[UNSUPERVISED CHANGE DETECTION TECHNIQUE BASED ON FULLY CONVOLUTIONAL NETWORK USING RGBD]

[JIANDA YAN]

[February, 2019]

SUPERVISORS:

[Dr, C, Persello]

[Dr, F.C, Nex]

[UNSUPERVISED CHANGE DETECTION TECHNIQUE BASED ON FULLY CONVOLUTIONAL NETWORK USING RGBD]

[JIANDA YAN]

Enschede, The Netherlands, [February, 2019]

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfillment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.

Specialization: [Name course (e.g. Applied Earth Sciences)]

SUPERVISORS:

[Dr. C. Persello]

[Dr. F.C. Nex]

THESIS ASSESSMENT BOARD:

[prof.dr.ir. A. Stein (Chair)]

[dr. F. Melgani (External Examiner, University of Trento, Department of Information Engineering and Computer Science)]

Etc.

DISCLAIMER

This document describes work undertaken as part of a programme of study at the Faculty of Geo-Information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the Faculty.

ABSTRACT

Due to the rapid process of urbanization, new build up areas and infrastructures like buildings, streets, bridges, and other man-made objects are changing the world all the time. Therefore, there is an increasing demand for detecting changes in urban areas. Traditional two-dimensional (2D) change detection methods are limited by different image perspective and illumination. This thesis describes a 3D change detection methodology by the joint use of height and spectral information. The proposed method follows the following steps. Firstly, the subtraction of DSMs and a morphological based post-processing are performed between image pairs. Then, the effect of several RGB-based methods is compared and analyzed based on the study area. Furthermore, we combine the DSM-based method and RGB-based. Finally, we define and calculate the 'reliability' for the labels obtained from the combined method, and selecting reliable labels as training samples according to the reliability to train the FCN network. This method enables the FCN architecture to work without manually labeled training samples, which is a high labor cost and is a time-consuming task. By using the FCN architecture, additional contextual information can be considered, and results derived from the joint of DSM-based and RGB-based method can be further improved. After checking the result, we find that the errors caused by shadows, different seasons, and the growth of vegetation are reduced. Also, the noises generated in the pixel-based method are also removed by the FCN architecture.

The study area is the city of Ecublens, Switzerland. The data is orthophoto acquired by the UAV, and the DSM data is generated from photogrammetry using the overlapping images. Images are obtained from three different times, and all experiments are performed three times, each time taking two images obtained from different times for change detection. In the end, our method is compared to the supervised FCN architecture, which uses manually labeled training samples to train the FCN architecture. Evaluation of the proposed approach in terms of accuracy, precision, recall, and F1 score is performed, showing that our result is even better than the result derived from the supervised FCN architecture, which uses ground truth as training samples.

Keywords

Change detection method, Unsupervised algorithms, Fully convolutional neural network, RGBD data

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my first supervisor, Dr. C. Persello, for his tremendous academic support, constructive criticism, and warm encouragement during the thesis. I also want to say thanks for my second supervisor, Dr. F.C. Nex, for critical comments and valuable discussion. I cannot finish this thesis without their help.

I want to extend my gratitude to all teachers in the GFM program for providing a comfortable learning environment.

I also want to thank all my friends in ITC who gave me companionship and support. They left me a deep memory in these 18 months

A very special thanks to my parents. They always support me and respect my choice. I want to give my thanks to my whole family who has supported me emotionally and financially.

TABLE OF CONTENTS

1 Introduction 1
1.1 Motivation and problem statement 1
1.2 Research identification 2
1.2.1 Research objective 2
1.2.2 Research question 2
1.2.3 Innovation aimed at 3
2 Literature review 4
2.1 The context of change detection 4
2.2 Related work 4
2.2.1 Change detection algorithm based on DSM 4
2.2.2 Change vector analysis (CVA) algorithm 5
2.3 The development of the CNN architectures 5
3 Method 10
3.1 The unsupervised change detection method 10
3.1.1 Change detection method based on DSM data 10
3.1.2 Change detection method based on RGB data 11
3.1.3 The strategy combines the change detection result from DSM and RGB 12
3.2 Process the unsupervised result 13
3.2.1 Define and calculate the reliability of the unsupervised result 13
3.2.2 Extract the reliability matrix from the unsupervised method 15
3.2.3 Remove less-reliable results 16
3.3 Training and configuring the FCN architectures 16
3.3.1 Input layer 16
3.3.2 Convolutional layer 16
3.3.3 Batch Normalization layer 17
3.3.4 Activation Functions 17
3.3.5 Softmax layer 18
3.3.6 Dropout layer 18
4 Experiment setup 19
4.1 Data preparation 19
4.1.1 Study area 19
4.1.2 Pre-processing 19
4.1.3 Annotation 21
4.2 Model parameter 24
4.2.1 The parameters in the unsupervised method 24

TABLE OF CONTENTS

4.2.2 Structure and parameters of the FCN architecture 24
4.3 Assessment 25
4.4 Software 26
5 Result and analysis 27
5.1 The result based on RGB data 27
5.1.1 The result of the CVA algorithm 27
5.1.2 The result of the SAM algorithm 29
5.1.3 The result of the CVA&SAM algorithm 30
5.1.4 Comparison of RGB-based algorithms 31
5.2 The result based on DSM data 31
5.3 Combining result from RGB-based and DSM-based method 34
5.4 The result of the FCN architecture 35
5.4.1 The result of different architectures 35
5.4.2 The result of the different proportion of training samples 37
5.4.3 Comparison of our unsupervised FCN result with supervised FCN result 40
6 Discussion 42
6.1 The unsupervised method 42
6.1.1 RGB-based method 42
6.1.2 DSM-based method 42
6.1.3 Reliability 42
6.2 Unsupervised FCN 43
7 Conclusions and recommendations 44
7.1 Conclusions 44
7.2 Recommendations 44
8 Reference 47
9 Appendix 51
9.1 Appendix 1 51
9.2 Appendix 2 54
9.3 Appendix 3 58
9.4 Appendix 4 61
9.5 Appendix 5 64

LIST OF FIGURES

Figure 2-1 Schematic representation of a basic system of ANN 6
Figure 2-2 The diagram of multilayer perceptron 6
Figure 3-1 Three types of changes in the real world in multi-spectral images 11
Figure 3-2 The flowchart of unsupervised algorithms 13
Figure 3-3 Representation of the changed and unchanged areas 14
Figure 3-4 The relation between reliability and difference from the DSM-based method 14
Figure 3-5 The relation between reliability and difference from the RGB-based method 15
Figure 3-6 Extract the reliable matrix 15
Figure 3-7 Schematic represents the way of concatenating 16
Figure 3-8 Schematic represents the dropout method 18
Figure 4-1 Workflow 19
Figure 4-2 Experimental images after processing 20
Figure 4-3 Four tiles of the first epoch 21
Figure 4-4 The annotation and corresponding of the first epoch 22
Figure 4-5 Labeled changed the detection reference and raw images 23
Figure 5-1 The CD13 of CVA 28
Figure 5-2 The CD13 of SAM 29
Figure 5-3 The CD13 of CVA&SAM 30
Figure 5-4 Comparison of three RGB-based algorithms 31
Figure 5-5 The CD13 based on DSM data 33
Figure 5-6 The CD13 of the CVA&DSM method 34
Figure 5-7 The results of the FCN using a different proportion of the training sample 39
Figure 5-8 Comparing the CD13 in the FCN and unsupervised result 40
Figure 5-9 The distribution of training samples and testing samples in the supervised FCN 41

LIST OF TABLES

Table 2-1 Representation of the LeNet-5 architecture 7
Table 2-2 Representation of the AlexNex architecture 8
Table 2-3 Representation of the VGG16 architecture 8
Table 4-1 The number of annotated pixels for the changed and the unchanged classes 23
Table 4-2 The architecture of FCN-DK6 with the kernel size of 5 × 5 24
Table 4-3 The matrix derived from the true class and predicted the class 25
Table 5-1 The result of the CVA algorithm 28
Table 5-2 The overall confusion matrix of three image pairs together using the CVA algorithm 28
Table 5-3 The result of the SAM algorithm 29
Table 5-4 The overall confusion matrix of three image pairs together using the CVA algorithm 30
Table 5-5 The result of the CVA&SAM algorithm 30
Table 5-6 The overall confusion matrix of three image pairs together using the CVA&SAM algorithm 31
Table 5-7 Comparing the overall result of the DSM-based method with different parameter 32
Table 5-8 The overall confusion matrix of three image pairs together using the DSM-based algorithm 33
Table 5-9 The result of the CVA&DSM algorithm 34
Table 5-10 The confusion matrix of three image pairs together using the unsupervised algorithm 35
Table 5-11 The architecture of FCN-DK6 with the kernel size of 3×3 35
Table 5-12 The architecture of FCN-DK12 with the kernel size of 3×3 35
Table 5-13 Comparing the overall result of four situations 36
Table 5-14 The result of the unsupervised method, unsupervised FCN and supervised FCN 41

1 Introduction

1.1 Motivation and problem statement

The increased rate of urban expansion in recent years has significantly changed urban landscapes all over the world. Understanding the changed areas allows the government to make a better city planning or solve the problems caused by changes. Change detection techniques aim to detect changes between two or more multi-temporal remote sensing images acquired over the same area, so as to monitor land cover changes. These techniques attracted much attention in recent decades. In the past, change detection has been mainly used to monitor agriculture and land cover changes primarily because of the limitation of resolution (Guerin, Binet, & Pierrot-Deseilligny, 2014). With the increase of spatial resolution, it is now applied on various applications, like land cover updating, urban expansion, water conservancy, environmental disaster and so on (Jiang et al., 2016).

With the development of remote sensing techniques like photogrammetry and various sensors, researchers can easily obtain remote sensing information from various platforms. The Landsat 8 satellite can achieve global coverage every 16 days, and the cycle reduces to only 5 days considering both Sentinel-2A and Sentinel-2B. For the Sentinel-2 mission alone, 3.4 petabytes of remote sensing data have been acquired already (Yokoya, Zhu, & Plaza, 2017).Moreover, platforms of the unmanned aerial vehicle (UAV) provides another way to acquire remote sensing information. Nowadays, studies on radiometric changes between optical or spectral images are popular research area, and most of the algorithms also proposed based on these data (Du, Liu, Gamba, Tan, & Xia, 2012). A systematic survey of these methods has been provided by Radke et al.(Radke, Andra, Al-Kofahi, & Roysam, 2005). However, high false alarm rates due to irrelevant radiometric changes is a big problem for these methods, which is caused by shadows, vegetation, and moving object.

Digital surface model (DSM) is a power indicator to detect changes. In urban areas, if the elevation of a district is changed significantly, there are always changes in man-made objects as well. This kind of change can be detected using DSM data. Moreover, DSM can be used to identify different types of vegetation based on the height properties of different vegetation. Without using spectral information, the influence caused by shadow and different illumination is not a problem as well. Recently, the development of various techniques like laser scanning and stereoscopic images also provide researchers with opportunities to acquire the height information on their study areas. However, most of the existing DSM-based algorithms face a problem when the land cover changes are not accompanied by the changes in height, and therefore, many study areas show a bad performance if only uses DSM data. Hence, a change detection method which can utilize the properties of DSM and mitigate the drawback of this data is needed. This thesis intends to explore the ability of DSM data in change detection and to overcome the problem caused by land cover changes without accompanying changes in elevation to some extent. To fulfill this target, external remote sensing information and latest technique for change detection should be considered.

Inspired by the architecture of the human brain, deep learning (DL) become more popular in recent years. As a branch of machine learning, DL algorithms try to understand the inner relation of the input information. In order to discover good representations, DL techniques learn a hierarchy of features from low-level features to high-level ones (Nogueira, Miranda, & Santos, 2015). In the area of image processing, convolutional neural networks (CNN) are the most effective deep architectures. This type of architectures was once hampered for several years, which is mainly due to the high computational costs involved in the period of training networks (Zeiler & Fergus, 2014). Researchers started to study this technique again mainly owing to the advance of GPU technology.

It is worth to mention that, as the domain area of computer vision, identifying images in the pixel-level is as important as identifying the whole image. In order to obtain dense pixel-wise labeling, a patch based algorithm was employed in the CNN architectures (Kim, Ha, & Kwon, 2018). This approach decomposes the entire image into several equal size small patches, and use CNN to predict and return a class label for every patch center. After obtaining labels for all patches, these patches can then be re-joined together and produce the pixel-wise labeling result. A shortcoming of this algorithm is the repetitive use of overlap patches which increases the computational costs. Lately, Long et al. proposed fully convolutional networks (FCN), which replaced the fully connected layers by one or multiple convolutional layers that upsample the feature maps to obtain same resolution as input (Long, Shelhamer, & Darrell, 2015). This algorithm is superior to patch-based CNN in three ways: i) The number of parameters is reduced while obtaining dense pixel-wise labeling; ii)allows the CNN architecture to understand structure and relation over the entire image instead of a small patch; iii) the size of the input image is arbitrary (Long et al., 2015).

Today, CNN architecture has attracted a lot of attention, and many researchers contribute their ideas to this area. Noh et al. adopted deconvolution and unpooling layers to identify pixel-wise class labels (Noh, Hong, & Han, 2015). Yu and Koltun introduced the dilated convolutional layers, which allows exponential expansion of the receptive field without loss of resolution or coverage (Yu & Koltun, 2015). This network has been adopted in FCN and improved by using six layers of dilated convolutions (Persello & Stein, 2017). Moreover, Melekhov et al. adopted Siamese CNN in the changing detection areas (Melekhov, Kannala, & Rahtu, 2016). Instead of concatenating two images into one input and using one stream to learn and predict labels, this network allows two images as the input and treats them using two streams sharing the same weight. Although lots of networks have been proposed, CNN inevitably relies heavily on the number of manually labeled training samples. These training samples are labor intensive and even insufficient in some situation, which limits the application of CNN in some situation.

In this thesis, we want to mitigate the drawback of the DSM data and propose an unsupervised change detection method for the analysis of urban areas. The high-resolution images obtained by UAV is adopted as a supplement. Furthermore, FCN architecture is applied to further improve the labels generated from joint use of DSM-based and RGB-based method.

1.2 Research identification

This thesis aims to propose an unsupervised change detection method based on RGB and DSM data, which we refer to as RGBD data. DSM data is not popular in change detection because it can only detect changes that accompany changes in elevation. Therefore, we develop an unsupervised technique to detect changes using both RGB and DSM data. The RGB data is used to mitigate the weakness of DSM data in the areas without height changes. This is a more versatile method in urban areas because it can effectively detect changes regardless of whether the study area has 3D changes. More importantly, we intend to utilize the CNN architecture to understand the images and optimize the unsupervised result. Hence, the labels obtained as a result of the unsupervised techniques are used to train the CNN, which frees the CNN architecture from the manual annotation. In this way, CNN can learn and understand the images without the support of manually labeled training samples.

1.2.1 Research objective

- Propose an unsupervised change detection method based on the RGBD data.
- Allow the CNN architectures to be applied without the manually labeled training samples.

1.2.2 Research question

Objective 1:

- How to process DSM data to get the change detection area?
- Which method can detect change based on RGB data?

- How to strategically combine the result generated from these two types of data?

Objective 2:

- Is it possible to treat the result from an unsupervised method as training samples for CNN?
- What proportion of results from the unsupervised method will be used as training samples?

1.2.3 Innovation aimed at

- Change detection methods for multi-spectral remote sensing images or synthetic aperture radar (SAR) images have been widely investigated. However, this thesis uses RGBD data, which combine RGB with height information.
- Although the DSM-based method is able to detect changes well in areas with elevation changes, it fails to detect changes in areas without elevation change. This thesis intends to propose a novel unsupervised method, which can mitigate this drawback of DSM data.
- Traditional CNN architecture needs a large number of manual interpretation, which increases the labor cost and reduces work efficiency. This thesis aims to free the CNN architecture from manually labeled training samples and can be applied like the unsupervised method.

2 Literature review

This chapter presents the background knowledge of this research. In section 2.1, the background of change detection is described; related works are reviewed in section 2.1; the development of the CNN architecture is provided in the last section.

2.1 The context of change detection

Change detection is core part of image processing, which aims at identifying the differences of the land cover by processing two remote sensing images acquired at different times from the same geographical area (Bruzzone & Bovolo, 2013). In the beginning, detecting is a manual task, which is labor-intensive and time-consuming, and then it is replaced by various change detection algorithms (Singh, 1989). Change detection algorithms can be roughly divided into supervised and unsupervised methods. The main advantage of the supervised methods is that these algorithms often have a better result. However, these methods are based on the availability of training data, which is not always possible. On the other hand, the accuracy of unsupervised methods is usually lower than the supervised algorithms, but it can be applied more widely. In order to adopt a general change detection algorithm in the urban areas, some unsupervised change detection algorithms are reviewed in the following section.

2.2 Related work

The advantage of the unsupervised change detection method is that these algorithms can be applied without prior knowledge of the study area (Moser, Moser, & Serpico, 2002). In order to mitigate the problem of DSM-based algorithm when the change in land cover is not accompanied by the change in height, external information or special methods are needed. Hence, in the following section, algorithms based on DSM and traditional unsupervised methods are reviewed and analyzed.

2.2.1 Change detection algorithm based on DSM

Land cover changes are accompanied by height changes, especially in urban areas where most of the changes are caused by man-made objects. This allows DSM data to become a naturally good indicator for detecting changes in urban areas. Another advantage of this data is that it may help the typical unsupervised change detection methods to exclude the influences caused by shadow or different illumination.

DSM data with different resolution suits for different study areas, but acquiring various types of DSM data is not an easy task before. For example, in order to detect topographic changes (Baldi, Fabris, Marsella, & Monticelli, 2005), low-resolution DSM images are enough because this type of changes usually corresponds to large displacements (Guerin et al., 2014). It turns different when it comes to the context of the urban areas where higher resolution images are required due to the density of man-made objects.

With the development of techniques, the DSM images can be obtained in many ways, like airborne laser scanning (ALS) and stereoscopic images. Many researchers are then gradually moving to these areas as well. Based on the shape of the objects derived from ALS data, two independent segmentations were performed to extract buildings (Voegtle & Steinle, 2004). Jung (2004) proposed a technique which aims to detect changes utilizing grey scale stereo pairs. In the method, images were classified into buildings or not building classes, and the classifier was combined by several decision trees. Segmentation is needed in both of these two algorithms, and in this way, the error that is caused by the segmentation will accumulate and propagate to the final result. Many researchers select DSM obtained from ALS data or aerial images as it contains a better signal-to-noise ratio (Ioannidis, Psaltis, & Potsiou, 2009). These data sometimes are not quickly accessible, and as a result, DSM generated from two stereo pairs are thus good alternatives (Guerin et al., 2014). A common way to extract

changed areas from DSM images is firstly using thresholding, and then filtering algorithms like normalized difference vegetation index mask or spatial filtering can be applied. All of these algorithms show a good result if these changes accompany the changes in elevation. Some researchers also try to detect different classes based on the shape features (edges, area, elongation, eccentricity) (Chaabouni-Chouayakh, d'Angelo, Krauss, & Reinartz, 2011), but the contextual knowledge of study areas is needed.

Nowadays, despite the abundance of ways of obtaining DSM data, there is a lack of study using this data type. This is because of the drawbacks of DSM data is apparent when the changes in the study areas are small or no 3-D changes. Hence, a method that can utilize the advantages of DSM and can also mitigate its disadvantages is needed. In order to solve this problem, some traditional change detection methods based on spectral information are reviewed in the next sub-section.

2.2.2 Change vector analysis (CVA) algorithm

CVA is a traditional change detection algorithm which was proposed in 1980 (Malila, 1980) and still being improved in recent years. Lu et al. described it as an enhanced band differencing algorithms and able to detect any kind of changes (2004). This algorithm calculates the spectral difference value for each pixel in the corresponding position from two multi-spectral images, and then a binary result can be acquired by comparing the difference value with a threshold. As a pixel-based algorithm, it can avoid error propagation and able to detect changes effectively even the spectral information is not too much.

After being proposed by Malila (1980), this algorithm was adopted in various application, like monitoring coastal environment (Michalek, Wagner, Luczkovich, & Stoffle, 1993), monitoring of land cover (Johnson & Kasischke, 1998) and monitoring of logging activities (Silva, Santos, Shimabukuro, Souza, & Graca, 2003). In 2000, an expanded CVA method was proposed by utilizing the information inside the vector's spherical statistics in the change extraction process (Allen & Kupfer, 2000). Later, in order to mitigate the shortcomings in the threshold selection, an improved change vector analysis (ICVA) was proposed to find an appropriate threshold for CVA method (Chen, Gong, He, Pu, & Shi, 2003). As a change detection algorithm which works on multidimensional data, ICVA improve the way of selecting a threshold of change magnitude and importing the cosines of change vectors, which shows a good result in many areas. Futhermore, Chen et al. (2011) analyzed the posterior probability space by using CVA to overcome radiometric errors.

Traditional CVA algorithms focus on calculating the magnitude difference between n-dimensional spectral vectors, and it is hard to distinguish changes if most of the changed vectors have similar direction cosine values. Spectral angle mapper (SAM) was then applied for detecting changes in Landsat-5 TM images (Moughal & Yu, 2014). Zhuang et al. (2016) employed the SAM in the traditional CVA algorithm, which mitigates the shortcoming of only considering the difference of magnitude or the difference of angle between two spectral vectors. Nowadays, CVA algorithms have been enhanced for solving different problems, the main property of this algorithm is that it can be applied even if the spectral information is insufficient, which could be a good supplement for the algorithms based on the DSM data.

2.3 The development of the CNN architectures

In machine learning areas, an artificial neural network (ANN) technique was inspired by the visual cortex of animals (Hubel & Wiesel, 1968). The elementary unit of this technique is a neuron, which receives input information from other neuron or from the outside. In order to obtain the output from a neuron, all the input information will be processed by the weight , bias and an activation function. Assuming the input vector is $\boldsymbol{x} = [x_1, x_2, x_3]$, and the corresponding weight and bias is $\boldsymbol{w} = [w_1, w_2, w_3]$ and $\boldsymbol{b} = [b_1, b_2, b_3]$ respectively. Equation 1 illustrates this process:

$$Y = f(\boldsymbol{w} \cdot \boldsymbol{x} + \boldsymbol{b}) \qquad \text{Equation 1}$$

In this equation, the input vector performs a linearly transform by weight and bias, and then a non-linearly activation function *f* is applied to decide the output information finally.

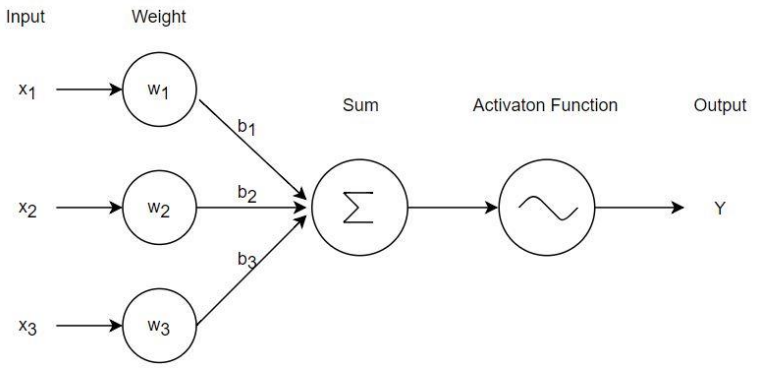


Figure 2-1 Schematic representation of a primary system of ANN

When combining these basic units, the architecture of a feedforward neural network is built up. In the beginning, feedforward neural network architectures can be divided into two groups according to with or without the hidden layer. The single-layer perceptron only consists of the input layer and the output layer, while the multi-layer perceptron is proposed by adding one or more hidden layers to this architecture. Figure 2-2 shows an example for the multilayer perceptron (MLP) architecture. In this figure, there are two hidden layers between the input and output layers. Each circle represents a primary processing neuron mentioned before.

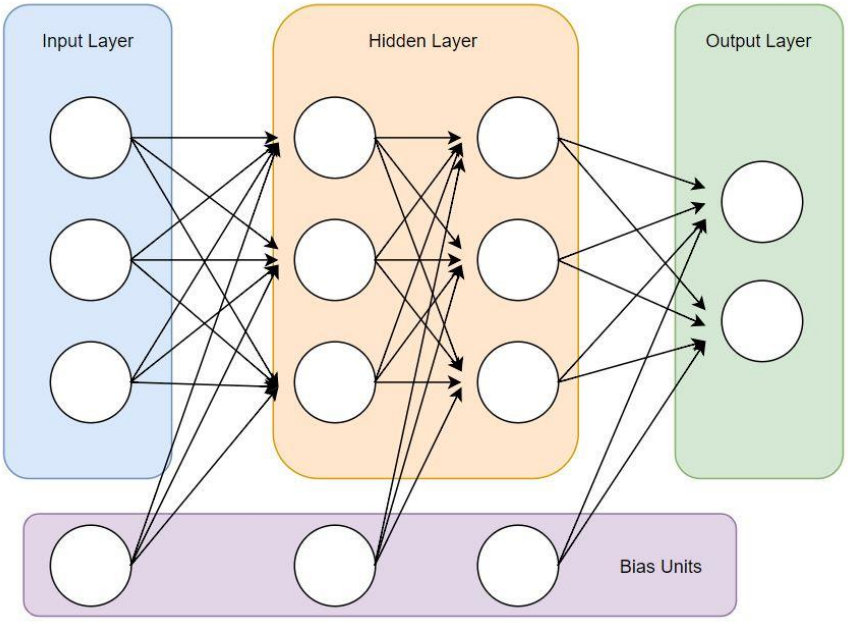


Figure 2-2 The diagram of multilayer perceptron

Meanwhile, the backpropagation (BP) algorithm allows the perceptron to optimize the objective function (Rumelhart, Hinton, & Williams, 1985). BP algorithm was designed to use the chain rule to compute derivatives in order to tune the weight by using gradient descent algorithms. A mathematical definition of gradient descent is given as below:

$$\Delta W(\tau) = -\eta(\tau)\frac{\partial E(\tau)}{\partial W(\tau)} + \alpha \Delta W(\tau - 1) \quad \text{Equation 2}$$

In this equation, ΔW and E represents for the weight updata and the error value, and therefore, $\frac{\partial E(\tau)}{\partial W(\tau)}$ represents the gradient. In addition, η and α represents the learning rate and the momentum rate respectively.

After several years, LeCun adopted this algorithm in the CNN marking the beginning of the CNN architectures (LeCun et al., 1989). In that paper, LeCun trained a multi-layer neural network using the BP algorithm to identify the handwritten numbers. Later, LeCun improved this architecture and proposed the LeNet-5 architecture (Table 2-1), which successfully recognized the visual patterns directly from the input images without processing (Lecun, Bottou, Bengio, & Haffner, 1998). However, this technique was hampered by two problems: (i) the BP algorithm requires a large amount of computation, which was hard to be satisfied by the hardware of that time; (ii) many shallow machine learning algorithms like support vector machine (SVM) attracted researchers' attention.

Table 2-1 Representation of the LeNet-5 architecture

Layer	Dimensions	Parameters			
		No. of filters	Filter dimensions	Stride	Pad
Input	32×32×1	---	---	---	---
Conv-1	28×28×6	6	5×5	1	0
Pooling-1	14×14×6	-	2×2	2	---
Conv-2	10×10×16	16	5×5	1	0
Pooling-2	5×5×16	---	2×2	2	-
Conv-3	1×1×120	120	5×5	1	0
FC-6	84 neurons	---	---	---	---
Output	10 neurons	---	---	---	---

CNN architectures were plagued by these problems in the following several years. In 2006, Hinton broke the silence and published an article on Science (G. E. Hinton & Salakhutdinov, 2006). He pointed out that neural networks with multiple hidden layers have good feature learning abilities and he also proposed that the complexity of training can be reduced by initializing the weight. In the meantime, the emergence of the GPU provides an opportunity for the development of deep learning and CNN. Later in 2012, Krizhevsky et al. proposed the AlexNet (Table 2-2) and won two first prizes in the ImageNet competition (Geoffrey E. Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012).The main properties of the AlexNet architecture are: (i) consists of 5 convolutional layers and 3 fully connected layers; (ii) utilizes of drop-out strategy to mitigate the overfitting problems; (iii) instead of using sigmoid activation function, this architecture adopts the rectified linear unit (ReLU), which allows neural networks find the optimal value at a fast pace.

Table 2-2 Representation of the AlexNex architecture

Layer	Dimensions	Parameters			
		No. of filters	Filter dimensions	Stride	Pad
Input	227×227×3	---	---	---	---
Conv-1	55×55×96	96	11 × 11	4	0
Max Pool-1	27×27×96	---	3×3	2	---
Norm-1	27×27×96	---	---	---	---
Conv-2	27×27×256	256	5×5	1	2
Max Pool-2	13×13×256	---	3×3	2	---
Norm-2	13×13×256	---	---	---	---
Conv-3	13×13×384	384	3×3	1	1
Conv-4	13×13×384	384	3×3	1	1
Conv-5	13×13×256	256	3×3	1	1
Max Pool-3	6×6×256	---	3×3	2	---
FC-6	4096 neurons	---	---	---	---
FC-7	4096 neurons	---	---	---	---
FC-8	1000 neurons	---	---	---	---

Furthermore, Zeiler and Fergus make some improvement on the AlexNet architecture by reducing the stride and the receptive field of the first convolutional layer (Zeiler & Fergus, 2014). A new VGG16 network which consists of 16 convolutional and 3 fully connected layers was also adopted (Simonyan & Zisserman, 2014).Table 2-3 shows the architecture of the VGG16, and all the convolutional layers are followed by a nonlinear algorithm (ReLU). This network provides a clue for the CNN architectures that a deeper layer network is more robust and could be a more accurate classifier. Instead of using traditional architectures, which stacking of layers, a novel network called GoogleNet adopted the multiscale architecture which dramatically reduces the computational costs (Ioffe & Szegedy, 2015). In the area of pixel-wise prediction, the FCN architecture significantly reduced redundancy of predicting labels, comparing with the 'patch-based' classification architectures (Long et al., 2015). This architecture replaces the traditional three fully connected layers in the CNN with convolutional layers and allows to get a pixel-by-pixel output. Then, Persello & Stein (2017) adopted an FCN-DK6 architecture to detect the informal settlement. Instead of using the traditional downsampling and upsampling, the FCN-DK6 architecture adopts increasing dilated factors based on the FCN architectures, which increases the receptive field without increasing the number of memory parameters.

Table 2-3 Representation of the VGG16 architecture

Layer	Dimensions	Parameters			
		No. of filters	Filter dimensions	Stride	Pad
Input	224×224×3	---	---	---	---
Conv 1-1	224×224×64	64	3×3×3	1	2
Conv 1-2	224×224×64	64	3×3×64	1	2
Max Pool-1	112×112×64	---	2×2	2	---

Conv 2-1	112×112×128	128	3×3×64	1	2
Conv 2-2	112×112×128	128	3×3×128	1	2
Max Pool-2	56×56×128	---	2×2	2	---
Conv 3-1	56×56×256	256	3×3×128	1	2
Conv 3-2	56×56×256	256	3×3×256	1	2
Conv 3-3	56×56×256	256	3×3×256	1	2
Max Pool-3	28×28×256	---	2×2	2	---
Conv 4-1	28×28×512	512	3×3×256	1	2
Conv 4-2	28×28×512	512	3×3×512	1	2
Conv 4-3	28×28×512	512	3×3×512	1	2
Max Pool-4	14×14×512	---	2×2	2	---
Conv 5-1	14×14×512	512	3×3×512	1	2
Conv 5-2	14×14×512	512	3×3×512	1	2
Conv 5-3	14×14×512	512	3×3×512	1	2
Max Pool-5	7×7×512	---	2×2	2	---
FC-6	1×1×4096	---	---	---	---
FC-7	1×1×4096	---	---	---	---
FC-8	1×1×1000	---	---	---	---

Nowadays, the computational costs have been substantially reduced, and this technique has been applied in various areas like image classification (Romero, Gatta, & Camps-Valls, 2016), object tracking (Fan, Xu, Wu, & Gong, 2010), change detection (Liu, Gong, Qin, & Zhang, 2018), text detection and recognition (Jaderberg, Vedaldi, & Zisserman, 2014) and so on. However, all of these architectures still show a bad performance when the training samples are insufficient. The ground truth is not always quickly available, and the labeling image is a time-consuming task. This not only slows down the pace of experiments but also puts a limitation on the study areas. Hence, a method to get reliable training samples on time is urgently needed.

3 Method

In this section, we will propose a novel unsupervised change detection method and this method will fulfill two goals: (i) combine the DSM with RGB data and proposes a more general change detection algorithm for the analysis of urban areas; (ii) take the CNN techniques out of the constraints of the manually labeled training samples. The main idea of this method is adopting the result from the unsupervised method and treating them as training samples of the CNN networks. The detail information will be presented in the following sub-sections. Section 3.1 defines a strategic way to combine the results from RGB and DSM data. In order to treat these results as training samples, section 3.2 presents the way to process them. The CNN architectures we adopted and the corresponding parameters are provided in section 3.3.

3.1 The unsupervised change detection method

3.1.1 Change detection method based on DSM data

In order to obtain changes from DSM, a threshold value *t* should be selected first. Then, the height difference in the corresponding pixels from the two images H_1 and H_2 should be calculated (Equation 3).If this difference of one pixel is equal to or greater than threshold value t this pixel will be recoginised as a potentially changed pixel, otherwise, this pixel will be considered as an unchanged pixel (Equation 4).

$$D_{DSM} = |H_1 - H_2| \quad \text{Equation 3}$$

$$CD_{DSM} = \begin{cases} 1 & if \quad D_{DSM} \geq t \\ 0 & if \quad D_{DSM} < t \end{cases} \quad \text{Equation 4}$$

In Equation 3, H_1 and H_2 represent the height value for two DSM images in the corresponding pixel. D_{DSM} represents the height differences between two images. CD_{DSM} represents the change detection result using the DSM-based method. It is worth to mention that, the threshold value should be carefully selected based on the study areas. If this value is small, moving object and growing vegetation will interface the detection result, while if it is very big, some changed pixels will be wrongly classified.

Most of the changes that occurred in urban areas are relevant. Therefore, changes with few pixels together are considered as noises and they are not candidates of changes. The morphological opening is then applied to remove isolated changes. As Equation 5, the changed areas are then going to be processed by the operation of erosion and then dilation. *B* is the structuring element. The size of the structuring element determines the size of smoothed areas, so it should be set according to the resolution and the smallest interested objects in the study areas.

$$(A \ominus B) \oplus B \quad \text{Equation 5}$$

At last, a binary change detection result can be extracted from DSM data. Based on the porpoises of the DSM data, changed labels are more reliable than the regions labeled as unchanged. Therefore, we consider that the areas detected as changed class are more reliable than areas detected as unchanged.

3.1.2 Change detection method based on RGB data

Most of methods in change detection areas need hyper-spectral images, and RGB only contains three bands, so they are not suit for our experiment. The CVA algorithm is a trditional change detection method which can be applied in this situation. This algorithm computes the multispectral difference and exploits its statistical distribution in spherical coordinates (Malila, 1980).

Let the $\boldsymbol{S_1} = (x_1, x_2, \dots, x_n)$ and $\boldsymbol{S_2} = (y_1, y_2, \dots, y_n)$ represent the spectral vector of two input images. Equation 6 shows the equation to calculate the difference of two bands based on CVA algorithm. D_{CVA} represents the difference of magnitude between two images. The x_m and y_m represents the spectral component in the band $m = (1,2, \dots, n)$ of multi-spectral images.

$$D_{CVA} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{m=1}^{n} (x_m - y_m)^2} \quad \text{Equation 6}$$

Finally, a threshold can be selected to compare with the D_{CVA} and generate changed or unchanged binary results for each pixel.

The SAM algorithm can also be applied if only three spectral bands are available (Moughal & Yu, 2014). It extracts the spectral angle θ according to the input vectors, and comparing this value with a threshold to generate a binary result.

$$\theta = arccos\left[\frac{(\sum_{m=1}^{n} x_m y_m)}{(\sqrt{\sum_{m=1}^{n} {x_m}^2}\sqrt{\sum_{m=1}^{n} {y_m}^2})}\right], \qquad \theta \in [0, 90^{\circ}] \quad \text{Equation 7}$$

From the literature review, we can know that if we treat the input images as vectors, the CVA algorithm detect changes according to the magnitude of two input vectors, while the SAM algorithm considers the angle between these two vectors. Figure 3-1 shows the difference between these two algorithms.

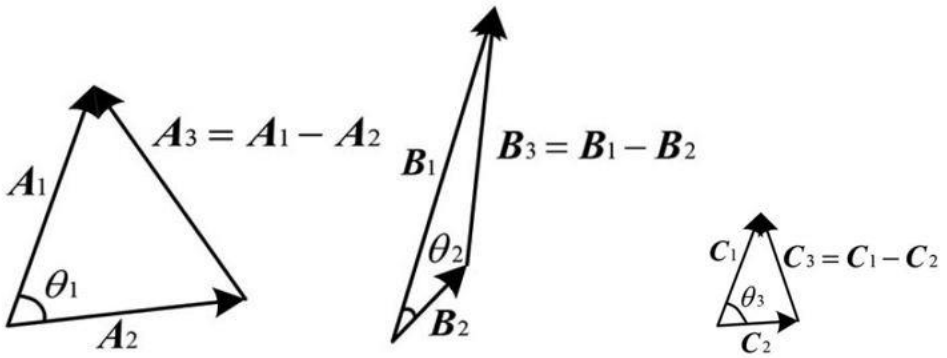


Figure 3-1 Three types of changes in the real world in multi-spectral images. (a) presents that changes caused by the large difference in magnitude and spectral angle; (b) illustrates the type of changes, which possess the large difference in magnitude but a small difference in spectral angle; (c) presents a kind of change with a small magnitude change but a large change in the spectral angle. $A_1, A_2, B_1, B_2, C_1, C_2$ represents the vectors generated from two images; vector A_3, B_3, C_3 represents the difference for the corresponding vectors; Angle $\theta_1, \theta_2, \theta_3$ presents the difference in angle; $\theta_1 =$

θ_3. Adapted from "*Strategies Combining Spectral Angle Mapper and Change Vector Analysis to Unsupervised Change Detection in Multispectral Images*," by H.Zhuang, 2016, *IEEE Geoscience and Remote Sensing Letters*, 13(5), p. 681. Copyright 2019 by Jianda

Figure 3-1 illustrates three types of changes in the real world. (i) Changed type presented in (a) can be detected using both CVA and SAM method; (ii) the changed type in (b) can only be detected using CVA as the difference of magnitude is large, but the angular difference is small in this situation; (iii) the changed type in (c) has a large angular difference but difference in the magnitude is small, so the SAM can be used here to detect changes.

In principle, CVA&SAM method, which combines the CVA and the SAM method (Zhuang et al., 2016), can mitigate the shortcoming of using CVA and SAM alone, and as a consequence, all the three types of changed categories presented in Figure 3-1 can be detected. In order to improve the change detection effect, this thesis compares the effect of the CVA, SAM and CVA&SAM algorithms in our study areas. Here, the range of the SAM result $\theta(x, y)$ is $[0, 90^{\circ}]$, while the range of the CVA result D_{CVA} is $[0, L]$ (L is the grayscale of the input image). Therefore, the range of $\theta(x, y)$ multiplied by the coefficient k to make them comparable. Coefficient can be obtained using Equation 8.

$$k = L/90 \qquad \text{Equation 8}$$

In the last part, an automatic threshold algorithm Otsu is adopted to obtain the binary change detection result (Otsu, 1979). Initially, this algorithm is used to classify the image into a background class and object class. It converts the greyscale image to monochrome, and then, calculates the optimal threshold to obtain the maximal inter-class variance and minimal intra-class variance. This threshold is then compared with each pixel value and pixel values that smaller than the threshold are classified as the unchanged class, while the other greater one is classified as the changed class.

3.1.3 The strategy combines the change detection result from DSM and RGB

In urban areas, the significant height variation in a large area can be a clue for changing in land cover. Therefore, the changed areas detected using DSM data in this thesis have been assigned as final changed areas directly. Due to the properties of the method used in DSM data, we know that areas labeled as unchanged are less reliable than the areas labeled as changed because changes can occur without elevation changes. Hence, the RGB-based method is then adopted in the changed areas detected by the DSM method. At last, the final binary result is generated. Figure 3-2 shows the flowchart of this unsupervised method.

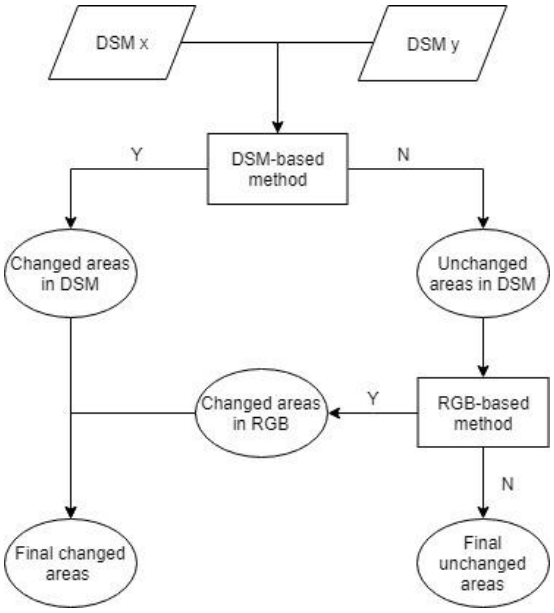


Figure 3-2 The flowchart of unsupervised algorithms

3.2 Process the unsupervised result

Before using the unsupervised results as training samples of the CNN, some processes are needed to apply to these results. In order to provide some improvement on the unsupervised result, we remove part of the labels generated from the unsupervised method and then use the FCN architecture to predict a new label for them again. For achieving this target, several steps are performed as following sub-sections.

3.2.1 Define and calculate the reliability of the unsupervised result

In the unsupervised part, all the results from RGB and DSM data are acquired by the thresholding method, which compares the difference of images with the threshold value. If the difference is greater than the threshold, the result of this pixel is changed and vice versa. In this sub-section, we want to assign a reliability value to all the results to represent the correct probability of this result.

Now we explained how to calculate the reliability based on the DSM-based method. Let t represent the threshold value, and H_1, H_2 represent input DSM images respectively.

$$D_{DSM} = |H_1 - H_2| \quad \text{Equation 9}$$

$$result = \begin{cases} changed & if \quad D_{DSM} \geq t \\ unchanged & if \quad D_{DSM} < t \end{cases} \quad \text{Equation 10}$$

D_{DSM} represents the difference of height between two images. If $D_{DSM} \geq t$, the result is judged as changed, and if $D_{DSM} < t$, it is classified as unchanged. Assuming the range of D_{DSM} is $[a, b]$, all the pixel value that greater than t will be classified as changed area, and pixels smaller than t are unchanged. Figure 3-3 presents a picture, which x axis D is represented the difference of height.

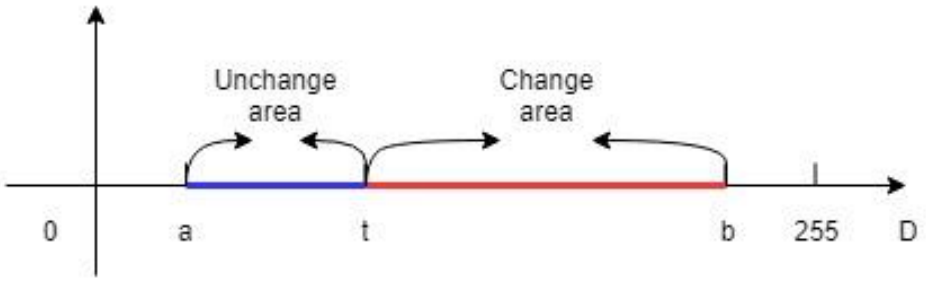


Figure 3-3 Representation of the changed and unchanged areas

Compared to other pixel values, pixel values near the threshold are associated with higher uncertainty. Therefore, we assume that the farther the pixel value is from the threshold, the more likely the pixel is correctly classified. This means that if the difference between the two images is far from the threshold, they should be considered more reliable, and deserve high reliability. If we consider the distance between the t and D_{DSM} as the reliability, we can find that for the changed areas the range of distance is $[0, b-t]$, while for the unchanged areas the range of result is $[0, t-a]$. In order to get the same range of results in both cases, we assume that for the unchanged areas if there is no change in elevation, then the DSM-based method should determine that the result has a reliability of 1. Correspondingly, for the changed areas if the height difference between the two images reaches a certain value, the DSM-based method should determine that the reliability of this result is 1 as well. This value is determined as 2 times the threshold at last, so that the slope of the reliability is the same in two cases. Figure 3-4 illustrate the relation between the difference D and the reliability R, and D here is the difference in elevation.

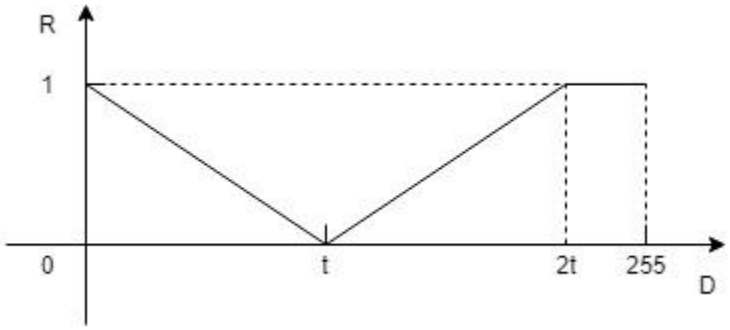


Figure 3-4 The relation between reliability and difference from the DSM-based method

For the RGB-based algorithm, the relationship between reliability and difference has a small difference. Instead of setting the threshold by ourselves, we use the Otsu algorithm to automatically generate the threshold in the RGB-based method. This method is classified by analyzing the whole image. Therefore, when the difference value obtained by this method is the smallest and the biggest in this image pair, the reliability reaches 1. Assuming the range of the difference between two images (in magnitude or angle) is $[a, b]$ as well, the relation between the reliability and difference in RGB is presented in Figure 3-5.

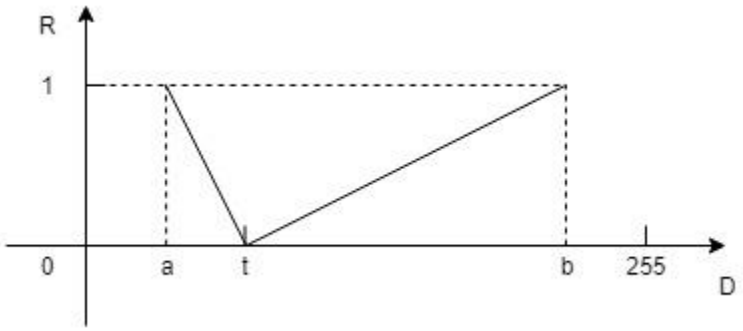


Figure 3-5 The relation between reliability and difference from the RGB-based method

In this way, we can obtain the reliability from the DSM-based method R_{DSM} and from the RGB-based method R_{RGB} in each pixel. We call it reliability matrixe, and the size of these two matrixes are the same as the size of images.

3.2.2 Extract the reliability matrix from the unsupervised method

From section 3.1.3, we know that the binary result from the unsupervised method is firstly generated from the RGB and the DSM method separately, and they are combined together following a specific rule. The way of generating the reliability matrix is similar to the method used in section 3.1.3. If a label is obtained from the RGB-based method, then the reliability matrix R_{RGB} in the corresponding pixel is used in this pixel; if the label is obtained from the DSM method, then the reliability matrix R_{DSM} is used.

Figure 3-6 Extract the reliability matrix. Graph (a) is the binary result from the DSM data; graph (b) is the binary result from the RGB data; graph (c) is the final binary result of the unsupervised method. red color represents for changed areas, and the green color represents for the unchanged areas

In Figure 3-6, we assume that the size of the unsupervised result is 4*4, color represents the binary result, and the value represents reliability. If one pixel is changed in (a), then this result and the corresponding reliability will be inherited by (c). Similarly, if one pixel is not changed in (a), all the corresponding result and reliability in (b) will be inherited by (c).

3.2.3 Remove less-reliable results

Due to the definition of reliability, it is easy to know that the larger of reliability value for a pixel, the more reliable label it is. Based on this rule, some less reliable labels can be removed to leave some space for improving the result from the unsupervised method.

In this thesis, we first decided the number of the result m to be removed. Furthermore, the reliability values obtained from the unsupervised method are sorted from smallest to largest on an image basis. Then, the lowest m reliability values are removed, and the other result can be treated as references to train the FCN architectures.

3.3 Training and configuring the FCN architectures

This thesis adopts the FCN-DK6s architecture proposed by Persello and Stein (2017), the detail information is presented in the following subsection.

3.3.1 Input layer

The raw images used in this thesis are combined from RGB and DSM images. Let W represents the height and width of images, and the image size is $W \times W \times 4$. These two images are concatenated together to generate an image with the size of $W \times W \times 8$, because of the FCN-DK6s architecture can only receive one image as input. In the end, a result with the size of $W \times W \times Nc$ is generated where Nc is the number of classes. Figure 3-7 provides a schematic representation for this step. The labels derived from the unsupervised method are used as training samples.

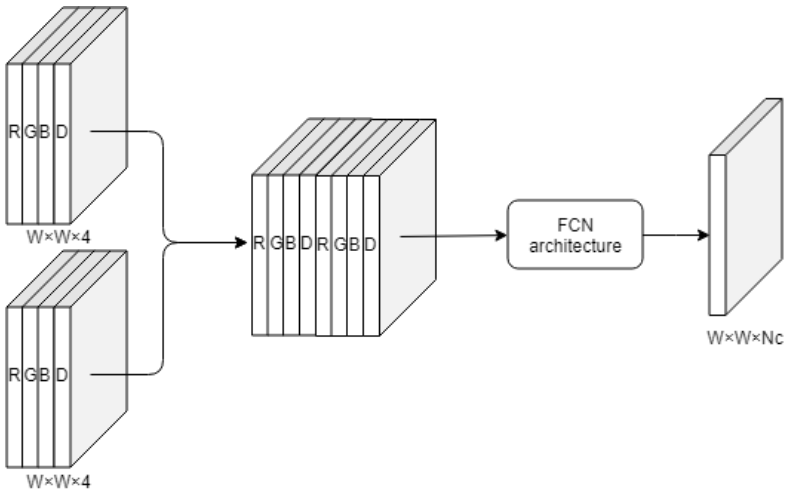


Figure 3-7 Schematic represents the way of concatenating

3.3.2 Convolutional layer

The convolutional layer is the core unit of the CNN architecture where most of the computation is involved. A convolutional layer consists of some learnable filters or kernels. Each filter is convolved across the feature maps during the forward pass to produce a separate 2-dimensional activation map. This map responses of every spatial position and generate the output. Here, the complexity of the networks is reduced since the neurons that lie in the same features maps are sharing the weight, which significantly reduces the number of parameters (Geoffrey E. Hinton et al., 2012). The number of activation maps is the same as the number of filters.

There is another kind of parameter called hyper-parameters, which are controlled by the researchers. The receptive field is one of them, which represents the spatial extent of sparse connectivity between the neurons of two layers (Aloysius & Geetha, 2017). Filter dimension controls the number of filters of the output volume.

The stride *s* represents the distance of kernel moving every time and the downsampling factor ($s = 1$ represents no downsampling). In addition, padding is used for filling borders, which can also shape the size of output. The 'zero-padding' means that all the value used to fill the border are zero. Furthermore, this thesis adopts the dilated convolutional layers, which allows an exponential expansion of the receptive field without loss of resolution.

Given an input image with the size of $W \times W$, and the size of the kernel is $K \times K$. Let s and p represent the stride and zero-padding respectively. The following equation presents the size of the output feature map.

$$\left[\frac{W - K + 2p}{s} + 1\right] \times \left[\frac{W - K + 2p}{s} + 1\right] \quad \text{Equation 11}$$

3.3.3 Batch Normalization layer

In the process of training, if the parameters of the previous layer are changed, the distribution of the next layer inputs may be affected. This can reduce training efficiency by requiring lower learning rates and careful parameter initialization. Ioffe & Szegedy (2015) solved this problem by performing the normalization for each training mini-batch. In this thesis, batch normalization layers are adopted after the convolutional layer blocks.

3.3.4 Activation Functions

Activation functions are non-linearity mathematical operations, which able to deal with more complex problems. Some common such functions are discussed here.

Sigmoid

This function takes a real-valued function as input and normalizes it into the range between 0 and 1.

$$f(x) = \frac{1}{1 + \mathrm{e}^{-x}} \quad \text{Equation 12}$$

In recent years, this function is not widely applied in the CNN architectures as two major drawbacks: (i) saturate and vanishes gradient at either tail, 0 or 1; (ii) the outputs of this function are not zero-centered, which causes the gradients to oscillate between positive and negative values during gradient descent.

Rectified Linear Units (ReLU)

The mathematical representation is provided in Equation 13.

$$f(x) = \begin{cases} 0 & if \quad x < 0 \\ x & if \quad x \geq 0 \end{cases} \quad \text{Equation 13}$$

From Equation 13, we can know that this function is a linear activation function, which has thresholding at zero. Krizhvsky et al. (2012) found that compared with the sigmoid, this function resulted in faster convergence of stochastic gradient descent. This is because of the non-saturating form when the input is greater than 0. However, this algorithm is sensitive to the large gradient and learning rate, since if the input is smaller than 0, the output gradient is always 0.

Leaky Rectified Linear Units (Leaky ReLU)

The Leaky ReLU activation function is modified from the ReLU algorithm. Instead of generating 0 when the input smaller than 0, this algorithm provides a small gradient. By doing this, the problem of the ReLU algorithm is solved while its merits are retained. The mathematical equation can be represented as:

$$f(x) = \begin{cases} 0.01x & if \quad x < 0 \\ x & if \quad x \geq 0 \end{cases} \qquad \text{Equation 14}$$

The activation function applied in this thesis is the Leaky ReLU algorithm.

3.3.5 Softmax layer

The aim of the softmax layers is to perform the classification, and it accepts a score value equal to the number of classes k. The sum of the output values is equal to 1, which can represent the probability distribution over K possible outcomes.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad for\ z = 1,2,3,\dots,k \qquad \text{Equation 15}$$

In Equation 15, z represents a vector of the class input, which is 2 in the binary change detection architectures.

3.3.6 Dropout layer

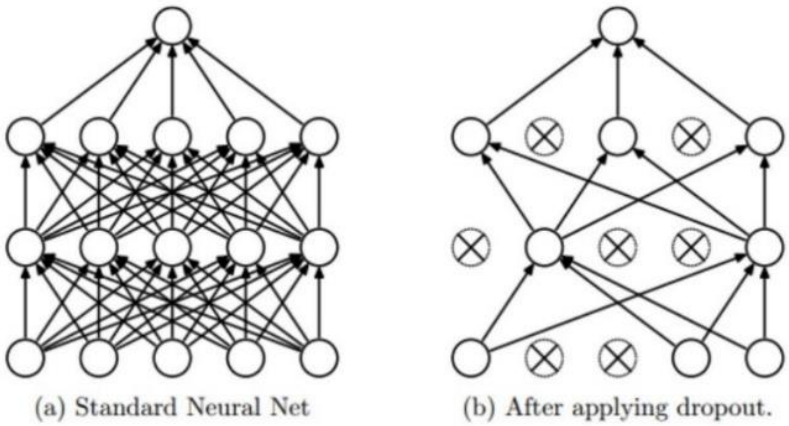


Figure 3-8 Schematic represents the dropout method. Adapted from "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," by Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014, Journal of Machine Learning Research, 15, p.1929–1958. Copyright 2019 by Jianda

Overfitting is a big problem in ANN architecture. Srivastava et al. (2014) adopt a method to skip some neurons in a certain probability during training the network. Dropout method allows a model to learn more robust feature and to obtain a more general result. This thesis adopts it at a rate of 0.5 in the last classification layer.

4 Experiment setup

This chapter presents how the dataset is prepared and how the parameters in the unsupervised method and in the FCN architectures are selected. Figure 4-1 describes the workflow of this experiment.

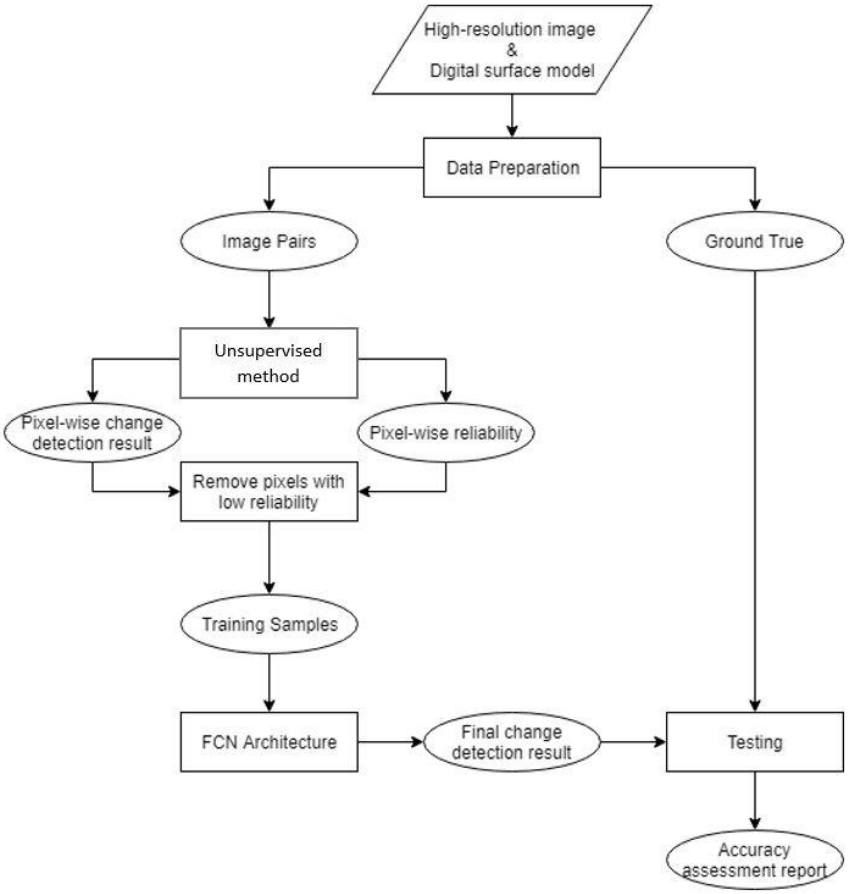


Figure 4-1 Workflow

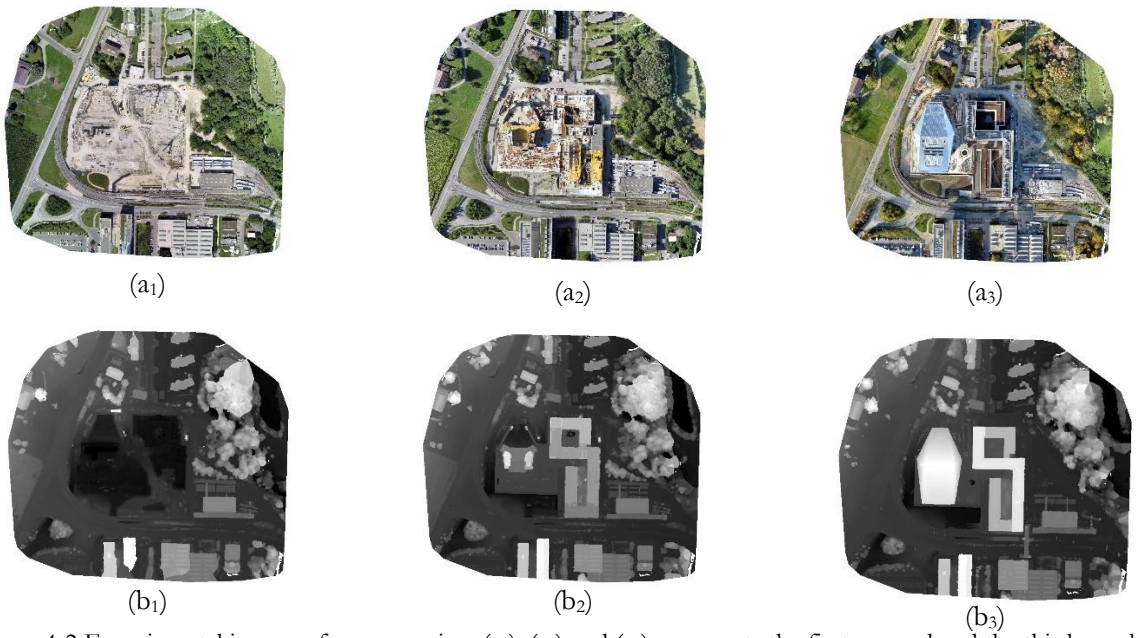
4.1 Data preparation

4.1.1 Study area

This research is applied to a municipality called Ecublens in Switzerland, which is located in the district of Ouest Lausannois in the canton of Vaud. The data is orthophoto acquired by the UAV, and the corresponding DSM data is produced from photogrammetry using the overlapping images acquired with the drone. These multi-temporal images consist of 3 epochs to monitor the changes in the study area. The center of the study area is a constructional area in the first epoch, and a hospital was built up in the last epoch. The size of the study area is about 32,830 m^2. The UAV images are acquired with a sampling distance of 5 cm. The owner of this data is Pixel4D.

4.1.2 Pre-processing

The data used in this thesis is high-resolution images, which limits the size of the research area under a certain number of pixels. Therefore, we firstly reduce the resolution of all orthophoto and DSM images to 14 cm. Then, these images were clipped to the same size. The result of all experimental images is presented in Figure 4-2.

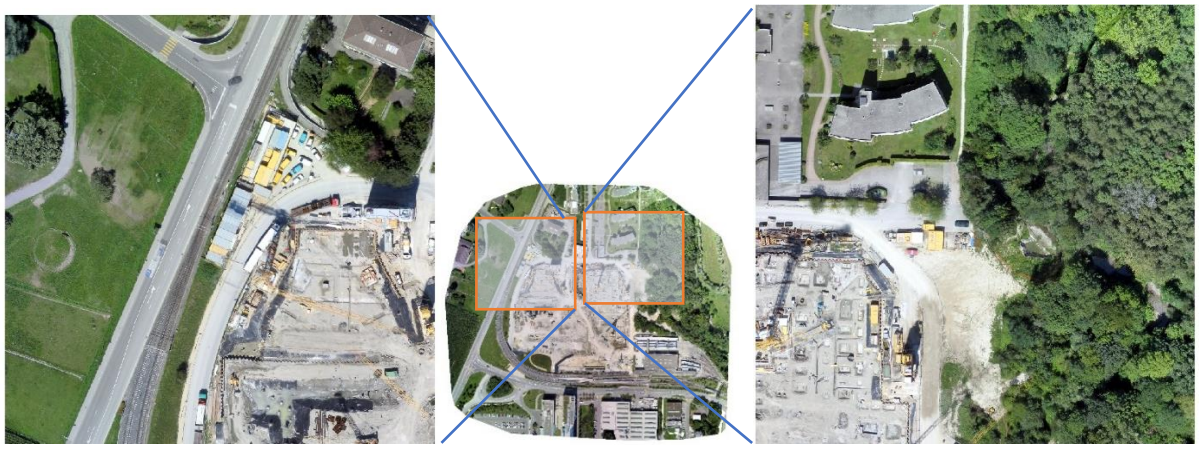


(a_1) (a_2) (a_3)

(b_1) (b_2) (b_3)

Figure 4-2 Experimental images after processing. (a_1), (a_2) and (a_3) represents the first, second and the third epoch of orthophotos respectively; (b_1), (b_2) and (b_3) represents the first, second and the third epoch of DSM data respectively

The size of these images is 9525×8524, which is too large to learn in the FCN architecture. Therefore, four tiles with the size of 1000×1000 were extracted from each image. Due to the main changed area in this study area is the hospital in the center, these four tiles are selected around the hospital to balance the number of changed and unchanged pixels. Figure 4-3 shows the four tiles in the first epoch.



(a) *(b)*

(c) *(d)*

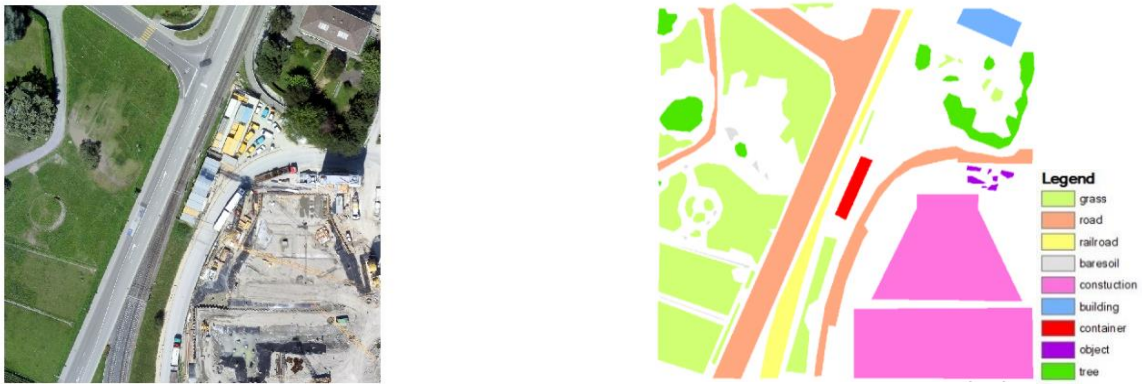
Figure 4-3 Four tiles of the first epoch. (a), (b), (c) and (d) is the first, second, third and fourth tile respectively

4.1.3 Annotation

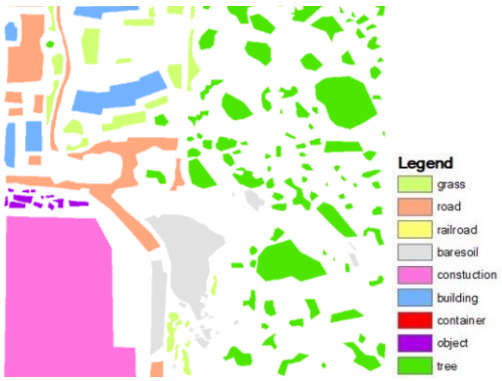
In order to assess the result of the unsupervised result and the result of the FCN architecture, pixel-wise ground truths were manually annotated and classified into the following seven land-cover categories: (i) Grass, (ii) Tree, (iii) Railroad, (iv) Bare soil, (v) Construction area, (vi) Container and (vii) Object. The rules for creating annotation are:

- Annotation is generated from the static objects on the ground while moving object is not a class (e.g., Cars running on the road, then the annotation is road class).
- The transition area was neglected (e.g., The transition from grass to bare soil keeps unlabeled).
- Shadow is labeled as the real land cover in this area (e.g., There is a shadow under the tree on the grass, then label it as grass class).
- The container is defined as an independent category as the regular shape.
- Except for the containers, the other elevated items on the ground are labeled as Object class (including stones, tube, constructional materials and so on).
- When the grass grows on the building roof, the annotation is Building class.

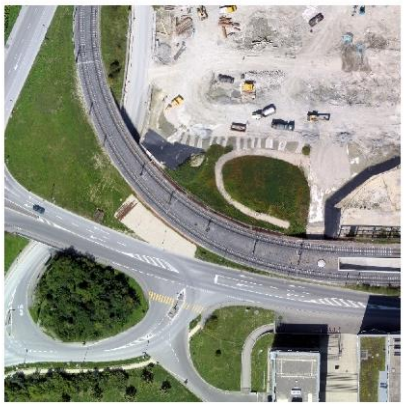
The result of annotation for the first epoch and the corresponding tile of the first epoch are presented in Figure 4-4. The other annotation results are provided in section 9.1.

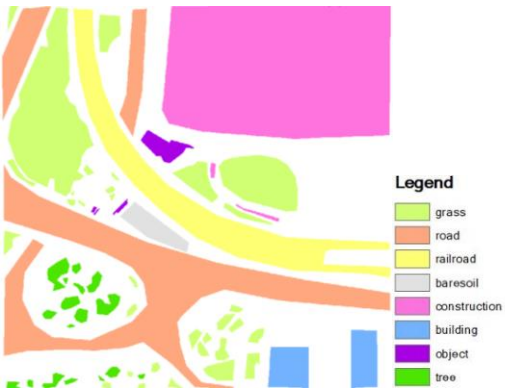


(a)

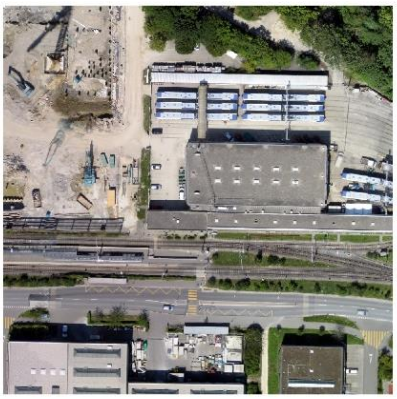


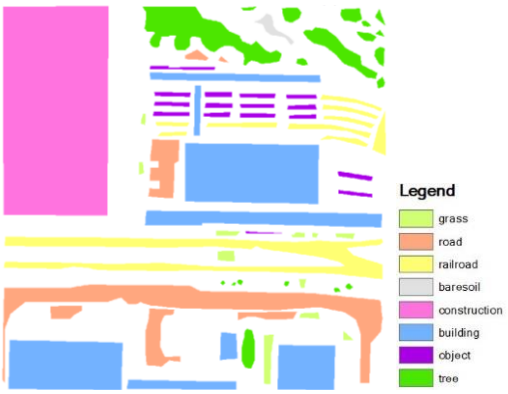
(b)





(c)





(d)

Figure 4-4 The annotation and corresponding of the first epoch. (a), (b), (c) and (d) represents the first, second, third and fourth epoch respectively

Then, the binary change detection result can be generated according to the annotated image pairs. Before generating the binary reference, we decide that trees growing with the time is not change, while if the height of building changes we label it as change. These three epochs can generate three change detection pairs. Here, the reference for detecting changes between the epoch 1 and the epoch 3 is provided in Figure 4-5. The other two references are provided in section 9.2.

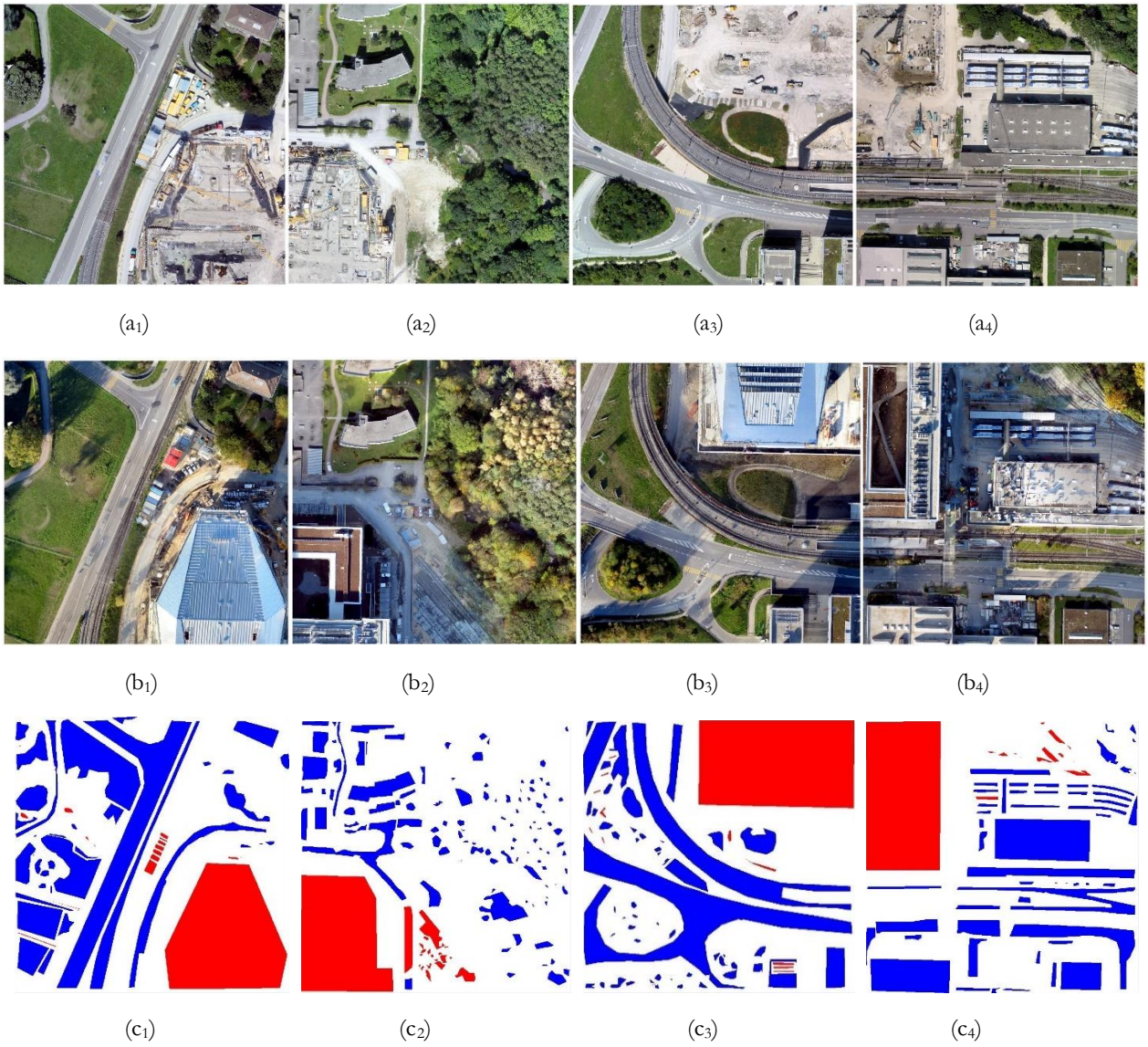
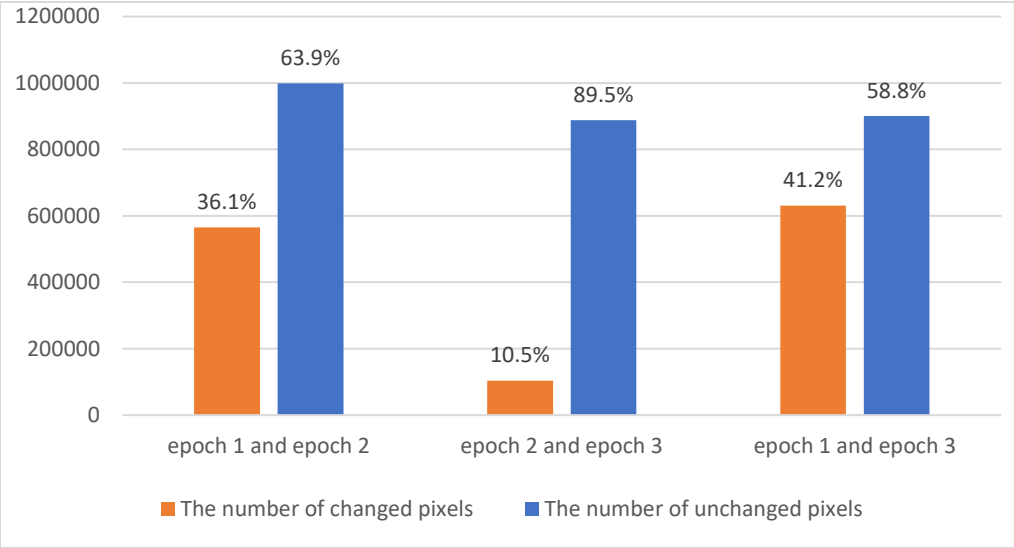


Figure 4-5 Change detection reference and raw images. (a_1), (a_2), (a_3) and (a_4) is the first, second, third and fourth tile of epoch 1; (b_1), (b_2), (b_3) and (b_4) is the first, second, third and fourth tile of epoch 3; (c_1), (c_2), (c_3) and (c_4) is the first, second, third, fourth tile of manually labeled reference, which red and blue color represents the changed and unchanged areas respectively, and the white color represents the areas without annotation

The number of labeled changed, and unchanged pixels for all three combinations is presented in Table 4-1.

Table 4-1 The number of annotated pixels for the changed and the unchanged classes



4.2 Model parameter

4.2.1 The parameters in the unsupervised method

The parameters in the DSM method are the thresholding value and the size of the structuring element for the morphological opening operation. Several combinations are tested, and finally, thresholding value equal to 3 meters and the window size equate to 10 pixels are used (detailed information and the comparison of the result are provided in section 5.2).

4.2.2 Structure and parameters of the FCN architecture

This thesis adopts the FCN-DK6 architecture (Persello & Stein, 2017) the architecture is provided in Table 4-2. This architecture contains 6 convolutional layers with the kernel size of 5 × 5. This thesis compares this architecture with the two other architectures. One of them is FCN-DK6 with the kernel size of 3 × 3, and another one is FCN-DK12, which contain 12 convolutional layers with the kernel size of 3 × 3. Each convolutional layer is followed by batch normalization, and the dilated factor is gradually increased from 1 to 6. In addition, the learning rate applied here is 1×10^{-4} at first, and then changed to 1×10^{-5} to find the global minimum.

Table 4-2 The architecture of FCN-DK6 with the kernel size of 5 × 5

Name of block	Layer	Weight(W×W×D×K))	Stride	Pad	Dilated factor
FCN-DK1	Conv-1	5 ×5 ×8 ×16	1	2	1
	BN-1	---	1	---	---
	LReLU1	---	1	---	---
FCN-DK2	Conv-2	5 ×5 ×16 ×32	1	4	2
	BN-2	---	1	---	---
	LReLU2	---	1	---	---
FCN-DK3	Conv-3	5 ×5 ×32 ×32	1	6	3
	BN-3	---	1	---	---
	LReLU3	---	1	---	---
FCN-DK4	Conv-4	5 ×5 ×32 ×32	1	8	4
	BN-4	---	1	---	---
	LReLU4	---	1	---	---
FCN-DK5	Conv-5	5 ×5 ×32 ×32	1	10	5

	BN-5	---	1	---	---
	LReLU5	---	1	---	---
FCN-DK6	Conv-16	5 ×5 ×32 ×32	1	12	6
	BN-4	---	1	---	---
	LReLU4	---	1	---	---
Classification	conv	1×1 ×32 ×2	1	0	1
	Dropout	---	---	---	---
	Softmax	---	---	---	---

4.3 Assessment

In this thesis, we use three parameters to assess the performance of the result, which is accuracy, recall, and f-1 score.

Table 4-3 The matrix derived from the true class and predicted the class

	True class		
Predicted class		Class=Changed	Class=Unchanged
	Class=Changed	True positive	False positive
	Class=Unchanged	False negative	True negative

True positive (TP): there are correctly predicted positive values.

True negative (TN): there are correctly predicted negative values.

False positive (FP): the actual class is unchanged, but predicted class is changed.

False negative (FN): the actual class is changed, but predicted class is unchanged.

Accuracy: It is the ratio of correctly predicted results to the total results.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad \text{Equation 16}$$

Precision: It is the ratio of correctly predicted changed results to the total predicted changed results.

$$Precision = \frac{TP}{TP + FP} \quad \text{Equation 17}$$

Recall: It is the ratio of correctly predicted changed results to the all changed results of the reference.

$$Recall = \frac{TP}{TP + FN}$$ Equation 18

F1 score: It is the weighted average of Precision and Recall.

$$F1\ Score = \frac{2 \times Recall \times Precision}{Recall + Precision}$$ Equation 19

4.4 Software

ENVI Classic 5.5 was applied to unify the resolution of images.

ArcMap 10.1.5 was used to shape the images into the same size.

Annotation was generated from the ENVI Classic 5.5, and the final binary result was extracted using Matlab 2018a

MatConvNet was used to train the FCN framework. The networks were trained using NVIDIA's CUDA GPU.

5 Result and analysis

All the parameters and results of comparison experiments are provided in this section. CD_{xy} represents the change detection result generated from epoch x and epoch y. For all the following figures, red and blue color represents the changed and unchanged areas respectively, and white color (existing in the annotation) is areas without annotation. In section 5.1, the change detection results based on RGB data are provided. Section 5.2 describes the results using DSM-based method. Then, section 5.3 combines the result derived from the RGB-based method and the DSM-based method. Furthermore, different proportions of training samples and different networks are presented in section 5.4.

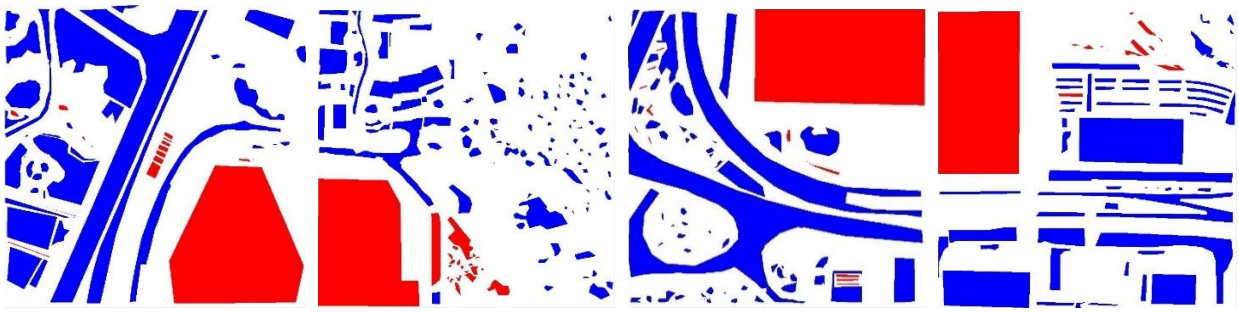
5.1 The result based on RGB data

5.1.1 The result of the CVA algorithm

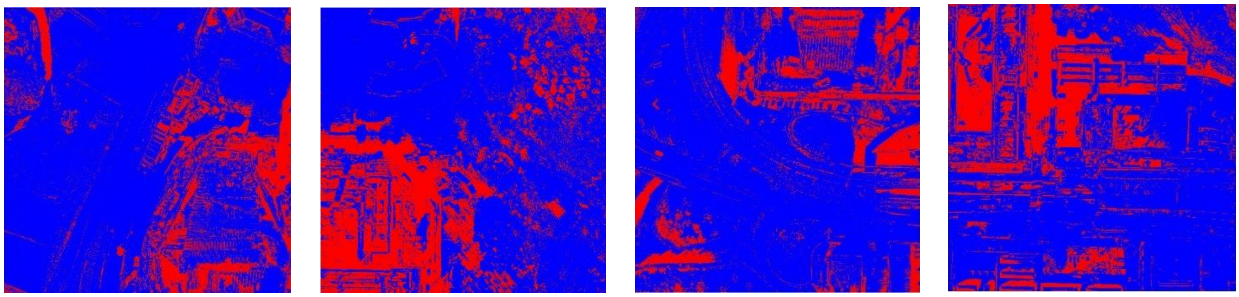
Figure 5-1 shows the change detection result from epoch 1 and epoch 3 using the CVA algorithm, and the other two combinations of change detection results are attached in section 9.2. In this study area, the main change is the constructing hospital, and it has been divided into four tiles to balance the number of changed and unchanged pixels in each tile. In the first tile, the change of building boundary was detected, but this algorithm failed to detect the changes inside the building. The road in the top left corner was wrongly classified as a changed area because of shadow. For the second tile, the changes of the building were successfully detected, while trees in the top right corner were wrongly classified as changed due to different seasons. A similar problem can also be found in the last two tiles, the shadow under the tree in the third tile and under the building in the last tile reduce the accuracy of the CVA accuracy. Furthermore, due to the lack of neighborhood information, many spikes and noise exist in the result of this algorithm.

(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4)



(c_1) (c_2) (c_3) (c_4)



(d_1) (d_2) (d_3) (d_4)

Figure 5-1 The CD13 of CVA. (a1), (a2), (a3) and (a4) is the first, second, third and fourth tile of the first epoch of RGB data; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the third epoch of RGB data; (c1), (c2), (c3) and (c4) is the first, second, third, fourth tile of ground truth respectively; (d1), (d2), (d3) and (d_4) is the first, second, third and fourth tile of the change detection result based on CVA algorithm respectively

Table 5-1 provides the assessment of these three results. From this table, we can know that the CVA algorithm can achieve 76.5%, and 54% on overall accuracy and precision respectively in this study areas. The highest performance of result obtained when detecting changes between the second and the third images, which reaches 79.7% on the accuracy and 67.4% on the F1 score.

Table 5-1 The result of the CVA algorithm

Image	Accuracy	Precision	Recall	F1 Score
CD_{12}	0.784	0.508	0.826	0.629
CD_{23}	0.797	0.647	0.711	0.674
CD_{13}	0.718	0.495	0.735	0.592
Overall	0.765	0.540	0.754	0.623

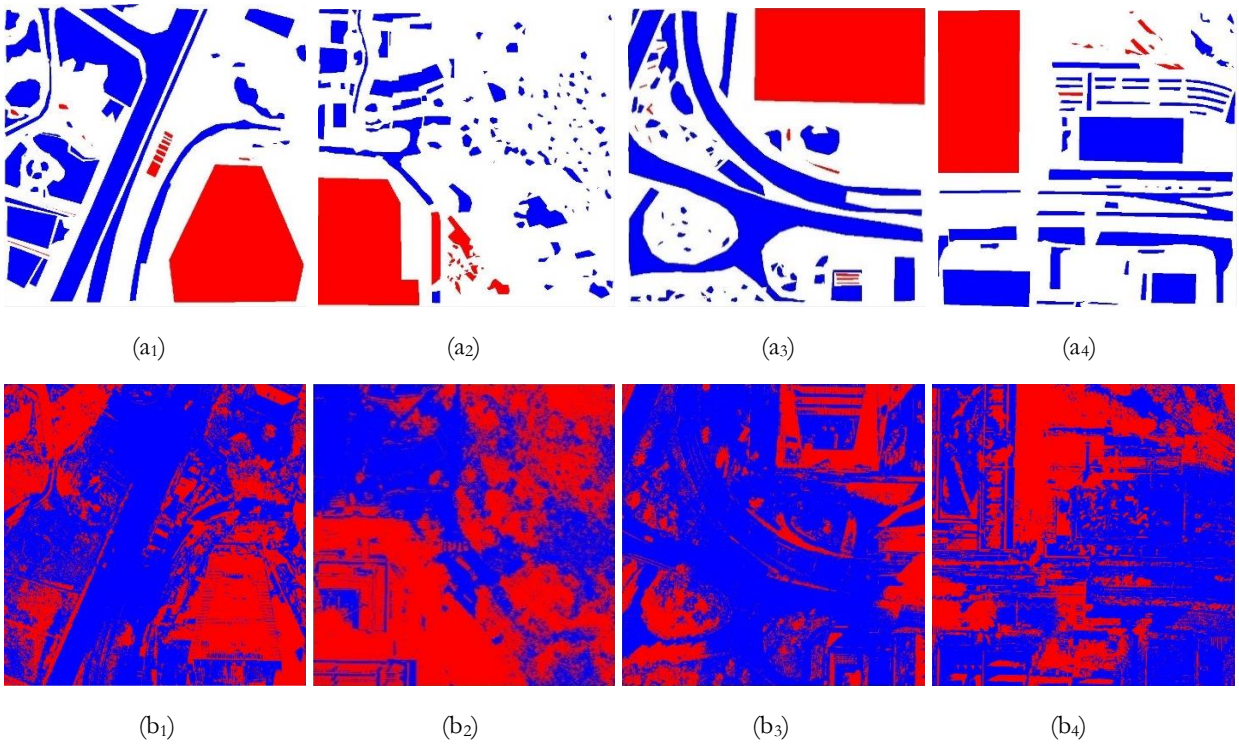
Table 5-2 shows the confusion matrix of all combinations of three epochs. Most of the unchanged areas are successfully detected using CVA, but type II error has a big problem, which leads to some of the changed pixels detecting as unchanged.

Table 5-2 The overall confusion matrix of three image pairs together using the CVA algorithm

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.200	0.065
Unchanged in the result	0.170	0.565

5.1.2 The result of the SAM algorithm

Comparing with the CVA, the result of the SAM algorithm has more changed pixels. Most of the building has been successfully detected using this algorithm, but at the same time, more unchanged areas also been falsely categorized. Problems caused by shadow and different seasons are exacerbated. Although more changed areas have been detected, many unchanged areas are wrongly detected as well. Figure 5-2 shows the change detection result between epoch 1 and epoch 3 using the SAM algorithm, results for the other combinations are provided in section 9.2.



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4)

Figure 5-2 The CD13 of SAM. (a1), (a2), (a3) and (a4) is the first, second, third, fourth tile of ground truth respectively; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the change detection result based on SAM algorithm respectively

The difference of accuracy for the combination of three epochs is not much which various between 68.6% to 76.3%, and the overall accuracy 73.1%. Similarly, the recall and F1 score do not have a big difference as well, and the overall result achieves 65.5% in the recall and 61.2% in the F1 score. However, the precision obtained from difference images shows a different pattern, which varies from 43.8% in CD_{12} to 75.4% in CD_{23}. Compared with the CVA method, the precision of the SAM method is better, but the effect of SAM on accuracy, recall, and the F1 score is not as good as CVA.

Table 5-3 The result of the SAM algorithm

Image	Accuracy	Precision	Recall	F1 Score
CD_{12}	0.748	0.438	0.765	0.557
CD_{23}	0.763	0.754	0.614	0.677
CD_{13}	0.686	0.572	0.632	0.600
Overall	0.731	0.574	0.655	0.612

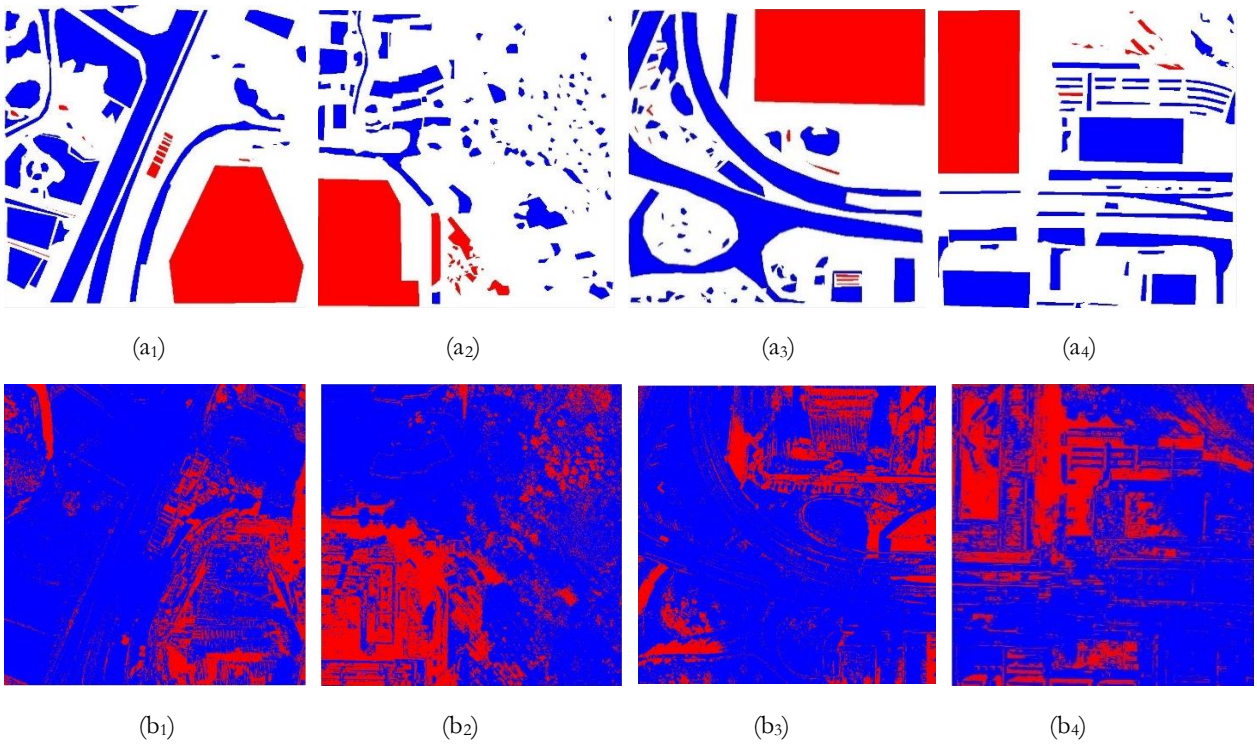
From Table 5-4, we found that most of the unchanged areas have been correctly classified, but the type I error and type II error are large in this situation. Comparing with the CVA algorithm, the type I error in the SAM algorithm is almost doubled.

Table 5-4 The overall confusion matrix of three image pairs together using the CVA algorithm

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.212	0.112
Unchanged in the result	0.157	0.519

5.1.3 The result of the CVA&SAM algorithm

Figure 5-3 shows the result of the CVA&SAM algorithm, and the problems of this algorithm are similar to the CVA algorithm. Results for the other combinations are provided in section 9.2.



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4)

Figure 5-3 The CD13 of CVA&SAM. (a1), (a2), (a3) and (a4) is the first, second, third, fourth tile of ground truth respectively; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the change detection result based on CVA&SAM algorithm respectively

The overall result of the CVA&SAM method is similar to the result derived from the CVA method alone. The result obtained from the second and the third tile achieve the best result except for the recall, and the best recall is 82.2% from the first two epochs. The overall accuracy and F1 score using this method is 76.3% and 62.1% respectively. The pattern of confusion matrix for the CVA&SAM method (Table 5-6) is similar to the other RGB-based methods.

Table 5-5 The result of the CVA&SAM algorithm

Image	Accuracy	Precision	Recall	F1 Score
CD_{12}	0.779	0.491	0.828	0.616
CD_{23}	0.796	0.628	0.718	0.670

CD_{13}	0.717	0.488	0.736	0.587
Overall	0.763	0.527	0.757	0.621

Table 5-6 The overall confusion matrix of three image pairs together using the CVA&SAM algorithm

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.195	0.062
Unchanged in the result	0.175	0.568

5.1.4 Comparison of RGB-based algorithms

Figure 5-4 shows the differences between these three algorithms on average accuracy, recall, and F-1 score. From this bar chart, the performance of result derived from the SAM algorithm is not as good as the result obtained from the other two algorithms. Although the strength of the CVA algorithm is not obvious, this algorithm is still the best in our study areas. In order to obtain a better result, we decide to adopt the CVA method alone to detect the changed areas as they fit our study areas more.

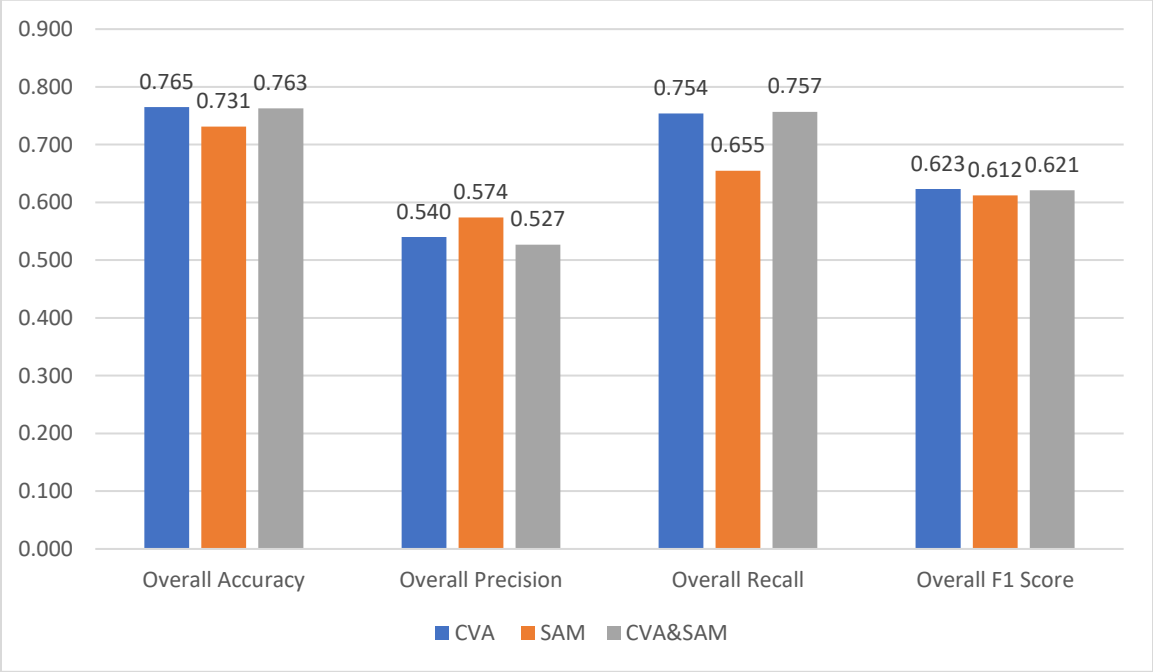


Figure 5-4 Comparison of three RGB-based algorithms

5.2 The result based on DSM data

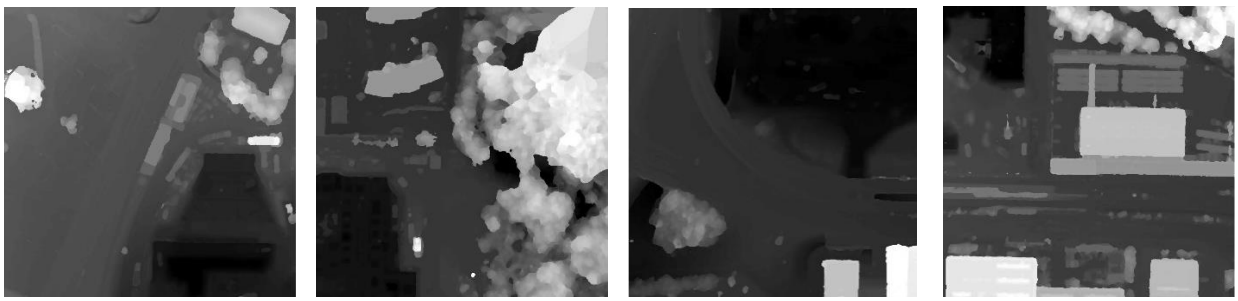
In order to find an appropriate threshold value and size of the structuring element, we did several experiments to compare the effects of different parameters. Before testing the effects of different parameters, the careful choice of parameters can increase working efficiency. For the selection of threshold value, we want to remove small changes like growing vegetation, moving people and vehicles, and try to keep the real changes. Therefore,

the threshold equals 1 m, 2 m, and 3 m are adopted. For the selection of the size of the structuring element, we want to neglect noise and moving objects. According to the resolution of our data, parameters equal to 10 and 20 pixels are adopted, which equivalents to 1.4 m and 2.8 m in the field. Table 5-7 shows the overall result of the testing. From this table, we can find that the difference in accuracy and precision is not significant, varying from 94.3% to 98.5% for the former one and from 89.3% to 93.1% for the other. The recall and F1 score achieve 90.4% at least, and the overall result is 98.3% and 93.6% for recall and F1 score. If we focus on the accuracy, we can find that leaving the size of the structuring element unchanged, the accuracy will get better as the threshold value grows. On the other hand, if we leave the threshold value unchanged, a larger size of structuring element can achieve a better accuracy, except when the threshold value is 3 meters. According to different evaluation indicators, the best performing parameter combinations are different. The threshold value and the size of the structuring element equaling to 3 meters and 10 pixels show the best result in accuracy and recall, and also the second highest in the recall. Therefore, we assume that these two parameters may suit our areas more, and the combination of these two parameters is used here.

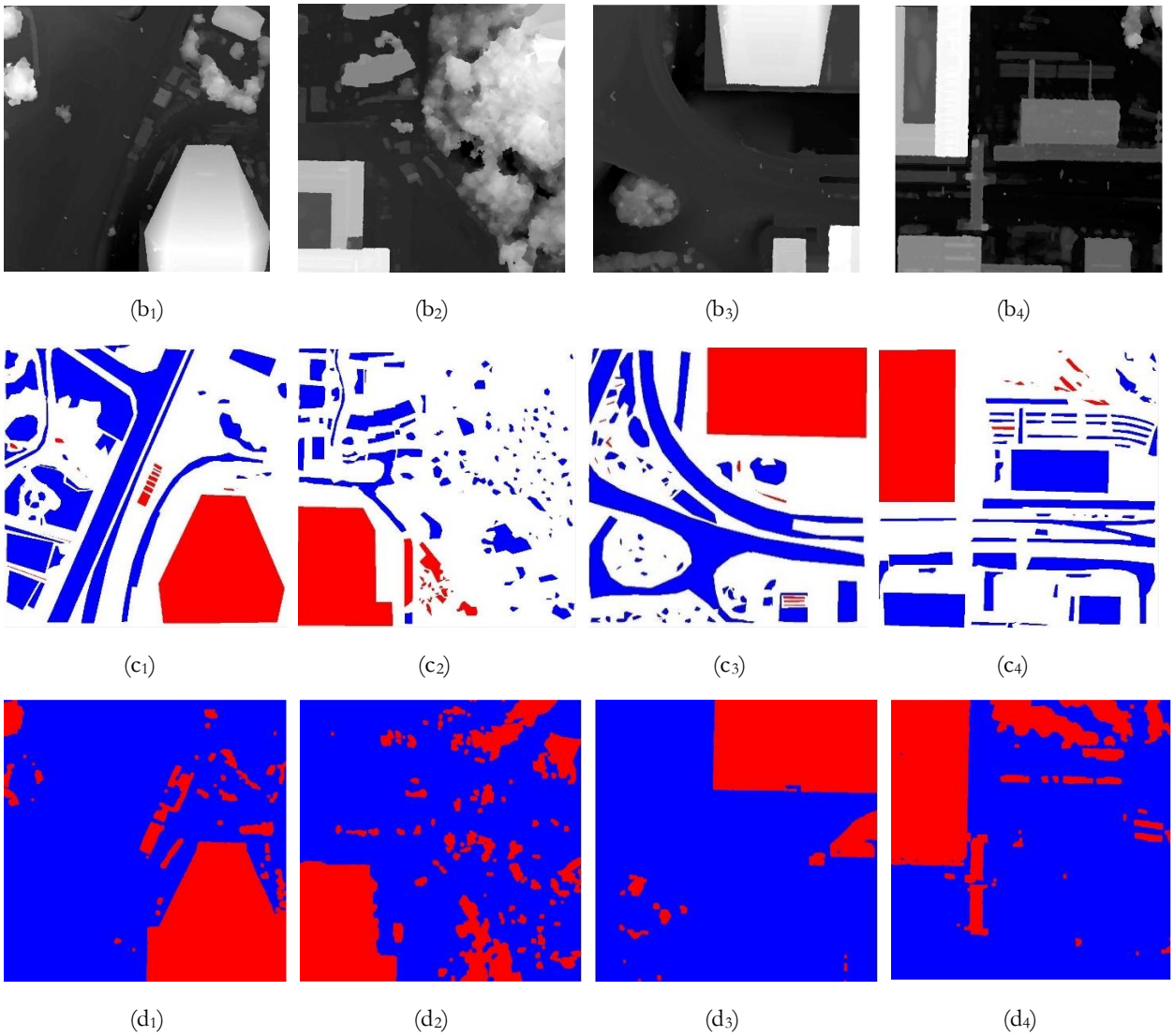
Table 5-7 Comparing the overall result of the DSM-based method with different parameter

Threshold value (meter)	Size of the structuring element (pixel)	Accuracy	Precision	Recall	F1 Score
1	10	0.943	0.932	0.904	0.918
1	20	0.950	0.904	0.958	0.930
2	10	0.951	0.894	0.966	0.929
2	20	0.953	0.898	0.973	0.934
3	10	0.955	0.898	0.979	0.937
3	20	0.955	0.893	0.983	0.936

Figure 5-5 shows the result using the DSM-based method between epoch 1 and epoch 3. From the result, we can find that the building has been clearly detected in all four tiles. In addition, comparing with the result from the RGB-based method, the result generated by the DSM-based method is no noise, and all the changed areas are cluster together. However, we can still find the error in the second tile, which tile has lots of false alarm due to the growth of the trees. The same problem can also be found in the bottom left corner of the third tile where trees grow above the road. The result for the other combinations has been presented in section 9.3.



(a_1) (a_2) (a_3) (a_4)



(b_1) (b_2) (b_3) (b_4)

(c_1) (c_2) (c_3) (c_4)

(d_1) (d_2) (d_3) (d_4)

Figure 5-5 The CD13 based on DSM data. (a1), (a2), (a3) and (a4) is the first, second, third and fourth tile of the first epoch of DSM data; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the third epoch of DSM data; (c1), (c2), (c3) and (c4) is the first, second, third, fourth tile of ground truth respectively; (d1), (d2), (d3) and (d_4) is the first, second, third and fourth tile of the change detection result based on DSM-based algorithm respectively

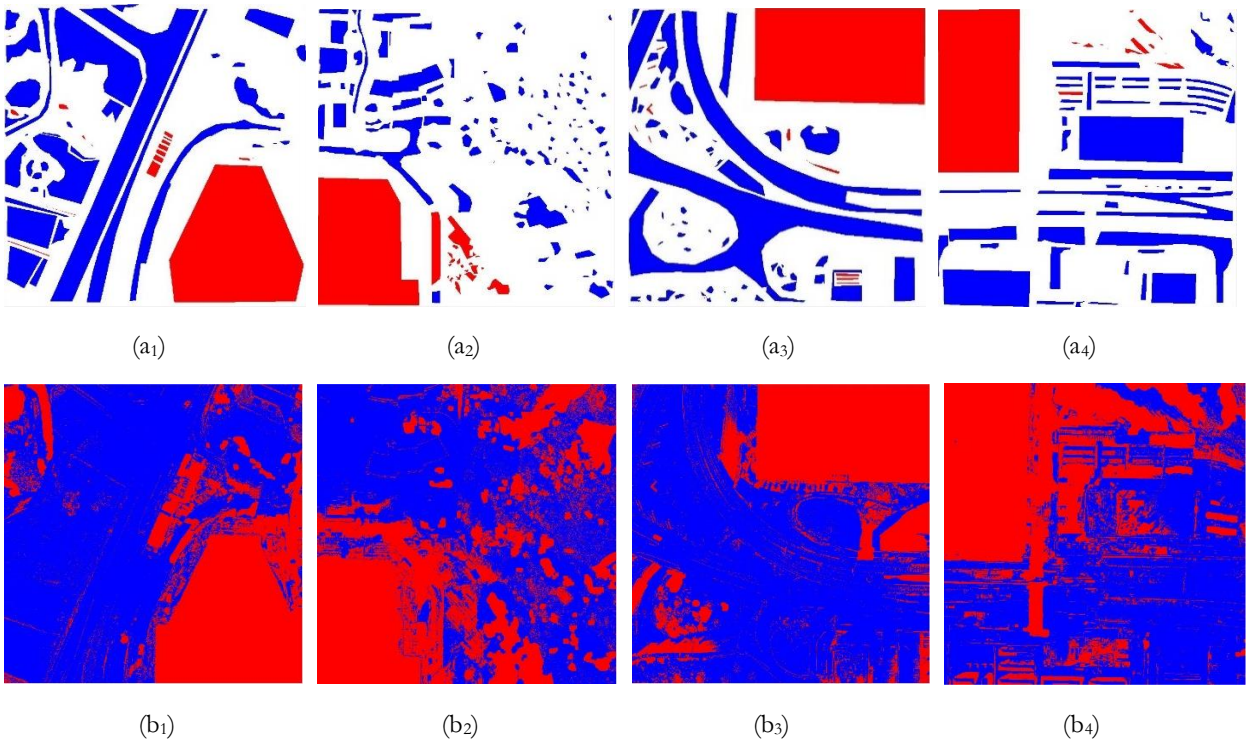
Table 5-8 shows the confusion matrix of CD13 using the DSM-based algorithm. Most of the pixels are correctly classified, and the type I error is just 0.9%. Due to the mainly changed areas in this study area is the under-construction areas, the type II error is small as well, which is only 3.8%.

Table 5-8 The overall confusion matrix of three image pairs together using the DSM-based algorithm

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.334	0.009
Unchanged in the result	0.038	0.621

5.3 Combining result from RGB-based and DSM-based method

Figure 5-6 shows the result of the CVA&DSM method, which combines the CVA algorithm and the DSM-based method. Compared with using the CVA method alone, this combining method significantly improves the effect of detecting buildings. However, this method also inherits some problems, such as false alarms caused by shadows and different seasons. Moreover, noise is still scattered in the image and affects the final result. Other results using the combination of CVA and DSM-based method are provided in section 9.2.



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4)

Figure 5-6 The CD13 of the CVA&DSM method. (a1), (a2), (a3) and (a4) is the first, second, third, fourth tile of ground truth respectively; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the CVA&DSM result respectively

From Table 5-9 we can find the overall accuracy is 91.9%, which is much higher than using the CVA method alone (76.5%) but lower than using a DSM-based algorithm alone (95.5%). The F1 score also shows that this combination makes an improvement from the CVA method but not as good as using the DSM-based method alone.

Table 5-9 The result of the CVA&DSM algorithm

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.944	0.970	0.886	0.926
CD23	0.893	0.947	0.777	0.853
CD13	0.916	0.985	0.839	0.906
Overall	0.919	0.970	0.837	0.898

Compared with using the CVA method alone, CVA&DSM improves the percentage of true positive significantly from 20% to 35.8%. Compared with the DSM method, true positive also increase by 2.4% and type II error decreases by 2.7%. However, this CVA&DSM method also has a bad aspect in the true negative, reducing from 62.1% in the DSM-based method to 56.1% and type I error increases from 0.9% to 7.0% correspondingly.

Table 5-10 The confusion matrix of three image pairs together using the unsupervised algorithm

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.358	0.070
Unchanged in the result	0.011	0.561

We observed that the results of the CVA&DSM method are actually better than the RGB-based method, but worst than the DSM-based method. The good performance of the DSM-based method is because most of the changes in this study area can be detected by DSM-based. If there is no or very little elevation change in other study areas, DSM-based method alone may not show good performance as this study area. In the end, we decided to use the CVA&DSM method as the result of our unsupervised part for the following experiments.

5.4 The result of the FCN architecture

5.4.1 The result of different architectures

In this sub-section, we compare the three difference architectures: (i) is the FCN-DK6 with the kernel size of 5 × 5 (Persello & Stein, 2017); (ii) is the FCN-DK6 with the kernel size of 3 × 3 (Table 5-11); (iii) is the FCN-DK12 with the kernel size of 3 × 3 (Table 5-12). Experiments are performed by the normalized images. We used 80% of the labels obtained from the unsupervised result as training samples, and testing result on the same images.

Table 5-11 The architecture of FCN-DK6 with the kernel size of 3×3

Name of block	Layer	Weight(W×W×D×K))	Stride	Pad	Dilated factor
FCN-DK1	Conv-1	3 ×3 ×8 ×16	1	1	1
	BN-1	---	1	---	---
	LReLU1	---	1	---	---
FCN-DK2	Conv-2	3 ×3 ×16 ×32	1	2	2
	BN-2	---	1	---	---
	LReLU2	---	1	---	---
FCN-DK3	Conv-3	3 ×3 ×32 ×32	1	3	3
	BN-3	---	1	---	---
	LReLU3	---	1	---	---
FCN-DK4	Conv-4	3 ×3 ×32 ×32	1	4	4
	BN-4	---	1	---	---
	LReLU4	---	1	---	---
FCN-DK5	Conv-5	3 ×3 ×32 ×32	1	5	5
	BN-5	---	1	---	---
	LReLU5	---	1	---	---
FCN-DK6	Conv-6	3 ×3 ×32 ×32	1	6	6
	BN-4	---	1	---	---
	LReLU4	---	1	---	---
Classification	conv	1×1 ×32 ×2	1	0	1
	Dropout	---	---	---	---
	Softmax	---	---	---	---

Table 5-12 The architecture of FCN-DK12 with the kernel size of 3×3

Name of block	Layer	Weight(W×W×D×K))	Stride	Pad	Dilated factor
FCN-DK1	Conv-1	3 ×3 ×8 ×16	1	1	1

	BN-1	---	1	---	---
	LReLU1	---	1	---	---
	Conv-2	3 ×3 ×16×32	1	1	1
	BN-2	---	1	---	---
	LReLU2	---	1	---	---
FCN-DK2	Conv-3	3 ×3 ×32 ×32	1	2	2
	BN-3	---	1	---	---
	LReLU3	---	1	---	---
	Conv-4	3 ×3 ×32 ×32	1	2	2
	BN-4	---	1	---	---
	LReLU4	---	1	---	---
FCN-DK3	Conv-5	3 ×3 ×32 ×32	1	3	3
	BN-5	---	1	---	---
	LReLU5	---	1	---	---
	Conv-6	3 ×3 ×32 ×32	1	3	3
	BN-6	---	1	---	---
	LReLU6	---	1	---	---
FCN-DK4	Conv-7	3 ×3 ×32 ×32	1	4	4
	BN-7	---	1	---	---
	LReLU7	---	1	---	---
	Conv-8	3 ×3 ×32 ×32	1	4	4
	BN-8	---	1	---	---
	LReLU8	---	1	---	---
FCN-DK5	Conv-9	3 ×3 ×32 ×32	1	5	5
	BN-9	---	1	---	---
	LReLU9	---	1	---	---
	Conv-10	3 ×3 ×32 ×32	1	5	5
	BN-10	---	1	---	---
	LReLU10	---	1	---	---
FCN-DK6	Conv-11	3 ×3 ×32 ×32	1	6	6
	BN-11	---	1	---	---
	LReLU11	---	1	---	---
	Conv-12	3 ×3 ×32 ×32	1	6	6
	BN-12	---	1	---	---
	LReLU12	---	1	---	---
Classification	conv	1×1 ×32 ×2	1	0	1
	Dropout	---	---	---	---
	Softmax	---	---	---	---

The normalized image refers to subtracting the average of the pixel values of each band by the mean of their pixel value so that all pixel values are centered at zero.

Table 5-13 shows the result of them. We can find the difference is small between different architectures. FCN-DK12 achieves the best recall, which is about 89.2%, and the FCN-DK6 with 5×5 kernel size is the best in the precision (96.0%). The highest value for the accuracy and the F1 score was obtained by FCN-DK6 with 3×3 kernel size which is 94.2% and 92.4% respectively. At last, the architecture of FCN-DK6 with 3×3 kernel size was selected as it obtained the best accuracy amongst them.

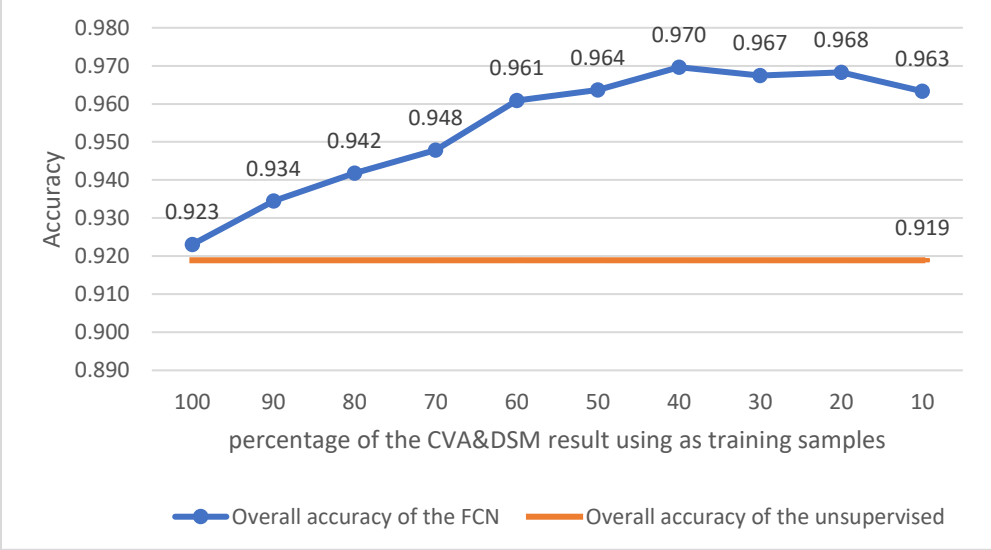
Table 5-13 Comparing the overall result of four situations

Type of architecture	Kernel size	Accuracy	Precision	Recall	F1 Score
Unsupervised result	---	0.919	0.970	0.837	0.898
FCN-DK6	5×5	0.940	0.960	0.891	0.924
FCN-DK6	3×3	0.942	0.960	0.891	0.924
FCN-DK12	3×3	0.942	0.959	0.892	0.924

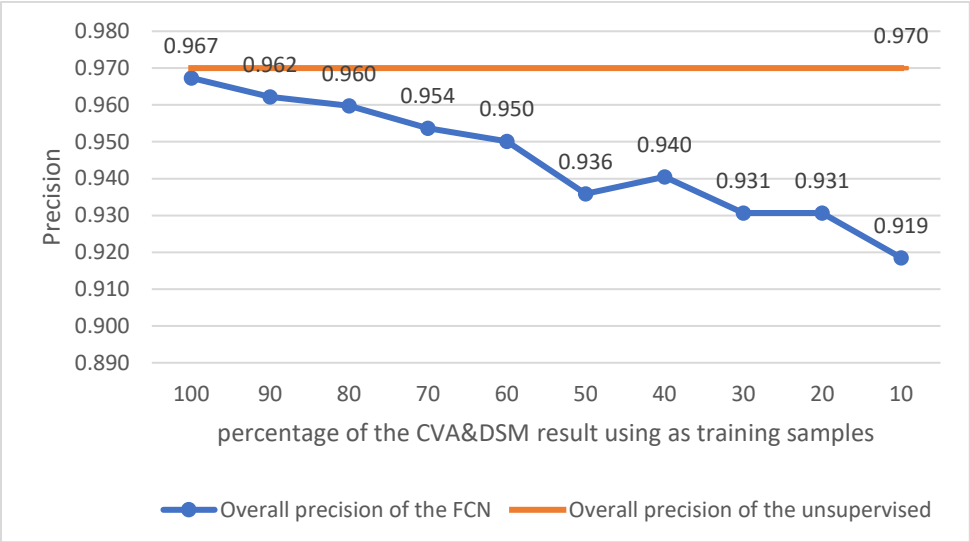
5.4.2 The result of the different proportion of training samples

In the previous sub-section, we compared four experiments under the assumption that 80% of unsupervised training results were used as training samples. However, we still want to know the different effects of using different proportions of the unsupervised classification results as training samples. In this sub-section, we will extract training samples from unsupervised results, taken from 100% at 10% intervals, and observe its trend. The training samples are selected based on the reliability of the unsupervised result. For example, if 80% of the labels obtained from the CVA&DSM method will be used as training samples, all the labels will be ordered according to the reliability firstly, and then labels belong to 80% of highest reliabilities will be used to train the FCN architecture.

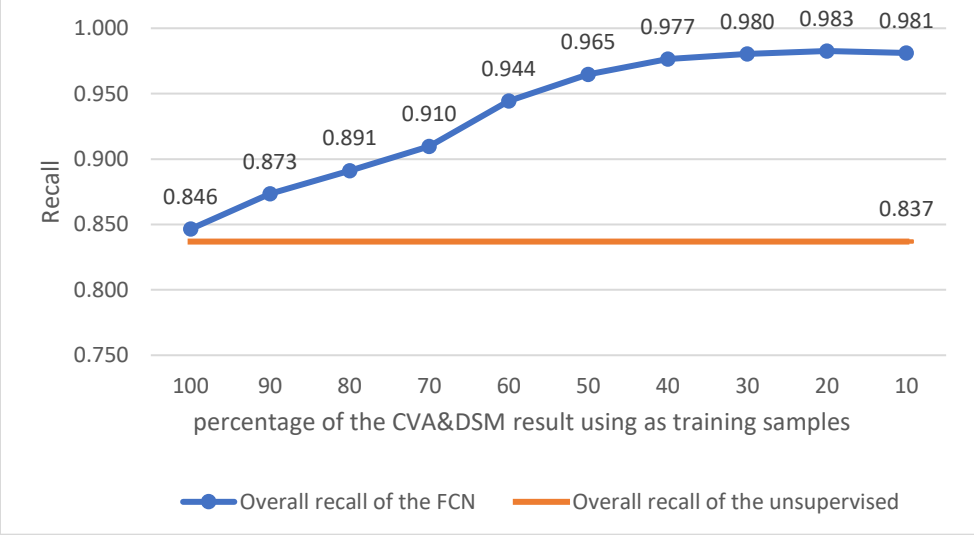
The accuracy, precision, recall and F1 score of the FCN results using a different proportion of unsupervised results are presented in Figure 5-7. In the following figures, the blue lines represent for the result of the FCN architecture, and the orange lines represent the unsupervised results. Detailed results of the FCN in different percentage of training samples have been provided in section 9.4.



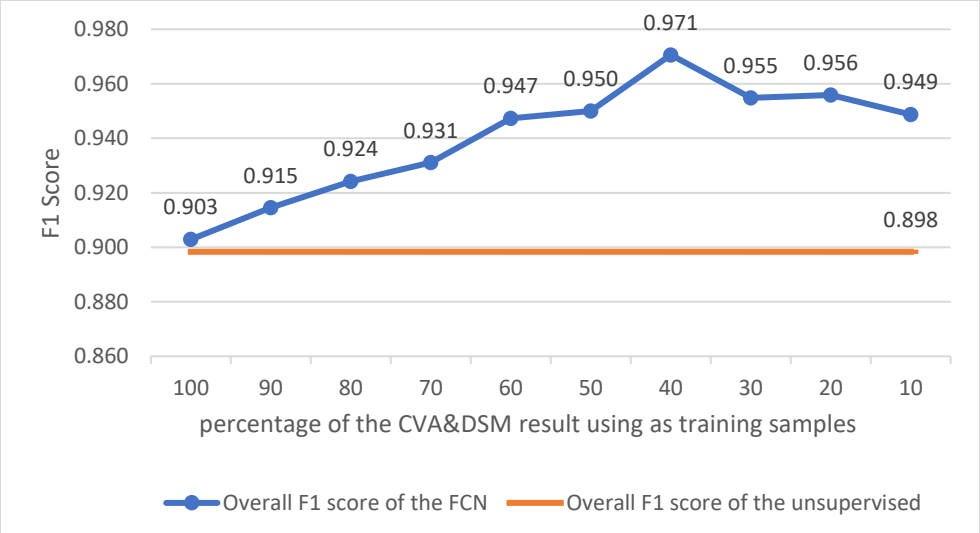
(a)


Precision
0.980
0.970
0.960
0.950
0.940
0.930
0.920
0.910
0.900
0.890
0.967
0.962
0.960
0.954
0.950
0.936
0.940
0.931
0.931
0.919
0.970
100 90 80 70 60 50 40 30 20 10
percentage of the CVA&DSM result using as training samples
Overall precision of the FCN
Overall precision of the unsupervised

(b)


Recall
1.000
0.950
0.900
0.850
0.800
0.750
0.846
0.873
0.891
0.910
0.944
0.965
0.977
0.980
0.983
0.981
0.837
100 90 80 70 60 50 40 30 20 10
percentage of the CVA&DSM result using as training samples
Overall recall of the FCN
Overall recall of the unsupervised

(c)

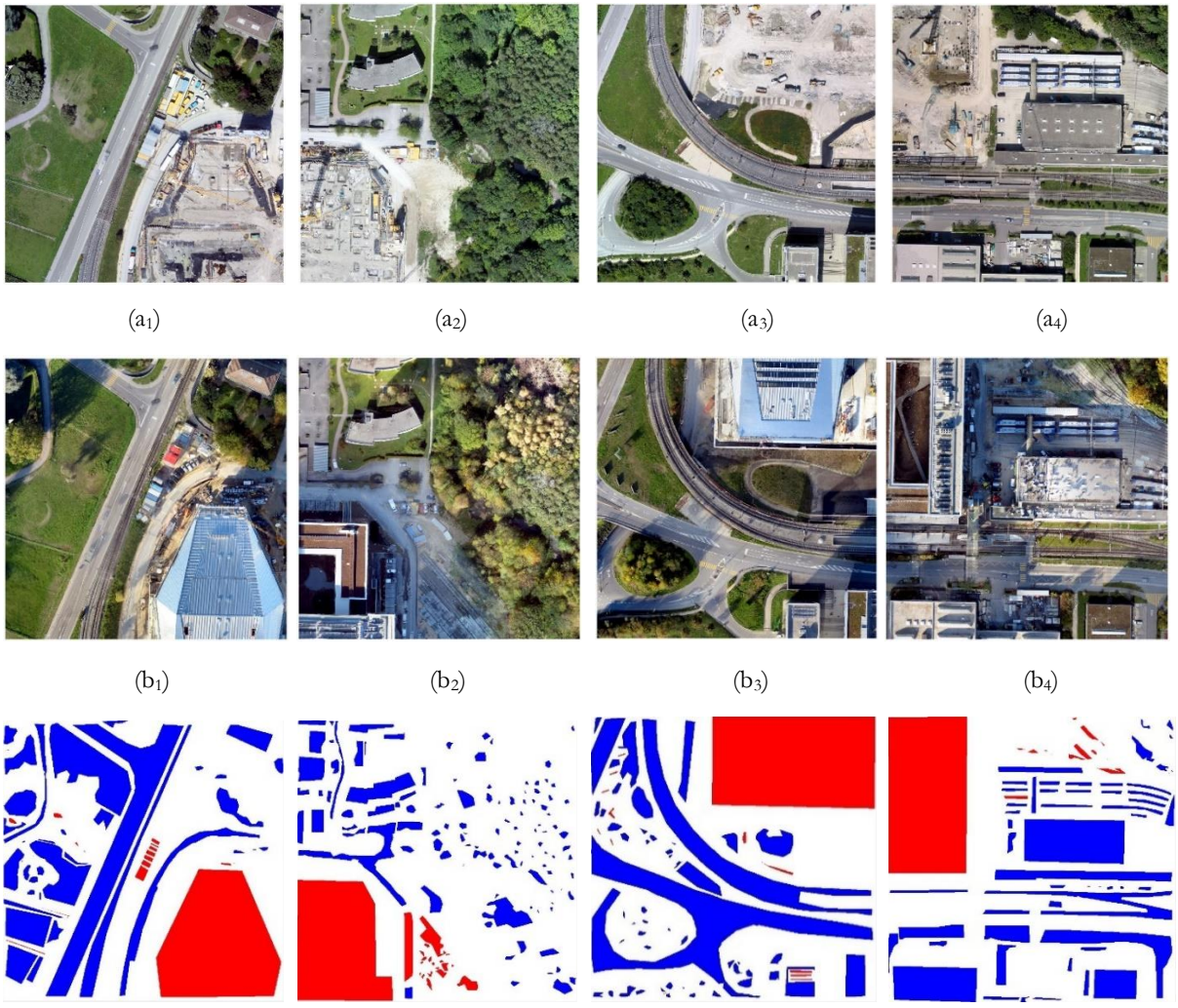

F1 Score
0.980
0.960
0.940
0.920
0.900
0.880
0.860
0.903
0.915
0.924
0.931
0.947
0.950
0.971
0.955
0.956
0.949
0.898
100 90 80 70 60 50 40 30 20 10
percentage of the CVA&DSM result using as training samples
Overall F1 score of the FCN
Overall F1 score of the unsupervised

(d)

Figure 5-7 The results of the FCN using a different proportion of the training sample. (a) compares the accuracy of the unsupervised results with the FCN using a different proportion of training sample; (b) compares the precision of the unsupervised results with the FCN using a different proportion of training sample; (c) compares the recall of the unsupervised results with the FCN using a different proportion of training sample; (d) compares the F-1 score of the unsupervised results with the FCN using a different proportion of training sample

From Figure 5-7, we can see that the trend of precision is different from the trend of the other indicators, which is decreasing as using a fewer number of training samples. Conversely, the performance assessed by the recall is increasing with fewer training samples, and get 98.1% finally. The accuracy and F1 score peak when using 40% of labels as training samples, then drops slightly. The highest value for these two indicators is about 97%.

Our final results are derived from using 40% of unsupervised results as training samples. The final result between epoch 1 and epoch 3 for the FCN architecture and the unsupervised have been provided in Figure 5-8. For the first tile, we can see that the false alarm on the road boundary is disappeared and the errors caused by the shadow also reduced significantly. Although in the second tile, the problem amongst trees caused by the different seasons still exists, it is reduced by nearly half. Similarly, in the third and fourth epochs, we can also find that the errors caused by the shadows of trees or buildings are largely eliminated. Furthermore, most of the noise was removed in the results from the FCN, and the output images become smoother. Comparing the result of these two methods between the first two and last two image pairs are presented in section 9.5.



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4)

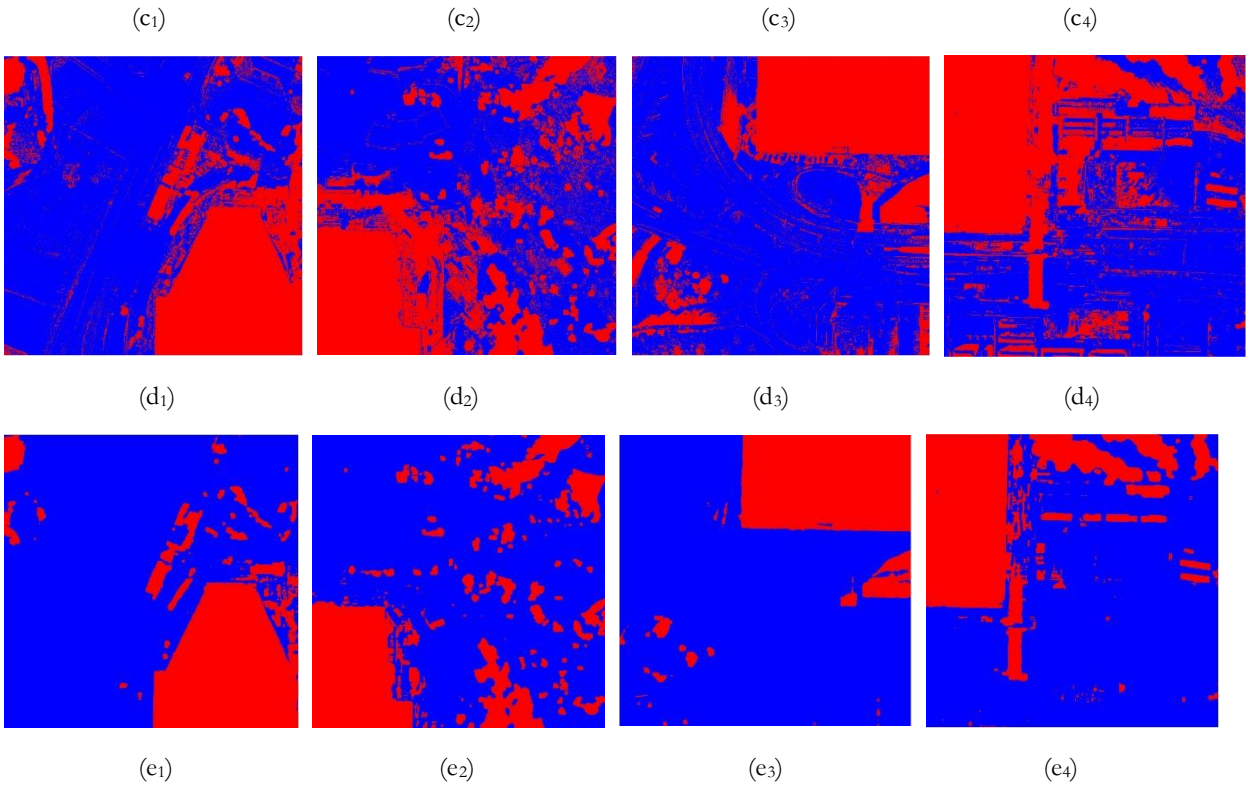


Figure 5-8 Comparing the CD_{13} in the FCN and unsupervised result. (a_1), (a_2), (a_3) and (a_4) is the first, second, third and fourth tile of the first epoch of RGB data; (b_1), (b_2), (b_3) and (b_4) is the first, second, third and fourth tile of the second epoch of RGB data; (c_1), (c_2), (c_3) and (c_4) is the first, second, third, fourth tile of ground truth respectively; (d_1), (d_2), (d_3) and (d_4) is the first, second, third and fourth tile of the CVA&DSM respectively; (e1), (e2), (e3) and (e4) is the first, second, third and fourth tile of the unsupervised FCN result respectively

5.4.3 Comparison of our unsupervised FCN result with supervised FCN result

In the previous, we adopted a system that allows the FCN architecture to train the network without ground truth point. In this section, we compare our system with common FCN architecture, which needs the manually labeled training samples. In order to distinguish, we name our FCN architecture as the 'unsupervised FCN and name the traditional FCN architecture as 'supervised FCN'.

The tile that uses to train the supervised FCN is not the same as the tile for predicting in real life. In order to use the supervised FCN as the way of using it in reality, we choose to get the results of the supervised method four times, and each time we use three tiles for training and one tile for testing (Figure 5-9). Then, the result for three combinations of image pairs can be obtained.

Figure 5-9 The distribution of training samples and testing samples in the supervised FCN

Here, the supervised FCN is used to compare two phases of our approach (Table 5-14). The performance of results derived from CVA&DSM method is slightly lower than the result from the supervised FCN architecture on each image pair, but our CVA&DSM method still achieved 91.9%, 83.7%and 89.8% of the average accuracy, recall, and F1 score. The CVA&DSM method achieves the highest overall precision which is 97%. The highest value for overall accuracy, recall, and F1 score is achieved by the unsupervised FCN, which is 97%, 97.7%, and 95.8% respectively. From Table 5-14, we can see that the overall result of the unsupervised FCN shows a better performance than the supervised FCN architecture.

Table 5-14 The result of the unsupervised method, unsupervised FCN and supervised FCN

Image pairs	Method	Accuracy	Precision	Recall	F1 Score
CD_{12}	Supervised FCN	0.962	0.935	0.976	0.955
	CVA&DSM method	0.944	0.970	0.886	0.926
	Unsupervised FCN	0.970	0.948	0.969	0.958
CD_{23}	Supervised FCN	0.905	0.958	0.796	0.869
	CVA&DSM method	0.893	0.947	0.777	0.853
	Unsupervised FCN	0.961	0.902	0.979	0.939
CD_{13}	Supervised FCN	0.923	0.939	0.882	0.910
	CVA&DSM method	0.916	0.985	0.839	0.906
	Unsupervised FCN	0.976	0.960	0.982	0.971
Overall	Supervised FCN	0.930	0.942	0.886	0.913
	CVA&DSM method	0.919	0.970	0.837	0.898
	Unsupervised FCN	0.970	0.940	0.977	0.958

6 Discussion

This thesis proposes a novel change detection system, which considers part of the labels from the unsupervised method as training samples and uses them to train an FCN according to different reliability. The unsupervised method used in this thesis is CVA&DSM, which is a pixel-based algorithm. By using the FCN in the last procedure, this system is able to consider contextual information and regenerate the change detection map. Our final result shows that after using this FCN architecture, false positive (ground truth is unchanged but predicting as changed) has been significantly reduced, and more pixels have been correctly classified. This improvement is because the influence of problems caused by shadow, different seasons and illumination are reduced. Simultaneously, the noises in the pixel-based algorithms have been removed after using the FCN architecture. This section discusses the strengths and weaknesses of the methods used in this thesis. Section 6.1 analysis the result of unsupervised methods, and section 6.2 analysis the relationship between unsupervised method, unsupervised FCN and supervised FCN.

6.1 The unsupervised method

6.1.1 RGB-based method

Comparing to the CVA method, the SAM method labels more results in the changed category in this study area, which enables the SAM algorithm to detected more true changes inside the building. However, the SAM method also suffer more problems caused by the shadow and different seasons. The difference between CVA and CVA&SAM results is hard to observe. By comparing the assessment indicators, CVA has better performance on the overall accuracy and F1 score, so the CVA method was chosen as the final method of combining with the DSM-based method.

From the result, we can observe that all of these unsupervised RGB-based methods have the following problems: (i) shadows from buildings, trees, and street lights will lead to false alarm; (ii) changes of seasons will change the color of vegetation, which will dramatically affect the final result; (iii) due to the lack of contextual information, many noises existing in the result. The third problem can be solved by adding neighborhood information, while the former two problems are common amongst the change detection algorithms based on the spectral information.

6.1.2 DSM-based method

When comparing the DSM methods with different parameters, we found that different thresholds and size of structuring elements did not have a big impact on the results. This may be because the small height variations are not common, and most of the height changes are large and gathering together in this study area. However, we can also found some false positive when detecting changes on the second tile, which may be caused by the growth of trees. Most of the changed areas have been correctly classified using this algorithm, especially for the under-constructing areas. In terms of noises of the result, the result obtained by the DSM-based method is relatively smoother due to the addition of the morphology opening algorithm.

6.1.3 Reliability

The method of obtaining reliability is established based on a different way of setting the threshold value. The threshold value of the RGB-based method is automatically generated after analyzing the classes in the whole image. In the difference obtained from RGB images, we define that the reliability of the maximum difference value is 1, and the reliability of the minimum difference value is 1 as well. However, when using DSM-based method to determine the reliability, if the reliability of the maximum difference is also assumed to be 1, same changed height from CD_{12} and CD_{13} may obtain different reliability. At last, we decided that when the changes obtained from the DSM data exceeds 2 times the threshold value, the reliability is 1. In this way, the same changes between different images can lead to the same reliability. On the other hand, the reliability curve has

the same rate of change in the part that exceeds the threshold and the part that is less than the threshold. The reliability of all unchanged regions comes from the RGB-based method because the changes of elevation can indicate the changes in land cover to a certain extent, but if there is no change in height, changes can also happen.

6.2 Unsupervised FCN

We first compare the impact of different network structures and kernel sizes on the results. It turns out that the difference between them is relatively small, and the possible reasons are listed below. (i) The training areas are the same as predicted areas so that the predicted results tends to the category of the training sample. If they can train and predict different data, they may obtain a bigger difference. (ii) In conducting the comparative trials, we first decided to use 80% of the unsupervised results for training, which made the maximum difference between the unsupervised results and the FCN results are 20%. If using less data for training and leave more space for the FCN architectures, different networks may have a larger difference. (iii) The learning task in this study area is relatively simple, and there is no need to learn complex textural patterns.

When observing the effects of different proportions of unsupervised results as training samples, we found that when all unsupervised results were used for training, the results obtained from the FCN were basically consistent with those used for training. With the higher reliability value being used as training samples, more and more pixels are correctly classified. The highest accuracy is achieved when 40% of the labels are used for training, exceeding the original unsupervised method by 5%. After this, the accuracy gradually flattens and begins to decline, which downward trend may be due to insufficient unsupervised results for training. Compared with the result derived from the CVA&DSM method, the noise of the entire image has been dramatically reduced by the use of the FCN architecture.

When comparing the unsupervised FCN with the supervised FCN, the training areas and the predicted areas for the supervised FCN is not the same. We select three tiles for training and one tile for predicting to imitate the way of supervised FCN in real life. After comparison, we found that unsupervised FCN has better performance in this research area. This because of the fact that the training samples used in the supervised FCN architecture are not sufficient for the network to fully understand the study area. As more ground truth is used for training, the accuracy of the result derived from the supervised FCN will gradually improve.

The unsupervised FCN network is not limited in using after CVA&DSM method. In theory, it can learn the results from all methods and predict them again. The upper limit of the unsupervised FCN result is using ground truth as a training sample and predicting the training areas. When we use all unsupervised results as training samples, we find that the predictions in the training areas are basically consistent with the training samples, which proves that the FCN network has a strong learning ability. Hence, when we use the ground truth as a training sample to predict the same area, the FCN method can restore the ground truth information to a great extent, and get outputs with little difference of ground truth. In theory, the upper limit is 100%. However, in the actual application process, if using all the results for training, the accuracy will not change, just like using 100% unsupervised results for training in this thesis. In this case, appropriately select of samples for training and discarding will have a significant impact on the FCN results, and all the methods that can calculate the 'reliability ' can be improved again using this unsupervised FCN architecture.

7 Conclusions and recommendations

7.1 Conclusions

In this thesis, we explore unsupervised changed detection algorithms based on the RGB and DSM data. The DSM data is used to detect 3-D changes in urban areas, and the RGB data is included as a supplement to detect changes in areas where there are no 3-D changes. Then, an unsupervised FCN system was proposed to learn the texture information from the image and further optimize the change detection performances. The experiment was performed on UAV images from three different times, and the DSM images were obtained by photogrammetry.

Four comparison trials are performed to get better experimental results in our study areas. The first one is to obtain a set of parameters for the DSM-based method. Then, a comparison between the three RGB-based methods is performed. Furthermore, the effect of unsupervised FCN architectures with different numbers of convolutional layers and different kernel size is compared to find a better architecture. At last, the result obtained from the supervised FCN architecture is compared with our experimental results to verify the effectiveness of our method.

The experimental results show that our method can effectively detect changes in urban areas. Compared with using spectral information for change detection alone, the combined approach reduces the effects of shadows and different seasons on the results to some extent. Compared with the DSM-based method alone, the combined method can be more effective in detecting areas without elevation changes. In addition, the unsupervised FCN framework can make an improvement based on unsupervised change detection results. This framework not only reduces the interference caused by shadows and different seasons on spectral information but also eliminates noise from the pixel-based algorithm.

Most importantly, our unsupervised FCN architecture can be applied to all the deep learning techniques. By doing this, we can effectively reduce the needs of manual interpretation, thereby improving the time efficiency and reducing the labor cost. At the same time, the problem of insufficient training samples in deep learning can be significantly alleviated as well. Moreover, by combining this framework with different unsupervised methods, the results of existing unsupervised methods are likely to be improved again. The unsupervised method here is not limited to the field of change detection. Algorithms in all fields (like image classification, boundary detection) can also be improved after combining this framework.

7.2 Recommendations

- The parameters applied in the DSM-based method should be selected according to the study area, and a more general set of parameters is likely to be obtained empirically after more experimental verification.
- The RGB-based methods and DSM-based method are performed separately, so following the same combining rule, the combinations between other DSM-based methods or RGB-based methods can also be tested.
- All deep learning techniques require a lot of ground truth information to train their architectures. This unsupervised FCN framework could also be applied to other deep learning architectures, allowing them to receive the non-manual annotation as training samples.
- The effect of the unsupervised FCN should be tested and analyzed after other algorithms,which can calculate the ‘reliability’.

- All the labels meet the requirement of the reliability are trained by the same weight in this thesis. However, we can also train the FCN architecture with different weights, and the weight can be obtained from the ‘reliability’ as well.

8 Reference

Allen, T. R., & Kupfer, J. A. (2000). Application of Spherical Statistics to Change Vector Analysis of Landsat Data: Southern Appalachian Spruce–Fir Forests. *Remote Sensing of Environment*, *74*(3), 482–493. https://doi.org/10.1016/S0034-4257(00)00140-1

Aloysius, N., & Geetha, M. (2017). A review on deep convolutional neural networks. In *2017 International Conference on Communication and Signal Processing (ICCSP)* (pp. 0588–0592). IEEE. https://doi.org/10.1109/ICCSP.2017.8286426

Baldi, P., Fabris, M., Marsella, M., & Monticelli, R. (2005). Monitoring the morphological evolution of the Sciara del Fuoco during the 2002–2003 Stromboli eruption using multi-temporal photogrammetry. *ISPRS Journal of Photogrammetry and Remote Sensing*, *59*(4), 199–211. https://doi.org/10.1016/J.ISPRSJPRS.2005.02.004

Bruzzone, L., & Bovolo, F. (2013). A Novel Framework for the Design of Change-Detection Systems for Very-High-Resolution Remote Sensing Images. *Proceedings of the IEEE*, *101*(3), 609–630. https://doi.org/10.1109/JPROC.2012.2197169

Chaabouni-Chouayakh, H., d'Angelo, P., Krauss, T., & Reinartz, P. (2011). Automatic urban area monitoring using digital surface models and shape features. In *2011 Joint Urban Remote Sensing Event* (pp. 85–88). IEEE. https://doi.org/10.1109/JURSE.2011.5764725

Chen, J., Chen, X., Cui, X., & Chen, J. (2011). Change Vector Analysis in Posterior Probability Space: A New Method for Land Cover Change Detection. *IEEE Geoscience and Remote Sensing Letters*, *8*(2), 317–321. https://doi.org/10.1109/LGRS.2010.2068537

Chen, J., Gong, P., He, C., Pu, R., & Shi, P. (2003). Land-Use/Land-Cover Change Detection Using Improved Change-Vector Analysis. *Photogrammetric Engineering & Remote Sensing*, *69*(4), 369–379. https://doi.org/10.14358/PERS.69.4.369

Du, P., Liu, S., Gamba, P., Tan, K., & Xia, J. (2012). Fusion of Difference Images for Change Detection Over Urban Areas. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *5*(4), 1076–1086. https://doi.org/10.1109/JSTARS.2012.2200879

Fan J, Xu W, Wu Y, & Gong Y. (2010). Human Tracking Using Convolutional Neural Networks. *IEEE Transactions on Neural Networks*, *21*(10), 1610–1623. https://doi.org/10.1109/TNN.2010.2066286

Guerin, C., Binet, R., & Pierrot-Deseilligny, M. (2014). Automatic Detection of Elevation Changes by Differential DSM Analysis: Application to Urban Areas. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(10), 4020–4037. https://doi.org/10.1109/JSTARS.2014.2300509

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507. https://doi.org/10.1126/science.1127647

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv*. https://doi.org/arXiv:1207.0580

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, *195*(1), 215–243. https://doi.org/10.1113/jphysiol.1968.sp008455

LIST OF REFERENCES

Ioannidis, C., Psaltis, C., & Potsiou, C. (2009). Towards a strategy for control of suburban informal buildings through automatic change detection. *Computers, Environment and Urban Systems*, *33*(1), 64–74. https://doi.org/10.1016/J.COMPENVURBSYS.2008.09.010

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Retrieved from http://arxiv.org/abs/1502.03167

Jaderberg, M., Vedaldi, A., & Zisserman, A. (2014). Deep features for text spotting. In *Computer Vision – ECCV 2014* (Vol. 8692 LNCS, pp. 512–528). Springer, Cham. https://doi.org/10.1007/978-3-319-10593-2_34

Jiang, Y., Wen, X., Xiang, D., Tan, D., Li, Z., Zhang, S., & Wan, Y. (2016). A change detection approach of high-resolution imagery combined the pre-classification with the post-classification comparison. *2016 5th International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2016*. https://doi.org/10.1109/Agro-Geoinformatics.2016.7577670

Johnson, R. D., & Kasischke, E. S. (1998). Change vector analysis: A technique for the multispectral monitoring of land cover and condition. *International Journal of Remote Sensing*, *19*(3), 411–426. https://doi.org/10.1080/014311698216062

Jung, F. (2004). Detecting building changes from multitemporal aerial stereopairs. *ISPRS Journal of Photogrammetry and Remote Sensing*, *58*(3–4), 187–201. https://doi.org/10.1016/J.ISPRSJPRS.2003.09.005

Kim, G., Ha, S., & Kwon, J. (2018). Adaptive Patch Based Convolutional Neural Network for Robust Dehazing. In *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 2845–2849). IEEE. https://doi.org/10.1109/ICIP.2018.8451252

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, *1*(4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Liu, J., Gong, M., Qin, K., & Zhang, P. (2018). A Deep Convolutional Coupling Network for Change Detection Based on Heterogeneous Optical and Radar Images. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(3), 545–559. https://doi.org/10.1109/TNNLS.2016.2636227

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html

Lu, D., Mausel, P., Brondízio, E., & Moran, E. (2004). Change detection techniques. *International Journal of Remote Sensing*, *25*(12), 2365–2401. https://doi.org/10.1080/0143116031000139863

Malila, W. (1980). Change Vector Analysis: An Approach for Detecting Forest Changes with Landsat. *LARS Symposia*. Retrieved from https://docs.lib.purdue.edu/lars_symp/385

LIST OF REFERENCES

Melekhov, I., Kannala, J., & Rahtu, E. (2016). Siamese network features for image matching. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (pp. 378–383). https://doi.org/10.1109/ICPR.2016.7899663

Michalek, J. L., Wagner, T. W., Luczkovich, J. J., & Stoffle, R. W. (1993). Multispectral change vector analysis for monitoring coastal marine environments. Retrieved from https://arizona.pure.elsevier.com/en/publications/multispectral-change-vector-analysis-for-monitoring-coastal-marin

Moser, G., Moser, G., & Serpico, S. B. (2002). Unsupervised change-detection methods for remote-sensing images. *Optical Engineering*, *41*(12), 3288. https://doi.org/10.1117/1.1518995

Moughal, T. A., & Yu, F. (2014). An Automatic Unsupervised Method Based on Context-Sensitive Spectral Angle Mapper for Change Detection of Remote Sensing Images. In *Advanced Data Mining and Applications* (pp. 151–162). Springer, Cham. https://doi.org/10.1007/978-3-319-14717-8_12

Nogueira, K., Miranda, W. O., & Santos, J. A. Dos. (2015). Improving Spatial Feature Representation from Aerial Scenes by Using Convolutional Networks. In *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images* (pp. 289–296). IEEE. https://doi.org/10.1109/SIBGRAPI.2015.39

Noh, H., Hong, S., & Han, B. (2015). Learning Deconvolution Network for Semantic Segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 1520–1528). IEEE. https://doi.org/10.1109/ICCV.2015.178

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Persello, C., & Stein, A. (2017). Deep Fully Convolutional Networks for the Detection of Informal Settlements in VHR Images. *IEEE Geoscience and Remote Sensing Letters*, *14*(12), 2325–2329. https://doi.org/10.1109/LGRS.2017.2763738

Radke, R. J., Andra, S., Al-Kofahi, O., & Roysam, B. (2005). Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, *14*(3), 294–307. https://doi.org/10.1109/TIP.2004.838698

Romero, A., Gatta, C., & Camps-Valls, G. (2016). Unsupervised Deep Feature Extraction for Remote Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(3), 1349–1362. https://doi.org/10.1109/TGRS.2015.2478379

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning Internal Representations by Error Propagation. Retrieved from https://apps.dtic.mil/docs/citations/ADA164453

Silva, P. G., Santos, J. R., Shimabukuro, Y. E., Souza, P. E. U., & Graca, P. M. L. A. (2003). Change vector analysis technique to monitor selective logging activities in Amazon. In *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477)* (Vol. 4, pp. 2580–2582). IEEE. https://doi.org/10.1109/IGARSS.2003.1294515

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. Retrieved from http://arxiv.org/abs/1409.1556

SINGH, A. (1989). Review Article Digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing*, *10*(6), 989–1003. https://doi.org/10.1080/01431168908903939

LIST OF REFERENCES

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958. Retrieved from http://jmlr.org/papers/v15/srivastava14a.html

Voegtle, T., & Steinle, E. (2004). Detection and Recognition of Changes in Building Geometry Derived from Multitemporal Laser scanning Data. Retrieved from https://www.semanticscholar.org/paper/Detection-and-Recognition-of-Changes-in-Building-Voegtle-Steinle/010fcba136866ea380e921308a07fd3cd792d2d3

Yokoya, N., Zhu, X. X., & Plaza, A. (2017). Multisensor Coupled Spectral Unmixing for Time-Series Analysis. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(5), 2842–2857. https://doi.org/10.1109/TGRS.2017.2655115

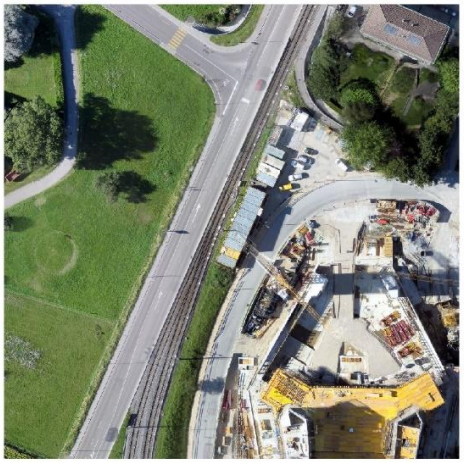
Yu, F., & Koltun, V. (2015). Multi-Scale Context Aggregation by Dilated Convolutions. Retrieved from http://arxiv.org/abs/1511.07122

Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In *Computer Vision – ECCV 2014* (pp. 818–833). Springer, Cham. https://doi.org/10.1007/978-3-319-10590-1_53

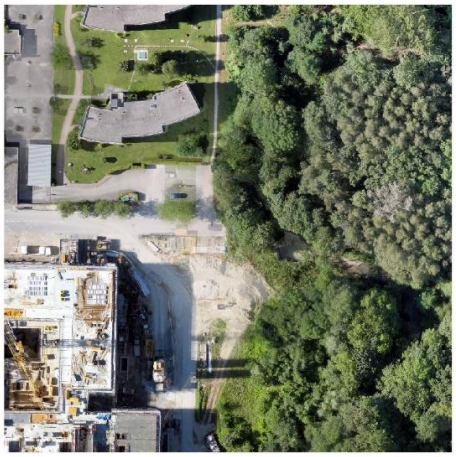
Zhuang, H., Deng, K., Fan, H., & Yu, M. (2016). Strategies Combining Spectral Angle Mapper and Change Vector Analysis to Unsupervised Change Detection in Multispectral Images. *IEEE Geoscience and Remote Sensing Letters*, *13*(5), 681–685. https://doi.org/10.1109/LGRS.2016.2536058

9 Appendix

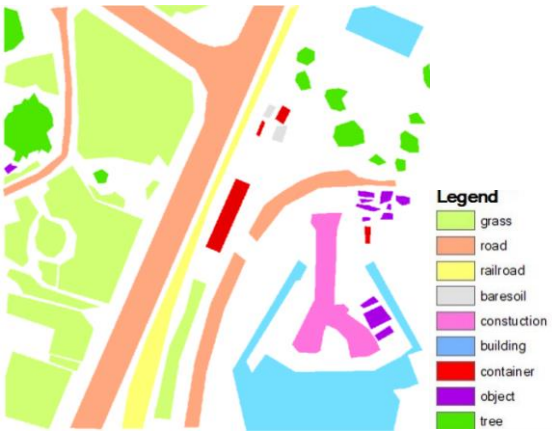
9.1 Appendix 1



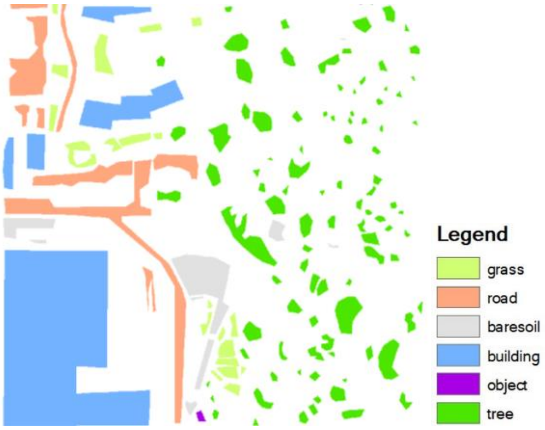
(a_1)



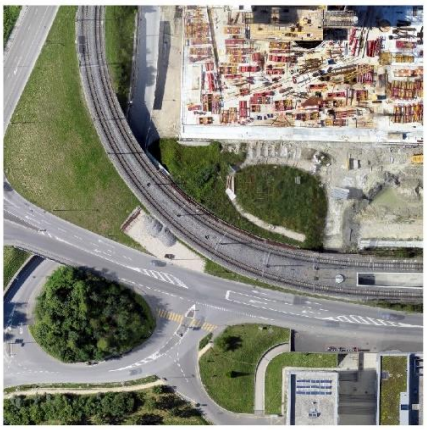
(b_1)

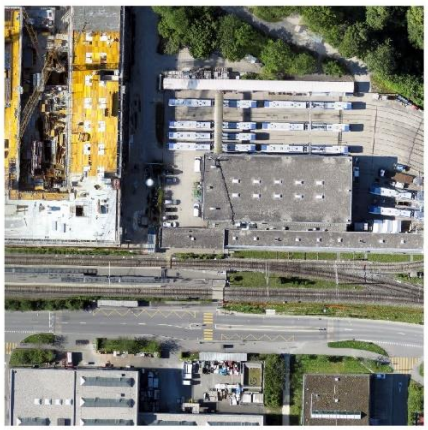


(a_2)

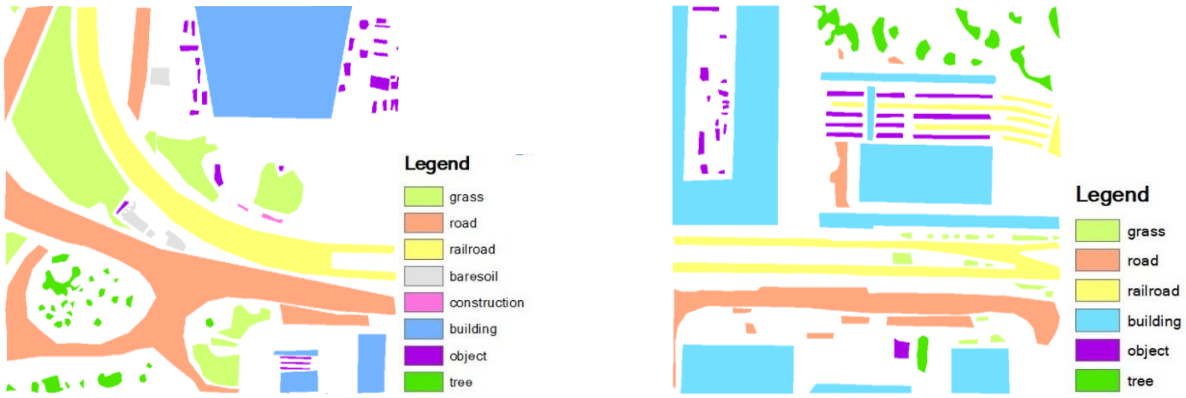


(b_2)



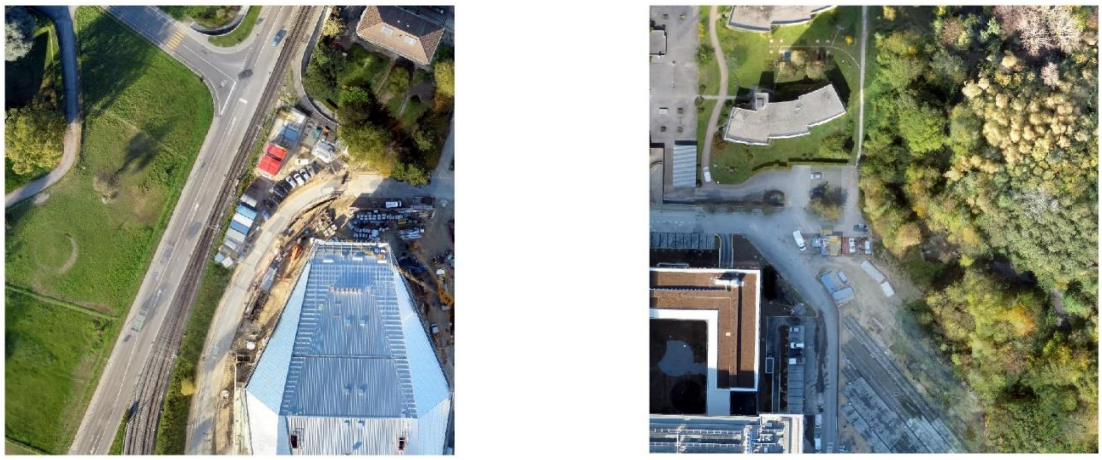


(c1) (d1)



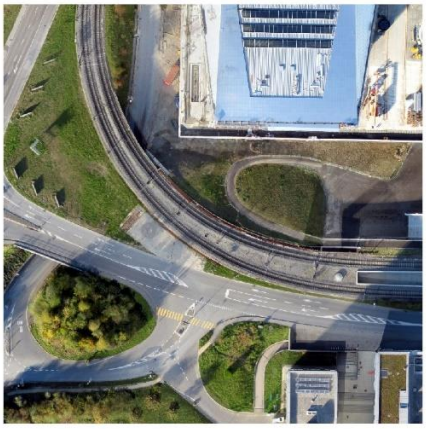
(c2) (d2)

The RGB images and corresponding annotation of four tiles for epoch 2

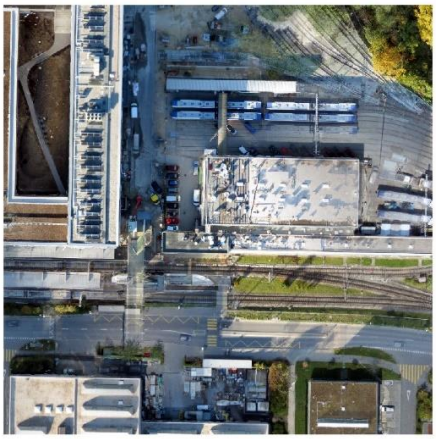


(a1) (b1)

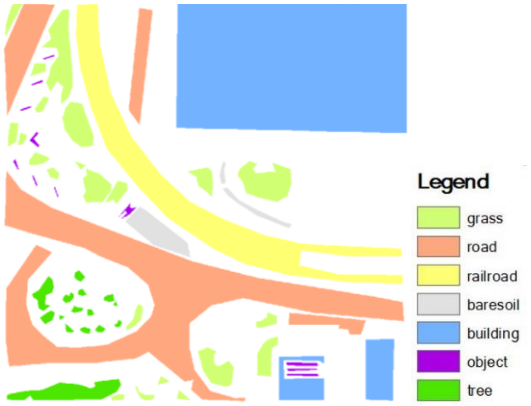
(a2) (b2)



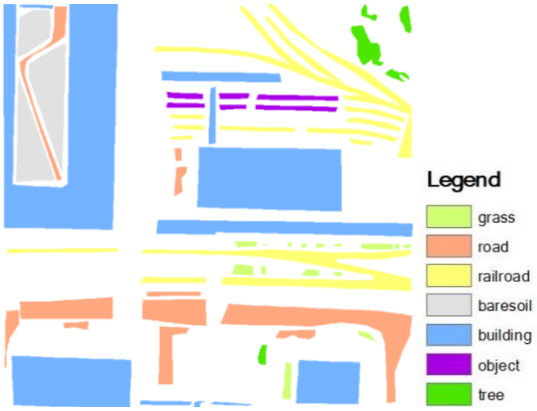
(c1)



(d1)



(c2)



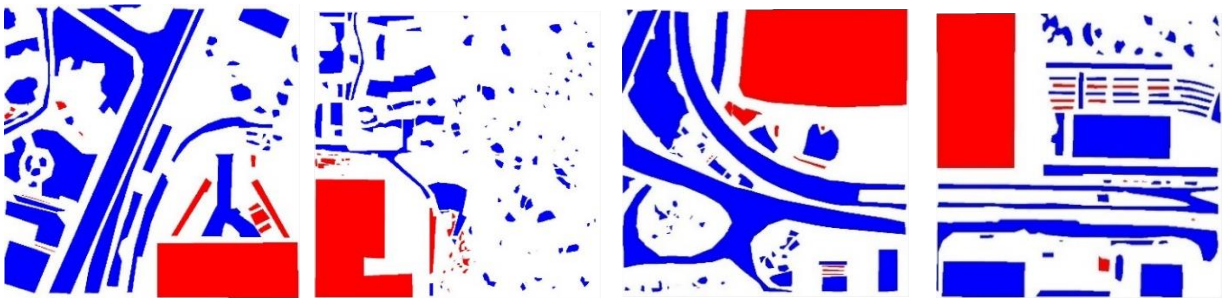
(d2)

The RGB images and corresponding annotation of four tiles for epoch 3

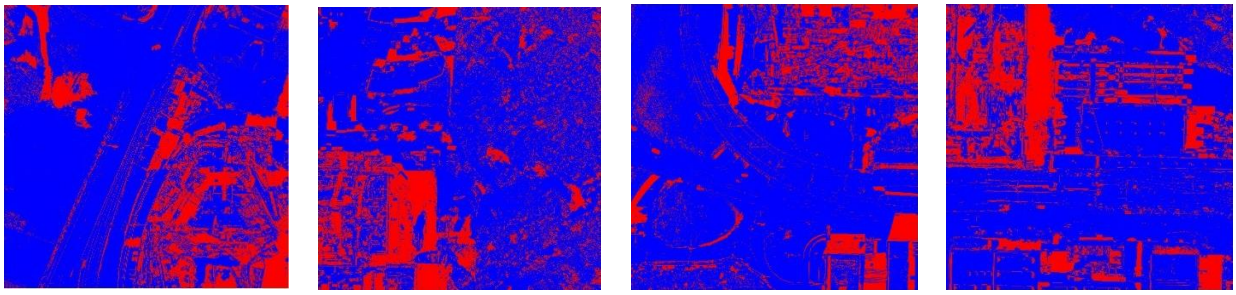
9.2 Appendix 2

(a_1) (a_2) (a_3) (a_4)

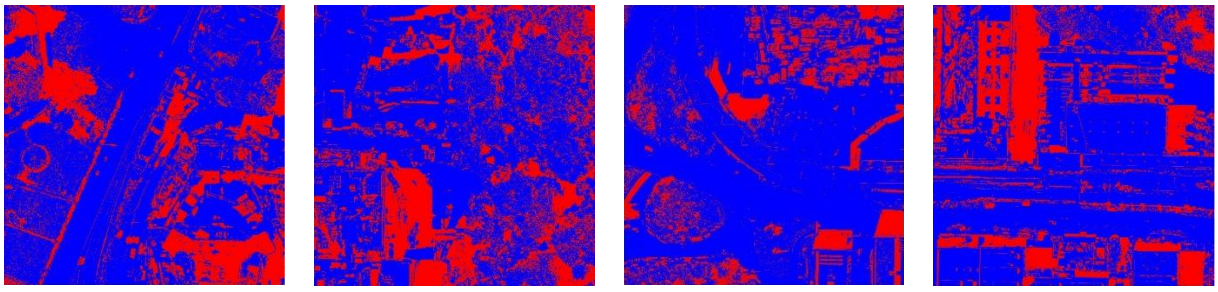
(b_1) (b_2) (b_3) (b_4)



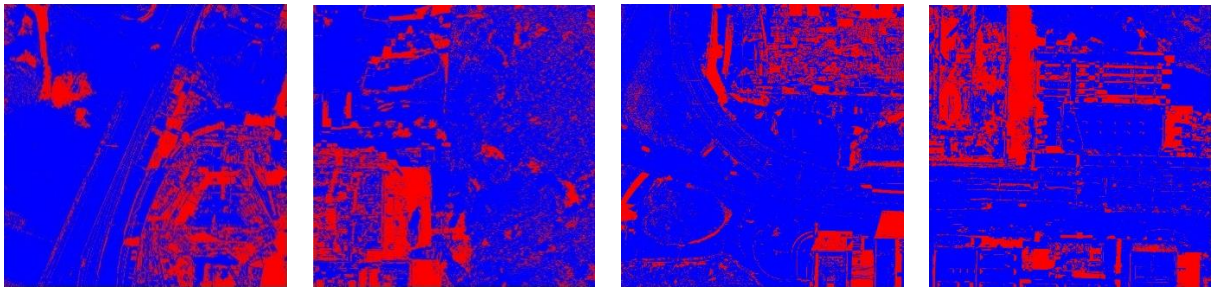
(c_1) (c_2) (c_3) (c_4)



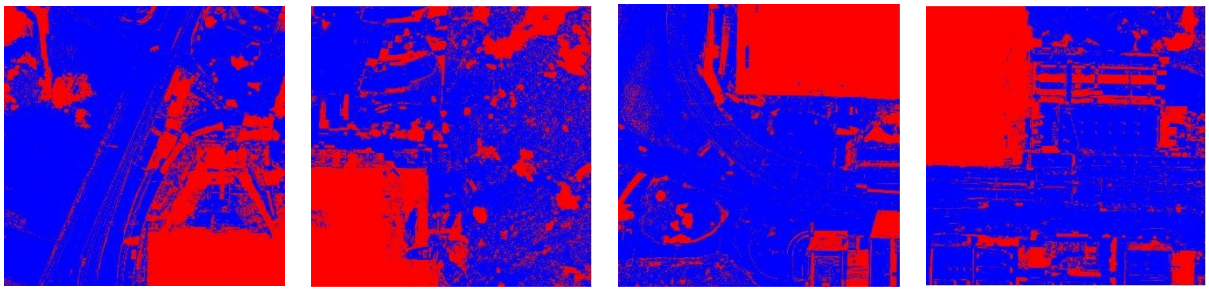
(d_1) (d_2) (d_3) (d_4)



(e_1) (e_2) (e_3) (e_4)

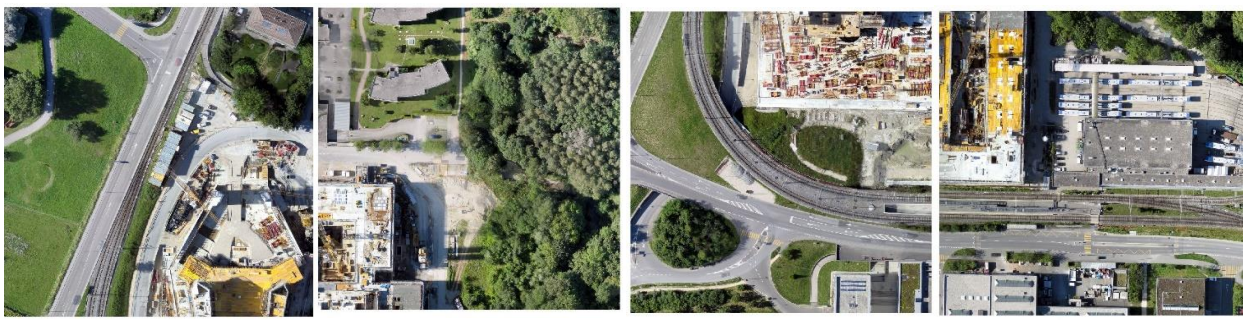


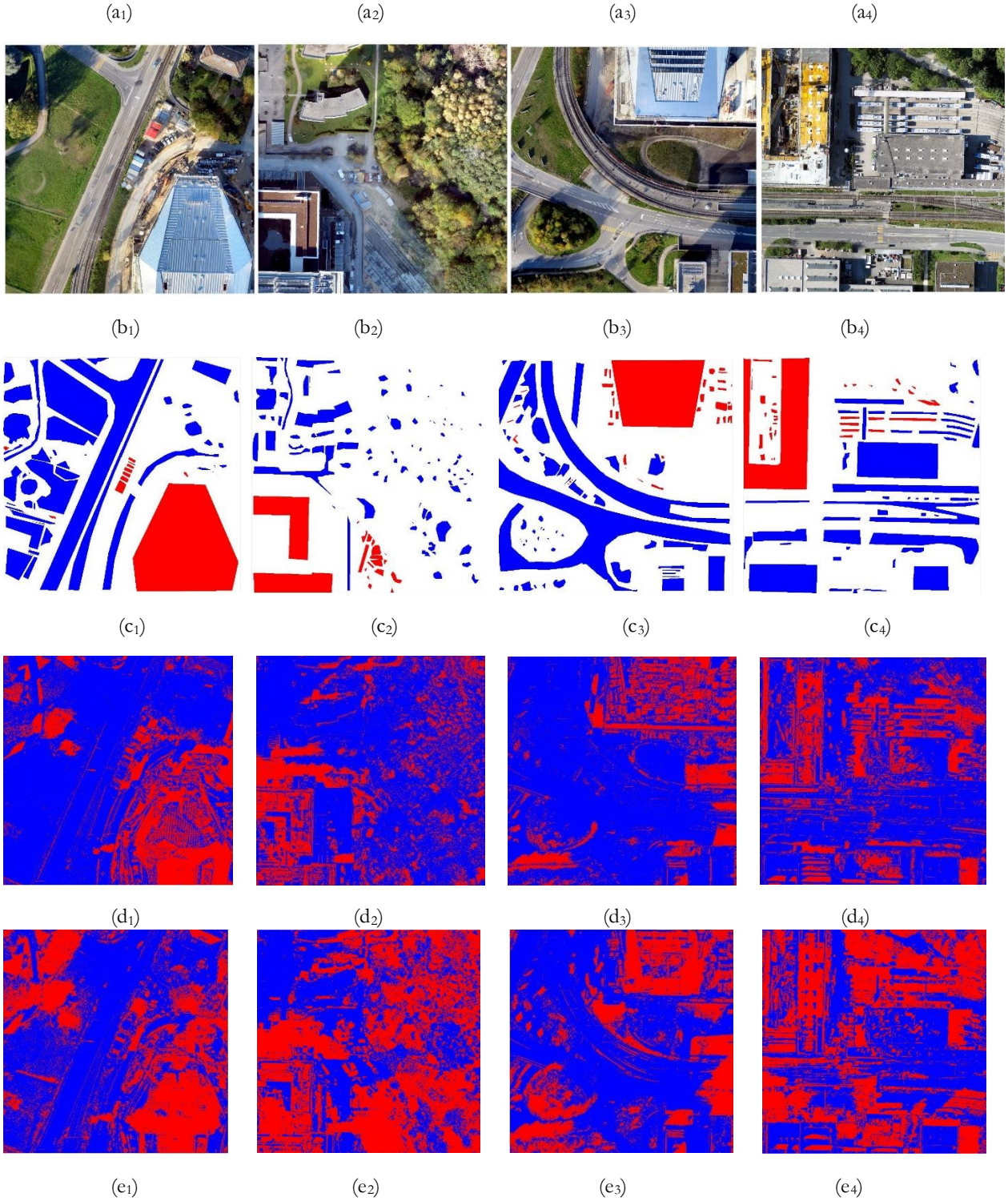
(f_1) (f_2) (f_3) (f_4)

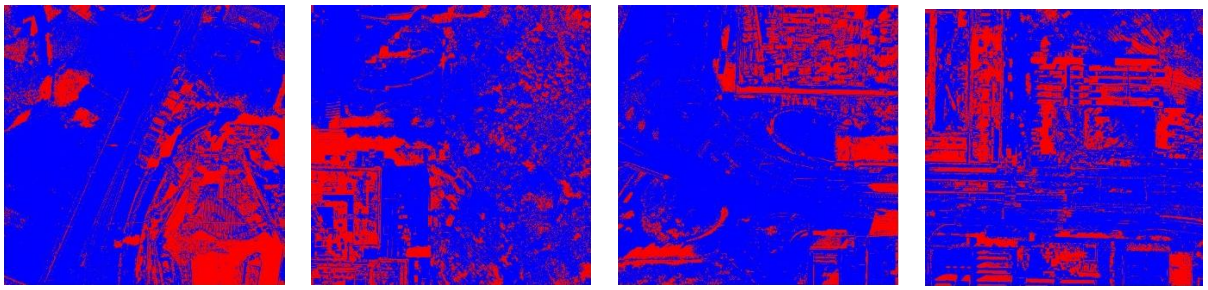


(g_1) (g_2) (g_3) (g_4)

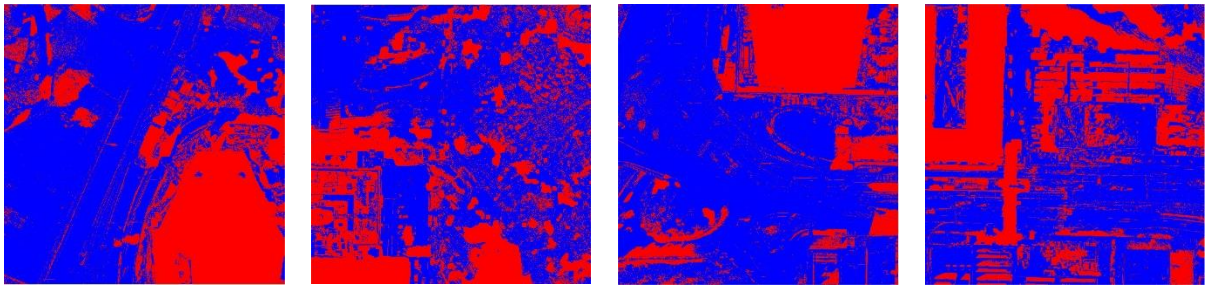
The change detection result between the first and the second epochs. (a1), (a2), (a3) and (a4) is the first, second, third and fourth tile of the first epoch of RGB data; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the second epoch of RGB data; (c1), (c2), (c3) and (c4) is the first, second, third, fourth tile of ground truth respectively; (d1), (d2), (d3) and (d4) is the first, second, third and fourth tile of the change detection result based on CVA algorithm respectively; (e_1), (e_2), (e_3) and (e_4) is the first, second, third and fourth tile of the change detection result based on SAM algorithm respectively; (f_1), (f_2), (f_3) and (f_4) is the first, second, third and fourth tile of the change detection result based on CVA&SAM algorithm respectively; (g_1), (g_2), (g_3) and (g_4) is the first, second, third and fourth tile of the change detection result based on the CVA&DSM algorithm respectively




(a1)
(a2)
(a3)
(a4)
(b1)
(b2)
(b3)
(b4)
(c1)
(c2)
(c3)
(c4)
(d1)
(d2)
(d3)
(d4)
(e1)
(e2)
(e3)
(e4)



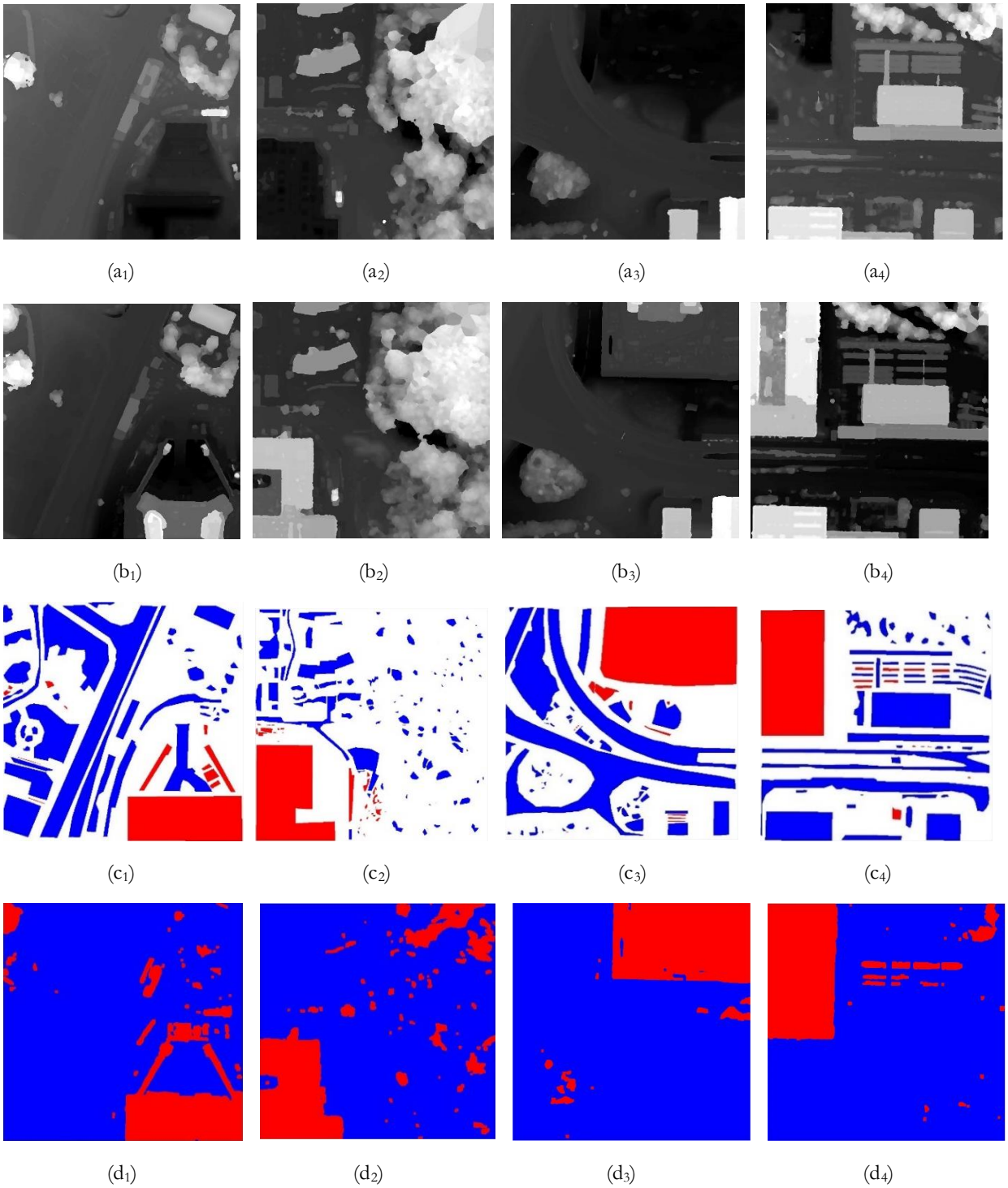
(f_1) (f_2) (f_3) (f_4)



(g_1) (g_2) (g_3) (g_4)

The change detection result between the second and the last epochs. (a1), (a2), (a3) and (a4) is the first, second, third and fourth tile of the second epoch of RGB data; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the third epoch of RGB data; (c1), (c2), (c3) and (c4) is the first, second, third, fourth tile of ground truth respectively; (d1), (d2), (d3) and (d4) is the first, second, third and fourth tile of the change detection result based on CVA algorithm respectively; (e_1), (e_2), (e_3) and (e4) is the first, second, third and fourth tile of the change detection result based on SAM algorithm respectively; (f1), (f2), (f3) and (f4) is the first, second, third and fourth tile of the change detection result based on CVA&SAM algorithm respectively; (g1), (g2), (g3) and (g4) is the first, second, third and fourth tile of the change detection result based on the CVA&DSM algorithm respectively

9.3 Appendix 3



(a_1) (a_2) (a_3) (a_4)

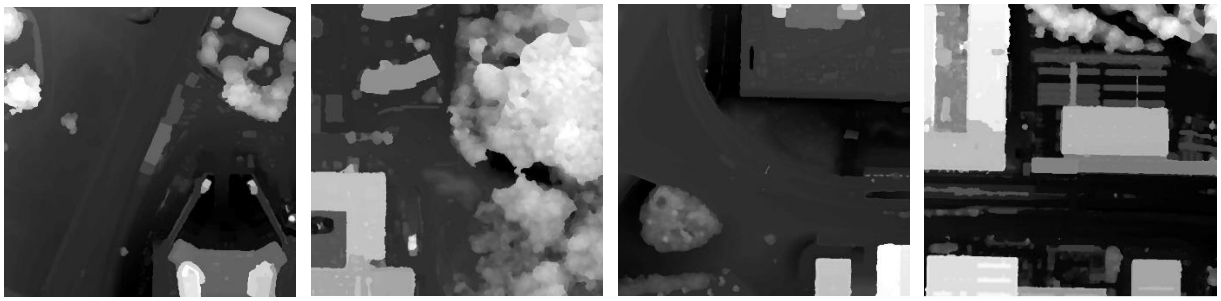
(b_1) (b_2) (b_3) (b_4)

(c_1) (c_2) (c_3) (c_4)

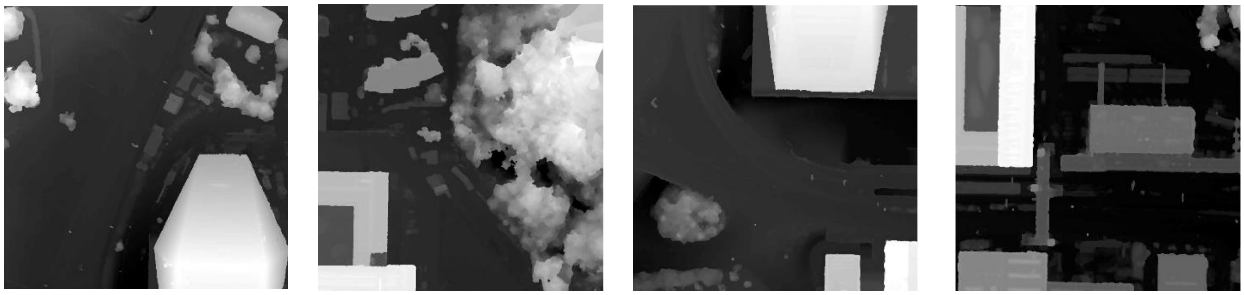
(d_1) (d_2) (d_3) (d_4)

The change detection results between the first and the second epochs. (a1), (a2), (a3) and (a4) is the first, second, third and fourth tile of the first epoch of DSM data; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the second epoch of DSM data; (c_1), (c_2), (c_3) and (c_4) is the first, second, third, fourth tile of ground truth respectively; (d_1),

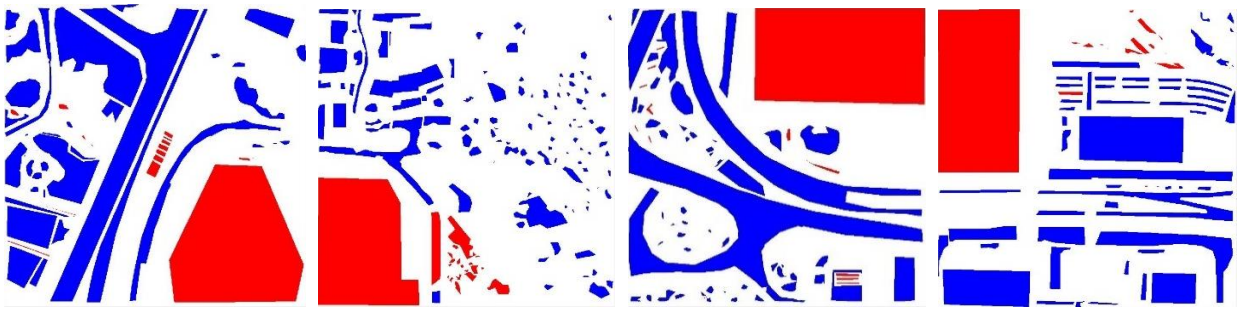
(d_2), (d_3) and (d_4) is the first, second, third and fourth tile of the change detection result DSM-based algorithm respectively



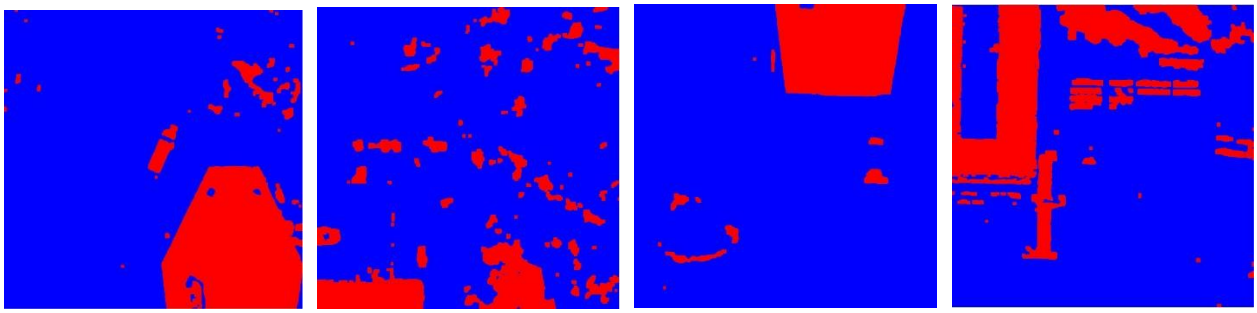
(a_1) (a_2) (a_3) (a_4)



(b_1) (b_2) (b_3) (b_4)



(c_1) (c_2) (c_3) (c_4)



(d_1) (d_2) (d_3) (d_4)

The change detection results between the second and the third epochs. (a1), (a2), (a3) and (a4) is the first, second, third and fourth tile of the second epoch of DSM data; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the third epoch of DSM data; (c_1), (c_2), (c_3) and (c_4) is the first, second, third, fourth tile of ground truth respectively; (d_1), (d_2), (d_3) and (d_4) is the first, second, third and fourth tile of change detection result of DSM-based algorithm respectively

9.4 Appendix 4

Result when receiving 100% unsupervised result as training samples

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.945567	0.967908	0.890884	0.9278
CD23	0.899088	0.945041	0.790306	0.860775
CD13	0.920924	0.982259	0.849438	0.911033
Average of above	0.923093	0.96733	0.846494	0.902887

The confusion matrix of the result when receiving 100% unsupervised result as training samples

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.35751	0.0648319
Unchanged in the result	0.012075	0.565583

Result when receiving 90% unsupervised result as training samples

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.95338	0.964739	0.911424	0.937324
CD23	0.914012	0.938925	0.824814	0.878178
CD13	0.932895	0.976009	0.875485	0.923018
Average of above	0.934479	0.962181	0.873405	0.915646

The confusion matrix of the result when receiving 90% unsupervised result as training samples

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.355607	0.0515433
Unchanged in the result	0.013978	0.5788716

Result when receiving 80% unsupervised result as training samples

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.960243	0.961575	0.930697	0.945884
CD23	0.922287	0.934473	0.846154	0.888123
CD13	0.939924	0.975681	0.889318	0.9305
Average of above	0.941823	0.959767	0.891192	0.924209

The confusion matrix of the result when receiving 80% unsupervised result as training samples

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.354716	0.043308
Unchanged in the result	0.014869	0.5871069

Result when receiving 70% unsupervised result as training samples

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.95953	0.95666	0.933035	0.9447
CD23	0.933716	0.924397	0.880714	0.902027

CD13	0.948161	0.971423	0.90905	0.939202
Average of above	0.947851	0.953725	0.909564	0.931121

The confusion matrix of the result when receiving 70% unsupervised result as training samples

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.352482	0.0350464
Unchanged in the result	0.017103	0.5953685

Result when receiving 60% unsupervised result as training samples

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.965796	0.953246	0.952151	0.952698
CD23	0.952843	0.919879	0.936152	0.927944
CD13	0.962895	0.96831	0.943182	0.955581
Average of above	0.960907	0.950129	0.94443	0.947271

The confusion matrix of the result when receiving 60% unsupervised result as training samples

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.351154	0.0206619
Unchanged in the result	0.018431	0.609753

Result when receiving 50% unsupervised result as training samples

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.963328	0.922255	0.974902	0.947848
CD23	0.957796	0.909315	0.960726	0.934314
CD13	0.968977	0.966473	0.958603	0.962522
Average of above	0.963626	0.935874	0.964654	0.950046

The confusion matrix of the result when receiving 50% unsupervised result as training samples

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.345885	0.0126736
Unchanged in the result	0.0237	0.6177414

Result when receiving 40% unsupervised result as training samples

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.970275	0.948274	0.968804	0.958429
CD23	0.961467	0.902416	0.979219	0.93925
CD13	0.97606	0.959841	0.98167	0.970633
Average of above	0.969638	0.94046	0.97652	0.958151

The confusion matrix of the result when receiving 40% unsupervised result as training samples

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.34758	0.0083575
Unchanged in the result	0.022005	0.6220574

Result when receiving 30% unsupervised result as training samples

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.970748	0.944416	0.973839	0.958902
CD23	0.956778	0.884992	0.982317	0.931118
CD13	0.973322	0.949922	0.984817	0.967055
Average of above	0.967451	0.93063	0.980303	0.954821

The confusion matrix of the result when receiving 30% unsupervised result as training samples

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.343947	0.0069107
Unchanged in the result	0.025638	0.6235042

Result when receiving 20% unsupervised result as training samples

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.9712	0.942151	0.977332	0.959419
CD23	0.957315	0.88214	0.987185	0.931711
CD13	0.974779	0.95395	0.984378	0.968925
Average of above	0.968277	0.930639	0.982607	0.955917

The confusion matrix of the result when receiving 20% unsupervised result as training samples

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.343951	0.0060883
Unchanged in the result	0.025635	0.6243257

Result when receiving 10% unsupervised result as training samples

Image	Accuracy	Precision	Recall	F1 Score
CD12	0.969863	0.942179	0.973565	0.957615
CD23	0.948016	0.852765	0.988126	0.915469
CD13	0.969886	0.942952	0.983303	0.962705
Average of above	0.96332	0.918532	0.981013	0.948745

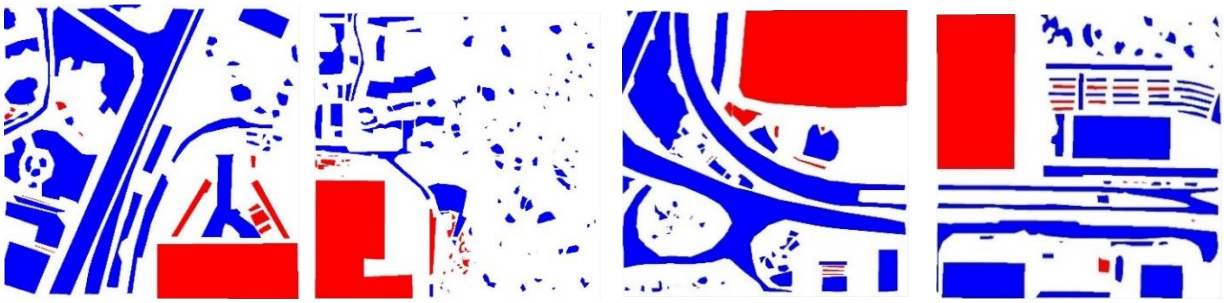
The confusion matrix of the result when receiving 10% unsupervised result as training samples

Overall	Changed in the reference	Unchanged in the reference
Changed in the result	0.339476	0.0065705
Unchanged in the result	0.030109	0.6238444

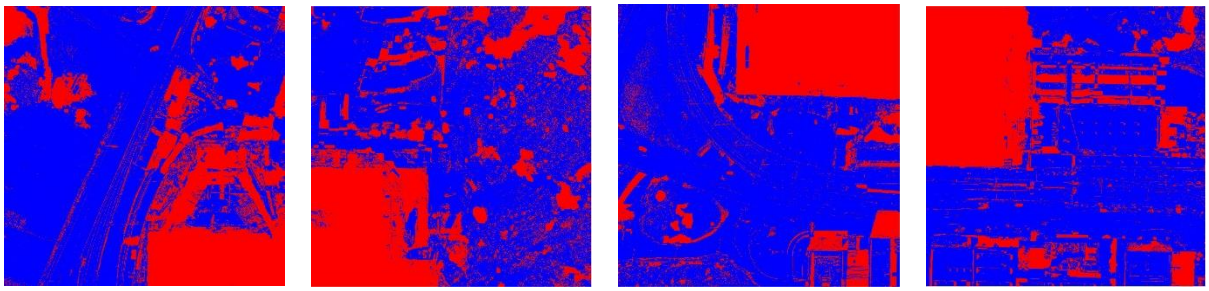
9.5 Appendix 5

(a_1) (a_2) (a_3) (a_4)

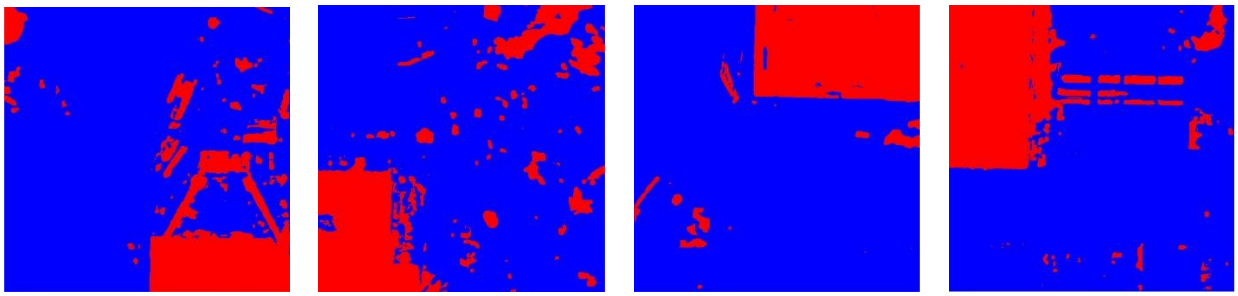
(b_1) (b_2) (b_3) (b_4)



(c_1) (c_2) (c_3) (c_4)

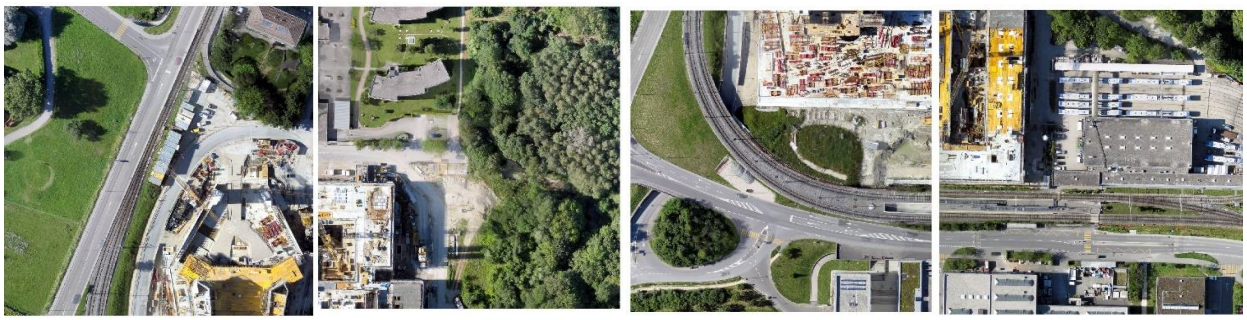


(d_1) (d_2) (d_3) (d_4)



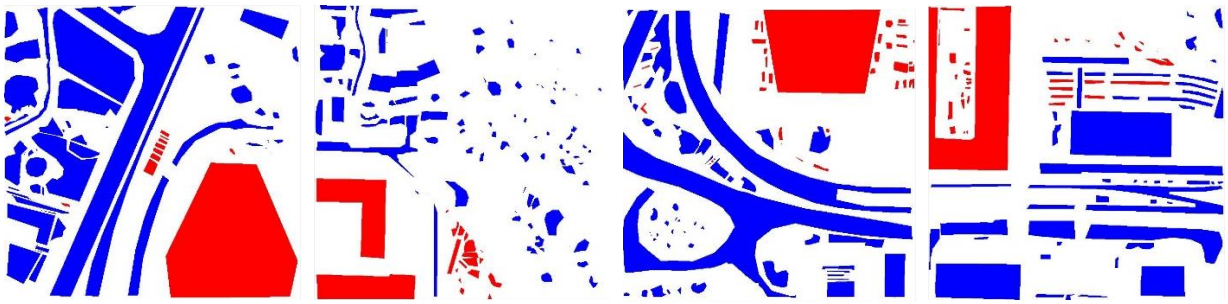
(e_1) (e_2) (e_3) (e_4)

The final result of the first two epoch. (a1), (a2), (a3) and (a4) is the first, second, third and fourth tile of the first epoch of RGB data; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the second epoch of RGB data; (c_1), (c_2), (c_3) and (c_4) is the first, second, third, fourth tile of ground truth respectively; (d_1), (d_2), (d_3) and (d_4) is the first, second, third and fourth tile of the unsupervised FCN result respectively

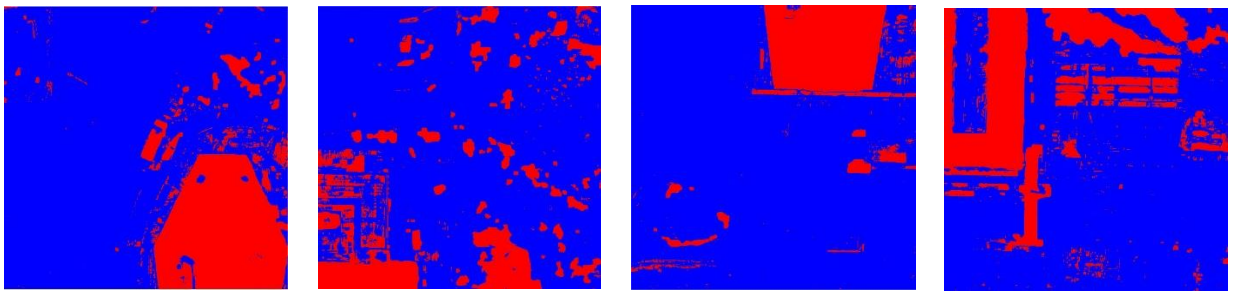


(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4)



(c_1) (c_2) (c_3) (c_4)



(d_1) (d_2) (d_3) (d_4)

The final result between the second and the third epochs. (a1), (a2), (a3) and (a4) is the first, second, third and fourth tile of the second epoch of RGB data; (b1), (b2), (b3) and (b4) is the first, second, third and fourth tile of the third epoch of RGB data; (c_1), (c_2), (c_3) and (c_4) is the first, second, third, fourth tile of ground truth respectively; (d_1), (d_2), (d_3) and (d_4) is the first, second, third and fourth tile of the unsupervised FCN result respectively