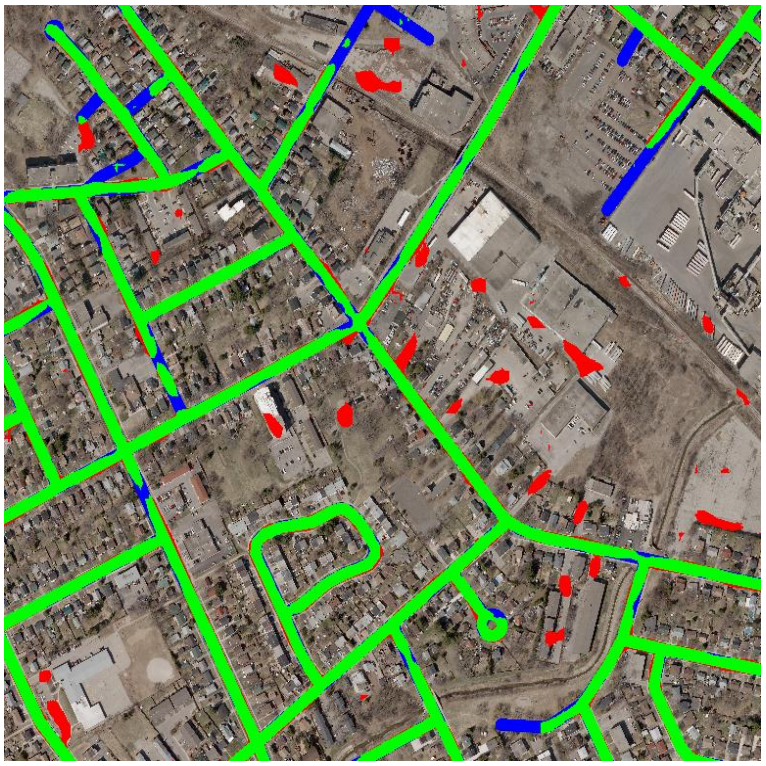
(b_2)

(c_2)

(d_2)

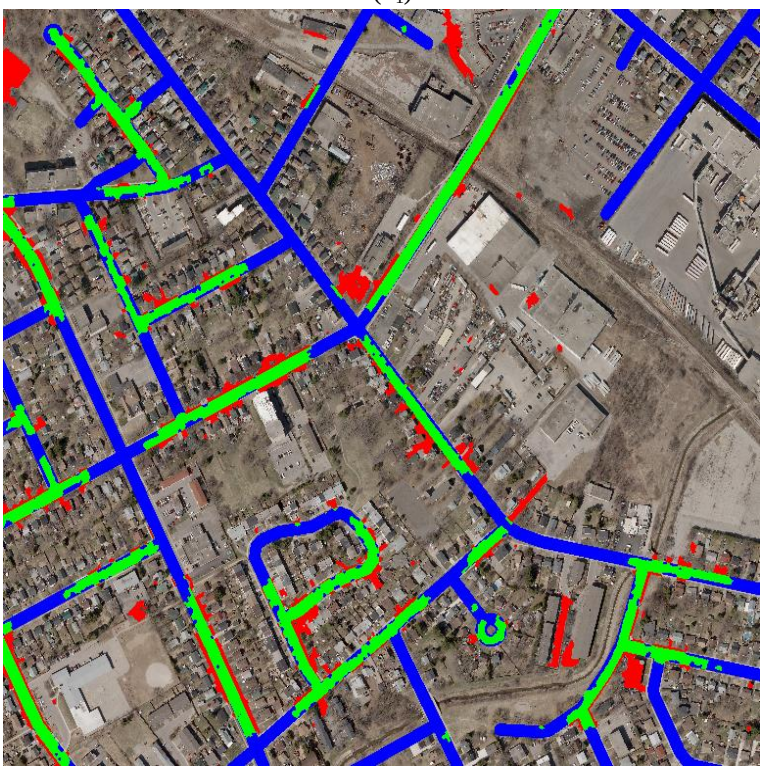
Figure 3.7. Comparison of building extraction. (a_i) Test image sample at SR of 0.48 m. (b_i) Result of patch-based CNNs. (c_i) Result of object-based CNNs. (d_i) Result of standard OBIA.

(a_1)



(b_1)

(c_1)



(d_1)

(a_2) (b_2)

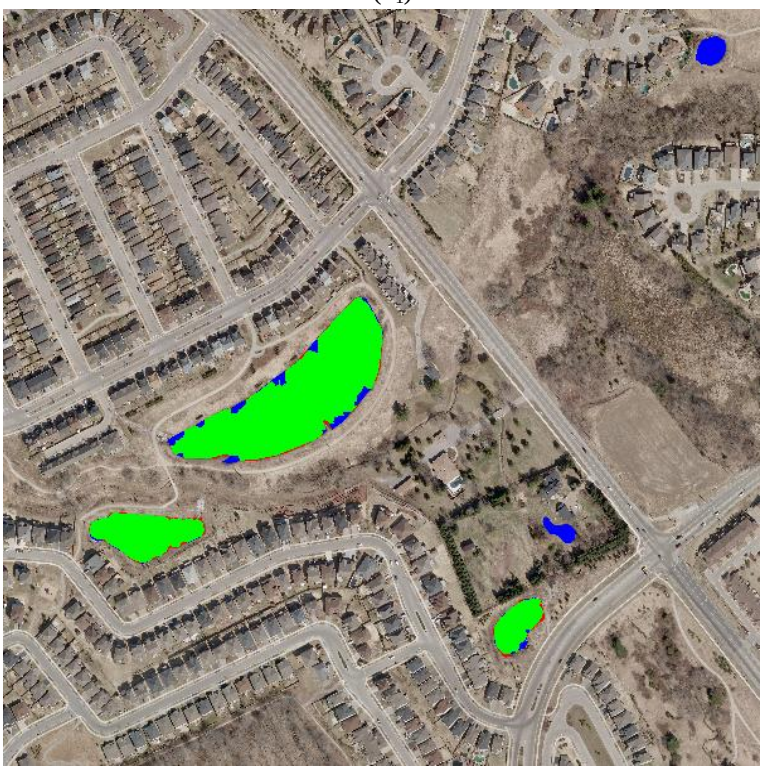
(c_1) (d_1)

Figure 3.8. Comparison of road extraction. (a_i) Test image sample at SR of 1.2 m. (b_i) Result of patch-based CNNs. (c_i) Result of object-based CNNs. (d_i) Result of standard OBIA.

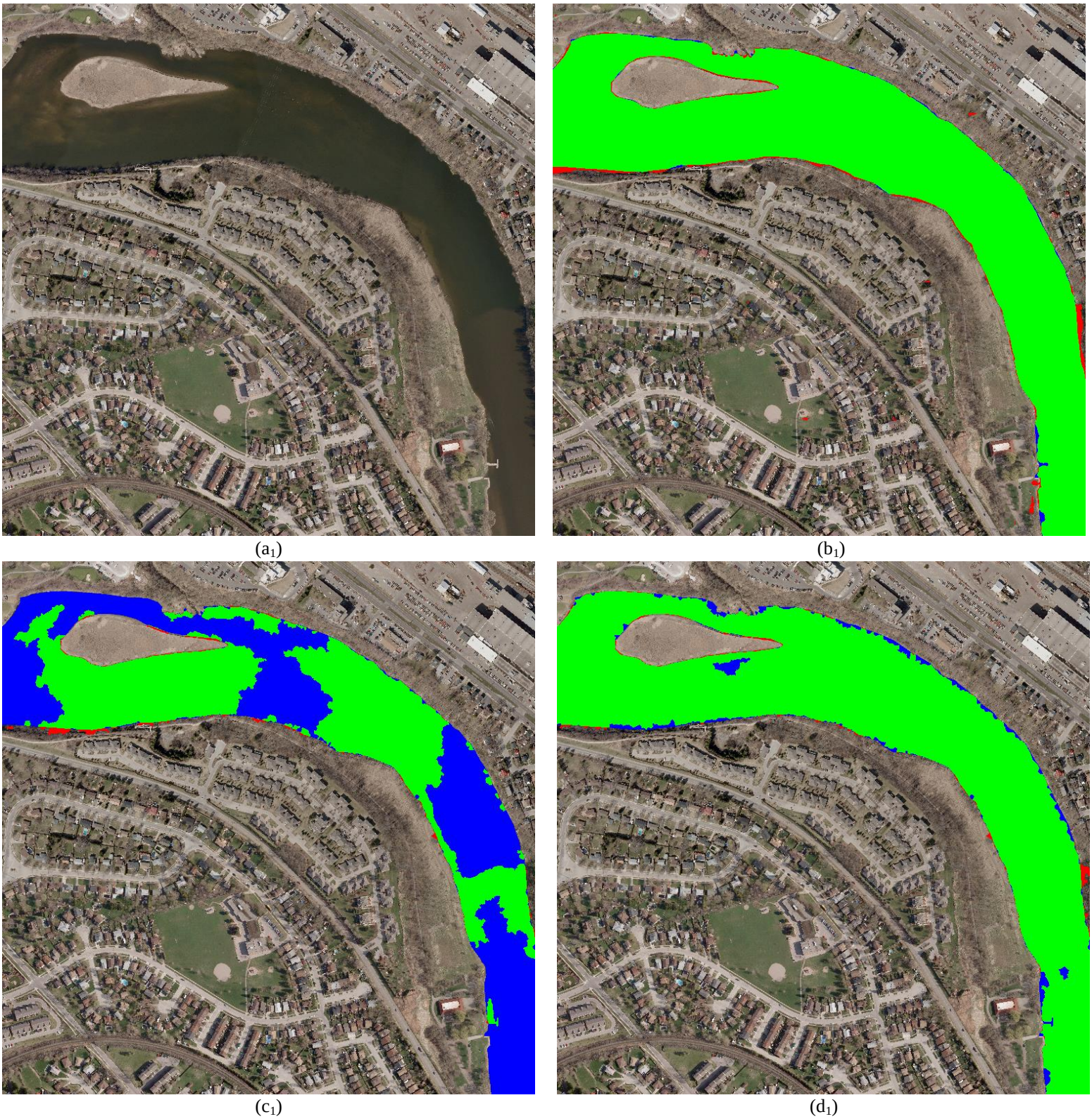
(a_1)

(b_1)

(c_1)



(d_1)



(a_1) (b_1)

(c_1) (d_1)

Figure 3.9. Comparison of waterbody extraction. (a_i) Test image sample at SR of 1.2 m. (b_i) Result of patch-based CNNs (c_i) Result of object-based CNNs. (d_i) Result of standard OBIA.

3.4.4 What's wrong with OBIA

The discovery above is very contradictory to common beliefs regarding OBIA, in which object-based methods are commonly thought of as a far superior framework for object extraction from HSR imagery than patch-based methods. This contradiction served as the catalyst for the re-examination of OBIA in this thesis: What's wrong with OBIA? What has caused object-based methods to be far less effective at leveraging the power of deep CNN features than patch-based methods? Two major issues have been observed during the course of this re-examination:

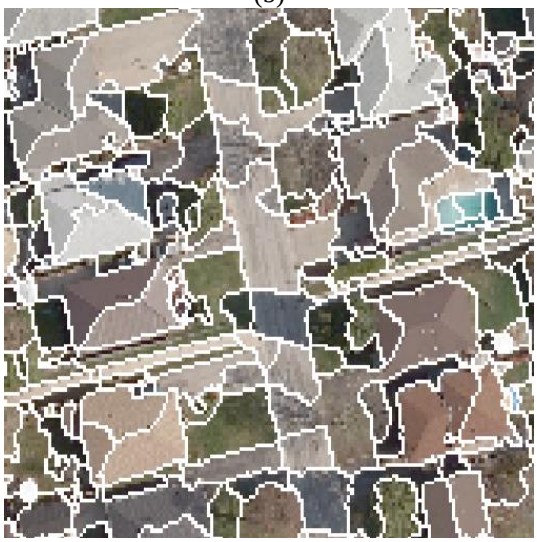
3.4.4.1 Over/Under Segmentation

The image segmentation in OBIA leverages the similarity between low-level features (e.g., spectra mean used by the region-merging algorithm described in this chapter) to group pixels into homogenous segments, with the hope that these generated regions correspond to potential objects. However, image segmentation is an ill-defined problem: the ambiguity of low-level features makes no guarantee that segments generated through such a process correspond to real objects. For example, the road in Figure 3.10(a) is partially blocked by the trees along its side. The region merging algorithm using only low-level features results in under-segmentation, in which the road segment is incomplete as shown in Figure 3.10(c). In another case shown in Figure 3.10(b), spectra features between the house rooftops and their driveways/surrounding grass are very similar. The region-merging algorithm tends to over-segment the image, which leads to the aggregation of the building rooftop with the pavement as shown in Figure 3.10 (d). Since object-based methods assign an object label to a segment as a whole, there is no way to

(a)

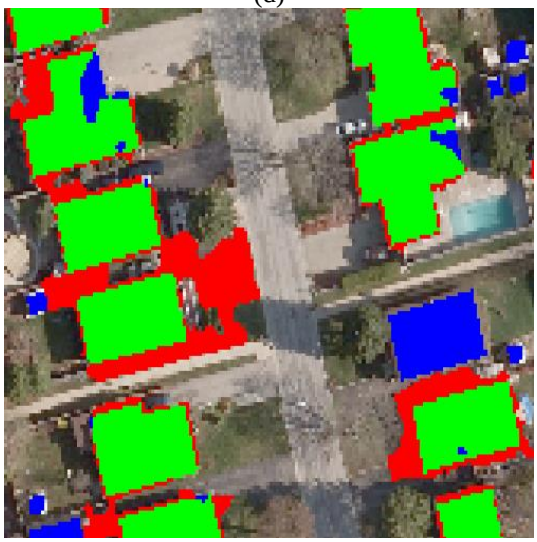
(b)

(c)

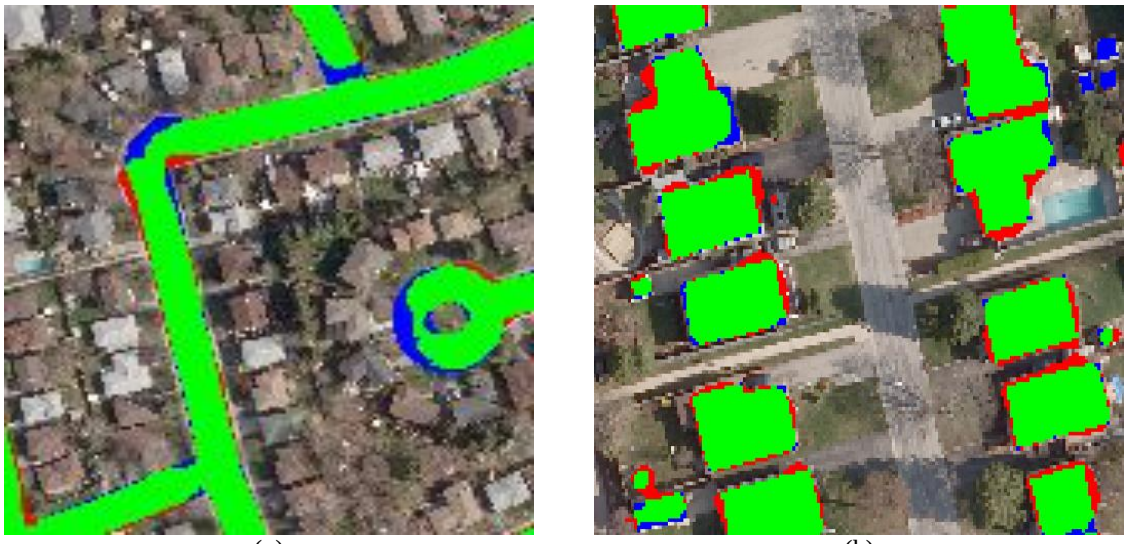


(d)

(e)



(f)



(g) (h)

Figure 3.10. Issues of under-segmentation and over-segmentation. (a) and (b) Sample images at SR of 1.2 m and 0.48 m. (c) and (d) Segmentation results: segment boundaries are overlaid on the images. (e) and (f) Object extraction results by object-based CNNs. (g) and (h) Object extraction results by patch-based CNNs.

deliver the correct result in either case where the segments are wrong as shown in Figure 3.10(e) and Figure 3.10(f). In summary, the issue of under-segmentation and over-segmentation has made the object-based methods prone to errors, particularly in the presence of shadows and occlusions. In contrast, patch-based methods leverage high-level features learned by deep CNNs to make a direct prediction of object labels pixel by pixel. When either of the above two cases happens, the ambiguity of low-level features can be easily clarified with the high-level features. Therefore, patch-based CNNs can achieve much more promising results in both of the cases, as demonstrated in Figure 3.10(g) and Figure 3.10(h).

3.4.4.2 Unstable Sample Generation

The process of training sample generation for object-based CNNs has been observed as very unstable when compared to patch-based CNNs across different object extraction tasks. Table 3.3 shows a comparison of the number of training samples generated by patch-based CNNs versus object-based CNNs. As can be seen, the number of positive samples generated by object-based CNNs drops dramatically from 1.3×10^5 for the task of building extraction to 1.1×10^4 for road extraction, and to 2.0×10^3 for waterbody extraction. In contrast, the number of positive samples generated by patch-based CNNs remains reasonably stable across the three tasks: 1.6×10^8 for building extraction, 1.3×10^7 for road extraction, and 1.1×10^7 for waterbody extraction[7]. In addition, the total number of training samples generated by object-based CNNs is much smaller than patch-based methods. As can be seen, the former contains 3 orders of magnitude samples than the latter.

	Patch-based CNNs		Object-based CNNs	
	Positive	Total	Positive	Total
Building extraction	1.6×10^8	2.0×10^9	1.3×10^5	1.1×10^6
Road extraction	1.3×10^7	2.8×10^8	1.1×10^4	2.4×10^5
Waterbody extraction	1.1×10^7	2.8×10^8	2.0×10^3	2.3×10^5

Table 3.3. Comparison of the number of training samples between patch-based CNNs and object-based CNNs.

The existence of such a difference is of course due to the fact that these two methods use very different sampling strategies: Patch-based methods generate object samples by moving a

[7] Considering the image size for road and waterbody extraction is about 4-times smaller than the one for building extraction, the change of training samples from road and waterbody extraction to building extraction in patch-based methods is small.

sampling window across the entire image at a fixed step, while object-based methods collect the samples by picking the regions produced by the process of image segmentation. As can be seen from the discussion of Section 3.4.3, these two sampling strategies work reasonably well for building extraction. However, the latter has proven to be very ineffective when it comes to road and watebody extraction. Figure 3.11 shows two exemplar results of image segmentation for road and waterbody extraction. As can be seen from Figure 3.11 (c)/(d), the whole waterbody/road is only segmented into a very few big regions due to their extreme similarity in spectra features. While it is debatable whether or not such a segmentation process provides any suitable regions for recognition of roads/waterbodies in general, the sharp drop of positive object samples as a result of this segmentation is clearly not in favor of deep CNNs. Deep CNNs are well known for their ravenous appetite for training samples (Sermanet et al., 2013). In contrast, the sample generation process of patch-based CNNs is much more stable. The number, scale, and density of training samples can be easily controlled by tuning the parameters of the patch window size and the moving step. This observation is supported by the fact that the model of object-based CNNs used for the task of road/waterbody extraction have to be fine-tuned from the model of patch-based CNNs as discussed in Section 3.3.2.1. Training such a model for object-based CNNs from the beginning with its own labeled samples has shown to produce very erroneous results during the extensive experiments. In contrast, the model of patch-based CNNs can be trained successfully from the beginning using only its own labeled samples. This observation also provides a possible explanation of why object-based CNNs largely outperform standard OBIA on the task of building extraction, but fail to do so for road and waterbody extraction: Standard OBIA using simple features and a KNN classifier can be trained with far fewer object samples than object-based CNNs. In summary, the issue of ill-defined image

segmentation leads to unstable sample generation, which again degrades the effectiveness of deep CNNs.

3.4.5 Role of Image Segmentation

So far, this chapter has shown that the performance of object-based CNNs is seriously affected by ill-defined image segmentation. Another interesting topic worth further discussion is the role of image segmentation with regard to object extraction. Conventionally, image segmentation is considered to be very important for deriving high-level features that are crucial for object extraction from HSR imagery. The entire concept of OBIA is built on top of image segmentation (Hay & Castilla, 2008; Blaschke, 2010). However, as can be seen from the experimental results in Section 3.4.3, powerful high-level features can be derived by deep CNNs under the framework of patch-based methods without using image segmentation. On the other hand, even with image segmentation, deriving useful high-level features is still problematic. Features derived from the segments in standard OBIA fail to deliver accurate results for building extraction; deep CNN features retrieved from segments in object-based CNNs fail to work effectively on road and waterbody extraction. If this is the case, what kind of role does image segmentation really play in object extraction?

(a) (b)

(c) (d)

Figure 3.11. Issues of unstable sample generation. (a) and (b) Sample images at SR of 1.2 m. (c) and (d) Segmentation results: the segment boundaries (white lines) are overlaid on the images.

This thesis suggests that image segmentation should be considered as a non-uniform sampling strategy, in contrast to the uniform sampling strategy of patch-based methods. The homogenous regions generated by image segmentation indicate possible locations of objects for sampling. However, in many cases, these segment samples are not the equivalent of real object samples or parts of real object samples. The task of assigning an object label to a segment as a whole is prone to errors due to the ambiguity of image segmentation. Such assignment is particularly unsuitable for the task of object extraction, where the labeling of each boundary pixel is of importance.

In addition, rather than suggest possible object locations, image segmentation provides inadequate insight on how high-level features can be further extracted from these locations. For example, when measuring region size, elongation, or Hu's moment on top of these segments, standard OBIA is too primitive to discriminate between complex object shapes. Also, given that segments may not correspond to the real objects, it is prone to error. Even when a segment happens to line up with an object perfectly, whether or not it presents the best form to recognize an object is still questionable. For example, the best scale to understand a road is probably not looking at the entire road segment but a fraction of it.

In summary, the concept of "object" in OBIA is very misleading; image segments are not real objects or parts of real objects, and segment-based analysis could be very different from real object-based analysis. Image segmentation in OBIA is not magic but one of many sampling strategies which provides a certain level of estimation as to where objects might be. The advocated treatment of OBIA as a new sub-discipline in studies (Hay & Castilla, 2008; Blaschke,

2010; Blaschke et al, 2014) is certainly over-exaggerating the function of image segmentation for object extraction and analysis. This thesis suggests that the name of object-based image analysis should be better rephrased as segment/region-based image analysis to reduce the confusion.

Chapter 4

Making High-quality Vector Maps

So far it has been shown that, with features learned by deep CNNs, a simple patch-based method can achieve promising results on a variety of challenging object extraction tasks from HSR imagery. However, producing high-quality vector maps from these results is yet another problematic task in itself, which limists the use of these achievements for many real applications. In this chapter, these problems are further tackled and a method for generating high quality vector maps from the results of patch-based deep CNNs is presented. The chapter starts by discussing the major issues encountered when transferring deep CNN results to vector maps in Section 4.1. The solution to these challenges is presented in Section 4.2, which combines bottom-up deep CNN prediction with top-down object reconstruction. Building vector map generation is chosen as an exemplar case to discuss. Possible ways of extending this method to other objects such as impervious surfaces and roads are also discussed. In Section 4.3, a number of experiments are conducted to evaluate the effectiveness of the proposed method for building extraction. Given that prediction and recall cannot truly refelct the performance of the proposed method for vectors map generation, a measure of post-editing time is further proposed in this study as a simple yet effetive substituted criterion. Results of building vector maps generated from very high spatial resolution (VHRS) imagery with the proposed method are presented and discussed based on this new criterion. Findings of this chapter is summarized in Section 4.4.

4.1 Problems of Generating High-quality Maps

Modern GIS systems often take vector maps as an input due to the advantages of vector format over raster format in terms of storage, visualization, and analysis (Longley, 2005). To leverage the power of modern GIS systems, most applications associated with the task of object extraction from HSR imagery require a vector map as the final format of the extraction. Although patch-based CNNs have shown to achieve very promising results on object extraction from HSR imagery in Chapter 3, generating high-quality vector maps from these results is yet another problematic task in itself. Major challenges come from two aspects:

(1) Converting raster to vector. To generate vector maps that meet the requirements of real-world applications, objects contained in the maps not only have to be precisely located, but also aesthetically pleasing. This is because, in practice, vector maps are used for both quantitative analysis and visualization. However, previous studies tend to pay attention only to the former while overlooking the latter. For example, most previous studies on object extraction would conclude with the results shown in Chapter 3; where each pixel is assigned to an object label, and accuracy is measured by correctness and completeness. When vector maps are required as the output, these methods simply vectorize the raster label map and then simplify the result using standard vector simplification algorithms such as the Douglas–Peucker algorithm (Douglas & Peucker, 1973). Figure 4.1 shows an example of building vector maps generated by such a method: Figure 4.1(a) shows the exemplar image; Figure 4.1(b) shows its corresponding label map obtained by the patch-based CNN; Figure 4.1(c) shows the vector map converted from the label map by using Douglas–Peucker algorithm. Figure 4.1(d) shows the ground truth overlaid with the image. As can be seen from Figure 4.1(c) and (d), the correctness of building outlines is

generally acceptable when compared to the ground truth. However, the appearance of the buildings is quite strange. When showing such a vector map to GIS managers from a number of different government organizations and companies, they all expressed that the quality of maps didn't meet their standards. The reason they gave was that the shapes of buildings generated didn't fit with peoples' prior knowledge of what buildings should look like buildings. For example, buildings usually have straight edges and right angles. However, the preservation of these key features is not accounted for in conventional studies of object extraction from HSR imagery, resulting in strange building shapes such as those in Figure 4.1(c).

(2) Extending to VHSR imagery. Although patch-based CNNs have shown to work effectively on object extraction from HSR imagery, applying this method to VHSR imagery (e.g., 12 cm aerial imagery) is not a straightforward task. As discussed in previous chapters, it is important to include enough spatial context in order to recognize an object in an image. However, to capture enough context from a VHSR image at spatial resolution of 12 cm, the 64×64-pixel window used for 0.5 m-resolution imagery must be extended to a 256×256-pixel window. And, even with modern GPUs, applying deep CNNs on such a large window is very computationally expensive.

4.2 Combining Bottom-up and Top-down

The causes of the above two issues are both tied to the fact that object prior knowledge has not been included the patch-based CNNs. Without the prior knowledge, everything has to be learned from the data. However, learning those key features in case (1) would requires much more training sample inputs which, in reality, may not always be available.

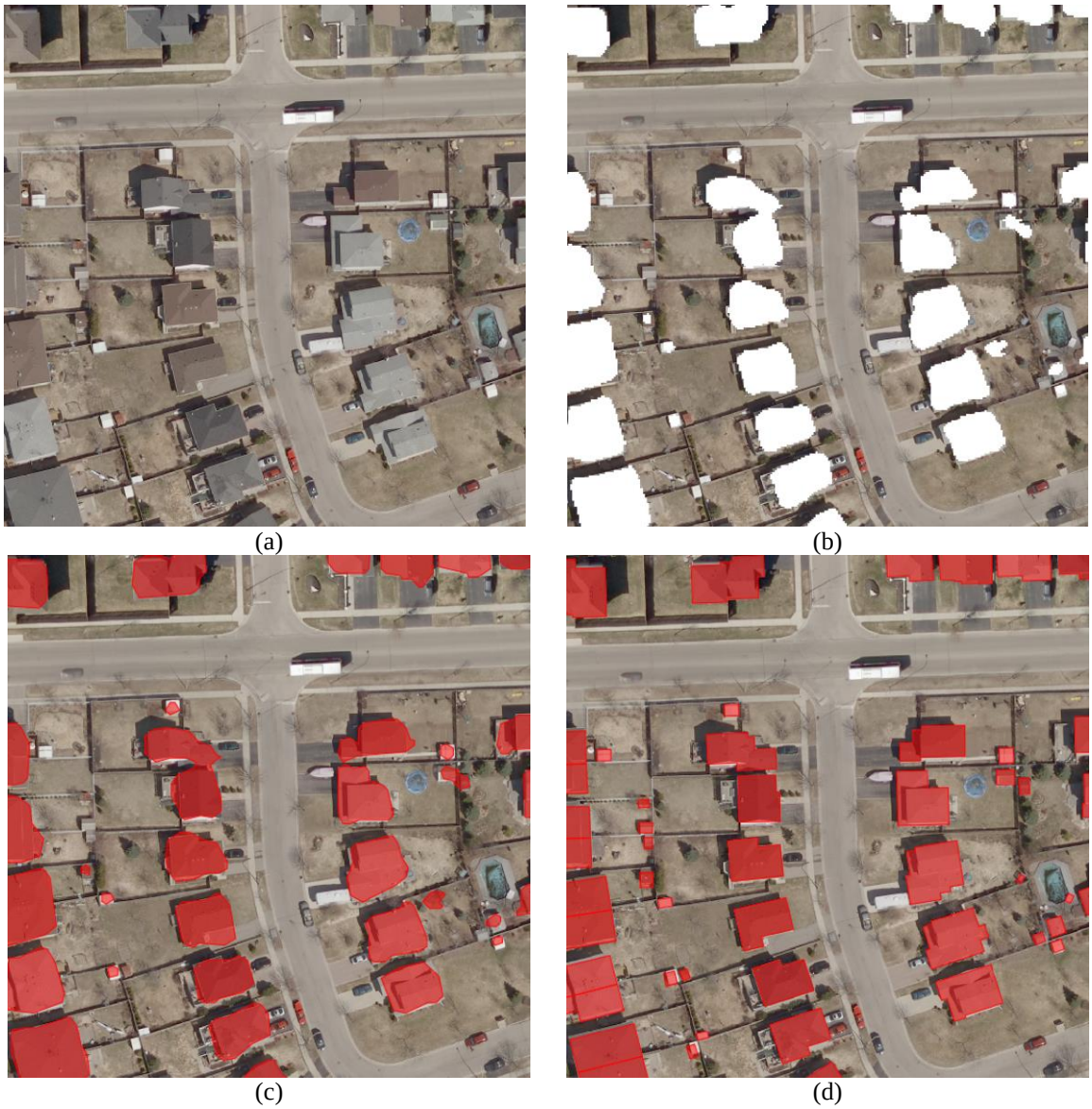


Figure 4.1. Issues with converting a label map to a vector map. (a) Sample image at SR of 12 cm. (b) Label map of the image. (c) Vector map converted from the label map overlaid with the image. (d) Ground truth overlaid with the image.

Learning object details from a much finer resolution as shown in case (2) can result in a large increase in required computation. Therefore, when imposing prior knowledge on top of the results obtained by patch-based CNNs, it can be expected that computational requirements will be mitigated. For example, issue (1) can be alleviated by leveraging prior knowledge of buildings'

geometric characteristics to force the results into straight lines and right angles. The effects of issue (2) can be lessened by first making a prediction with deep CNNs on a coarse resolution, and then inferring the exact location of the buildings on a finer resolution with prior knowledge of buildings' spectral characteristics. Incorporating this prior knowledge is also expected to further help reduce the adverse effects of shadows and occlusions as discussed by the "chicken and egg" problem in Section 2.3.3.

One possible way to integrate prior knowledge into object extraction is through a top-down modeling process as discussed in Section 2.3.3. However, top-down process has been known to be notoriously computationally expensive due to the need to search for model matching in multiple locations, orientations, and scales. This issue is expected be mitigated when harnessing the power of patch-based CNNs. As can be seen from Section 3.4, the results of patch-based CNNs already provide a sufficient estimation of the likely location of each object. Leveraging this initial estimation can help top-down modeling avoid exhaustive searching for model matching, largely reducing the required computation. Therefore, this thesis proposes to combine bottom-up deep CNN prediction with top-down object modeling for the generation of high-quality vector maps from VHSR imagery.

However, one question that needs to be further addressed is how to model prior knowledge of an object into the top-down process. Modeling such knowledge in a way is a very challenging problem, and is beyond the scope of this study. The focus of this study is to demonstrate that (1) prior knowledge is needed for deep CNNs (and presumably for any automated object extraction method) to produce high-quality vector maps (2) the combination of bottom-up and top-down

processes for object extraction is a superior framework in comparison to the individual processes. Given limited time, this thesis chooses building extraction as an example for the purpose of this study with prior knowledge of buildings modeled in a simplified category-specific way. The extension of this method to other objects of interests is also briefly discussed in the latter part of chapter, with several preliminary sample results shown in the Appendix. Figure 4.2 demonstrates the combined bottom-up and top-down process for building extraction:

In the bottom-up process, the input image is first resized to a coarse scale where that enough spatial context can be captured within a 64×64-pixel window. The patch-based deep CNNs are then applied to predict the probability of each pixel being a building on the coarse scale. The predicted probability map is then warped to the original fine scale. At the last step, a threshold is applied to the probability map to locate building candidates for the top-down modeling.

In the top-down process, buildings' spectral prior is used to infer more precise labeling for each building candidate on the fine scale from the result of coarse scale. The generated label result is further converted into polygons based on buildings' geometric prior. In the last step, individual building polygons are simply assembled into a building vector map.

The details of the combined bottom-up and top-down process are presented in the following subsections.

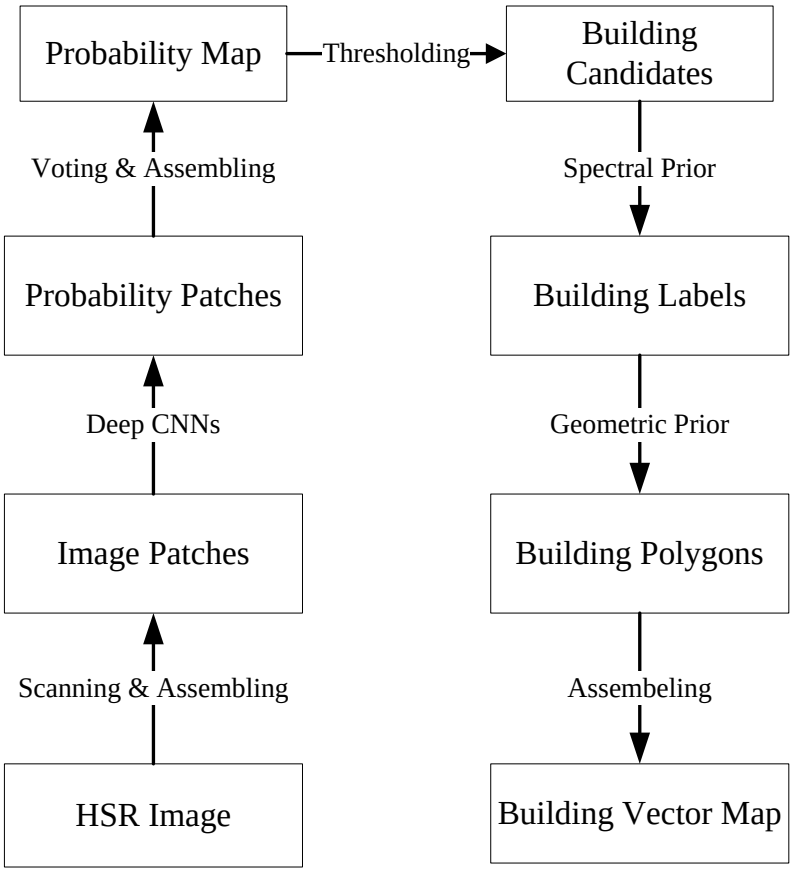


Figure 4.2. Combined bottom-up and top-down process for building extraction.

4.2.1 Bottom-up Prediction

Patch-based CNNs is used for the bottom-up prediction here. It follows nearly the exact same process as presented in Section 3.2, except that they operate on a shrunken resolution *z* rather than the original in Section 3.3. *z* is set to 0.48 m in the experiments of this Chapter, which are found to be an appropriate scale for extracting residential buildings. Tuning the scaling factor *z* would allow capturing spatial context for object extraction on different scales, which could lead to different prediction results. The details of the influence of the scaling factor are presented in Section 4.3.3.1.

After the probability map has been generated on the shrunken scale, it is resized back to the original scale to locate possible building candidates. More specifically, a threshold T_p is first applied to the probability map to generate a label map on the fine scale. This label map is generated by marking each pixel with a probability above the threshold as a building, and those below the threshold as background. Connected building regions are then found in the label map; regions with a size bigger than a second threshold of T_a are treated as building candidates, while regions below the threshold are filtered out. These procedures for locating building candidates are under the assumption that the building candidates should each be a reasonably large size, and have a high associated probability of being a building. For all the experiments in this chapter, T_p and T_a are set to be 0.5 and 20 square feet respectively. Empirically, these thresholds were found to provide good building extraction results on a large variety of images. For each candidate a tight bounding box is found, which is then expanded by 100 pixels to include more context within the box. The image data *I* and its corresponding probability map *P* and label map *L* within the bounding box will be used for modeling the building candidate in subsequent sections.

4.2.2 Spectral Prior

Given a building candidate and its associated image *I* and probability map *P* on the fine scale, prior knowledge of buildings' spectral characteristics is used to infer a precise building label. Two types of prior knowledge can be used to guide the labeling process: first, internal pixels of buildings likely have similar spectral values; and second, boundary pixels of buildings likely have large image gradients. This prior knowledge is derived from the fact that objects made of the same material are likely to have similar spectral values in a respective image. Within this context of building labeling, a good label should conform both to the prediction of the deep

CNNs, and the prior knowledge of building appearance. The challenge of deriving high quality labels is a classic object segmentation problem. It can be formulated as energy minimization, with energy function *E* defined in a way such that its minimum corresponds to "the good labeling".

Among the various ways of defining the energy function *E*, this thesis follows the idea of "Grabcut" (Rother et al., 2004) with some new developments to fit it to the purpose herein. Let *I* denote the input image defined on the domain Ω and l denote a labeling that partitions the image domain into two disjoint pairwise regions $\Omega_k = \{x|l(x) = k\}$, with $\Omega = \cup_{k=0}^{1} \Omega_k$, $\Omega_0 \cap \Omega_1 = \phi$. For a binary classification, l takes a value belonging to $\{0, 1\}$. The energy function can be defined as:

$$E(l) = \sum_{k=0}^{1} \lambda_1 \int_{\Omega_k} f(l)\, dx + \lambda_2 \int_{\Omega_k} f_2(l)\, dx + f_3(\Omega_k) \qquad (4.1)$$

The functions f_1, f_2, and f_3 are called bottom-up, homogeneity, and edge terms, respectively, which are defined as in the subsequent sections. λ_1 and λ_2 are the constant weights, which determine the influence of each term on the labeling result. Empirically, λ_1 and λ_2 are set to be 0.3 and 0.7, which are found to provide good results on a variety of HSR images in the extensive experiments.

(a) Bottom-up Term: The bottom-up term $f_1(l)$ influences the labeling process to resemble the prediction produced by the bottom-up process. $f_1(l; O)$ is chosen such that, given the building outline O, pixels that exist near the outline are more likely to have a building label than pixels lying far from the outline. More specifically, the bottom-up potential is defined as

$$f_1(l;O) = -\log(\Pr(l|O)) \tag{4.2}$$

Inspired by the work of Kumar et al. (2010), $\Pr(f_i|\Omega)$ is given by

$$\Pr(l(x)=k|O) = \begin{cases} \frac{1}{1+\exp(\alpha * dist(x,O))} & if\ k=1 \\ 1-\frac{1}{1+\exp(\alpha * dist(x,O))} & if\ k=0 \end{cases} \tag{4.3}$$

where $dist(i,O)$ is the spatial distance between a pixel i and its nearest pixel on the building outline O. It is negative if the pixel i is inside O, and positive if it is outside. The weight α determines how many of the pixels inside O will be penalized compared to the pixels outside O. Following Kumar et al. (2010), it is set to 0.3 in this chapter. The building outline O from the label map L has been produced by the bottom-up prediction described in Section 4.2.1[8].

(2) Homogeneity Terms: The homogeneity term $f_2(l)$ encourages internal building pixels to have similar spectral values. Considering that a building may contain different parts, which are made of different materials, Gaussian Mixture Models (GMMs) with S (S = 5 in this chapter) components are chosen to model the spectral distributions of the building and its surrounding background. The homogeneity term $f_2(l;I)$ is defined as

$$f_2(l;I) = -\log D(l(x)=k|I(x)) \tag{4.4}$$

where D is the function of the likelihood that a pixel belongs to the building/background. It is given by

$$D(l(x)=k|I(x)) = \sum_{s=1}^{S} \pi_{s,k} \frac{1}{\sqrt{det\sum_{s,k}}} e^{\left(-\frac{1}{2}|I(x)-\mu_{s,k}|^T \sum_{s,k}^{-1}|I(x)-\mu_{s,k}|\right)} \tag{4.5}$$

[8] The building outline can be derived from its corresponding label map by simply tracing the boundary of the label region.

where $\mu_{s,k}$ and $det \sum_{s,k}$ denote mean and the determinant of the covariance matrix of the s-th GMM component of class k, respectively; $\pi_{s,k}$ is a weight coefficient of the s-th component of class k. The parameters of the GMM for the building class are estimated from pixels with probability greater than threshold T_h, while the parameters of the GMM for the background class are estimated from pixels with probabilities smaller than threshold T_l. Setting $T_h(T_l)$ too small (large) would result in misclassified pixels to be included for the GMM parameter estimation. Through the experiments, 0.9 and 0.1 are found to be appropriate values for T_h and T_l, respectively.

Although a standard approach to estimate the parameters of a GMM is achieved through Expectation Maximization algorithm (Dempster et al., 1977), this thesis uses k-Means clustering algorithm (Bishop, 2006) instead in order to accelerate the parameter estimation process. More specially, each pixel used for the parameter estimation of building/background is assigned a unique GMM component by the k-Means algorithm. Given the assignment, the mean $\mu_{s,k}$ and the determinant of the covariance matrix $det \sum_{s,k}$ are then computed by employing a maximum likelihood estimation (Bishop, 2006); the weight coefficient $\pi_{s,k}$ is calculated by the ratio of the number of pixels assigned to the k-th component to the number of pixels assigned to the building/background class.

(3) Edge Term: The edge potential $f_3(\Omega_k)$ urges building boundary pixels to coincide with large image gradients. It is defined such that two pixels are more likely to belong to two different object labels (building/background in this case) if the image gradient between the two pixels is large. More specifically, edge potential $f_3(\Omega_k)$ is defined as:

$$f_3(\Omega_k) = \frac{1}{2} Per_g(\Omega_k) \quad (4.6)$$

where $Per(\Omega_k)$ denotes the perimeter of each set Ω_k measured with an edge-dependent metric defined by non-negative function. Following Rother et al. (2004), the function is defined as

$$g(x) = \gamma \exp(\frac{-|\nabla I(x)|^2}{\sigma}) \quad (4.7)$$

where γ and σ are positive constants. Following Rother et al. (2004) and Blake et al. (2004), γ is set to be 50; σ is chosen to be $\sigma = (2\langle \nabla I(x)|^2 \rangle)^{-1}$ where $\langle \cdot \rangle$ denotes expectation over an image example.

4.2.2.1 Minimization via Convex Relaxation

Until now, the energy function *E* has been completely defined. The minimization of the energy function is known to be NP hard. However, its approximate solution can be obtained by means of a convex relaxation strategy (Zach et al., 2008; Nieuwenhuis et al. 2013). The key idea is to encode the region Ω_k with the indicator function $u \in BV(\Omega, \{0,1\})$, where

$$u_k(x) = \begin{cases} 1, & if\ l(x) = k \\ 0, & otherwise \end{cases} \quad \forall\ k = 0,1 \quad (4.8)$$

Here *BV* denotes the functions of bounded variation. Let Du_k denote the distributional derivative of u_k and $\xi_k \in C_c^1(\Omega, \mathbb{R}^2)$ the due variable with the space C_c^1 of smooth functions with compact support. Following Federer (1996) and Zach et al. (2008), it can be shown the weighted perimeter of Ω_k is equivalent to the weighted total variation

$$\frac{1}{2} Per_g(\Omega_k) = \frac{1}{2} Per_g\{x | u_k(x) = 1\} \quad (4.9)$$

$$= \frac{1}{2} TV_g(u_k) \quad (4.10)$$

$$= \frac{1}{2}\int_{\Omega} g|Du_k| \tag{4.11}$$

$$= \frac{1}{2} sup_{\xi_k \in \kappa_g}\left(-\int_{\Omega} u_k div\, \xi_k dx\right) \tag{4.12}$$

with $\kappa_g = \left\{\xi_k \in C_c^1(\Omega, \mathbb{R}^2) || \xi_k(x)| \leq \frac{g(x)}{2}, x \in \Omega\right\}$

Finding a labeling *l* to minimizing is equivalent to

$$\min_{u\in\beta} E(u) = \tag{4.13}$$

$$\min_{u\in\beta} \sup_{\xi_k \in \kappa_g} \{\sum_{k=0}^{1} \lambda_1 \int_{\Omega} u_k f_1(l)\, dx + \lambda_2 \int_{\Omega} u_k f_2(l)\, dx - \int_{\Omega} u_k div\, \xi_k dx\} \tag{4.14}$$

with $\beta = \{u_k \in BV(\Omega, \{0,1\}) | \sum_0^1 u_k = 1\}$

To obtained a relaxed convex optimization problem that can be minimized globally, the set can be relaxed to the convex set

$$\tilde{\beta} = \{u_k \in BV(\Omega, [0,1]) | \sum_0^1 u_k = 1\} \tag{4.15}$$

The relaxed convex optimization problem can be solved numerically by employing a primal dual-algorithm (Pock et al., 2009). Essentially, this algorithm alternates a projected gradient descent in the primal variables *u* with a projected gradient ascent in the dual variable ξ in the course of updating the two variables. It also contains an over-relaxation step in the primal variables giving rise to auxiliary variables $\tilde{u}$, which assures fast convergence of the algorithm

$$\xi^{t+1}(x) = \Pi_{\kappa_g}(\xi^t + \tau_d \nabla \tilde{u}^t)$$

$$u^{t+1} = \Pi_{\tilde{\beta}}(u^t - \tau_p(\lambda_1 f_1 + \lambda_2 f_2 - div\xi^{t+1}))$$

$$\tilde{u}^{t+1} = 2u^{t+1} - u^t \tag{4.16}$$

where Π denotes the projections onto the respective convex sets; τ_d and τ_p denote the primal and dual step sizes, respectively. The projection onto κ_g is carried out by simple clipping, while the projection onto the simplex $\tilde{\beta}$ is given in Michelot (1986). The step sizes τ_d and τ_p are set to 0.5 and 0.25 respectively in this thesis, which allow the algorithm (4.16) to provably converge to a minimizer of the relaxed problem as shown in Pock et al. (2009). The primal-dual gap can be further computed to formulate suitable convergence criteria for the algorithm according to Pock et al. (2009). However, to avoid the computation caused by the convergence test at each step, this thesis takes a simple approach; iterations are terminated if the maximum number of the iterations T = 300 is reached. Experiments show this is a versatile setting for a wide variety of images. To further obtain a binary label at the end of the iterations, one can simply threshold u to 0 or 1.

4.2.3 Geometric Prior

After the precise label has been obtained for a building candidate on the finer scale, it is then converted into a vector polygon based on the prior knowledge of buildings' geometric characteristics presented in this section. Various prior knowledge regarding geometric characteristics may be used to guide the building vectorization process. For example, buildings usually have straight lines, right angles, and symmetrical shapes. This prior knowledge is subsequent to the fact that buildings are man-made objects and have very structured elements. In fact, developing sophisticated geometric models to integrate all of this prior knowledge for the building vectorization could be another topic of research in itself (Musialski et al., 2013). However, this study simplifies the problem by focusing on modeling each building with only two principal directions that are perpendicular to each other. This simplification is also due to the observation that the majority of buildings in North America are of perpendicular construction.

And, as can be seen from Section 4.3.3.3, having a good model for these types of buildings already helps save a large amount of manual digitization time in many real-world applications.

Therefore, the algorithm for geometric building modeling proposed in this section assumes buildings only have two principal directions that are perpendicular to each other. Figure 4.3 shows the pipeline of applying the modeling to a building candidate. It contains three major steps: first, the principal directions of a building are estimated based

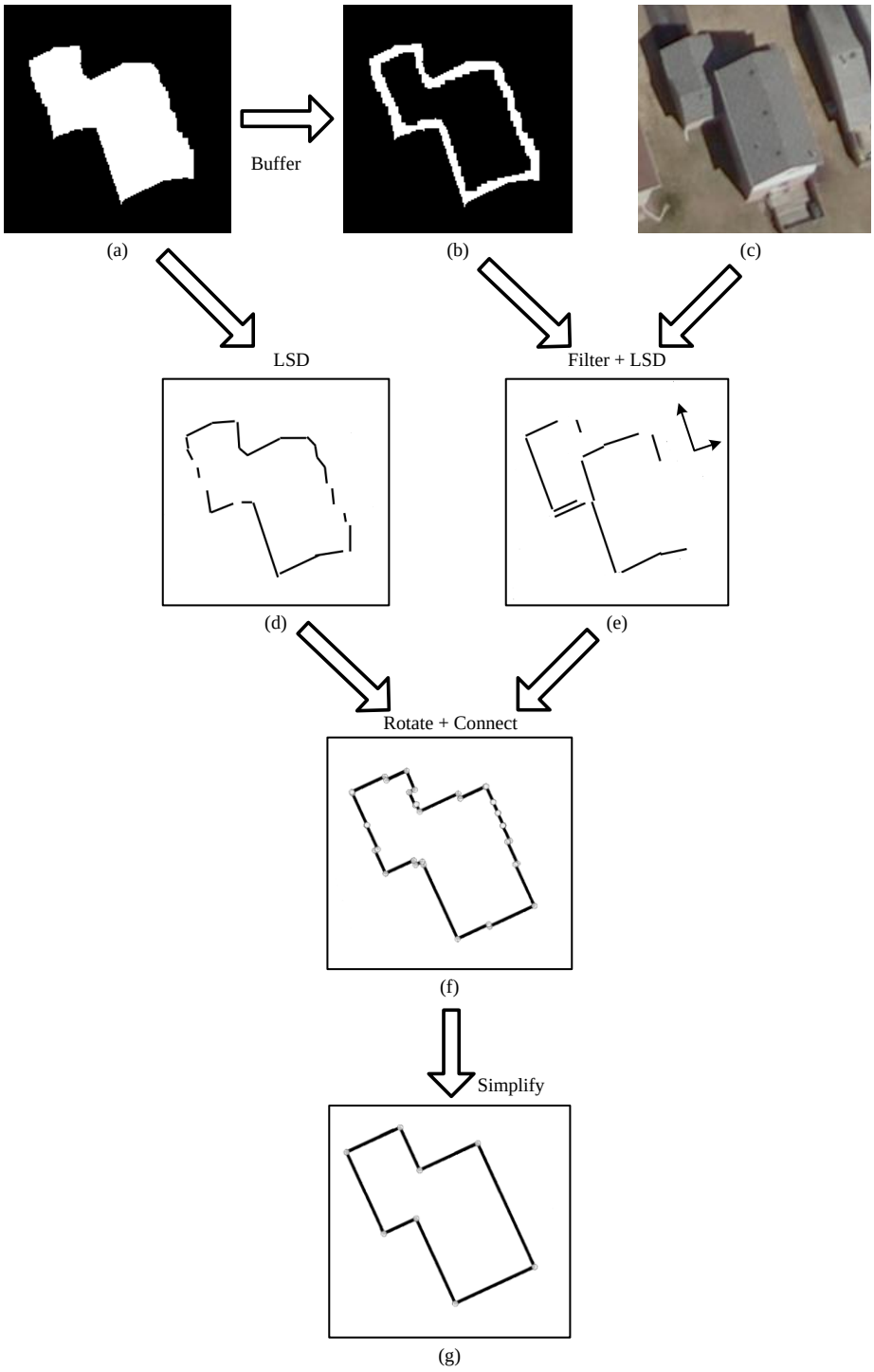


Figure 4.3. Pipeline of applying geometric prior. (a) Label map. (b) Buffer. (c) Image. (d) Decomposed line segments. (e) Principal directions. (f) Rotated outline. (g) Simplified outline.

on the building image data and its corresponding label map; second, the original building outline is rotated to one of the two estimated directions; and last, the rotated building outline is simplified by a greedy building simplification algorithm. The details of each step are as follows:

(1) Finding principal directions: a 5-pixel buffer (Figure 4.3(b)) is first created around the building outline, which is derived from the building label map (Figure 4.3(a)) obtained from the result of Section 4.2.2. Line segments that fall into the buffer area are then detected from the image data (Figure 4.3(c)) by using a line segment detection (LSD) algorithm (Von Gioi et al., 2010). A histogram of direction angles is then built from the detected line segments with the number of the angles weighted by the length of the respective detected line segments. Lastly, the principal directions of the building (Figure 4.3 (e)) are determined by finding two perpendicular directions whose angles conform to the most angles in the histogram. Note that principal directions are estimated from the image data rather than the label map because the lines detected from the image are found to be more stable than the lines detected from the label map. The latter can be easily affected by the result of mis-classification and thus is not suitable for principal direction estimation.

(2) Rotating building outlines: This building outline is then decomposed into a set of line segments (Figure 4.3(d)) using a LSD algorithm. Each line segment is then rotated to fit one of the two principal directions as shown. In the final step, two consecutive line segments are connected by defining the shortest path within the principal directions to form a new building outline (Figure 4.3(f)).

(3) Simplifying building outlines: A building simplification algorithm is then applied to merge the rotated building outline. This merging follows the rules such that two consecutive parallel line segments will be merged if (a) the vertical distance between them is smaller than a vertical threshold T_v, or (b) the length of the shorter line segment is less than a horizontal threshold T_h. After merging, the shorter line will be shifted to form an extension of the longer line. This merging occurs iteratively in a greedy fashion until no line segments can be further merged. At each iteration, only the longest active[9] line segment is allowed to merge. The reason for designing such merging rules is that longer lines are considered to be more stable than the shorter lines in terms of estimating the true location of building outline. Therefore, the algorithm should always try to maintain the position of longer lines and merge others with them first. Figure 4.3(g) shows the final result of simplification. This result can be easily converted into a vector polygon by recording the coordinates of each line segment in a clockwise order.

The vertical and horizontal thresholds T_v and T_h control the degree of the building simplification. Figure 4.4 shows a series of results achieved by using different settings of the thresholds. As can be seen, setting the thresholds too small leads to results being prone to the adverse effects of shadows and mis-classification (Figure 4.4 (b)); setting the thresholds too large would over-constrain the results and lead to artificial structures (Figure (f)); setting the thresholds T_v and T_h within the range approximately from 0.5 m to 1 m and 1 m to 2 m respectively provides suitable results for building extraction (Figure (c), (d), and (e)). Following such an observation, T_v and T_h are set to 0.96 m and 1.92 m respectively in the experiments of this chapter.

[9] A line segment is defined as active if merging can occur with a consecutive parallel line.

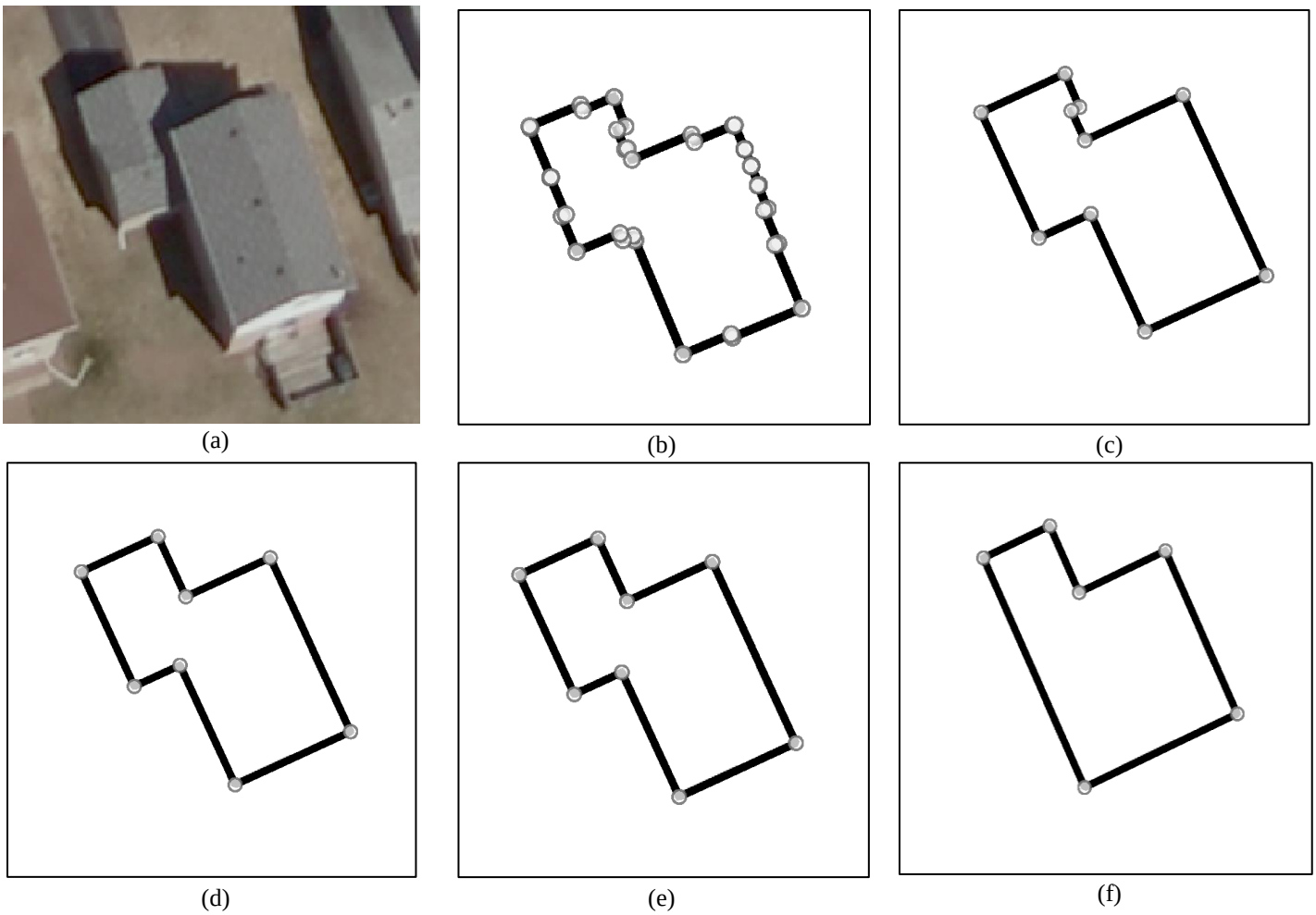


Figure 4.4. Building geometric modeling with different parameter setting. (a) Test image at SR of 12 cm. (b) Result with $T_v = 0$ m and $T_h = 0$ m. (c) Result with $T_v = 0.24$ m and $T_h = 0.48$ m. (d) Result with $T_v = 0.48$ m and $T_h = 0.96$ m. (e) Result with $T_v = 0.96$ m and $T_h = 1.92$ m (f) Result with $T_v = 1.92$ m and $T_h = 3.84$ m.

4.2.4 Extension to Other Objects

At this point, the entire top-down building modeling process has been presented. Although this process is designed for building modeling only, some of the ideas can be borrowed to model other objects of interest, and generate their respective vector maps from VHSR imagery. For example, vector maps of natural objects such as waterbodies, forests, agricultural land, and

impervious surfaces can be generated by simply applying the same spectral prior knowledge presented in Section 4.2.2 to the results of bottom-up prediction. Road centerlines can also be produced by applying geometric prior of connectivity to the road label map obtained from the bottom-up perdition. Figure 7.1 and 7.2 in the Appendix show some preliminary results of impervious surface and road centerline maps generated by such top-down modeling processes. The details of the processes are left out of this thesis.

4.3 Experiments

A number of experiments are designed to verify the effectiveness of the proposed method, which combines bottom-up deep CNN prediction with top-down object modeling. Unless otherwise stated, the parameters of the proposed method used in the experiments are by default the ones described through Section 4.2. The results of the experiments are discussed in the following subsections.

4.3.1 Dataset

The image data used for generating high-quality building vector maps in this chapter is the same as data used for the comparison of patch-based and object-based CNNs in Chapter 3: It contains 753 aerial images, each being 8350 × 8350 pixels at a spatial resolution of 12 cm in bands R, G, and B, collected from the Region of Waterloo, Ontario, Canada. Among them, 100 images are selected for training; 10 images are used for validation; and 18 images are chosen for testing. The corresponding ground truth is created via manual digitization. However, in contrast to the chapter 3, the experiments within this chapter are conducted on the original 12 cm resolution rather than the resized 0.48 m resolution used in Chapter 3.

4.3.2 Evaluation Methods

Two types of evaluation methods are used to asssess the experiment results. The first is the correctness and completeness method presented in Section 3.4.2, which works reasonablely well for assessing the performance of different object extraction methods in a general sense. However, it has been found ineffective when attempting to express the quality of vector maps achived by different methods in many real-world applications. In these applications, the decision of whether or not a vector map can be used is usually very strict. A vector map containing any errors beyond a certain tolerance are required to be post-edited to achieve a final format. Therefore, to measure the contribution of an automated object extraction method to any given application, it is important to understand the associated time reduction in the post-editing process. Unfortunately, correctness and completeness fails to deliver such information very well. Figure 4.5 gives two examples where the effectiveness of the correctness and completeness measure breaks down. In the case of Figure 4.5(a), the extracted building outline is very close the ground truth, but not visually appealing. The measurement of correctness and completeness will be very high in this case. However, this building outline will not meet the standards in many real-world applications, which require building outlines to be aesthetically pleasing as discussed in Section 4.1. Therefore, this outline would have to be repaired. However, the additional time required to repair this outline summed with the automated extraction process duration may be equal to or greater than the time required to manually digitizing the outlines from scratch. This is because none of parts of the automated extracted contour meet the standards of building digitization and thus are useless to the post-editing process. As can be seen from this example, although the automated method scores very high correctness and completeness, it in fact saves no time for the application. In another case as shown in Figure 4.5(b), the measurement of

correctness and completeness will be very low because a large part of extracted contour deviates from the ground truth. However, the remainder of the contour is located precisely and aesthetically pleasing. Therefore, to repair the error of this extraction is as easy as drawing a short straight line (denoted by green line in Figure 4.5(b)) to close the gap. Compared to manually digitizing the whole outline from the scratch, automated extraction paired with post-editing in this case indeed leads to a large reduction in end-to-end time. However, the correctness and completeness criteria again fail to reflect such information.

Developing effective criteria to better evaluate performance is just as tricky as developing sophisticated methods to improve the modeling of prior knowledge. The fundamentals of these two problems are more or less the same; to evaluate how the result is aesthetically pleasing, the criteria would have to assess how the result fits within the objects' prior knowledge according to Section 4.1. This is essentially the problem of modeling the prior knowledge, which is nontrivial and discussed in Section 4.2. Considering these challenges, this study proposes to bring human operators to assist the evaluation and directly measure post-editing time as the evaluation criterion. More specifically, a set of sample results obtained by an automated extracted method are given to a number of human operators who are required to further post-edit the results until the output is acceptable. These operators are trained to learn the required output standards, and to perform the post-editing at a relatively consistent speed. The time these human operators spend on post-editing is then averaged and used to assess the performance of the automated method. Due to the uncertainty of human operators in this evaluation process, the criterion is indeed limited to measuring subtle advantages of one method over another However, it has been found to work effectively in real-world applications, because it can provide direct estimation on how

much labor cost and time would be saved if an automated extraction method were used. Therefore, this thesis chooses to use the post-editing time in conjunction with correctness and completeness to access the performance of the proposed method.

(a) (b)

Figure 4.5. Issues with correctness and completeness when assessing the quality of a building vector map. Blue lines indicate the ground truth. Red lines indicate the automated extracted lines. The green line indicates the line that is needed to repair the automated extracted line. (a) High correctness and completeness, yet lengthy post-editing process. (b) Low correctness and completeness, yet efficient post-editing process.

4.3.3 Results

Figure 4.6 shows one exemplar result of building extraction obtained by the proposed combination of bottom-up and top-down processes at each step: Figure 4.6 (a_i) shows the sample aerial imagery at a resolution of 12cm; Figure 4.6(b_i) shows the probability map obtained by the patch-based CNNs in the bottom-up process; Figure 4.6(c_i) shows the label map obtained by the object segmentation with prior knowledge of spectra; Figure 4.6(d_i) shows the vector map achieved by incorporating geometric building

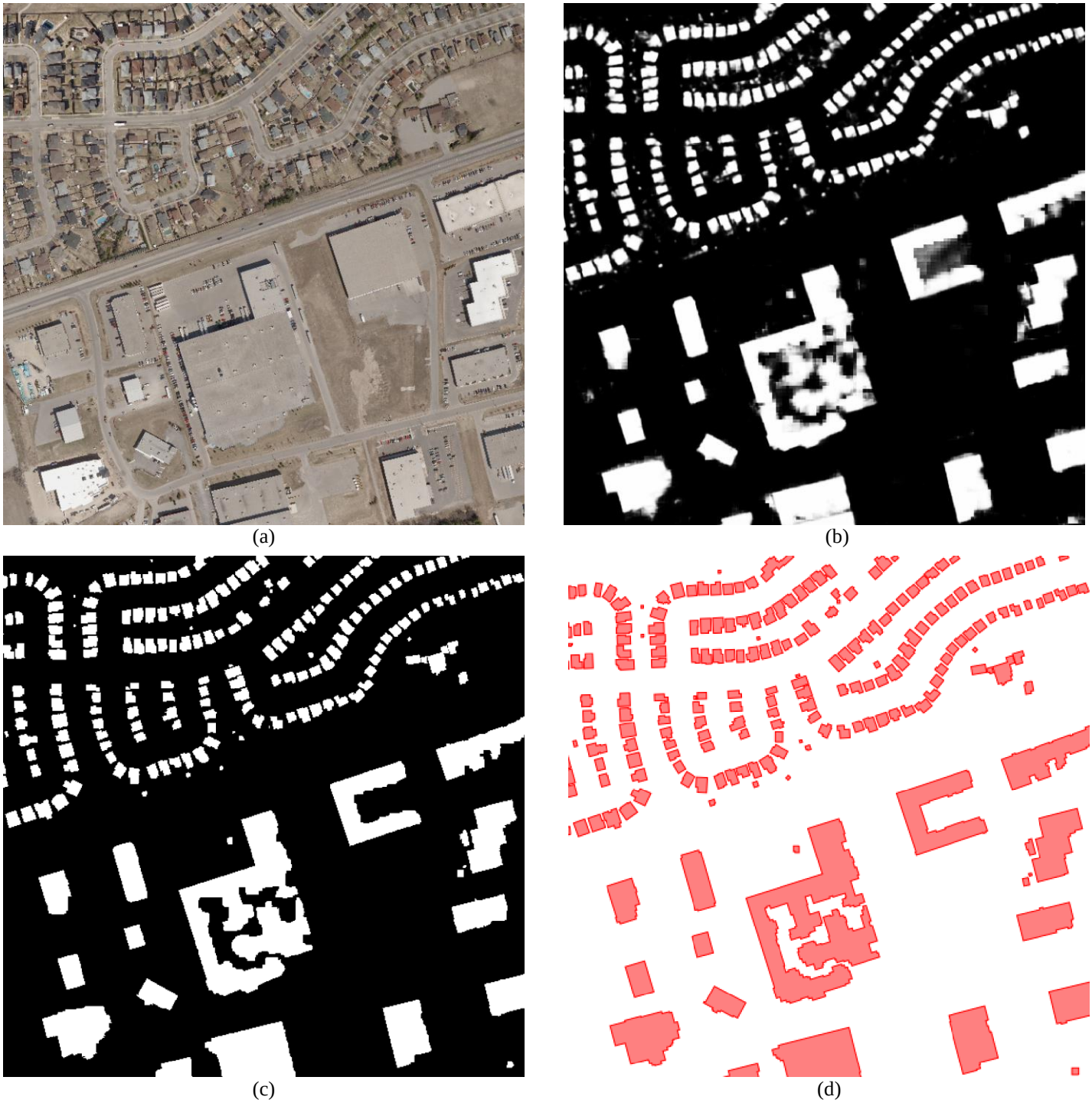


Figure 4.6. Results of the proposed method at each step. (a) Test image at SR of 12 cm; (b) Bottom-up prediction; white color indicates high probability and dark indicates low probability. (c) Object segmentation with spectral prior; white pixels indicate buildings and dark pixels indicate non-buildings. (d) Final building vector map.

modeling. The proposed method is implemented on a single Geforce GTX Titan GPU using CUDA and C++. Experiments show that the GPU implementation is able to process about 756 km^2 of 12 cm aerial images in approximately 30 hours, which is sufficient for use in real-world applications.

4.3.3.1 Influence of Scale

Figure 4.7 shows the probability maps of bottom-up prediction obtained on different scales by tuning the scaling factor *z*. As can be seen from the comparison, prediction on the fine scale (Figure 4.7 (b)) displays better potential for extracting small residential buildings, but is less suitable for big commercial buildings. As the scale become larger (Figure 4.7 (c)), the prediction for commercial buildings becomes better; however, it starts losing the details of residential buildings. As the scale further increases (Figure 4.7 (d)), more details are lost and different buildings start merging into one piece. Therefore, choosing an appropriate scale is important for building extraction. Overall, the fine scale (z = 0.48 m) displays the best potential for building extraction with 12 cm aerial images as shown in the Table 4.1; it outperforms the scale z = 0.96 m by 0.064 in *F*-measure at threshold of 0.5, and the scale z = 1.44 m by 0.137. Therefore, z = 0.48 m is used in the rest of the experiments for building extraction. The relevance of object extraction on the scale also reveals the needs for multi-scale analysis. However, this is a non-trivial task. The details of the discussion are presented as future work in Section 5.2.

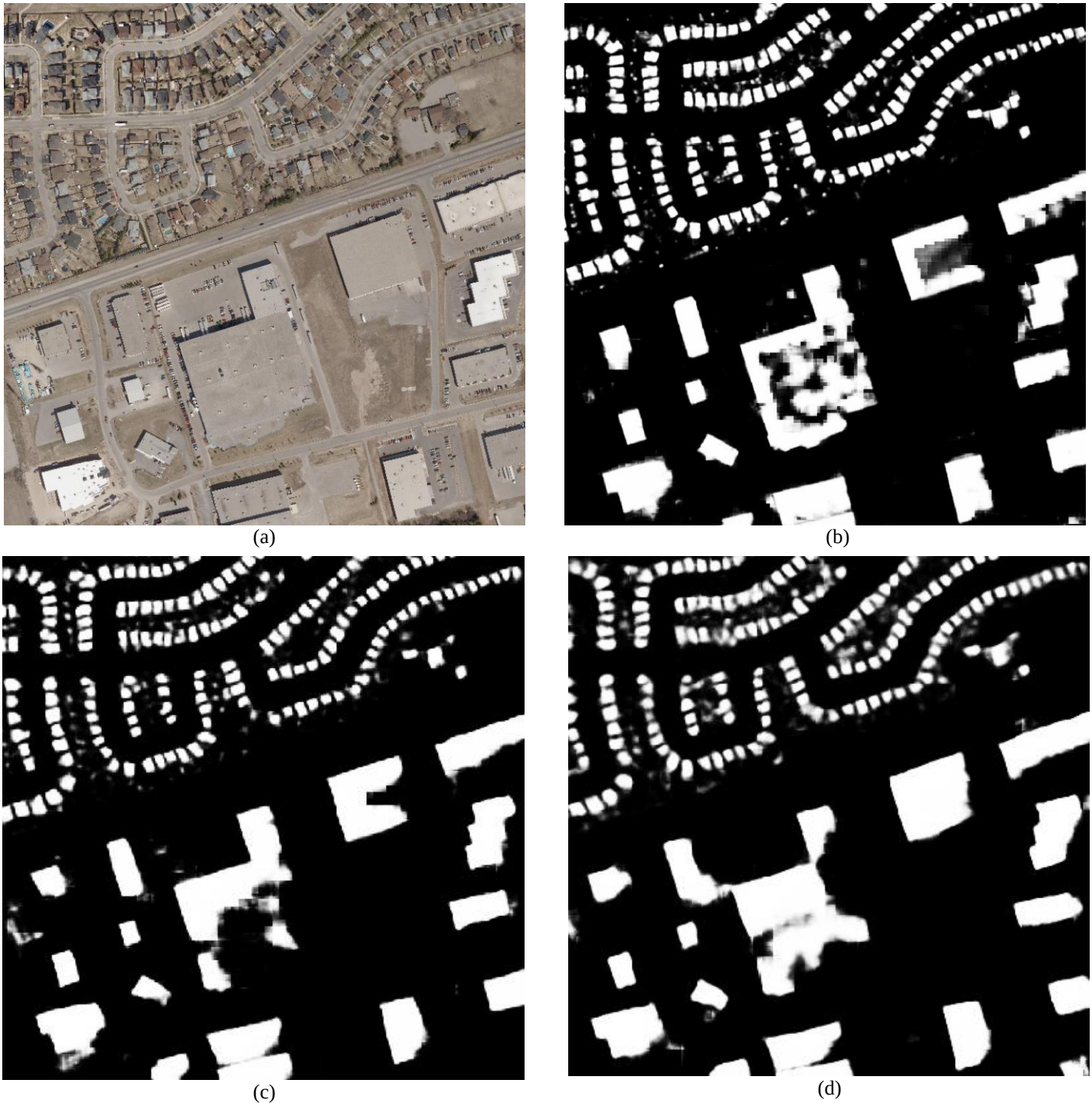


Figure 4.7. Results of the bottom-up prediction with different scale factors. (a) Test image at SR of 12 cm. (b) Result with scaling factor z = 0.48 m. (c) Result with scaling factor z = 0.96 m. (d) Result with scaling factor z = 1.44 m.

	z = 0.48 m	z = 0.96 m	z = 1.44 m
Correctness	0.886	0.787	0.719
Completeness	0.859	0.831	0.752
F-measure	0.872	0.808	0.735

Table 4.1. Comparison of the accuracy of building extraction on different scales with a threshold of 0.5.

4.3.3.2 Influence of Spectral Prior

Figure 4.8 shows the comparison between the results with and without spectral prior on one sample image. The result without using the prior (Figure 4.8(b)) is produced by simply applying a threshold of 0.5 to the corresponding probability map generated by the bottom-up prediction. The result using the prior (Figure 4.8(c)) is obtained by applying object segmentation with spectral constraints to the same probability map generated by the process described in Section 4.2.2. As can be seen from the comparison, building outlines generated by object segmentation with spectral prior conform much better to the image edges than the ones without spectral prior. This observation is also supported by the comparison of correctness and completeness between the two methods on the entire testing dataset, which is shown in Table 4.2. The method using spectral prior outperforms the method without using spectral prior by 0.017 in *F*-measure.

	With spectral prior	No spectral prior
Correctness	0.890	0.886
Completeness	0.888	0.859
F-measure	0.889	0.872

Table 4.2. Comparison of the method using spectral prior and the method without using spectral prior for building extraction.

Figure 4.8. Influence of spectral prior. White pixels indicate the pixels labeled as building pixels. (a) Test image at SR of 12 cm. (b) Result without using spectral prior. (c) Result using spectral prior.

4.3.3.3 Influence of Geometric Prior

Figure 4.9 shows the influence of geometric prior. Figure 4.9(b) shows the result without spectral or geometric prior. These results were obtained by applying a threshold of 0.5 to the probability map generated by bottom-up prediction, and then converted to a vector map using the Douglas–Peucker algorithm. Figure 4.9(c) shows the result using spectral prior only (but without

geometric prior). It is obtained by applying object segmentation with spectral prior to the probability map generated by the bottom-up prediction, and then converted to a vector map using the Douglas–Peucker algorithm. Figure 4.9(d) shows the result using both spectral prior and geometric prior, following the method described in Section 4.2.1 to Section 4.2.3. As can be seen from the comparison, the building outlines generated with the geometric prior are much more visually appealing than the ones without such prior. Also, geometric prior helps to further reduce the adverse effects of shadows as shown by the comparison of the areas circled in green in Figure 4.9(a). Table 4.3 further shows a comparison of the method with both spectral and geometric prior, the method with spectral prior only, and pure manual digitation for building extraction. As can be seen, the performance of different methods are very close under the measurement of correctness and completeness; the method using spectral only even exceed 0.011 in *F*-measure as compared to the method using both geometric spectral prior only. However, there is a large difference in their post-editing time; using the method employing both spectral and geometric prior to generate building vector maps results in speed enhancements of about 2-times in comparison to the method with spectral prior only and about 4-times in comparison to the pure manual digitization method. The existence of such differences in performance measurement is essentially due to the fact that conventional boundary based metrics haven't properly encoded geometric prior into their measurements. However, this inclusion is considered very important to the quality of vector maps in practical applications

	Spectral and Geometric	Spectral only	Manual digitization
Correctness	0.876	0.890	−
Completeness	0.881	0.888	−
F-measure	0.878	0.889	−
Editing Time	36 hours	80 hours	165 hours

Table 4.3. Comparison of manual digitization, the method with spectral prior only, and the method with both spectral and geometric prior for building extraction.

4.3.4 A Word on Chicken and Egg

As discussed in previous chapters, object extraction from HSR imagery is by nature a "chicken and egg" problem: given the outline of an object, recognition becomes easier. But in order to get the object's proper outline, recognition is first needed to determine the type of object. The experiments in this chapter prove that a combination of bottom-up and top-down processes provides a superior framework to resolve this dilemma in comparison to the individual processes.

As can be seen from the experiments of this chapter, the bottom-up deep CNN prediction provides the possible locations that the top-down building model requires to avoid exhaustive searching for model matching, resulting in very efficient process for building extraction. The top-down building model offers the prior knowledge that the bottom-up deep CNN prediction desires to cope with unexpected scenery and produce building vector maps that fit with peoples' prior knowledge.

(a) (b)

(c) (d)

Figure 4.9. Influence of geometric prior. (a) Test image at RS of 12 cm. (b) Result of bottom-up prediction. (c) Result after spectral prior applied. (d) Result after geometric prior applied.

Chapter 5

Conclusions and Future Work

This chapter summarizes conclusions of this thesis and gives recommendations for future work.

5.1 Conclusions

This thesis investigates the issues surrounding the development of automated object extraction methods for HSR imagery, and the design of methods that work reliably for large-scale real-world applications with HSR imagery. It is built on the foundation of deep learning, a recent groundbreaking technology in machine learning, with the effort of further deepening the insight regarding the use of deep learning to develop automated object extraction methods for HSR imagery.

Four major issues have been identified which make the development of automated methods for extracting objects from HSR imagery extremely challenging. They include “large intra-class variation”, “effect of shadows and occlusions”, “chicken and egg”, and “large-scale dataset”. Chapter 2 reviews, in chronological order, the methods developed by the remote sensing community over the past decade in response to these problems. The chapter reveals that the fundamental issue preventing previous methods from attaining reliable performance on challenging real-word datasets is a lack of effective maneuvers for retrieving powerful features from the imagery. Deep learning has shed light on this problem.

Encouraged by the development of deep learning, Chapter 3 re-examines the knowledge the remote sensing community has developed regarding the problem of automated object extraction from HSR imagery in the context of deep learning. Attention is given to OBIA, which is currently considered to be the prevailing framework for this problem and has had a far-reaching impact on the history of remote sensing. A comparison of the effectiveness of patch-based methods and object-based methods for leveraging the power of hierarchies of features learned by deep CNNs is conducted on the tasks of building, road, and waterbody extraction from challenging HSR imagery. The results confirm the effectiveness of deep CNN features for handling the issues of "large-intra variation", "effects of shadows and occlusions", and "large-scale datasets", and adaptability to different object extraction tasks. However, in contrast to common beliefs, the results show that object-based methods suffer seriously from ill-defined image segmentation and are significantly less effective at leveraging the power of deep CNN features than patch-based methods. This observation acted as the catalyst for the re-examination of the entire concept of OBIA that the remote sensing community has built throughout the course of the past decade. It argues that the role of image segmentation has been over-stretched in past studies on object extraction from HSR imagery. Image segmentation is not magic but one of many sampling strategies which provide a certain level of estimation regarding object locations. Image segments generated by this process are not real objects, and any analysis on top of the segments may differ from analysis on real objects. An understanding of the limitations of image segmentation is necessary in order to apply it correctly to object extraction, particularly in the context of deep learning.

Chapter 4 studies ways to further improve the accuracy of object extraction with deep CNNs. Given that vector maps are required as the final format in many applications, the chapter focuses on addressing the issues of generating high-quality vector maps using deep CNNs. Incorporating objects' prior knowledge has been found to be the key to addressing these issues. A method combining bottom-up deep CNN prediction with top-down object modeling is proposed for building extraction. This method also exhibits the potential to extend to other objects of interest. Given that conventional criteria, correctness and completeness, are ineffective at evaluating the accuracy of extracted vector maps, the measurement of post-editing time is proposed as a simple yet effective substitute. Experimental results show that incorporating buildings' spectral and geometric prior helps to further improve the accuracy of deep CNNs, resulting in high-quality vector maps, which meet the standards of real-world applications. Implementing the proposed method on a single GPU results in the capability of processing 756 km^2 of 12 cm aerial imagery in about 30 hours. For the task of producing high-quality building vector maps, post-editing on top of the resulting automated extraction results in speed enhancements of 4-times in comparison to conventional manual digitization methods. Experimental results also show the combined bottom-up and top-down process is a superior framework succeeds at solving the "chicken and egg" dilemma where the individual processes fail.

5.2 Recommendations for Future Research

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

First, it has been observed from the experimental results in Chapters 3 and 4, that the proposed methods work more effectively on small buildings than large buildings. This is likely due to two reasons: (1) Large buildings have more inter-class variation and fewer training samples than small buildings; and (2) Large buildings require a bigger patch-window to include more context than the patch-window used in Chapters 3 and 4. The first issue may be solved by collecting more training samples for large buildings and balancing the ratio between the number of large buildings and small buildings in the training data. The second issue indicates the need for multi-scale analysis. Instead of training a single deep CNNs on one scale, training multiple deep CNNs on multiple scales and voting their results may help mitigate the issue. However, this voting needs to be carefully designed and, as discovered through our extensive experiments, is not as trivial as one may expect. Further, using multiple deep CNNs may increase computational requirements, which has to be considered when assessing feasibility for real-world applications.

Secondly, although the deep CNNs presented in this thesis have been proven sufficient for the task of object extraction on a city scale, larger scale applications of this method have been found to be problematic. For example, the accuracy of building extraction drops when applying the deep CNNs to simultaneously predict five cities containing very different building styles. This is likely due to the fact that the deep CNN model used in this thesis is not complex enough to learn the variation of the buildings in each city. Therefore, a much larger deep CNN model will likely be required for much larger scale predictions. However, increasing the size of deep CNNs will largely increase the computation expense and become more prone to over-fitting. Scaling up deep CNNs is still an open problem in the field of machine learning.

Lastly, the object prior in this thesis is modeled in a category-specific way. As a result, the modeling algorithm may have to be redesigned for every new category. This is very inconvenient for practical use. It would be much better if the prior can be modeled in more general way. Also, this thesis only studies the method for modeling buildings with two perpendicular directions. It would be desired to see much complex building structures modeled effectively in the future.

References

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 898-916.

Arbelaez, P., Pont-Tuset, J., Barron, J. T., Marques, F., & Malik, J. (2014). Multiscale Combinatorial Grouping. In *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus.

Agarwal, S., Vailshery, L. S., Jaganmohan, M., & Nagendra, H. (2013). Mapping urban tree species using very high resolution satellite imagery: comparing pixel-based and object-based approaches. *ISPRS International Journal of Geo-Information*, *2*(1), 220-236.

Andrefoueet, S., Kramer, P., Torres-Puliza, D., Joyce, K. E., Hochberg, E. J., Garza-Perez, R., … & Muller-Karger, F.E. (2003). Multi-site evaluation of IKONOS data for classification of tropical coral reef environments. *Remote Sensing of Environment, 88*(1), 123-14.

Baatz, M., & Schape, A. (2000). Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation. *Angewandte Geographische Informations-Verarbeitung XII*, 12-23

Benediktsson, J.A., Pesaresi, M., & Arnason, K. (2003). Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing, 41*(9), 1940-1949.

Benedek, C., Descombes, X., & Zerubia, J. (2012). Building development monitoring in multitemporal remotely sensed image pairs with stochastic birth-death dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(1), 33-50.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems*, *19*, 153.

Benz, U.C., Hofmann, P., Willhauck, G., Lingenfelder, I., & Heynen, M. (2004). Multi-resolution object-oriented fuzzy analysis of remote sensing data for GIS-ready information. *ISPRS Journal of Photogrammetry and Remote Sensing*, *58*(3-4), 239-258.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: springer.

Blaschke, T., & Strobl, J. (2001). What's wrong with pixels? Some recent developments interfacing remotes sensing and GIS. *Proceedings of GIS–Zeitschrift fur Geoinformationsysteme*, *14*(6), 12-17.

Blaschke, T., Burnett, C., & Pekkarinen, A. (2004). New contextual approaches using image segmentation for object-based classification. *In: De Meer, F., de Joneg, S. (Eds.), Remote Sensing Image Analysis: Including the Spatial Domain*, 211-236. Kluver Academic Publishers, Dordrecht.

Blaschke, T. (2010). Object-based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, *65*, 2-16.

Blaschke, T., Hay, G. J., Kelly, M., Lang, S., Hofmann, P., Addink, E., ... & Tiede, D. (2014). Geographic Object-Based Image Analysis–Towards a new paradigm. *ISPRS Journal of Photogrammetry and Remote Sensing*, *87*, 180-191.

Blake, A., Rother, C., Brown, M., Perez, P., & Torr, P. (2004). Interactive image segmentation using an adaptive GMMRF model. In *European Conference on Computer Vision*, Prague.

Bruzzone, L., & Carlin, L. (2006). A multilevel context-based system for classification of very high spatial resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, *44*(9), 2587-2600.

Brunner, D., Lemoine, G., & Bruzzone, L. (2010). Earthquake damage assessment of building using VHR optical and SAR imagery. *IEEE Transactions on Geoscience and Remote Sensing, 48*(5), 2403-2420.

Burnett, C., & Blaschke, T. (2003). A multi-scale segmentation/object relationship modeling methodology for landscape analysis. *Ecological Modeling 168*(3), 233-249.

Carleer, A., & Wolf, E. (2004). Exploitation of very high resolution satellite data for tree species identification. *Photogrammetric Engineering and Remote Sensing, 70*(1), 135-140.

Chanussot, J., Benediktsson, J.A., & Fauvel, M. (2006). Classification of remote sensing image from urban areas using a fuzzy possibilistic model. *IEEE Transactions on Geoscience and Remote Sensing, 3*(1), 40-44.

Cracknell, A.P. (1998). Synergy in remote sensing– What's in a pixel? *International Journal of Remote Sensing, 19*(11), 2025-2047.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning*, Helsinki.

Colomina, I., & Molina, P. (2014). Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, *92*, 79-97.

Coops, N.C., Johnson, M., Wulder, M.A., & White, J.C. (2006). Assessment of QuickBird high spatial resolution imagery to detect red attack damage due to mountain pine beetle infestation. *Remote Sensing of Environment, 103*(1), 67-80.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(1), 30-42.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B 39*(1), 1-38.

DigitalGlobe. (2014). < http://www.digitalglobeblog.com/tag/worldview-3/>.

Doxani, G., Karantzalos, K., & Strati, M. T. (2012). Monitoring urban changes based on scale-space filtering and object-oriented classification. *International Journal of Applied Earth Observation and Geoinformation, 15*, 38-48.

Douglas, D. H., & 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), 112-122.

Ehrlich, D., Kemper, T., Blaes, X., & Soille, P. (2013). Extracting building stock information from optical satellite imagery for mapping earthquake exposure and its vulnerability. *Natural Hazards, 68*(1), 79-95.

Eikivl, L., Aurdal, L., & Koren, H. (2009). Classification-based vehicle detection in high-resolution satellite images. *ISPRS Journal of Photogrammetry and Remote Sensing, 64*(1), 65-72.

FAA. (2014). http://www.faa.gov/news/press_releases/news_story.cfm?newsId=16354

Federer, H. (1996). *Geometric Measure Theory*. New York: Springer.

Fernandes, M. R., Aguiar, F. C., Silva, J., Ferreira, M. T., & Pereira, J. (2014). Optimal attributes for the object based detection of giant reed in riparian habitats: A comparative study between Airborne High Spatial Resolution and WorldView-2 imagery. *International Journal of Applied Earth Observation and Geoinformation, 32*, 79-91.

Flanders, D., Hall-Beyer, M., & Pereverzoff, J. (2003). Preliminary evaluation of eCognition object-based software for cut block delineation and feature extraction. *Canadian Journal of Remote Sensing, 29*(4), 441-452.

Garrity, S. R., Allen, C. D., Brumby, S. P., Gangodagamage, C., McDowell, N. G., & Cai, D. M. (2013). Quantifying tree mortality in a mixed species woodland using multitemporal high spatial resolution satellite imagery. *Remote Sensing of Environment, 129*, 54-65.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus.

Gong, B., Im, J., & Muntrakis, G. (2011). An artificial immune network approach to multi-sensor land use/land cover classification. *Remote Sensing of Environment, 115*(2), 600-614.

Hansen, M. C., Potapov, P. V., Moore, R., Hancher, M., Turubanova, S. A., Tyukavina, A., ... & Townshend, J. R. G. (2013). High-resolution global maps of 21st-century forest cover change. *Science, 342*(6160), 850-853.

Hay, G. J., & Castilla, G. (2008). Geographic Object-Based Image Analysis (GEOBIA): A new name for a new discipline. In *Object-based Image Analysis*, 75-89. Springer Berlin Heidelberg.

Hinton, G., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527-1554.

Hinton, G., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313(5786)*, 505-507.

Huang, X., & Zhang, L. (2012). 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), 161-172.

Huang, C., Davis, L.S., & Townshend, J.R.G. (2002). An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing, 23*, 725-749.

Jensen, J.R. (2005). *Introductory Digital Image Processing: A Remote Sensing Perspective (3rd ed.).* Upper Saddle River: Prentice-Hall.

Jin, X., & Davis C.H. (2007). Vehicle detection from high-resolution satellite imagery using morphological shared-weight neural networks. *Image and Vision Computing, 25*, 1422-1431.

Kampouraki, M., Wood, G.A., & Berwer, T.R. (2008). Opportunities and limitations of object based image analysis for detecting urban impervious and vegetated surfaces using true-colour aerial photography. *In Object-Based Image Analysis-Spatial Concepts for Knowledge-Driven Remote Sensing Applications, T. Blaschke, S. Lang and G. Hay (Eds.),* 555-569. Berlin: Springer-Verlag.

Katartzis, A., & Sahli, H. (2008). A stochastic framework for the identification of building rooftops using a single remote sensing image. *IEEE Transactions on Geoscience and Remote Sensing*, *46*(1), 259-271.

Katartzis, A., Sahli, H., Pizurica, V., &, Cornelis, J. (2001). A model based approach to the automatic extraction of linear features from airborne images. *IEEE Transactions on Geoscience and Remote Sensing 39*(9), 2073-2079.

Key, T., Warner, T.A., McGraw, J.B., & Fajvan, M.A. (2001). A comparison of multispectral and multitemporal information in high spatial resolution imagery for classification of

individual tree species in a temperate hardwood forest. *Remote Sensing of Environment, 75(1),* 100-112.

Kolmogorov, V., & Zabih, R. (2004). What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(2),* 147-159.

Kosaka, N., Akiyama, T., Tsai, B., & Kojima, T. (2005). Forest type classification using data fusion of multispectral and panchromatic high resolution satellite imageries. *Proceedings of IEEE International Geoscience and Remote Sensing Symposium 4,* 2980-2983.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, Lake Tahoe.

Kumar, M. P., Torr, P. H., & Zisserman, A. (2010). Objcut: Efficient segmentation using top-down and bottom-up cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(3), 530-545.

Lacoste, C., Descombes, X., & Zerubia, J. (2005). Point processes for unsupervised line network extraction in remote sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10), 1568-1579.

Lacoste, C., Descombes, X, & Zerubia, J. (2010). Unsupervised line work network extraction in remote sensing using a polyline process. *Pattern Recognition, 43*(4), 1631-1641.

Lafarge, F., Gimel'farb, G., & Descombes, X. (2010). Geometric feature extraction by a multimarked point process. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(9), 1597-1609.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, New York.

Leblon, B. (2001). Forest wildfire hazard monitoring using remote sensing: a review. *Remote Sensing Reviews, 20,* 1-43.

Le, Q. V., Zou, W. Y., Yeung, S. Y., & Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *IEEE Conference Computer Vision and Pattern Recognition*, Colorado.

LeCun, Y. (1989). Generalization and network design strategies. In *Connectionism in Perspective.*

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Li, Y., & Li, J. (2010). Oil spill detection from SAR intensity image using a marked point process. *Remote Sensing of Environment, 114*(7), 1590-1601.

Li Y., Li J., & Lu Y. (2008). A fuzzy segmentation based approach to extraction of coastlines from IKONOS imagery. *Geomatica*, *62*(4), 407-417.

Liu, H., & Jezek, K.C. (2004). Automated extraction of coastline from satellite imagery by integrating Canny edge detection and locally adaptive thresholding methods, *International Journal of Remote Sensing, 25*(5), 937-958.

Liu, D., & Xia, F. (2010). Assessing object-based classification: advantages and limitations. *Remote Sensing Letters, 1*(4), 187-194.

Longley, P. (2005). *Geographic information systems and science*. England: John Wiley & Sons, Ltd

Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91-110.

Mallinis, G., Koutsias, N., Tsakiri-Strati, M., & Karteris, M. (2008). Object-based classification using Quickbird imagery for delineating forest vegetation polygons in a Mediterranean test site. *ISPRS Journal of Photogrammetry and Remote Sensing, 25*(4), 347-356.

McNairn, H., Champagne, C., Shang, J., Holmstrom, D., & Reichert, G. (2009). Integration of optical and Synthetic Aperture Radar (SAR) imagery for delivering operational annual crop inventories. *ISPRS Journal of Photogrammetry and Remote Sensing, 64*(5), 434-449.

Mena, J.B. (2003). State of the art on automatic road extraction for GIS update: a novel classification. *Pattern Recognition Letters, 24*, 3037-3058.

Miao, Z., Shi, W., Zhang, H., & Wang, X. (2013). Road centerline extraction from high-resolution imagery based on shape features and multivariate adaptive regression splines. *Geoscience and Remote Sensing Letters, IEEE, 10*(3), 583-587

Michelot, C. (1986). A finite algorithm for finding the projection of a point onto the canonical simplex of r^n. *Journal of Optimization Theory and Applications, 50*(1), 195-200.

Mitri, G. H., & Gitas, I. Z. (2013). Mapping post-fire forest regeneration and vegetation recovery using a combination of very high spatial resolution and hyperspectral satellite imagery. *International Journal of Applied Earth Observation and Geoinformation, 20*, 60-66.

Mnih, V. (2013). *Machine Learning for Aerial Image Labeling* (Doctoral dissertation, University of Toronto).

Moeller, M.S., & Blaschke, T. (2006). Urban change extraction from high resolution satellite image. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vienna, Austria, XXXVI,* 151-156.

Moller, M., Lymburner, L., & Volk, M. (2007). The comparison index: a tool for assessing the accuracy of image segmentation. *International Journal of Applied Earth Observation and Geoinformation, 9*, 311-321.

Mountrakis, G., Im, J., & Ogole, C. (2011). Support vector machine in remote sensing: a review. *ISPRS Journal of Photogrammetry and Remote Sensing, 66,* 247-259.

Mumby, P. J., Skirving, W., Strong, A. E., Hardy, J. T., LeDrew, E. F., Hochberg, E. J., Stumpf, R. P. and David, L. T. (2004). Remote sensing of coral reefs and their physical environment. *Marine Pollution Bulletin 48,* 219-22.

Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., Gool, L., & Purgathofer, W. (2013). A survey of urban reconstruction. In *Computer Graphics Forum 32*(6), 146-177.

Mumford, D., & Desolneux, A. (2010). *Pattern Theory: The Stochastic Analysis of Real-World Signals.* Natick, MA: A K Pters, Ltd.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, Haifa.

Nieuwenhuis, C., Toppe, E., & Cremers, D. (2013). A survey and comparison of discrete and continuous multilabel segmentation approaches. *International Journal of Computer Vision*, *104*(3), 223-240.

Ortner, M., Descombes, X., & Zerubia, J. (2008). A marked point process of rectangles and segments for automatic analysis of Digital Elevation Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(1), 105-119.

Ouma, Y. O., Ngigi, T. G., & Tateishi, R. (2006). On the optimization and selection of wavelet texture for feature extraction from high-resolution satellite imagery with application towards urban-tree delineation. *International Journal of Remote Sensing, 27,* 73-104.

Pacifici, F., Chini, M., & Emery, W.J. (2009). A neural network approach using multi-scale textural metrics from very high-resolution panchromatic imagery for urban land-use classification. *Remote Sensing of Environment, 113*(6), 1276-1292.

Pal, M. (2005). Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, *26*(1), 217-222.

Park, J.H., Tateishi, R., Wikantika, K., &, Park, J.G. (1999). The potential of high resolution remotely sensed data for urban infrastructure monitoring. *Proceedings of IEEE International Geoscience and Remote Sensing Symposium, 2*, 1137-1139.

Perrin, G., Descombes, X., & Zerubia. J. (2005). A marked point process model for tree crown extraction in plantations. *Proceedings of IEEE International Conference on Image Processing.*

Pock, T., Cremers, D., Bischof, H., & Chambolle, A. (2009). An algorithm for minimizing the piecewise smooth Mumford-Shah functional. In *IEEE International Conference on Computer Vision*, Kyoto.

Powers, R. P., Hermosilla, T., Coops, N. C., & Chen, G. (2015). Remote sensing and object-based techniques for mapping fine-scale industrial disturbances. *International Journal of Applied Earth Observation and Geoinformation*, *34*, 51-57.

Porway, J., Wang, Q., & Zhu, S. C. (2010). A hierarchical and contextual model for aerial image parsing. *International Journal of Computer Vision*, *88*(2), 254-283.

Puissant, A., Hirsch, J., & Weber, C. (2005). The utility of texture analysis to improve per-pixel classification for high to very spatial resolution imagery. *International Journal of Remote Sensing*, *26*(4), 733-745.

Robinson, D. J., Redding, N. J., and Crisp, D. J. (2002). *Implementation of a fast algorithm for segmenting SAR imagery,* Scientific and Technical Report, 01 January 2002. Australia: Defense Science and Technology Organization.

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, *23*(3), 309-314.

Seelan, S.K., Laguette, S., Casady, G.M., & Seielstad, G.A. (2003). Remote sensing applications for precision agriculture: a learning community approach. *Remote Sensing of Environment, 88*(1-2), 157-169.

Senay, G. B., Lyon, J. G., Ward, A. D., & Nokes, S. E. (2000). Using high spatial resolution multispectral data to classify corn and soybean crops. *Photogrammetric Engineering and Remote Sensing, 66*(3), 319-328.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229.*

Sexton, J. O., Song, X. P., Huang, C., Channan, S., Baker, M. E., & Townshend, J. R. (2013). Urban growth of the Washington, DC–Baltimore, MD metropolitan region from 1984 to 2010 by annual, Landsat-based estimates of impervious cover. Remote Sensing of Environment, *129*, 42-53.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: visualizing image classification models and saliency maps. *arXiv preprint arXiv:1312.6034.*

Sirmacek, B., & Unsalan, C. (2009). Building detection using local Gabor features in very high resolution satellite images. *Proceedings of Recent Advances in Space Technologies,* 283-286.

Sirmacek, B., & Unsalan, C. (2011). A probabilistic framework to detect buildings in aerial and satellite images. *IEEE Transactions on Geoscience and Remote Sensing, 49*(1), 211-221.

SpaceNews. (2014). <http://www.spacenews.com/article/civil-space/40874us-government-eases-restrictions-on-digitalglobe>.

Stoica, R., Descombes, X., & Zerubia, J. (2004). A Gibbs point process for road extraction in remotely sensed images. *International Journal of Computer Vision, 57*(2), 121-136.

Stramondo, S., Biqnami, C., Pierdicca, N., & Tetulliani, A. (2006). Satellite radar and optical remote sensing for earthquake damage detection: result from different case studies. *International Journal of Remote Sensing, 27*(20), 4433-4447.

Sohn, G., & Dowman, I. (2007). Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction. *ISPRS Journal of Photogrammetry and Remote Sensing, 62*(1), 43-63.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199.*

Tauberer, J. (2014). *Open government data: the book (2nd ed.).* < https://opengovdata.io>.

Thomas, J., Kareem, A., & Bowyer, K. W. (2014). Automated Poststorm Damage Classification of Low-Rise Building Roofing Systems Using High-Resolution Aerial Imagery. *IEEE Transactions on Geoscience and Remote Sensing, 52*(7), 3851-3861.

Tuia, D., Pacifici, F., Kanevski, M., & Emery, W.J. (2009). Classification of very high spatial resolution imagery using mathematical morphology and support vector machine. *IEEE Transactions on Geoscience and Remote Sensing, 47*(11), 3866-3879.

Tokarczyk, P., Wegner, J. D., Walk, S., & Schindler, K. (2015). Features, color spaces, and boosting: new insights on semantic classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing, 53*(1), 280-295.

van der Sande, C.J., De Jong, S.M., &, De Roo, A.P. (2003). A segmentation and classification approach of IKONOS-2 imagery for land cover mapping for assist flood risk and flood damage assessment. *International Journal of Applied Earth Observation and Geoinformation, 4*(3), 217-229.

Verdie, Y., & Lafarge, F. (2014). Detecting parametric objects in large scenes by Monte Carlo sampling. *International Journal of Computer Vision*, *106*(1), 57-75.

Von Gioi, R. G., Jakubowicz, J., Morel, J. M., & Randall, G. (2010). LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(4), 722-732.

Wiedemann, C., Heipke, C., Mayer, H., & Jamet, O. (1998) Empirical evaluation of automatically extracted road Axes. *In: CVPR Workshop on Empirical Evaluation Methods in Computer Vision,* 172-187.

Wulder, M.A., Dymond, C.C., White J.C., Leckie, D.G., & Carroll, A.L. (2006). Surveying mountain pine beetle damage of forests: A review of remote sensing opportunities. *Forest Ecology and Management, 221*(1-3), 27-41.

Wulder, M., Niemann, K.O., & Goodenough, D.G. (2000). Local maximum filtering for the extraction of tree locations and basal area from high spatial resolution imagery. *Remote Sensing of Environment, vol. 73*(1), 103-114.

Xu, H. (2013). Rule-based impervious surface mapping using high spatial resolution imagery. *International Journal of Remote Sensin*g, *34*(1), 27-44.

Kim, Y., & Kim Y. (2014). Improved Classification Accuracy Based on the Output-Level Fusion of High-Resolution Satellite Images and Airborne LiDAR Data in Urban Area. *IEEE Geoscience and Remote Sensing Letters, 11*(3), 636-640.

Yu, Q., Gong, P., Clinton, N., Biging, G., Kelly, M., & Schirokauer, D. (2006). Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery. *Photogrammetric Engineering & Remote Sensing*, *72*(7), 799-811.

Zach, C., Gallup, D., Frahm, J. M., & Niethammer, M. (2008). Fast global labeling for real-time stereo using multiple plane sweeps. In *Vision Modeling and Visualization Workshop (VMV)*, Konstanz.

Zeiler, M.D., & Fergus, R. (2013). Visualizing ad understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901.*

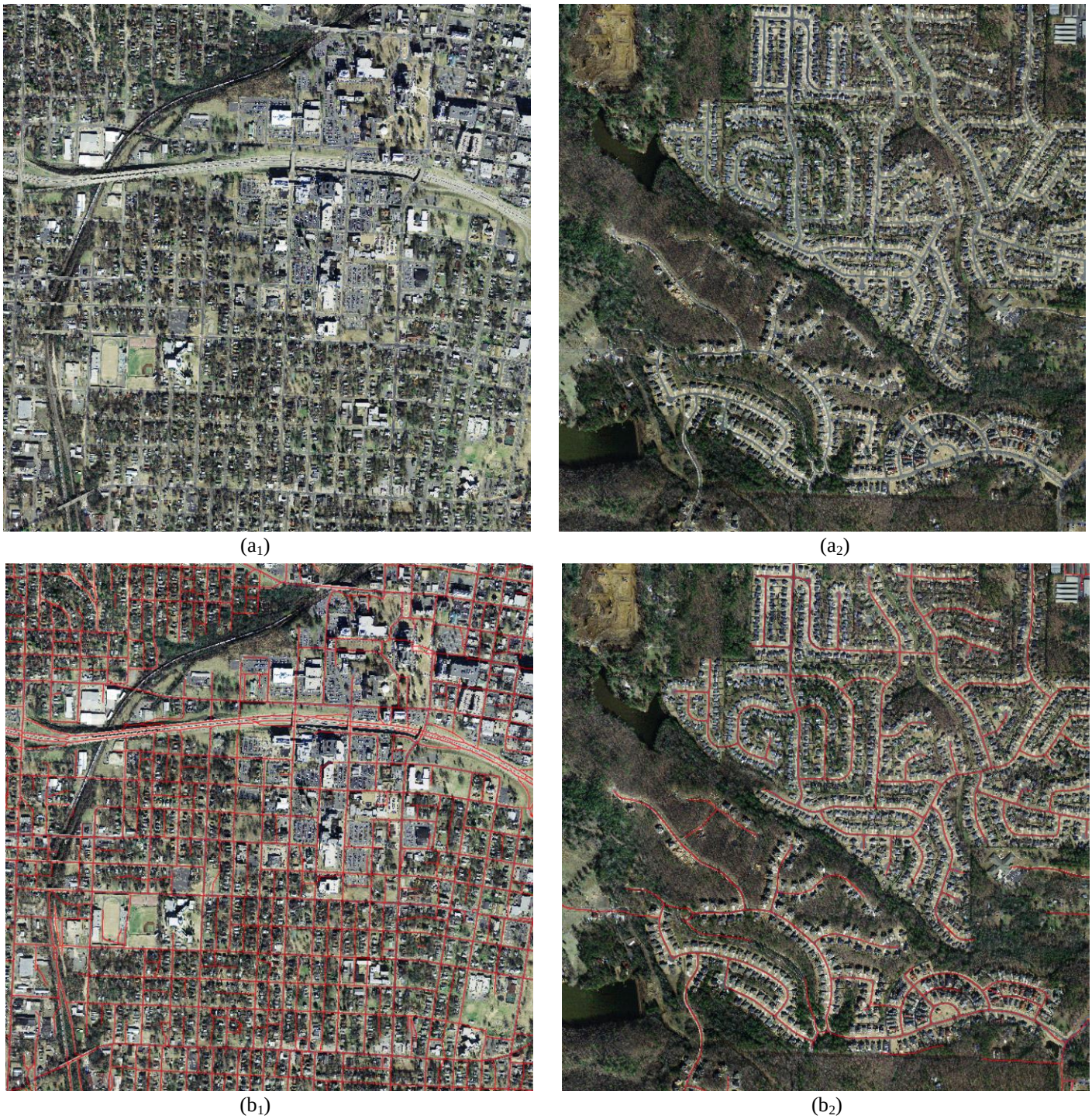
Zhang, Y. (2001). Texture-integrated classification of urban treed areas in high-resolution color-infrared imagery. *Photogrammetric Engineering & Remote Sensing, 67*(12), 1359-1365.

Zhang, P., Lv, Z., & Shi, W. Z. (2013). Object-based spatial feature for classification of very high resolution remote sensing images. *IEEE Geoscience and Remote Sensing Letters, 10*(6), 1572-1575

Zhang, H., Shi, W., Wang, Y., Hao, M., & Miao, Z. (2014). Classification of very high spatial resolution imagery based on a new pixel shape feature set. *IEEE Geoscience and Remote Sensing Letters, 11*(5), 940-944.

Zhong, Y., Zhao, J., & Zhang, L. (2014). A hybrid object-oriented conditional random field classification framework for high spatial resolution remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing, 52*(11), 7023-7037.

Appendix



(a_1) (a_2)

(b_1) (b_2)

Figure 7.1 Sample results of road extraction. (a_i) Sample images at SR of 25 cm (courtesy of Sanborn Inc.); (b_i) Result of road centerlines overlaid with the image.

(a_1) (a_2)

(b_1) (b_2)

Figure 7.2 Sample results of impervious surface extraction. (a_i) Sample images at SR of 12 cm (courtesy of Region of Waterloo); (b_i) Result of impervious surface overlaid with the image