Master's Thesis of Engineering

Domain Adaptation Framework for Deep Learning based Change Detection in Remotely Sensed Data under Prior Probability Shift

원격탐사 영상의 딥러닝 기반 변화탐지 기술을 위한 사전확률변화 도메인 적응 프레임워크

February 2020

Department of Civil and Environmental Engineering
Seoul National University

Hunsoo Song

Abstract

The success of deep learning algorithms applied on large-scale remotely sensed imagery rests on the representativeness of the samples used to train the model. When the training samples in the source domain do not fully represent the samples in the target domain where prior information is unavailable, the trained model may fail to adapt to the target domain and will therefore require a manual and time-consuming process of labelling the samples to ensure sample representativeness. However, the vast majority of the algorithms in literature have not considered the distribution gap between the sampled domains, and in turn, have failed to generalize their performance on a newly given area.

Domain adaptation can provide a solution to this representative issue. The goal of domain adaptation is to bridge the distribution gap between the source domain and the target domain. This thesis focuses on the distribution gap given unbalanced class proportions of two domains, defined as the prior probability shift, and proposes a novel framework to resolve the reduction in performance of deep learning models in change detection under prior probability shift. The proposed framework estimated class distributions via an unsupervised approach and adjusted the change threshold of the softmax output in the deep learning model. To validate the algorithm's performance, the most widely used deep learning model, Convolutional Neural Network (CNN), and unsupervised algorithm, Change Vector Analysis (CVA), were implemented

in the proposed framework. The framework was examined using three sub-regions composed of various proportions of the "change class". The datasets used in this study were acquired from bi-temporal Unmanned Aerial Vehicle (UAV) imagery including RGB channels and Digital Surface Model (DSM). To train the CNN, two different sample sizes (i.e., a total of 1,000 and 2,000) with two sets of different sampling designs (i.e., the base sampling design and the additional sampling design), were drawn from each of the three sub-regions (source domain). Subsequently, the trained CNN was adapted using the patch-based CVA and tested on the two other sub-regions (target domains) of different change/no-change ratios. The proposed framework was applied separately to RGB and RGB + DSM imagery.

The results demonstrated that the proposed framework successfully rectified the biased change threshold and improved the change detection performance under prior probability shift. Furthermore, the deep learning model was found to be susceptible to prior probability shift, particularly when pre-trained on greater amount of data with richer content, which thus highlighted the importance of domain adaptation in current trends of deep learning applications. Last but not least, the proposed framework is simple to implement and can be applied to a wide variety of classifiers.

Keyword: Domain Adaptation, Change Detection, Deep Learning, Prior Probability Shift, Convolutional Neural Network, Change Vector Analysis

Student Number: 2018-27253

Table of Contents

1. Introduction

1.1. Motivation and Objective of Study

With great advances in remote sensing technology, the amount of remotely sensed imagery produced by spaceborne and airborne sensors is growing enormously. To deal with this rapidly increasing amounts of data, techniques to effectively interpret the Earth's surface are developing. Change detection is one of the most important techniques in remote sensing and is widely used to update changes to manage a large-scale area (Hansen and Loveland, 2012; Tewkesbury *et al.,* 2015). Significant knowledge for urban and environmental studies can be achieved from accurate and timely change detection (Rahman, 2016; Sexton *et al.,* 2015). In spite of numerous struggles to develop change detection techniques, however, there is still no universal change detection algorithm with high accuracy that can keep the pace of rapidly accumulating remotely sensed data (Hussain *et al.,* 2013; Qin *et al.,* 2016). Therefore, the advancement of change detection algorithms is still required.

Recently, deep neural networks have shown successful performance in change detection applications (Zhu *et al.,* 2017; Song *et al.,* 2018; Mou *et al.,* 2018; Ma *et al.,* 2019). Deep neural networks with multiple non-linear layers effectively learn high-level abstract features in the multi-temporal images and outperform conventional change detection algorithms (Ma *et*

al., 2019; Heydari and Mountrakis, 2019). However, a large number of training samples are required in general to perform deep learning successfully. Plus, the success of deep learning algorithms largely depends on the representativeness of the samples used to train the model (LeCun *et al.,* 2015). In many applications for remotely sensed imagery, however, collecting ground truths of good representation is expensive, labor-intensive, and even impossible occasionally (Foody, 2002; Jin *et al.,* 2014). To alleviate the burden of collecting ground truths, there have been studies reusing the trained model or the collected training samples to the newly acquired scenes where there is no existing knowledge (Persello and Bruzzone, 2014). However, directly applying the trained model or the collected training samples to an unseen area is prone to fail (Tuia *et al.,* 2016).

The performance degradation is likely to occur when the trained model, through any machine learning algorithms, is applied to a new region (*target domain*) that is different from the region (*source domain*) where the training samples are collected. This is because of the *dataset shift* where the joint distribution of inputs and outputs differs between the training and test datasets (Moreno-Torres *et al.,* 2012). To tackle the problem where the distribution in the source domain does not fully represent the distribution in the target domain, there have been studies to overcome the discrepancy between two domains; this is known as *domain adaptation* (Patel *et al.,* 2015).

In order for deep learning to be used timely and cost-effectively in remote sensing, domain adaptation techniques are imperative. However, domain adaptation has not received much attention in remote sensing society, while numerous studies have focused on developing deep learning models to achieve higher and faster performance. Especially in change detection, only a few studies have researched on domain adaptation for deep learning models (Yang *et al.,* 2019). Therefore, the objective of this study lies in highlighting domain adaptation for change detection, which is important but often overlooked, and is to present a domain adaptation framework to maximize the performance of the deep learning based change detection algorithm.

1.2. Organization of the Thesis

This thesis is organized as follows: in Section 2, change detection algorithms for remote sensing are presented. Also, a theoretical explanation of domain adaptation and its methodology are described. In Section 3, the proposed domain adaptation framework for change detection is explained in two steps, the estimation of the prior probability and the identification of changed pixel. The experimental design is illustrated in Section 4. Then, experimental results and discussion are provided in Section 5. Finally, the conclusion of the thesis is given in Section 6.

2. Background and Related Works

2.1. Change Detection Algorithms

Numerous change detection algorithms have been developed for a long time in the field of remote sensing (Hussain *et al.,* 2013; Qin *et al.,* 2016). The change detection algorithms can be categorized according to both the change detection output and the classification methodology (Tewkesbury *et al.,* 2015). In terms of the change detection output, the algorithms can be first classified according to whether the changed pixel has a class or not. If the changed pixel has a class, there is a clear advantage that a land cover map can be directly generated using the change detection result; however, this approach has disadvantages in that it requires prior information on each class and intra-class changes can be omitted. On the other hand, change detection algorithms, which only determine whether the target has changed or not, have an advantage that they do not necessitate the information about the land cover class; however, an additional post-classification process is required if a complete land cover map is of interest. The second method of categorization (i.e., according to the classification methodology) will be detailed in Subsection 2.1.1.

In this thesis, a change detection algorithm, which simply identifies whether the target has changed or not (i.e., binary classification), was

considered. The contents of Section 2.1. hereafter are as follows: first, change detection algorithms are illustrated by being grouped into two families (i.e., unsupervised and supervised), and their pros and cons are described in Subsection 2.1.1. Then, Change Vector Analysis (CVA) and Convolutional Neural Network (CNN), the most commonly used supervised algorithm and deep learning model, respectively, are illustrated in Subsection 2.1.2. and 2.1.3. This thesis only covers the CVA and CNN in detail among the change detection algorithms since the proposed domain adaptation framework adopted CVA and CNN. Please see the following papers (Hussain *et al.,* 2013; Tewkesbury *et al.,* 2015) for the technical details of other change detection algorithms.

2.1.1. Unsupervised and Supervised Algorithms

In terms of the classification methodology, change detection algorithms can be divided into unsupervised and supervised methods according to the presence of prior knowledge in the investigation area. Supervised methods learn the decision boundary of change/no-change by training the ground truth data; however, unsupervised methods do not require prior knowledge of the data (Tewkesbury *et al.,* 2015). Although unsupervised methods have an advantage that they do not require training samples, it is difficult to set a certain threshold of change, and the accuracy of the clustering method depends too much on the parameters chosen (Hussain *et al.,* 2013).

Since change detection algorithms have their pros and cons, there is no consensus on a particularly superior algorithm. Therefore, the selection of an algorithm should be made with careful consideration of the distribution and characteristics of the applied data. However, in these days, remotely sensed data are often obtained from multimodal sensors, namely, optical sensors, synthetic aperture radars, and LiDAR, so the properties and the dimensions of the data differ (Wegner *et al.,* 2010; Bovolo and Bruzzone, 2015). Moreover, multimodal data consist of data with varying precisions and, in most cases, have different abilities for solving a given task (Byun *et al.,* 2013; Jung *et al.,* 2014; Chang *et al.,* 2015). Therefore, an algorithm that can consider non-linear features and assign a different

weight to each of the data therein is desirable (Bovolo and Bruzzone, 2015; Mou *et al.,* 2018). However, most of the unsupervised algorithms lose the information of the original data because most of them simplify the original feature space to solve the problem (Bovolo and Bruzzone, 2015) and cannot automatically identify the features of importance. Thus, a learning-based method that can learn complex and non-linear behavior considering complicated light scattering mechanisms and multimodal data sources is desirable provided that proper training samples are available.

2.1.2. Change Vector Analysis

CVA is one of the most commonly used unsupervised algorithms for change detection (Bovolo and Bruzzone, 2006). CVA identifies changes based on its magnitude and direction (Tewkesbury *et al.,* 2015). Considering corresponding pixels of bi-temporal data, the two pixels can be mapped on an n-dimensional feature space. Then, three vectors can be considered: the feature vector at time 1, the feature vector at time 2, and an interconnecting vector (i.e., change vector). With these three vectors, both magnitude and direction between data at time 1 and time 2 can be calculated. Figure 1 illustrates the geometry of a CVA in bi-dimensional feature space. The magnitude of change vector indicates the degree to which the feature (i.e., digital number in case of image) has been changed, while the direction can assist to explain the type of change. In general, the use of magnitude is sufficient to determine whether the pixel has been changed or not, and its threshold is empirically determined.

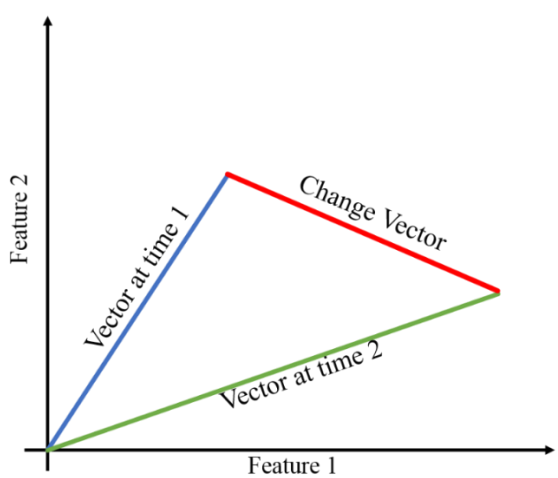

Figure 1. The Geometry of CVA in Bi-dimensional Feature Space

2.1.3. Convolutional Neural Network

Deep learning has accomplished many breakthroughs in a variety of applications. Deep learning surpasses traditional neural networks by enabling multiple hidden layers to learn deep and complex features in data. Deep learning also has gained a lot of attention in remote sensing society. Among deep learning models, CNN is the most widely adopted algorithm and has become the state-of-the-art model in many remote sensing applications because of its distinctive ability to process image data (Zhu *et al.,* 2017; Ma *et al.,* 2019). With the concept of local connectivity and parameter sharing through convolutional layers, CNN can effectively learn intricate features and can prevent overfitting problem which is the biggest problem of conventional neural networks (LeCun *et al.,* 2015).

Figure 2 illustrates the classic architecture of CNN for image classification. CNN normally consists of convolutional layers and fully-connected layers. The main difference between a CNN and traditional neural networks is the presence of a convolutional layer. The convolutional layer extracts features from the input data using filters. The filters behave like a moving window and learn their weights by computing a dot product with the input. The size and number of filters can be determined by the user. A notable characteristic of CNN's operation is that each filter has different weights but keeps each of its parameters constant during the calculation of a dot product with a moving window (i.e., parameter

sharing). Also, the dot product between the input and the filter enables neurons to connect to only a local region of the input volume (i.e., local connectivity). With the parameter sharing and local connectivity, CNN can effectively extract features and greatly reduce the learnable parameters. After the convolutional layers, fully-connected layers that connect all neurons in one layer to all neurons in the next layer usually follow. Based on the way traditional neural networks operate, each neuron of the last CNN layer contains scores calculated by consecutive dot products using the trained parameters from previous layers. Then, the last neural layer passes to the softmax layer, which transforms the score of each neuron into a probability P_i of belonging to each class by the following formula (1):

$$P_i = Softmax(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)} \quad (1)$$

where x_i represents the scores of class i calculated by the parameters in the neural network. In the case of a binary problem, such as change detection, the softmax layer returns two classes: change and no-change. The sum of the probabilities assigned to each label will be 1, and the CNN determine the class which has the highest probability. In addition to this process, various advances, such as pooling techniques, activation functions, and optimization functions, have improved the performance of CNNs and other deep learning algorithms (Krizhevsky *et al.,* 2012).

In the field of remote sensing, CNN also outperforms conventional

machine learning algorithms in many applications, including land cover classification, object detection, change detection, etc (Zhu *et al.,* 2017; Ma *et al.,* 2019). For each of the applications, several CNN models have been devised. Most CNNs have deep and complex networks because of their distinct advantage to efficiently process big data (Canziani *et al.,* 2016). However, deep and complex networks are not always good because the increase in the number of layers and filters increases the number of parameters to be learned, which thus also raises computational costs and necessitates a large number of training samples (LeCun *et al.,* 2015). If not properly designed, CNN might lose its advantage that effectively extracts features from a large amount of data and reduces learnable parameters. Therefore, CNN of appropriate complexity should be designed considering the objective and the data type to be learned.

For change detection in remote sensing, a fairly large number of CNN models, including variants such as fully convolutional networks, faster region-based convolutional network (Faster-RCNN), and siamese convolutional network, have been developed and have shown promising performance. Daudt *et al.* (2018) presented two CNN models to detect changes in an urban area using Sentinel-2 satellite images. Khan *et al.* (2017) proposed a deep CNN model modified from VGG-16 network for forest change detection using Landsat images. Zhan *et al.* (2017) proposed a deep siamese convolutional network for optical aerial images. Liu *et al.*

(2016) proposed a convolutional coupling network jointly using optical images and synthetic aperture radar images. Gong *et al.* (2017) compared several change detection algorithms including CVA, object-based change detection, post-classification comparison, and other deep learning methods, then confirmed that Faster-RCNN obtained the highest overall accuracy in high-resolution satellite images. Despite a large number of studies, very few studies considered the performance of CNN under the dataset shift in change detection (the examples of dataset shift will be described in detail in Subsection 2.2.2.). Thus, this thesis is dedicated to verifying and improving the performance of CNNs in change detection under dataset shifts.

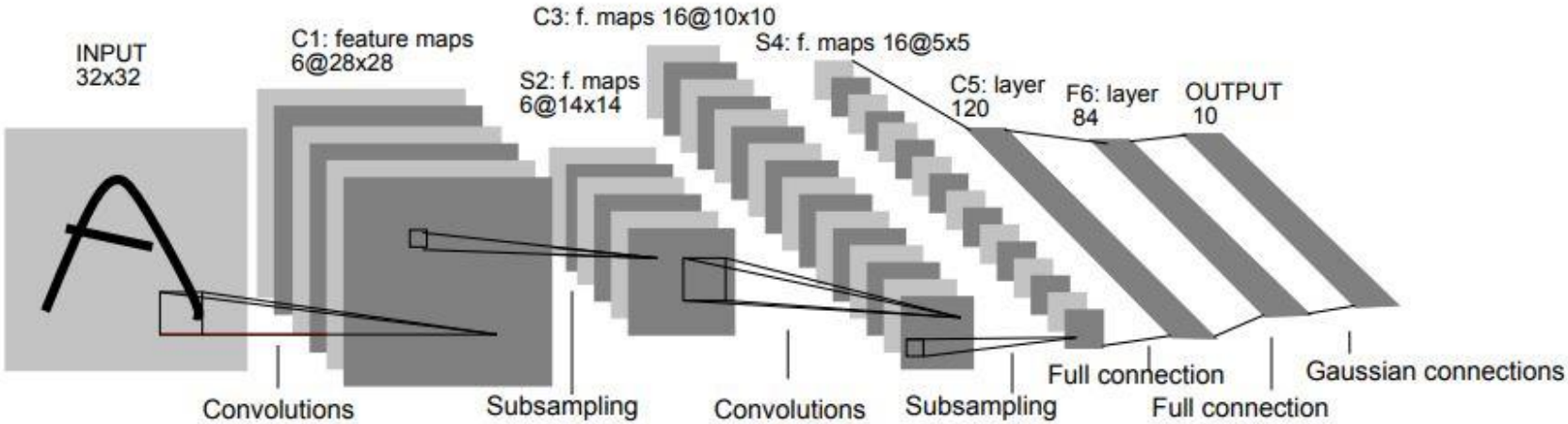

Figure 2. Classic Architecture of Convolutional Neural Network (LeNet-5, Lecun *et al.,* 1998)

2.2. Domain Adaptation

The success of supervised deep learning, applied to remotely sensed imagery, strongly relies on the representativeness of the samples used to train the model. When training data acquired from an image or a spatial region do not represent the data that is used for mapping, the difference of the data distribution (i.e., dataset shift) may deteriorate the performance of supervised algorithms. Such dataset shifts are very common in remote sensing problems because remotely sensed data are often acquired from multi-sensor, from a different time with different atmospheric conditions, and from a very large area (Gomez *et al.,* 2016; Zhu *et al.,* 2019). Such a non-transferability issue can be overcome by acquiring additional training samples in the newly given area. However, collecting training samples of good representation for every image acquisition is not realistic (Tuia *et al.,* 2016). Therefore, the reuse of training samples or trained model is mandatory to enhance the effectiveness of supervised algorithms.

Domain adaptation is a study to address the problem where the distribution in the source domain does not fully represent the distribution in the target domain (Patel *et al.,* 2015). By overcoming the discrepancy between two domains, domain adaptation helps to reuse the collected training samples or trained models and prevents time-consuming labeling and training processes. Domain adaptation is a specific branch of the

transfer learning problem with multitask learning and self-taught learning (Pan and Yang, 2009; Patel *et al.,* 2015). Among the branches of transfer learning, the domain adaptation problem refers to the case where the joint probabilities distribution of the two domains describing the relationship between the input variable and the output variable are similar but not identical (Patel *et al.,* 2015; Tuia *et al.,* 2016). Therefore, the goal of domain adaptation methods is to bridge the distribution gap so that the algorithms working in the source domain perform well in the target domain as well.

The contents of Section 2.2. hereafter are as follows: in Subsection 2.2.1., the four categories of domain adaptation methodologies are introduced. Next, Subsection 2.2.2. presents a concept and examples of dataset shift. Lastly, in Subsection 2.2.3., prior probability shift in change detection, one type of dataset shift, is illustrated.

2.2.1. Adaptation Methodology

In remote sensing literature, various domain adaptation methods have been presented to address the dataset shift problem. Adaptation methods can be grouped into four categories (Tuia *et al.,* 2016). The first category methods involve finding and utilizing invariant features during the shift between the two domains (Bruzzone and Persello, 2009). The second category methods involve aligning the distribution of the source domain with the distribution of the target domain and perform the adaptation using the same classifier (Yang and Crawford, 2015). The third category methods involve applying the trained model from the source domain to the target domain leaving both distributions intact (Rajan *et al.,* 2006; Bahirat *et al.,* 2011). The last category methods involve assuming the presence of a few labeled samples in the target domain, and by utilizing these, making the classifier adapt to the target domain (Crawford *et al.,* 2013; Yu *et al.,* 2017). To solve the domain adaptation problem, it is important to select an appropriate method considering the available data and the given type of dataset shift.

2.2.2. Concept and Examples of Dataset Shift

The dataset shift appears when distributions of source domain and target domain are different (Moreno-Torres *et al.,* 2012). In a classification problem, a set of features and a class variable can be defined by X and Y, respectively. Then, dataset shift refers to the case where $P_s(X,Y) \neq P_t(X,Y)$ (the subscript 's' and 't' represent the source domain and the target domain, respectively). This shift can be interpreted in terms of $P(X)$ and $P(Y)$, which correspond to the two respective shifts: *covariate shift* and *prior probability shift*. Covariate shift specifically refers to the case where the distribution $P(X)$ changes between the two domains but $P(Y|X)$ is the same. On the other hand, the prior probability shift is the reverse case of the covariate shift. The prior probability shift refers to the case where the distribution $P(Y)$ changes but $P(X|Y)$ is the same in the two domains.

In remote sensing, the dataset shift is very common and occurs in various forms with intertwined manners. This occurs when the images to be dealt with share common characteristics, but they were acquired at different times, from different sensors or from different areas (Karpatne *et al.,* 2016; Bovolo and Bruzzone, 2015). To elaborate in detail, in case of land cover mapping, if the images for land cover classification were acquired at different times, the joint distribution between spectral bands

and land cover class could be different among the images (Bovolo and Bruzzone, 2015; Gomez *et al.,* 2016). Thus, a trained classifier in each image may not work well for the other images acquired at different periods. Moreover, although images were taken in the same area at the same time, the dataset shift can occur if the sensors were different. Since retrieved digital numbers from different sensors might differ due to the difference in view angles and spectral response functions (Huang *et al.,* 2017), the optimal classification boundaries set by each image might differ too. Furthermore, the dataset shift can occur among different areas due to the difference in feature characteristics (Zhu and Woodcock, 2014) and the class distribution (Mellor *et al.,* 2015; Maxwell *et al.,* 2018). In addition to the aforementioned factors, other various factors such as illumination conditions and atmospheric conditions can cause dataset shifts. These dataset shifts usually occur simultaneously and correlate with each other. Therefore, domain adaptation of proper strategy is required with careful consideration of the type of dataset shift and the given condition.

2.2.3. Prior Probability Shift in Change Detection

The dataset shift commonly occurs in change detection. However, the vast majority of change detection algorithms have not considered their performance in inter-domains, and in turn, failed to generalize their performance under the dataset shift. Thus, in this study, domain adaptation for change detection problem under the dataset shift is addressed. Specifically, among the dataset shifts, this study concentrated on the prior probability shift that occurs among spatially disjoint areas acquired in the same time zone. This is because, in remote sensing, the rate of change varies locally in large-scale remotely sensed imagery (Hansen and Loveland, 2012; Zhu and Woodcock, 2014), thus the prior probability shift was considered to be prevalent in change detection problem. For example, urban development does not change at the same rate over the entire area, but the rate of change varies considerably from region to region. Likewise, changes caused by disasters also vary greatly in each region. In this situation, obviously, the prior probability shift occurs and severely degrades the change detection performance. Thus, it is assumed that the different prior probabilities among the domains could be the decisive factor of the dataset shift in change detection. However, only a few studies have considered the prior probability shift in change detection (Peng *et al.,* 2019; Zerrouki *et al.,* 2019).

Figure 3 illustrates examples of the prior probability shifts in two-dimensional feature spaces. The samples of input variables X are represented as points in the feature spaces with the two different class Y_1 and Y_2. The goal

of the change detection algorithm is to find an optimal decision boundary to identify whether the incoming data change or not. The class conditional probability density of each class is the same in all domain, i.e., $P_s(X|Y_i) = P_{t1}(X|Y_i) = P_{t2}(X|Y_i)$, where $i = 1, 2$. However, the prior probability of each class is different in all domains. The prior probability ratio for each class is 50:50 in the source domain, whereas the ratios of Y_1 to Y_2 are 20:80 in the target domain 1 and 80:20 in the target domain 2, i.e., $P_s(Y_i) \neq P_{t1}(Y_i)$ and $P_s(Y_i) \neq P_{t2}(Y_i)$. As a result, the joint probabilities of the source and target domain are different, i.e., $P_s(X|Y_i)P_s(Y_i) \neq P_{t1}(X|Y_i)P_{t1}(Y_i)$ and $P_s(X|Y_i)P_s(Y_i) \neq P_{t2}(X|Y_i)P_{t2}(Y_i)$. Therefore, the decision boundary in the source domain should be adjusted to attain the optimal result in the target domain. This problem is of crucial importance since the rate of change greatly differs in remotely sensed imagery.

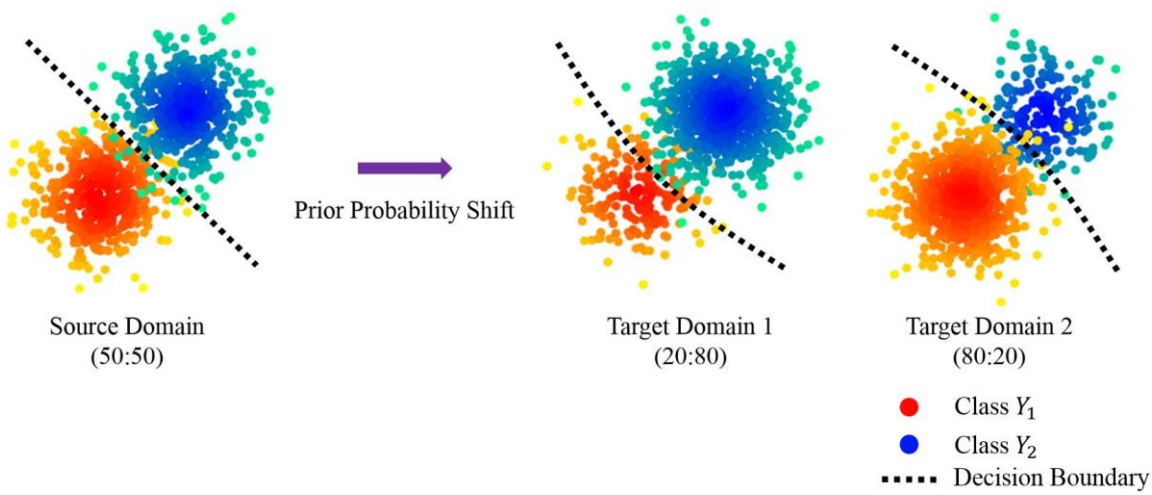

Figure 3. Examples of the Prior Probability Shift in Binary Classification

3. The Proposed Change Detection Framework

Since only the prior probability shift was considered in this thesis, the experimental setup supposed the case where the proportions of the "change class" (referred to hereafter as *change ratio*) vary among spatially disjoint areas acquired in the same time zone. In addition, it is assumed the training sample of the target domain is not available. Thus, among the domain adaptation methodology illustrated in Subsection 2.2.1, it was determined that the selection of the third category method, where the trained model is adapted to the target domain, was thought to be desirable.

In short, the proposed framework adapted the trained model to the target domain considering its prior probability estimated by an unsupervised algorithm. CNN was adopted as a training model and the patch-based CVA was used for estimating the prior probability of the target domain. The proposed framework estimates the change ratio of the target domain by the patch-based CVA and adjusts the decision boundary by ranking the softmax output of CNN to adapt well to the target domain. The proposed framework was examined under various conditions of prior probability shift with real-world data, bi-temporal Unmanned Aerial Vehicle (UAV) imagery with 0.1-m resolution including RGB channels and Digital Surface Model (DSM).

The rest of this Section 3 is organized as follows: first, in Subsection

3.1., the workflow of the proposed domain adaptation framework is briefly described. Subsection 3.2. and 3.3. explain how the framework estimates the prior probability of the target domain and identifies the changes in detail, respectively.

3.1. Workflow

The flowchart of the proposed framework can be summarized as follows (Figure 4).

1) Let the CNN trained from the source domain infer the target domain and rank the probabilities of change for each input derived from the softmax output of the trained CNN.

2) Calculate the threshold that maximizes the overall accuracy in the source domain using the patch-based CVA on imagery.

3) Apply the obtained threshold to the target domains and estimate the change ratio of the target domain.

4) Assign the changed pixel in order of the highest probability derived from the CNN, assuming that the change in the target domain has been made by the change ratio estimated by CVA.

5) Then, a change detection map can be generated for every pixel in the target domain.

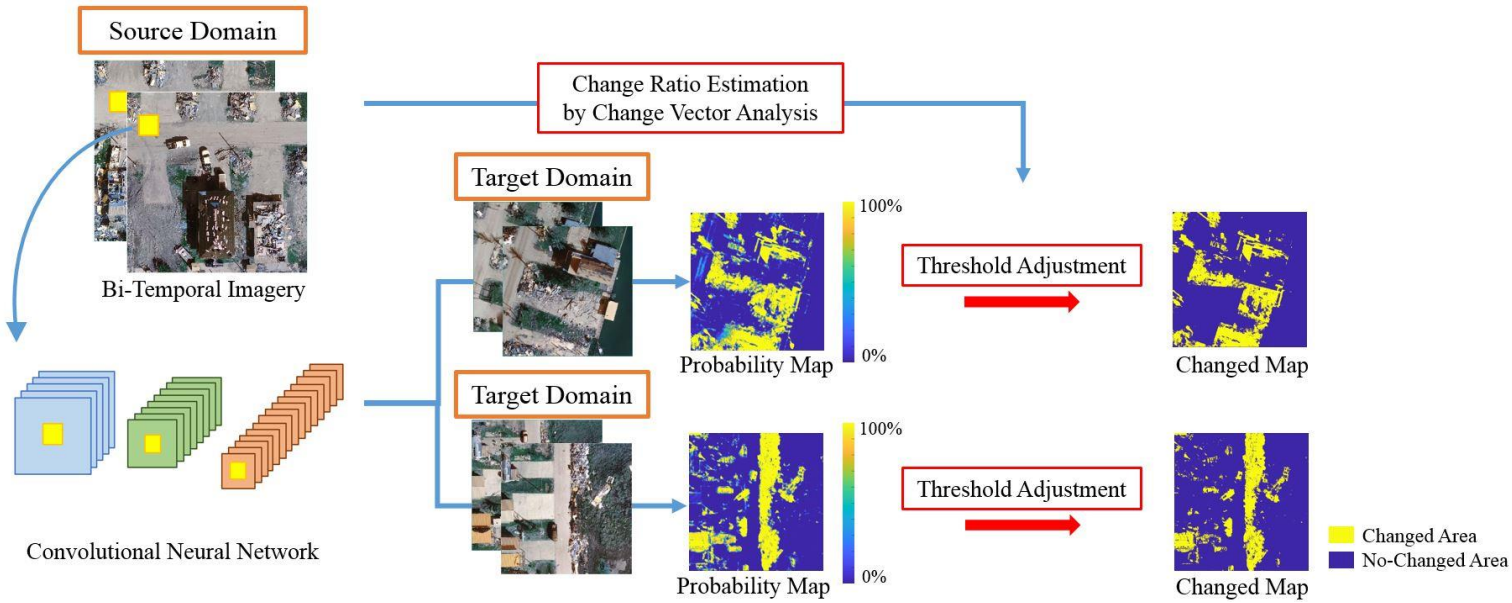

Figure 4. Overview of the Proposed Domain Adaptation Framework

3.2. The Prior Probability Estimation

3.2.1. Patch-based Change Vector Analysis

In this experiment, a patch-wise performing CVA was adopted and only its magnitude was considered to identify changed pixels because the interest is not the identification of the type of change but the estimation of change ratio. Since a pixel-based CVA is liable to illumination variations, which yield salt and pepper noises, a patch-based CVA was considered to be better to estimate the change ratio accurately by averaging the noise effects. The patch size was determined to be 9 pixels by 9 pixels, approximately covering the area of 1-m by 1-m in this experimental area, in order to find the optimal threshold. The magnitude $M(i,j)$ of the patch-based CVA was calculated by the following formula (2):

$$M(i,j) = \sum_m \sum_n \sum_b \left(X_b(i+m, j+n) - X'_b(i+m, j+n) \right) \quad (2)$$

where (i,j) is the pixel coordinate of the image, and $m,n = -4,-3,-2,-1,\ 0,\ 1,\ 2,\ 3,\ 4$. X_b and X'_b are the digital number of the b^{th} band of bi-temporal imagery. With the patch-based CVA, the estimation of target domain can be performed because CVA does not require a training sample, and it is more insensitive to the distribution shift than a CNN.

3.2.2. Estimation of Change Ratio with Patch-based CVA

The method for estimating the change ratio of the target domain via the patch-based CVA is as follows. First, find the optimal magnitude of the patch-based CVA that maximizes the overall accuracy in the source domain. To automatically find the optimal magnitude in this process, the samples which will be used to train CNN was used to empirically find the optimal threshold that yields the highest overall accuracy in the source domain. Then, the threshold maximizing overall accuracy in the source domain is directly applied to the target domain to estimate its change ratio. In this process, for RGB-D imagery, a weight of three was assigned to DSM compared to the weight of one assigned to each RGB band. This is because the DSM information is more resilient to the illumination variations than the RGB channel's spectral information in general (Tian, 2013).

3.3. Changed Pixel Identification

3.3.1. Lightweight Convolutional Neural Network

In this thesis, the lightweight CNN that has low computational cost is used due to its much lighter design compared to conventional CNNs for image classification tasks. This is why it is named as “lightweight” CNN. Most CNNs have a deep and complex network with a large number of trainable parameters because the distinctive advantage of CNN is its ability to deal with big and intricate data (Canziani *et al.,* 2016). However, models should be designed to find the proper balance between the capacity and the complexity to avoid overfitting but maintain sufficient performance. Considering the given task of performing a binary classification with a small size of patch as an input, a lightweight CNN was considered to have enough complexity. Plus, it does not require a large number of training samples for the network to perform well without overfitting due to its low complexity (He *et al.,* 2016; Song *et al.,* 2019).

The adopted lightweight CNN consists of only two convolutional layers, and each layer has 16 and 32 filters, respectively. The filter size of the convolutional layers was 3 by 3. The CNN takes a three-dimensional patch as input. The patch size of the CNN was also determined to be 9 pixels by 9 pixels, same as the patch-based CVA. Two images, acquired at different times, were stacked to conserve the full information in original

images and binary classification was performed as widely adopted in supervised change detection (Volpi *et al.,* 2013; Daudt *et al.,* 2018). As two sets of stacked bi-temporal RGB imagery and RGB-D imagery were used, the CNN takes the input either with a size of 9 by 9 by 6 for stacked RGB imagery or with a size of 9 by 9 by 8 for stacked RGB-D imagery. After taking this input, each CNN learns the decision boundary to solve a binary problem. Then, the trained CNN can produce the probability of whether the center pixel of the patch has changed or not with the softmax output (Alshehhi *et al.,* 2017; Xu *et al.,* 2018). The pooling layer, which reduces the dimensions of the data by down-sampling the input, was not used because the lightweight CNN takes a small size of patch as an input. Also, the fully-connected layer is excluded to lower the network complexity. Rectified Linear Unit (ReLU) and the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.001 were used.

3.3.2. Assignment of Changed Pixel

The baseline threshold of change in CNN is 50.0%. Therefore, if the softmax output of the CNN is greater than 50.0%, CNN determines it as a change whether or not CNN experienced a dataset shift. Despite the failure of CNN to produce accurate posterior probability, the rank of the probabilities of change, inferred from the softmax output in the CNN, is still valid provided only a prior probability shift has occurred. Therefore, it was assumed that the dataset shift problem can be resolved by estimating the prior probability in the target domain from the supplementary information produced from the source domain.

Based on the logic presented above, the proposed framework lets CNN identify the changed pixels overcoming the prior probability shift problem. To elaborate, first, the change ratio of the target domain is estimated as described in Subsection 3.2.2. Then, the trained lightweight CNN, illustrated in Subsection 3.3.1., infers the target domain and produces the probabilities of change using the softmax layer. Subsequently, it is determined whether the sample has changed in the order of the highest probability under the assumption that the target domain has changed by the estimated change ratio using the patch-based CVA. With this process, it is possible to adjust the biased threshold of CNN under prior probability shift, thereby identifying the changed pixels.

Figure 5 illustrates two examples of threshold adjustments. The histograms on the left and right represent the cases where the change ratio of the target domain is higher and lower than the source domain, respectively. In the case of the left histogram, CNN underestimated the probability of change for samples in the target domain because of the prior probability shift. This case is highly likely to occur when the change ratio in the source domain is smaller than that of the target domain. Therefore, the threshold should be adjusted to less than 50.0% to compensate for the underestimation. As for the opposite case, the histogram on the right illustrates where the threshold should be adjusted to a value larger than 50.0%, which can occur when the CNN overestimates the probability of change, from training in the source domain where the change ratio is higher than that of the target domain.

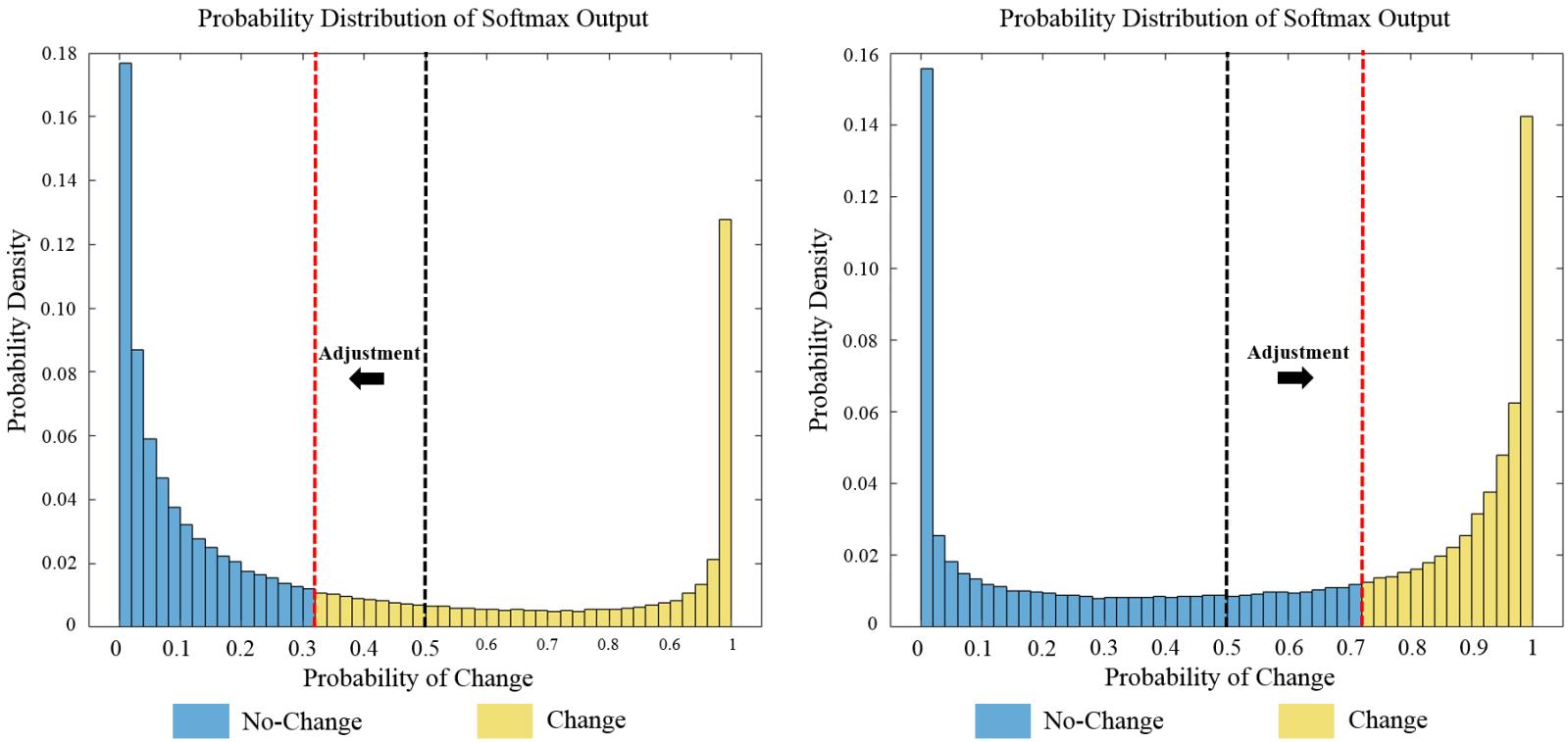

Figure 5. Examples Explaining Threshold Adjustments

4. Experimental Design

4.1. Experimental Area

To evaluate the proposed domain adaptation framework, the real-world data acquired from UAV imagery was used. The experimental area, a Holiday Beach in Rockport in southern Texas, had experienced significant changes in buildings and vegetation during the recovery from the damage of Hurricane Harvey (Figure 6). Since changes had taken place in very large areas and the change ratio varies locally, it is laborious to obtain high-quality training samples in the full range. Therefore, to perform change detection efficiently, the domain adaptation that can fully use collected training samples is required.

Bi-temporal UAV images were processed to generate the orthomosaic RGB imagery and DSM using the Structure from Motion (SfM) algorithm (Snavely *et al.,* 2006; Westoby *et al.,* 2012) in Agisoft Photoscan Pro software. Three sub-regions (i.e., A, B, and C) were clipped and ground truths were acquired (Figure 6). Also, the three sub-regions were down-sampled to 0.1-m resolution to lower the amount of computations. As a result, three sub-regions A, B, and C consisted of 500 by 500 pixels each and had different ratios of the changed areas (i.e., 20.0 %, 45.0 %, and 35.0 %), respectively.

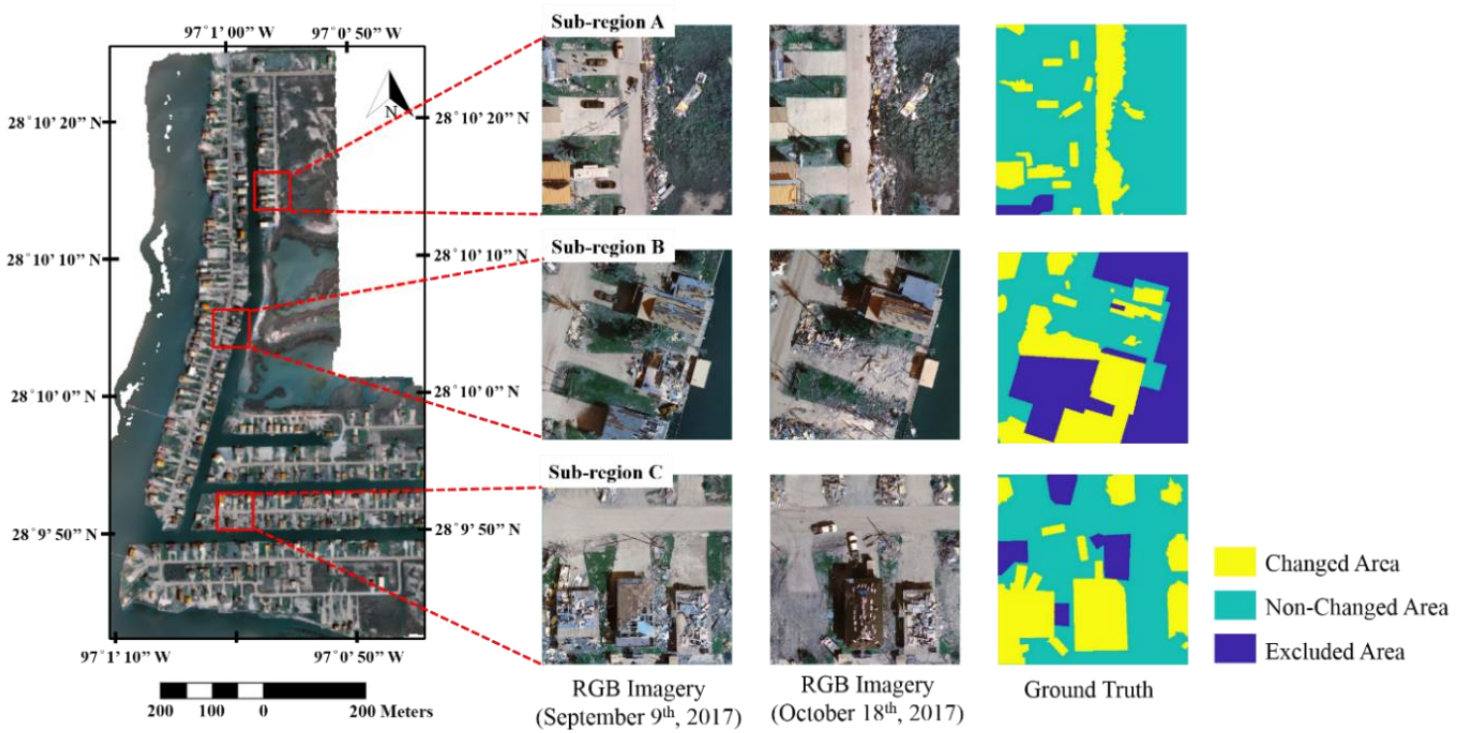

Figure 6. Experimental Area

4.2. Data Pre-processing

Water bodies were excluded from the experiment because orthomosaic images and DSM over the water bodies are generally of poor quality and not reliable. Several areas scattered with small, insignificant objects were also excluded because it is difficult to identify whether they had been changed. All values in RGB and DSM were rescaled to the range of [0, 1] by min-max normalization before the operation of the CNN and CVA. Since both bi-temporal images were taken under clear sky, no other radiation correction was performed.

4.3. Sampling and Evaluation Design

Basically, the CNN was trained using the training samples drawn from one of each three sub-regions (source domain). Subsequently, the trained CNN was adapted and tested on the two other sub-regions (target domain). Therefore, the proposed framework was assessed on six dataset shifts (i.e., from sub-region A to B and C, from sub-region B to A and C, and from sub-region C to A and B) that have different change/no-change ratios. In addition, two different sizes of training samples, a total of 1,000 and 2,000 samples, respectively, were collected for every case. The training samples were randomly sampled from the source domain. Then, all pixels in the target domain were tested. Also, the experiment was performed on both RGB and RGB-D images. The implementation of CNN was supported by Google's Tensorflow framework (Pedregosa *et al.,* 2011).

First, the experiment was performed by a sampling method called "base sampling design". The training samples were drawn from each class with the same proportion to the class ratio of each sub-region. For example, suppose the case of dataset shifts from sub-region A (change ratio: 20.0%) to sub-regions B and C. If 1,000 training samples are selected from sub-region A, this then produces 200 samples for the change class and 800 samples for the no-change class to train the CNN model. Subsequently, trained model is applied on sub-regions B and C to identify changed pixels.

Also, the training and test samples are three-dimensional patches as 9 by 9 by 6 for RGB imagery and as 9 by 9 by 8 for RGB-D imagery, respectively, and their classes are defined by the class of the center pixel.

In addition, other sampling method was also tested. This sampling design was referred to as "additional sampling design". Training samples composed of three different change/no-change ratios of 25.0%, 50.0%, and 75.0% from each of the three sub-regions were collected. Since the change ratios of the three sub-regions A, B, and C were 20.0%, 45.0%, and 35.0%, respectively, the differences in the change ratios among the three sub-regions vary from 10.0% to 25.0% in the base sampling design. For real-world application, however, the variation may be more drastic. Therefore, to confirm that the proposed framework works well under more severe prior probability shift cases, three more training sample sets (i.e., additional sampling design) with different change/no-change ratios from each sub-region were considered. Similar to the case of base sampling design, the experiments were performed on both RGB and RGB-D images, and the CNN was trained with two different training sample sizes (i.e., a total of 1,000 and 2,000 samples).

To sum up, the proposed framework was evaluated with the two sets of sampling design (i.e., the base sampling design and the additional sampling design) and the two different training sample sizes (i.e., a total of 1,000 and 2,000 samples). Additionally, the proposed framework was applied to both RGB and RGB-D imagery.

5. Experimental Results and Discussion

5.1. Domain Adaptation Results

5.1.1. Base Sampling Design

In the base sampling design, the CNN was trained with the training samples consisting of the same proportion to the class ratio of each sub-region, and were then tested on other remaining sub-regions. To evaluate the performance of the proposed domain adaptation framework, the overall accuracy (OA) before and after applying the proposed framework was compared. OA is calculated by dividing the sum of the number of correct classifications by the total number of samples. The results of OA from the six dataset shifts (i.e., from A to B and C, from B to A and C, and from C to A and B) with the two different training sample sizes in both RGB and RGB-D imagery are shown in Table 1. To increase the statistical confidence, random sampling was repeated 10 times in all cases and all values in Table 1 represent their average values.

In both two sets of images, the proposed framework increases the accuracy except for few cases. For RGB imagery, the proposed framework increased the mean OA from 74.68% to 76.24% for a training sample size of 1,000, and from 75.82% to 78.19% for a training sample size of

2,000. For RGB-D imagery, the proposed framework increased the mean OA from 75.0% to 78.8% for a training sample size of 1,000, and from 75.23% to 79.85% for a training sample size of 2,000.

Note that the mean OA with the training sample size of 1,000 after the domain adaptation is higher than the mean OA without domain adaptation using the training sample size of 2,000. It indicates that the proposed domain adaptation was more effective than doubling the training sample sizes. Moreover, doubling the training size increased the mean OA only by 0.9% when domain adaptation was not conducted, while the mean OA increased by 1.9% after domain adaptation. These results show that deep learning models reach their limit in performance easily when simply trained with a large amount of data, and thus confirm that domain adaptation is compulsory and effective to further improve the performance of deep learning.

Table 1-1. Comparison of OA with Base Sampling Design (RGB, training sample size = 1,000)

	w/o Domain Adaptation			w/ Domain Adaptation		
Source \ Target	A	B	C	A	B	C
A		0.6921	0.7259		0.7171	0.7319
B	0.7721		0.7413	0.8172		0.7415
C	0.8029	0.7466		0.8148	0.7519	
Mean OA		**0.7468**			**0.7624**	

OA: Overall Accuracy

Table 1-2. Comparison of OA with Base Sampling Design (RGB, training sample size = 2,000)

	w/o Domain Adaptation			w/ Domain Adaptation		
Source \ Target	A	B	C	A	B	C
A		0.7295	0.7399		0.7719	0.7577
B	0.7502		0.7517	0.8164		0.7531
C	0.8176	0.7602		0.8281	0.7645	
Mean OA		**0.7582**			**0.7819**	

OA: Overall Accuracy

Table 1-3. Comparison of OA with Base Sampling Design (RGB-D, training sample size = 1,000)

	w/o Domain Adaptation			w/ Domain Adaptation		
Source \ Target	A	B	C	A	B	C
A		0.7470	0.7402		0.7882	0.7582
B	0.7321		0.7570	0.8112		0.7574
C	0.7455	0.7781		0.8342	0.7787	
Mean OA		**0.7500**			**0.7880**	

OA: Overall Accuracy

Table 1-4. Comparison of OA with Base Sampling Design (RGB-D, training sample size = 2,000)

	w/o Domain Adaptation			w/ Domain Adaptation		
Source \ Target	A	B	C	A	B	C
A		0.7724	0.7628		0.8041	0.7784
B	0.7335		0.7788	0.8217		0.7772
C	0.6896	0.7768		0.8349	0.7749	
Mean OA		**0.7523**			**0.7985**	

OA: Overall Accuracy

5.1.2. Additional Sampling Design

The additional sampling design acquires training samples of 25.0%, 50.0%, and 75.0% of the total sample from each of the three sub-regions. As a result, experiments were carried out not only when the “change class” had the same ratio as the “no-change class”, but also when the “change class” accounted for both the majority case (75.0%) and the minority case (25.0%). Every case was replicated five times using random sampling to increase the statistical confidence of the results.

Table 2 reports the improvement of mean OA after the proposed framework was applied, shown together with its increasing rate. The proposed framework successfully improved the accuracy in all additional sampling scenarios regardless of training sample sizes in both images. As the change ratio increases to 25.0%, 50.0%, and 75.0%, the mean OA decreases significantly. This is because the “change class” accounts for about 33.3% in the three sub-regions on average. Thus, it confirmed that the CNN fails to produce accurate decision boundary as the prior probability gap between the two domains widens. Moreover, as the change ratio of the training samples increases, the increasing rate continuously increases overall. This indicates that the domain adaptation is particularly required when the prior probability shift is severe and the proposed framework worked as well at varying degrees of prior probability shift.

Table 2-1. Comparison of Mean OA and Its Increasing Rate (RGB, training sample size =1,000)

Change vs No-Change	25% vs 75%		50% vs 50%		75% vs 25%	
Domain Adaptation?	No	Yes	No	Yes	No	Yes
Mean OA	75.29%	77.04%	74.43%	76.51%	68.73%	72.10%
Increasing Rate	2.31%		2.78%		4.91%	

OA: Overall Accuracy

Table 2-2. Comparison of Mean OA and Its Increasing Rate (RGB, training sample size =2,000)

Change vs No-Change	25% vs 75%		50% vs 50%		75% vs 25%	
Domain Adaptation?	No	Yes	No	Yes	No	Yes
Mean OA	77.42%	79.39%	74.87%	78.01%	69.28%	73.72%
Increasing Rate	2.55%		4.18%		6.41%	

OA: Overall Accuracy

Table 2-3. Comparison of Mean OA and Its Increasing Rate (RGB-D, training sample size =1,000)

Change vs No-Change	25% vs 75%		50% vs 50%		75% vs 25%	
Domain Adaptation?	No	Yes	No	Yes	No	Yes
Mean OA	78.58%	80.17%	73.06%	78.39%	67.77%	72.66%
Increasing Rate	2.03%		7.30%		7.21%	

OA: Overall Accuracy

Table 2-4. Comparison of Mean OA and Its Increasing Rate (RGB-D, training sample size =2,000)

Change vs No-Change	25% vs 75%		50% vs 50%		75% vs 25%	
Domain Adaptation?	No	Yes	No	Yes	No	Yes
Mean OA	76.84%	81.13%	75.08%	78.97%	68.10%	74.74%
Increasing Rate	5.58%		5.18%		9.75%	

OA: Overall Accuracy

5.2. Estimated Change Ratio and Adjusted Threshold

In this Section, the specific results of each step in the proposed framework were investigated. The estimated change ratio, adjusted threshold, and the final difference in accuracy after application of the proposed framework were shown in Table 3. The results from a total of six dataset shifts with the base sampling design were reported.

In Table 3, when the source domain is sub-region A, the CNN learned its decision boundary to optimize the dataset in this particular source domain with changed samples accounting for 20.0% of the total sample size. As a result, the CNN underestimated the change ratio of B and C where the actual change ratio is 45.0% and 35.0%, respectively. After domain adaptation, however, the underestimations were compensated close to the actual value. The baseline threshold is also lowered below 50.0%.

On the contrary, when CNN was trained in sub-region B or C (source domain) and tested on sub-region A (target domain), the change ratio of sub-region A is prone to be overestimated than its actual change ratio of 20.0%. In this case too, the estimated change ratio was adjusted after the domain adaptation. The threshold decreased from the default value of 50.0% to compensate for the underestimation of changed samples, while the threshold increased to compensate for the overestimation. Furthermore,

mean absolute error (MAE), an average of the absolute errors between the estimated change ratio and the true change ratio, was calculated for each of the six dataset shift. As a result, it can be confirmed that the MAE is clearly reduced after the domain adaptation. A few exceptions adversely estimating the change ratio were also made, especially between B and C where the difference of change ratio was not relatively large. However, in most cases, the prior probabilities of the target domain were effectively adjusted through the proposed domain adaptation framework.

Another noteworthy point in the results is that MAE increases as the training sample size gets bigger and additional information (i.e., DSM) was added. This means that deep learning models can fail in spite of richer data, if the data in the source domain do not represent well the data in the target domain. This tendency is known to be explicit as the deep learning model becomes more complex (Guo *et al.,* 2017; Buda *et al.,* 2018). Considering the complexity of the lightweight CNN used in this experiment, the decline in performance without domain adaptation may be more severe with a deeper and more complex network. These results indicate that the application of domain adaptation is crucial, not only for complex networks, but also when using large amounts of input data.

Table 3-1. Difference in Change Ratio, Threshold, and OA after the Proposed Framework was Applied (RGB, training sample=1,000)

			Change Ratio		Threshold		OA	
	Domain Adaptation?		No	Yes	No	Yes	No	Yes
	Source	Target						
RGB	A (20%)	B (45%)	0.2444	0.3814	0.5000	0.1712	0.6921	0.7171
		C (35%)	0.2427	0.3135		0.3067	0.7259	0.7319
	B (45%)	A (20%)	0.3453	0.2855		0.6807	0.7721	0.8172
		C (35%)	0.4034	0.4179		0.4427	0.7413	0.7415
	C (35%)	A (20%)	0.2865	0.2752		0.4928	0.8029	0.8148
		B (45%)	0.4047	0.4762		0.3026	0.7466	0.7519
	Mean Absolute Error		**0.1072**	**0.0600**	Mean OA		**0.7468**	**0.7624**

OA: Overall Accuracy

Table 3-2. Difference in Change Ratio, Threshold, and OA after the Proposed Framework was Applied (RGB, training sample=2,000)

			Change Ratio		Threshold		OA	
	Domain Adaptation?		No	Yes	No	Yes	No	Yes
	Source	Target						
RGB	A (20%)	B (45%)	0.2549	0.3636	0.5000	0.1920	0.7295	0.7719
		C (35%)	0.2439	0.2963		0.3378	0.7399	0.7577
	B (45%)	A (20%)	0.3813	0.2961		0.7352	0.7502	0.8164
		C (35%)	0.4267	0.4335		0.4924	0.7517	0.7531
	C (35%)	A (20%)	0.2791	0.2629		0.5436	0.8176	0.8281
		B (45%)	0.4065	0.4583		0.3740	0.7602	0.7645
	Mean Absolute Error		**0.1136**	**0.0652**	Mean OA		**0.7582**	**0.7819**

OA: Overall Accuracy

Table 3-3. Difference in Change Ratio, Threshold, and OA after the Proposed Framework was Applied (RGB-D, training sample=1,000)

			Change Ratio		Threshold		OA	
	Domain Adaptation?		No	Yes	No	Yes	No	Yes
	Source	Target						
RGB-D	A (20%)	B (45%)	0.2730	0.3715	0.5000	0.1466	0.7470	0.7882
		C (35%)	0.2025	0.3041		0.1333	0.7402	0.7582
	B (45%)	A (20%)	0.3858	0.2812		0.7318	0.7321	0.8112
		C (35%)	0.4079	0.4121		0.4821	0.7570	0.7574
	C (35%)	A (20%)	0.3809	0.2647		0.7121	0.7455	0.8342
		B (45%)	0.4546	0.4607		0.4500	0.7781	0.7787
	Mean Absolute Error		**0.1256**	**0.0572**	Mean OA		**0.7500**	**0.7880**

OA: Overall Accuracy

Table 3-4. Difference in Change Ratio, Threshold, and OA after the Proposed Framework was Applied (RGB-D, training sample=2,000)

			Change Ratio		Threshold		OA	
	Domain Adaptation?		No	Yes	No	Yes	No	Yes
	Source	Target						
RGB-D	A (20%)	B (45%)	0.2899	0.3670	0.5000	0.2477	0.7724	0.8041
		C (35%)	0.2224	0.2994		0.2379	0.7628	0.7784
	B (45%)	A (20%)	0.3954	0.2779		0.7689	0.7335	0.8217
		C (35%)	0.4104	0.4065		0.5276	0.7788	0.7772
	C (35%)	A (20%)	0.4499	0.2661		0.7806	0.6896	0.8349
		B (45%)	0.4531	0.4624		0.4656	0.7768	0.7749
	Mean Absolute Error		**0.1327**	**0.0577**	Mean OA		**0.7523**	**0.7985**

OA: Overall Accuracy

6. Conclusion

6.1. Summary

In this thesis, a domain adaptation framework for deep learning based change detection in remotely sensed data under prior probability shift is proposed. The most widely used deep learning model and unsupervised algorithm, CNN and CVA, respectively, were adopted to confirm the performance of the proposed framework. The proposed framework was evaluated with the two sets of bi-temporal remotely sensed data, RGB and RGB-D, including two different sampling designs and two different training sample sizes, to validate the effectiveness of the proposed method in various conditions. The experiments demonstrated that the proposed framework successfully adjusts the biased threshold and improves the change detection performance.

In addition, the susceptibility of deep learning under the prior probability shift was verified. Also, it is confirmed that the adjustment of biased softmax output can prevent performance degradation. Furthermore, experiments on both RGB and RGB-D images with the two different sample sizes showed that the domain adaptation was particularly required when deep learning models are pre-trained on greater amount of data with richer content.

6.2. Concluding Remarks and Future Works

With the increase in the amount of remotely sensed data, previous studies excessively focused on designing deep complex networks and training the models with massive data. However, the results of this thesis indicate that deep learning with a large amount of training data has obvious limitations in performance and its performance can be improved with domain adaptation. This study showed that domain adaptation is mandatory to fully utilize the acquired training samples and to process a large amount of data effectively. In addition, this study focused on the prevailing problem of prior probability shift in change detection. Changes in remotely sensed imagery occur globally and differ regionally. Thus, the proposed framework being capable of resolving the prior probability shift problem in change detection has advantages. Last but not least, the proposed framework is not limited to CNN and CVA, but is applicable to a wide variety of classifiers. The use of this framework with the combination of other supervised algorithms being able to provide classification results as a probability and other unsupervised algorithms being able to estimate the change ratios can further increase the change detection accuracy.

Several findings in this thesis can be extended. A lightweight CNN model and CVA were adopted in this experiment. However, other deep

learning models for change detection, namely, recurrent neural network and fully convolutional network, or unsupervised algorithm implemented with clustering fashion could also be adopted. Moreover, to clarify and address the impact of the prior probability shift, the experiments in this thesis supposed that only the prior probability shift has occurred. However, the covariate shift could be also involved even though the three sub-regions used in this experiment were taken at the same time zone with the same sensor. Lastly, the proposed framework was developed for the case when the trained model is directly applied to other domains. However, other methodologies mentioned in Subsection 2.2.1 can be jointly used and further improve the change detection performance if the additional information is available.

References

Alshehhi, R., Marpu, P. R., Woon, W. L., & Dalla Mura, M., (2017). Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing, 130,* 139-149.

Bahirat, K., Bovolo, F., Bruzzone, L., & Chaudhuri, S., (2011). A novel domain adaptation Bayesian classifier for updating land-cover maps with class differences in source and target domains. *IEEE Transactions on Geoscience and Remote Sensing, 50*(7)*,* 2810-2826.

Bovolo, F., & Bruzzone, L., (2006). A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain. *IEEE Transactions on Geoscience and Remote Sensing, 45*(1)*,* 218-236.

Bovolo, F., & Bruzzone, L., (2015). The time variable in data fusion: A change detection perspective. *IEEE Geoscience and Remote Sensing Magazine, 3*(3)*,* 8-26.

Bruzzone, L., & Persello, C., (2009). A novel approach to the selection of spatially invariant features for the classification of hyperspectral images with improved generalization capability. *IEEE Transactions on Geoscience and Remote Sensing, 47*(9), 3180-3191.

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks, 106,* 249-259.

Byun, Y., Choi, J., & Han, Y., (2013). An area-based image fusion scheme for the integration of SAR and optical satellite imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6*(5), 2212-2220.

Canziani, A., Paszke, A., & Culurciello, E., (2016). An analysis of deep neural network models for practical applications. arXiv preprint arXiv:1605.07678.

Chang, A., Jung, J., & Kim, Y., (2015). Estimation of forest stand diameter class using airborne lidar and field data. *Remote Sensing Letters, 6*(6), 419-428.

Crawford, M. M., Tuia, D., & Yang, H. L., (2013). Active learning: Any value for classification of remotely sensed data?. Proceedings of the IEEE, 101(3), 593-608.

Daudt, R. C., Le Saux, B., Boulch, A., & Gousseau, Y. (2018, July). Urban change detection for multispectral earth observation using convolutional neural networks. In IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, 2115-2118.

Foody, G. M. (2002). Status of land cover classification accuracy assessment. *Remote Sensing of Environment, 80*(1)*,* 185-201.

Gomez, C., White, J. C., & Wulder, M. A. (2016). Optical remotely sensed time series data for land cover classification: A review. *ISPRS Journal of Photogrammetry and Remote Sensing, 116,* 55-72.

Gong, M., Zhan, T., Zhang, P., & Miao, Q. (2017). Superpixel-based difference representation learning for change detection in multispectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing, 55*(5)*,* 2658-2673.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017, August). On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (1321-1330). JMLR. org.

Hansen, M. C., & Loveland, T. R., (2012). A review of large area monitoring of land cover change using Landsat data. *Remote Sensing of Environment, 122,* 66-74.

He, K., Zhang, X., Ren, S., & Sun, J., (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.

Heydari, S. S., & Mountrakis, G., (2019). Meta-analysis of deep neural networks in remote sensing: A comparative study of mono-temporal classification to support vector machines. *ISPRS Journal of Photogrammetry and Remote Sensing, 152,* 192-210.

Huang, X., Wen, D., Li, J., & Qin, R. (2017). Multi-level monitoring of subtle urban changes for the megacities of China using high-resolution multi-view satellite imagery. *Remote Sensing of Environment, 196,* 56-75.

Hussain, M., Chen, D., Cheng, A., Wei, H., & Stanley, D., (2013). Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing, 80,* 91-106.

Jin, H., Stehman, S. V., & Mountrakis, G. (2014). Assessing the impact of training sample selection on accuracy of an urban classification: a case study in Denver, Colorado. *International Journal of Remote Sensing, 35*(6)*,* 2067-2081.

Jung, J., Pasolli, E., Prasad, S., Tilton, J. C., & Crawford, M. M., (2014). A framework for land cover classification using discrete return LiDAR data: Adopting pseudo-waveform and hierarchical segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7*(2)*,* 491-502.

Karpatne, A., Jiang, Z., Vatsavai, R. R., Shekhar, S., & Kumar, V. (2016). Monitoring land-cover changes: A machine-learning perspective. *IEEE Geoscience and Remote Sensing Magazine, 4*(2)*,* 8-21.

Khan, S. H., He, X., Porikli, F., & Bennamoun, M. (2017). Forest change detection in incomplete satellite images with deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing, 55*(9), 5407-5423.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (1097-1105).

LeCun, Y., Bengio, Y., & Hinton, G., (2015). Deep learning. *Nature, 521*(7553), 436.

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

Liu, J., Gong, M., Qin, K., & Zhang, P. (2016). 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.

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing, 152,* 166-177.

Maxwell, A. E., Warner, T. A., & Fang, F. (2018). Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing, 39*(9)*,* 2784-2817.

Mellor, A., Boukir, S., Haywood, A., & Jones, S. (2015). Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin. *ISPRS Journal of Photogrammetry and Remote Sensing, 105,* 155-168.

Moreno-Torres, J. G., Raeder, T., Alaiz-RodríGuez, R., Chawla, N. V., & Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition, 45*(1)*,* 521-530.

Mou, L., Bruzzone, L., & Zhu, X. X., (2018). Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Transactions on Geoscience and Remote Sensing, 57*(2)*,* 924-935.

Pan, S. J., & Yang, Q., (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345-1359.

Patel, V. M., Gopalan, R., Li, R., & Chellappa, R., (2015). Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine, 32*(3), 53-69.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. & Vanderplas, J., (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*(Oct), 2825-2830.

Peng, D., Zhang, Y., & Guan, H. (2019). End-to-End Change Detection for High Resolution Satellite Images Using Improved UNet++. *Remote Sensing, 11*(11), 1382.

Persello, C., & Bruzzone, L., (2014). Active and semisupervised learning for the classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing, 52*(11), 6937-6956.

Qin, R., (2014). Change detection on LOD 2 building models with very high resolution spaceborne stereo imagery. *ISPRS Journal of Photogrammetry and Remote Sensing, 96,* 179-192.

Qin, R., Tian, J., & Reinartz, P., (2016). 3D change detection–approaches and applications. *ISPRS Journal of Photogrammetry and Remote Sensing, 122,* 41-56.

Rahman, M., (2016). Detection of land use/land cover changes and urban sprawl in Al-Khobar, Saudi Arabia: An analysis of multi-temporal remote sensing data. *ISPRS International Journal of Geo-Information, 5*(2), 15.

Rajan, S., Ghosh, J., & Crawford, M. M., (2006). Exploiting class hierarchies for knowledge transfer in hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing, 44*(11), 3408-3417.

Redko, I., Courty, N., Flamary, R., & Tuia, D., (2018). Optimal transport for multi-source domain adaptation under target shift. arXiv preprint arXiv:1803.04899.

Sexton, J. O., Noojipady, P., Anand, A., Song, X. P., McMahon, S., Huang, C. & Townshend, J. R., (2015). A model for the propagation of uncertainty from continuous estimates of tree cover to categorical forest cover and change. *Remote Sensing of Environment, 156,* 418-425.

Snavely, N., Seitz, S. M., & Szeliski, R. (2006, July). Photo tourism: exploring photo collections in 3D. In ACM Transactions on Graphics, 25(3), 835-846.

Song, A., Choi, J., Han, Y., & Kim, Y., (2018). Change detection in hyperspectral images using recurrent 3d fully convolutional networks. *Remote Sensing, 10*(11), 1827.

Song, H., Kim, Y., & Kim, Y., (2019). A Patch-Based Light Convolutional Neural Network for Land-Cover Mapping Using Landsat-8 Images. *Remote Sensing, 11*(2), 114.

Tewkesbury, A. P., Comber, A. J., Tate, N. J., Lamb, A., & Fisher, P. F., (2015). A critical synthesis of remotely sensed optical image change detection techniques. *Remote Sensing of Environment, 160,* 1-14.

Tian, J., Reinartz, P., d'Angelo, P., & Ehlers, M., (2013). Region-based automatic building and forest change detection on Cartosat-1 stereo imagery. *ISPRS Journal of Photogrammetry and Remote Sensing, 79,* 226-239.

Tuia, D., Persello, C., & Bruzzone, L., (2016). Domain adaptation for the classification of remote sensing data: An overview of recent advances. *IEEE Geoscience and Remote Sensing Magazine, 4*(2), 41-57.

Volpi, M., Tuia, D., Bovolo, F., Kanevski, M., & Bruzzone, L., (2013). Supervised change detection in VHR images using contextual information and support vector machines. *International Journal of Applied Earth Observation and Geoinformation, 20,* 77-85.

Wegner, J. D., Hansch, R., Thiele, A., & Soergel, U., (2010). Building detection from one orthophoto and high-resolution InSAR data using conditional random fields. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 4*(1), 83-91.

Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012). 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology, 179,* 300-314.

Xu, Y., Wu, L., Xie, Z., & Chen, Z., (2018). Building extraction in very high resolution remote sensing imagery using deep learning and guided filters. *Remote Sensing, 10*(1), 144.

Yang, H. L., & Crawford, M. M., (2015). Domain adaptation with preservation of manifold geometry for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9*(2), 543-555.

Yang, M., Jiao, L., Liu, F., Hou, B., & Yang, S. (2019). Transferred Deep Learning-Based Change Detection in Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(9), 6960-6973.

Yu, H., Yang, W., Hua, G., Ru, H., & Huang, P., (2017). Change detection using high resolution remote sensing images based on active learning and Markov random fields. *Remote Sensing, 9*(12), 1233.

Zerrouki, N., Harrou, F., Sun, Y., & Hocini, L. (2019). A Machine Learning-based Approach for Land Cover Change Detection Using Remote Sensing and Radiometric Measurements. *IEEE Sensors Journal, 19*(14), 5843-5850.

Zhan, Y., Fu, K., Yan, M., Sun, X., Wang, H., & Qiu, X. (2017). Change detection based on deep siamese convolutional network for optical aerial images. *IEEE Geoscience and Remote Sensing Letters, 14*(10), 1845-1849.

Zhang, Y., Yang, H. L., Prasad, S., Pasolli, E., Jung, J., & Crawford, M., (2014). Ensemble multiple kernel active learning for classification of multisource remote sensing data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8*(2), 845-858.

Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F., (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine, 5*(4), 8-36.

Zhu, Z., & Woodcock, C. E. (2014). Continuous change detection and classification of land cover using all available Landsat data. *Remote Sensing of Environment, 144,* 152-171.

Zhu, Z., Zhang, J., Yang, Z., Aljaddani, A. H., Cohen, W. B., Qiu, S., & Zhou, C. (2019). Continuous monitoring of land disturbance based on Landsat time series. *Remote Sensing of Environment, 111116.*

국문 초록

원격탐사에서의 딥러닝 (deep learning)의 성공적인 활용은 학습에 사용되는 표본이 목표 대상을 얼마나 잘 대표하는지에 달려있다. 참조자료 (ground truth)가 취득되는 소스 도메인 (source domain)의 데이터 분포가 훈련된 모델이 적용되는 타겟 도메인 (target domain)의 데이터 분포를 충분히 대표하지 못하는 경우, 훈련된 모델은 타겟 도메인에서 성능이 급격히 저하된다. 이로 인해, 훈련된 모델은 재사용되기 어렵고, 사용자는 새로운 실험 지역마다 매번 참조자료를 새롭게 취득하고 훈련하는 절차를 반복해야만 한다. 이러한 데이터 분포 변화 (dataset shift)에 따른 문제를 해결하는 연구 분야를 도메인 적응 (domain adaptation)이라 하며, 넓은 지역에 분포한 대상물을 분석하는 원격탐사 분야에서 특히나 중요하다. 하지만 대부분의 변화탐지 선행연구는 딥러닝 알고리즘을 적용할 때 데이터 분포 변화에 따른 문제를 고려하지 않아, 새로운 지역에 바로 적용할 경우 정확도가 저하되는 문제가 존재한다.

이러한 문제를 해결하기 위해 본 논문은 변화탐지에서 발생하는 데이터 분포 변화 문제 중, 두 도메인 간의 변화율이 일치하지 않는 경우에 발생하는 사전확률변화 (prior probability shift) 문제에 주목하였다. 본 논문은 딥러닝 알고리즘이 변화탐지에서 사전확률변화로 인해 정확도가 저하되는 문제를 해결하기 위한 프레임워크를 제안하였다. 제안한 프레임워크는 비지도 방식을 이용하여 타겟 도메인의 변화율을 예측한 뒤, 소스 도메인에서 학습된 딥러닝 알고리즘의 소프트맥스 (softmax) 출력의 변화 임계 값을 조절하는 방식으로 정확도 저하를 방지한다.

본 연구에서는 제안한 프레임워크의 성능을 검증하기 위해 가장 널리 사용되는 딥러닝 모델인 합성곱신경망 (convolutional neural network)과 대표적 비지도 변화탐지 방법인 변화벡터분석 (change vector analysis)을 채택하였다. 실험 데이터는 RGB (Red, Green, Blue) 채널과 DSM (Digital Surface Model) 데이터를 포함하는 두 시기의 UAV (Unmanned Aerial Vehicle) 영상을 사용하였으며, 변화율이 다양한 세개의 하위지역으로 실험 지역을 나누어 도메인 간의 변화율 차이가 다양한 조건하에 실험을 진행하였다. 아울러, 훈련 표본을 세 개의 하위지역 각각에서 취득하여 합성곱신경망을 학습시킨 뒤, 나머지 두 개의 하위지역에 적용하는 방식의 실험을 수행하였다. RGB와 RGB+DSM 데이터 각각에 대해 실험을 수행하였고, 훈련의 표본 수가 총 1,000개인 경우와 2,000개인 경우에 대해서도 실험을 수행하였다. 또한, 제안한 프레임워크의 성능을 다양한 조건하에서 확인하기 위해 변화율이 급변하는 샘플링 조건도 추가하여 분석하였다.

실험 결과, 본 연구에서 제안한 프레임워크가 사전확률변화로 인해 편향된 변화 임계 값을 효과적으로 조절함으로써 변화탐지 정확도를 향상시킨 것이 확인되었다. 추가적으로, 딥러닝 모델이 더 많은 정보를 통해 학습되었을 때 사전변화확률에 특히나 더 취약하다는 것이 확인되었다. 본 실험의 결과는 점차 방대한 양의 데이터로 딥러닝의 성능을 개선시키는 현재의 딥러닝 추세에 도메인 적응의 필요성을 강조한다. 마지막으로, 제안한 프레임워크는 이행 방법이 간단하며 다양한 종류의 변화탐지 알고리즘과 함께 융합되어 활용될 수 있다는 장점이 있다.

주요어: 도메인 적응, 변화탐지, 딥러닝, 사전변화확률, 합성곱신경망, 변화벡터분석

학번: 2018-27253