Embedding Domain-Invariant Building Segmentation Information On Change Detection Model

Minh-Khoa Le[1], Dinh Phong Vo and Bac Le[1]

[1]Vietnam Vietnam National University, Ho Chi Minh City, Vietnam
[1]University of Science, Ho Chi Minh City
18120415@student.hcmus.edu.vn; dpvo@tuta.io; lhbac@fit.hcmus.edu.vn

***Abstract*—The use of remote sensing data in combination with deep learning methods has resulted in an impressive performance on tasks such as building segmentation and change detection. While some research has demonstrated that supervised deep learning approaches outperform traditional methods, these methods are often limited by small datasets and can struggle with distribution shifts. Advances in the field of domain adaptation have led to the development of architectures with improved ability to learn domain-invariant features. In this paper, we propose a domain-invariant segmentation model and enhance its training process to improve performance. We also incorporate building segmentation information into our change detection model, demonstrating the benefits of incorporating building data into the model.**

***Index Terms*—deep learning, remote sensing, change detection, build segmentation**

I. Introduction

Satellite imagery is a rich source of providing information to solve many problems in the human world [36], [29], [32]. As time passes, many satellites have been deployed with more modern equipment, bringing a vast amount of high-quality data. Now, more satellite data can be accessed easier and freely [14]. As a result, the goal of quickly detecting or calculating insights from satellite images becomes a reality, making the fantasy of quickly identifying or assessing damage from calamities a reality.

Building segmentation and change detection are two crucial problems in remote sensing fields. Building segmentation is a classification task that extracts building pixels from a satellite image [24]. Change detection is the process of identifying objects' differences in the same geographical location at different times [28]. A change detection system based on earth observation image's purpose is to assign a label of changed or unchanged per pixel for each pair of coregistered images. Building segmentation and change detection is widely used in many applications, such as damage assessment, urban planning and agricultural monitoring [16].

Due to the explosive growth in the quantity and quality of satellites, deep learning models have shown outstanding performance in remote sensing tasks [6], [26] [10]. Both segmentation and change detection task is solved well by deep learning models. However, supervised deep learning power is limited due to the lack of supervised remote sensing datasets [7]. Moreover, distribution shifts between them cause the model's poor performance across multiple datasets. On the other hand, each task is usually solved separately from scratch [13].

Recently, the relationship between segmentation and change detection has been explored. Image segmentation technique provides significant advantages for remote sensing imagery analysis, pointing out many buildings in different scales [18]. Researchers have improved their model's performance by embedding segmentation masks or network architecture in change detection [31], [35]. Among various methods, we present a method combining remote sensing images with generated segmentation masks without further labeling efforts.

To summarize, in this paper, we collected a segmentation dataset from existing sources. We also enhanced the segmentation model architecture and training process to better evaluate unseen data with distribution shifts. Moreover, we presented a simple method by embedding building information, which helps our model perform well on the change detection dataset.

II. Background and Related Work

A. Remote sensing

As the number of satellites has increased, the amount of remote sensing data available has also grown rapidly, providing a wealth of valuable information that can be applied to a variety of human problems [16]. The vast amount of remote sensing images has spurred the use of deep learning methods in many applications, such as building and road segmentation, as well as change detection and damage assessment.

In 2018, Microsoft released building footprints datasets in the United States [23]. They used a deep convolutional neural network and polygonization algorithms to generate 129,591,852 building footprints in 50 states. The neural network has a pixel recall/precision at $95.5\%/94.0\%$. The number of footprints is even higher when they release building footprints worldwide in 2022 [22]. There are about 982M buildings between 2014 and 2022. Although the generated dataset is imperfect, it is still beneficial for developing many satellite

image-based tasks: building segmentation, loss catastrophic loss assessment.

One important characteristic of satellite images is the off-nadir angle, which is the angle between the satellite and the target object at the time of collection. Satellites typically capture images at large off-nadir angles to collect more data. However, this can result in poor geometric consistency for tall objects when multiple off-nadir angles are introduced. There are few datasets that consist of multiple-angle images, resulting in a significant amount of satellite data going unused. Shen et al. have introduced a new dataset called S2Looking [27] for change detection tasks, which includes off-nadir angles ranging from -35° to 35°. This dataset primarily focuses on rural areas with sparse building updates, and complex rural environments [27]. In addition, the registration process for bi-temporal images is not entirely accurate, making change detection even more difficult.

B. Building segmentation for satellite imagery

Building segmentation is a binary semantic segmentation task that aims to extract buildings from satellite images. Many researchers have designed their networks based on U-Net architecture because of its excellent performance. There are also many efforts to enhance the U-Net architecture for better performance. Iglovikov et al. introduce the TernausNetV2 [18] with the improvement from U-Net by using WideResNet-38 network with in-place activated batch normalization as the encoder, showing superior performance in the building segmentation task.

However, U-Net architecture considers features extracted from every layer have the same importance. The proposed Squeeze-and-Excitation (SE) network [17] has solved this problem by allowing the network to perform feature recalibration and produce better performance with just little overhead computation resources. Many authors have shown that the combination of SE blocks and U-Net performs well on the segmentation task. Chatterjee et al. proposed a network architecture that combines the U-Net architecture, Dense blocks, and Squeeze-and-Excitation (SE) blocks [5]. This combination results in improved prediction accuracy compared to the original U-Net. Abdollahi et al. modify the original U-Net by embedding densely connected convolution between the encoder and decoder parts, the SE function in the expansive part and BConvLSTM in the skip connections. This modification makes the U-Net architecture utilize global information to suppress useless features and selectively emphasize informative ones [1].

C. Change detection

Change detection became possible after the existence of satellite images. In the past, image differencing was the primary technique for change detection [28]. However, it easily suffers from changes that are not in favor [37]. For example, sessional changes can create differences in color at different times, dramatically decreasing the image differencing method performance. Recently, deep learning methods have developed, and improved accuracy with the rise of Convolutional Neural Networks (CNNs) [6], [3]. Some of them are post-classification methods whose structures contain two classifiers to classify the bi-temporal images [4], [34]. Then, the change map can be obtained by comparing classification maps. However, those methods are not always possible when only the change label is available.

On the other hand, many attempts have done to train CNNs directly and generate change maps from satellite images. They focus on improving the model's feature extraction and feature discrimination abilities. Daudt et al. introduce three fully convolutional models based on simplified U-Net, and Siamese Net [3]. In the siamese version, they use the skip connection, which takes input from two encoders, and concats or subtracts them before feeding them to the decoder. The self-attention mechanism also has been employed to capture rich global spatial-temporal features [7], [8]. The architecture usually combines of CNNs feature extractor at the beginning. Following that is the attention module and at the end is the prediction head.

D. Domain adaptation

In this section, we briefly review some types of domain adaptation. One of the main domain adaptation methods is domain-invariant feature learning. Domain-invariant feature learning methods focus on aligning source and target domains by producing domain-invariant feature representation [33]. In detail, a network can produce domain-invariant feature representation if the output feature follows the same distribution, whether input comes from the source or target domain. To archive domain-invariant feature representation, researchers have exploited three methods.

The divergence-based method can align source and target domains by minimizing their divergence. For example, Rozantsev et al. proposed a two-stream architecture: one works on the source domain and the other on the target domain. The model can produce transferable features across domains in supervised, semi-supervised or unsupervised manners through a combination of classification, regularization, and domain discrepancy losses. Kang et al. proposed contrastive domain discrepancy that measures the difference between conditional data distributions across domains without requiring both source and target domain labels [19]. They also trained a deep neural network with respect to classification and discrepancy loss. As a result, source and target data distributions are aligned, and the inter-class domain discrepancy is maximized.

Another approach, named adversarial-based domain adaptation, also aims to minimize cross-domain disparency via an adversarial objective. Some use the generative adversarial network (GAN) to generate synthetic target data related to the source domain to train the network [21], [38]. The CoGAN model, proposed by Liu and Tuzel, consists of 2 GANs, each accountable for producing images in one domain [21]. In the training phase, they share a subset of parameters in the first few layers of the generative models and the last few layers of the discriminative models. This setting allows

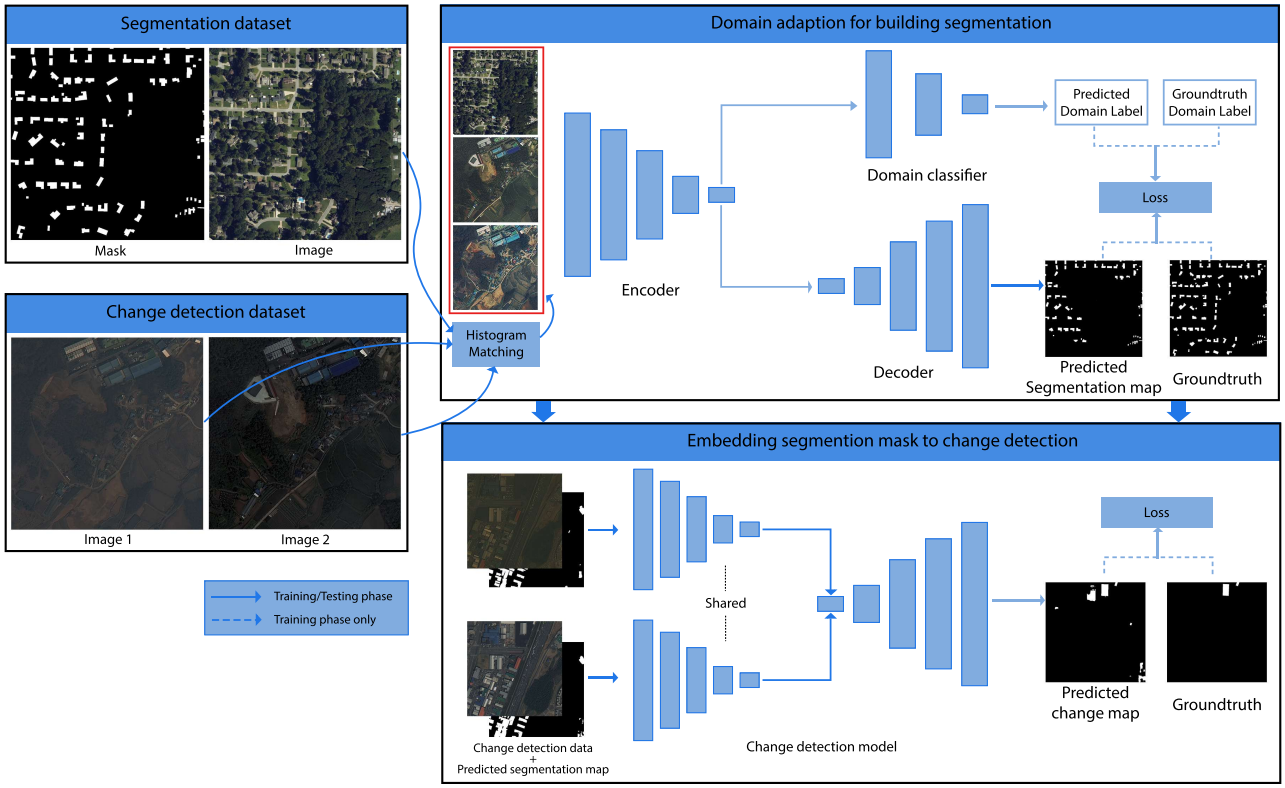


Fig. 1. Overview of our proposed method including two parts: domain adaptation for building segmentation and embedding segmentation mask to change detection

CoGAN to learn a joint distribution of images without labels. As a result, CoGAN can generate a pair of images that shares the same high-level features while differing in low-level realizations. Another method called Domain-Adversarial Training of Neural Networks (DANN) uses domain confusion loss to confuse high-level layers about the input's domain [13]. The proposed architecture consists of 3 components: feature extractor, domain classifier and label predictor. By minimizing the label prediction loss while maximizing the domain classification loss, the feature extractor is trained to generate domain-invariant feature representation and perform well on the label prediction task.

III. Proposed Methods

Our idea is to combine building segmentation information with RGB images to make the change detection model more aware of building entities in images. We use a baseline Unet [2] architecture as the segmentation model in the first step. However, we observed that the training and test data come from two different domains, which may make the model perform poorly. We introduce the domain adaptation technique in the building segmentation task to address this problem. Moreover, pre-processing methods are applied to try to improve the model further.

In the next step, we use segmentation masks generated from our segmentation model as the input image's fourth channel, making our change detection model more aware of building entities in the image.

A. Domain adaption for building segmentation

1) Baseline segmentation model: A baseline model inspired by TernausNetV2 [18] is chosen to solve domain adaption building segmentation. Concretely, the model has an encoder-decoder architecture, and the backbone is ResNet-50 [15] which was pretrained on ImageNet. ResNet architecture has been proposed and is well-known for the ease of training networks substantially deeper than those used previously [15]. Moreover, long skip connections are added at each convolution block, allowing low-feature from the input image and encoder to flow directly to high-level feature maps, further improving localization accuracy and speeding up the training process [25].

2) Domain adaptation segmentation model: On the baseline segmentation model, we apply the Domain-Adversarial Neural Networks (DANN) technique [13] to construct a segmentation model called Domain-Adaptation Segmentation Model (DASM). The expectation is to improve the baseline's accuracy on the test set and minimize the effects of distribution shifts. We aim to confuse the decoder module about the image domains and generate the features invariant to domains. After the encoder module, a domain adaptation component - a domain classifier is added to classify images from different domains (the domain classifier in figure 1). The domain

classifier is a simple neural network with only one hidden layer. Moreover, a dropout layer is added at the beginning of the neural network to simplify the classifier and avoid easily overfitting.

Denote that $G_e(\cdot;\theta_e)$ is the encoder with parameters θ_e, $G_s(\cdot;\theta_s)$ is the decoder with parameters θ_s, and $G_c(\cdot;\theta_c)$ is domain classifier with parameter θ_c. For an input image x_i with the domain label d_i and the building mask y_i, segmentation loss and domain classification loss is defined as:

$$L_s^i(\theta_e,\theta_s) = L_s(G_s(G_e(x_i;\theta_e);\theta_s), y_i)$$

$$L_c^i(\theta_e,\theta_c) = L_c(G_c(G_e(x_i;\theta_e);\theta_c), d_i)$$

For N sample images with the first n samples with labels from source domain - $\mathcal{D}_S$ and the last $N-n$ samples without labels from target domain - $\mathcal{D}_T$, total loss of the model is defined as:

$$\begin{aligned} E(\theta_e,\theta_s,\theta_c) &= L_s(\theta_e,\theta_s) - L_c(\theta_e,\theta_c) \\ &= \frac{1}{n}\sum_{i=1}^{n} L_s^i(\theta_e,\theta_s) \\ &- \lambda\left(\frac{1}{n}\sum_{i=1}^{n} L_c^i(\theta_e,\theta_c) + \frac{1}{N-n}\sum_{i=n+1}^{N} L_c^i(\theta_e,\theta_c)\right) \end{aligned} \tag{1}$$

with λ is a hyperparameter that controls the importance of the domain loss. We can minimize equation 1 by finding the saddle point $\hat{\theta}_e, \hat{\theta}_s, \hat{\theta}_c$ with respect to

$$(\hat{\theta}_e, \hat{\theta}_s) = \underset{\theta_e,\theta_s}{\operatorname{argmin}}\ E(\theta_e,\theta_s,\hat{\theta}_c) \tag{2}$$

$$\hat{\theta}_c = \underset{\theta_c}{\operatorname{argmax}}\ E(\hat{\theta}_e,\hat{\theta}_s,\theta_c) \tag{3}$$

Equation 1 calculates the loss function by subtracting domain loss from segmentation loss. We expect the subtraction makes the loss converge to a saddle point, where domain loss is maximized while minimizing the segmentation loss. Such subtraction can be implemented easily by a gradient reversal layer [13]. This layer has no parameters and reverses the gradient's sign by multiplying a negative constant during the backpropagation phase. By applying the gradient reversal layer, a saddle point can be found using the gradient descent algorithm:

$$\theta_e \longleftarrow \theta_e - \mu\left(\frac{\partial L_s^i}{\partial \theta_e} - \lambda\frac{\partial L_c^i}{\partial \theta_e}\right) \tag{4}$$

$$\theta_s \longleftarrow \theta_s - \mu\frac{\partial L_s^i}{\partial \theta_s} \tag{5}$$

$$\theta_c \longleftarrow \theta_c - \mu\lambda\frac{\partial L_c^i}{\partial \theta_c} \tag{6}$$

with μ is learning rate. Converging to the saddle point, the encoder fails to distinguish between image domains. As a result, the encoder generates domain-invariant features.

3) Pre-processing methods to improve domain adaption segmentation model: Despite adding a domain classifier to make the output features of the encoder invariant to domains, there is still a significant distribution shift between image domains that can not be excluded. In order to reduce the gap between distributions, we use per-image normalization and histogram matching in pre-processing. We call the combination of the DASM and pre-processing methods as Improved Domain-Adaptation Segmentation Model (I-DASM).

a) Per-image normalization: Image normalization is simple and the first process to solve the domain shift problem. It brings the images with different ranges of pixel intensity into the same range. As a result, images from different domains share the same contrast.

$$\hat{x_i} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{7}$$

In the equation 7, μ, σ is mean and standard deviation of image, $\hat{x_i} \in [-1, 1]$ is the pixel value at a position i after normalizing. It is calculated by subtracting the original pixel value x_i with μ over the image and dividing by the σ.

b) Histogram matching: Histogram matching is the process of transforming an image to match its color histogram to a specified histogram of another image. In different lighting conditions, atmospheres, and satellite sensors, the observed colors may vary and differ from the actual colors [30]. By adjusting gamma, saturation, and contrast via histogram matching, we can correct colors to produce an output image that shares an equivalent color distribution with the input image while preserving the content of the input image.

Given a pair of images, a grayscale source image S that needed to be applied histogram matching and a grayscale reference image R. Each image has a probability density function $p^s(a)$ and $p^r(a)$ where a is a grayscale value and $p^s(a)$ and $p^r(a)$ is the probability of that value for source and reference image. The probability can be computed from its color histogram:

$$\begin{aligned} p^s(a_i) &= \frac{n_i^s}{n^s} \\ p^r(a_i) &= \frac{n_i^r}{n^r} \end{aligned} \tag{8}$$

where n_i is the frequency of the pixel value a_i, and n is the total number of pixels in the image. To apply histogram matching, we need to compute its cumulative distribution function for each image.

$$\begin{aligned} F^s(a_j) &= \sum_{i=0}^{j} p^s(a_i), \quad j = 0, 1, ..., G-1 \\ F^r(a_j) &= \sum_{i=0}^{j} p^r(a_i), \quad j = 0, 1, ..., G-1 \end{aligned} \tag{9}$$

where G is the total number of pixel values. For each source pixel value $G_i^s \in [0, 255]$, we need to find a reference pixel value G_j^r for which $F^s(G_i^s) = F^r(G_j^r)$, which is the result of a histogram matching function $M(G_i^s) = G_j^r$. Finally, to

find the output image, we apply $M()$ for each pixel of the reference image.

We choose a collection of images from the Alabama dataset as reference images, consisting of various images from rural to industrial areas. Every reference image has unique content and color distribution, ensuring we can always find a similar reference image with every source image. Each source image is compared with the collection, and select one most similar to the source image. To compare two images, we use Pearson's Correlation Coefficient [12] as the similarity metric - $s(H_1, H_2)$ with H_1, H_2 are the histogram of images, respectively. The higher the metric, the more similar they are.

$$s(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}} \tag{10}$$

$$\bar{H_k} = \frac{1}{N} \sum_{J=1}^{N} H_k(J) \tag{11}$$

where I is the index of histogram bins, N is the total number of histogram bins.

B. Embedding Segmentation to Change Detection

1) Baseline change detection model: To proceed with the Change Detection task, we use the Siamese UNet [25] architecture with ResNet-50 [15] as the backbone. The network employs an encoder-decoder design based on FC-Siam-Diff [10]. The encoding layers of the network are separated into two streams of similar structure with shared weights as in a traditional Siamese network [11] to extract features from two input images. Each has 5 convolution layers and 5 max-pooling layers to continuously extract features from input images by increasing the number of channels and decreasing the size of images twice in each layer.

Skip connections are added as an improvement to the architecture. Moreover, each skip connection takes input features from both encoders and processes them before feeding them to the decoders. Instead of concatenating input features and feeding them to a convolution block, we choose to take the absolute value of their difference. It makes the network simpler and helps it learn to compare the differences between images [10].

2) Embedding segmentation information to improve change detection model: We decide to embed the segmentation mask into the input images. It is the most straightforward way to help our change detection model emphasize the differences between images, especially the object differences. The input image is extended by adding one more channel of segmentation mask (figure 1). Concretely, the segmentation mask is the output of the I-DASM.

By keeping each mask pixel as the probability of being the object pixel $p \in [0, 1]$, our model may use object differences information selectively and robust to false-positive generated by the segmentation model. Moreover, the mask is also normalized by per-image normalization. In the decoding phase, the two input images, including segmentation masks, are subtracted before feeding to the last decoder.

IV. Experiments

A. Experiments details

We use Alabama, and S2Looking [27] datasets to evaluate our methods in this work. All 10,200 samples from the Alabama dataset are used for the segmentation task as the training set and reference images for the histogram matching algorithm. For the test set, we use every S2Looking image patch pair.

However, there are no ground-truth segmentation labels in the S2Looking dataset, so we use the change detection mask as the label to compare segmentation models. With each pair of input RGB images from the change detection dataset, the segmentation models result in 2 segmentation masks. We use the Hungarian algorithm [20] to generate a change detection mask from 2 segmentation masks. For more details, we extract every building mask from segmentation masks into a set of objects. And then using the Hungarian algorithm [20] to map building masks between two sets. Every mask that is not mapped is classified as a changed building and presents in the change detection mask.

To evaluate the performance on both tasks, Precision, Recall, and F1-Score can be used as evaluation metrics:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

$$\text{F1-Score} = \frac{2 \times TP}{(TP + FP) + (TP + FN)} \tag{14}$$

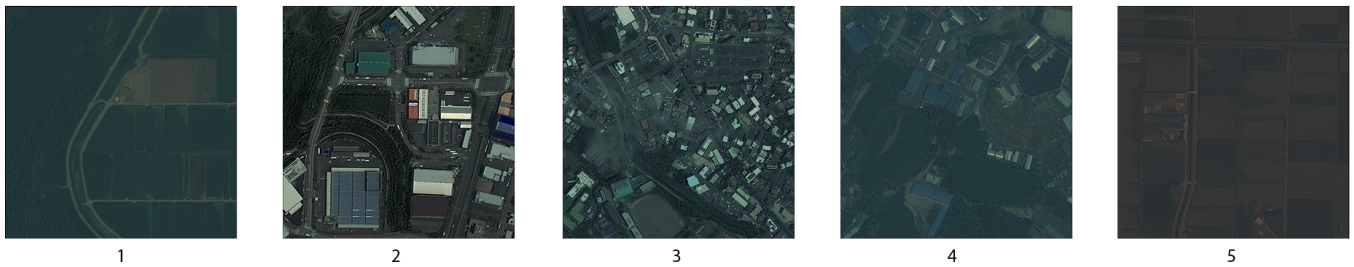
with TP, FP, FN correspond to the number of true-positive, false-positive, and false-negative predicted pixels for class 1 and 0.

In the training phase on both tasks, we use the Adam optimizer with a learning rate of 0.01. After every ten steps, the learning rate will be decreased ten times. Due to the memory limitation, we split a 1024×1024 image into four images with a size of 512×512. For the change detection task, we randomly select six 512×512 parts for each pair of images and the change detection mask using the random crop regarding the mask to avoid overfitting because of the class-imbalance phenomenon. A hard augmentation pipeline, including random shift, scale and rotation, random RGB shift, brightness and contrast, is embedded in pre-processing phase to improve the robustness of the model. Every method was trained for 20 epochs in this setting.

B. Experiments on segmentation models

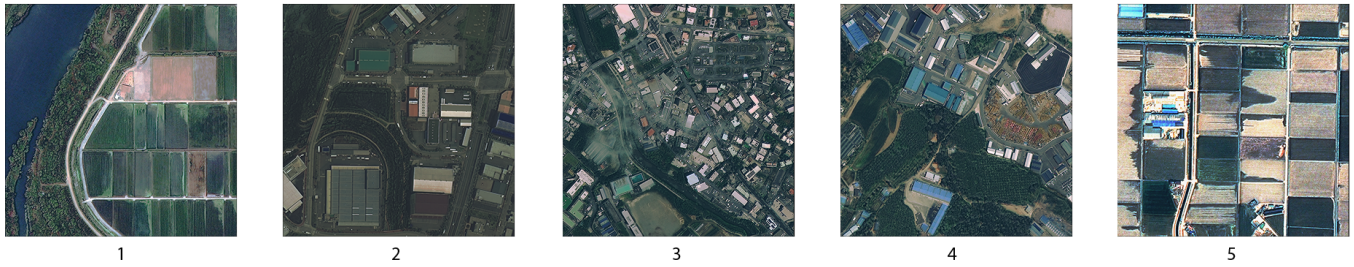
We conduct multiple experiments to compare methods for the segmentation task. The training set is the Alabama dataset, and the test sets are both Alabama and S2Looking [27] datasets. The training setting was described in the section IV-A.

In the table I, we show the F1-Score of our models on two different tasks: segmentation in the Alabama dataset and change detection in the S2Looking [27] dataset. Although the



(a) S2Looking source images

(b) Alabama reference images



(c) Result images

Fig. 2. Histogram matching algorithm: Each sample numbered ordinally consists of three images: a source, a reference and an output, respectively shown in each row.

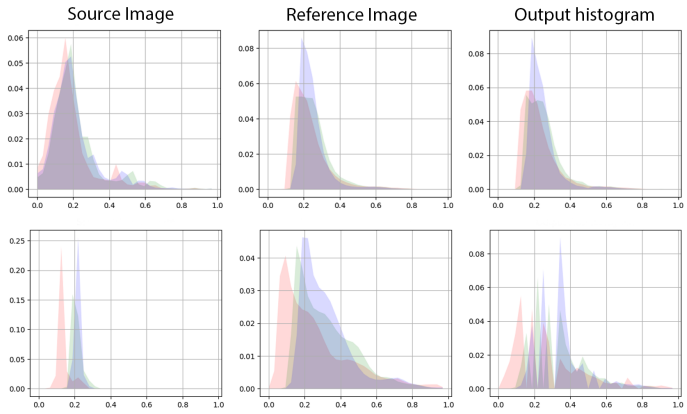


Fig. 3. Histograms of source, reference and output images. Each graph has red, green, and blue histograms corresponding to the red, green, and blue channels.

baseline shows excellent performance on the segmentation task - at 0.8421, it is not good at the change detection in the S2Looking [27] dataset with a deficient F1-Score. Our Domain Adaptation Segmentation Model improves the change

TABLE I
RESULT FOR THE EVALUATED SEGMENTATION METHODS

Method	**Alabama F1-Score**	**S2Looking F1-Score**
Baseline (Section III-A1)	0.8412	0.1520
DASM (Section III-A2)	0.8020	0.1950
I-DASM (Section III-A3)	0.8336	0.2084

detection F1-Score while preserving a high segmentation F1-Score. I-DASM has performed better on both two tasks than DASM with pre-processing methods. It significantly improves the change detection F1-Score with a small segmentation performance trade-off.

Figure 4 illustrates that the baseline model easily suffers from the image colors. It recognizes the landscape excellently in case there is no building in this area. However, when some buildings and backgrounds overlap, it fails to distinguish and classifies background pixels as building pixels, especially when the contrast is low in color between buildings and backgrounds. The domain-adversarial neural networks technique performs excellently. Figure 4 shows that DASM performs far better than the baseline Unet model. It no more suffers

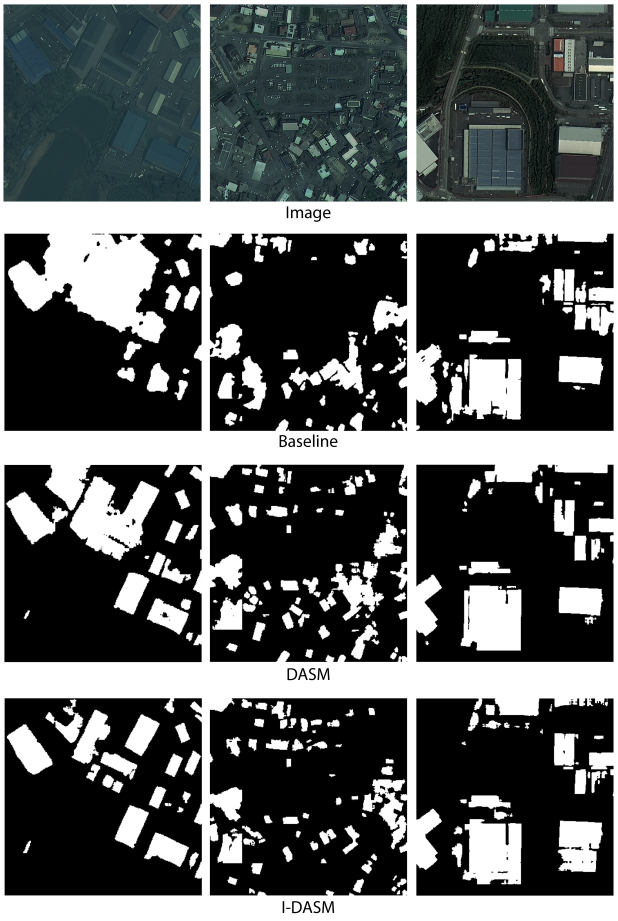


Fig. 4. Sample images from S2Looking dataset and our object-level change detection model's predictions. Each row's label is the index of each sample from S2Looking dataset.

from the landscape colors and recognizes the border between buildings and backgrounds very well. The score improvement proves that our encoder has produced features partly invariant to the input domain. The most interesting part is that we do not need to provide the model with any segmentation label. Nevertheless, the DASM still suffers from the low contrast between buildings and backgrounds. There are still many unrecognized instances, especially when the colors of buildings are the same as the background ones.

With dark, low-contrast source images from S2Looking and reference images from the Abalama dataset, histogram matching results in images with the same color distribution as reference images, better brightness and contrast. The first sample in figure 3 shows that the output histogram resembles the reference one. As seen in the second row of figure 3, the algorithm results in significant deviations between some neighboring bins in the histogram when the difference between source and reference images is vast. Moreover, if there is anything abnormal in either source or reference image, the cumulative distribution will be skewed, causing the histogram matching to perform poorly. On the other hand, histogram matching does not give attention to the image's context and even tries to match the building colors with landscape colors. It lowers the contrast between buildings and background, making our model harder to recognize building instances.

C. Experiments on change detection models

We experiment to check the performance of different change detection models. Each model is trained for 20 epochs on 3500 S2Looking images and evaluated on 1500 S2Looking images. A combination of cross-entropy loss and Dice loss, the same as the loss function in training segmentation, is the objective function. The training setting is the same as segmentation experiments. In the training phase, a hard augmentation pipeline, including random shift, scale and rotation, random gamma, blur, RGB shift, brightness and contrast. Moreover, we randomly select six parts for each pair of images and mask using the random crop regarding the mask to make our change detection model more robust to input images and avoid overfitting.

TABLE II
RESULT OF OUR CHANGE DETECTION METHODS AND OTHER METHODS ON S2LOOKING DATASET.

Method	Precision	Recall	F1-Score
FC-EF [10]	**0.8136**	0.0895	0.1613
FC-Siam-Diff [10]	**0.8329**	0.1576	0.2650
STANet-Base [9]	0.2575	**0.5629**	0.3534
CDNet [6]	0.6748	0.5493	**0.6056**
I-DASM (Section III-A3)	0.1393	0.4133	0.2084
Baseline (Section III-B2)	0.5731	0.5514	0.5621
OL-CDM (Section III-B2)	0.5454	**0.6051**	**0.5737**

Looking at the table II, we observed that our baseline achieved a good F1-Score, at 0.5621, even higher than some previous methods, such as FC-Siam-Diff [10]. By providing the baseline building information, our method is improved by over 1%, getting a higher F1-Score at 0.5737. Compared to other methods, our methods archive far higher F1-Scores than previous methods, such as FC-Siam-Diff [10], and STANet-Base [9]. In the S2Looking [27] dataset, CDNet [6] performs better than our model, at an F1-Score of 0.6056, around 3% higher than our object-level change detection model.

Figure 5 shows our object-level change detection model predictions on the S2Looking dataset. We observe that our model performs well and points out many change pixels. It is robust to the difference between two images: color distribution and brightness. Moreover, our model can discriminate between the change in construction and the change in non-construction. Although there are changes in buildings and background in the blue rectangle in sample 3, the CD model can distinguish them pretty well.

On the other hand, our method still suffered from the off-nadir angle. Even though there is no change in buildings, the off-nadir angles are various in the two images. The phenomenon is even more severe when the buildings are tall, such as landmarks or apartments. In the second sample, the CD model fails to recognize changes in the off-nadir angles, not the buildings. Our model also fails to recognize buildings when clouds cover buildings in an image. In sample 5 in figure

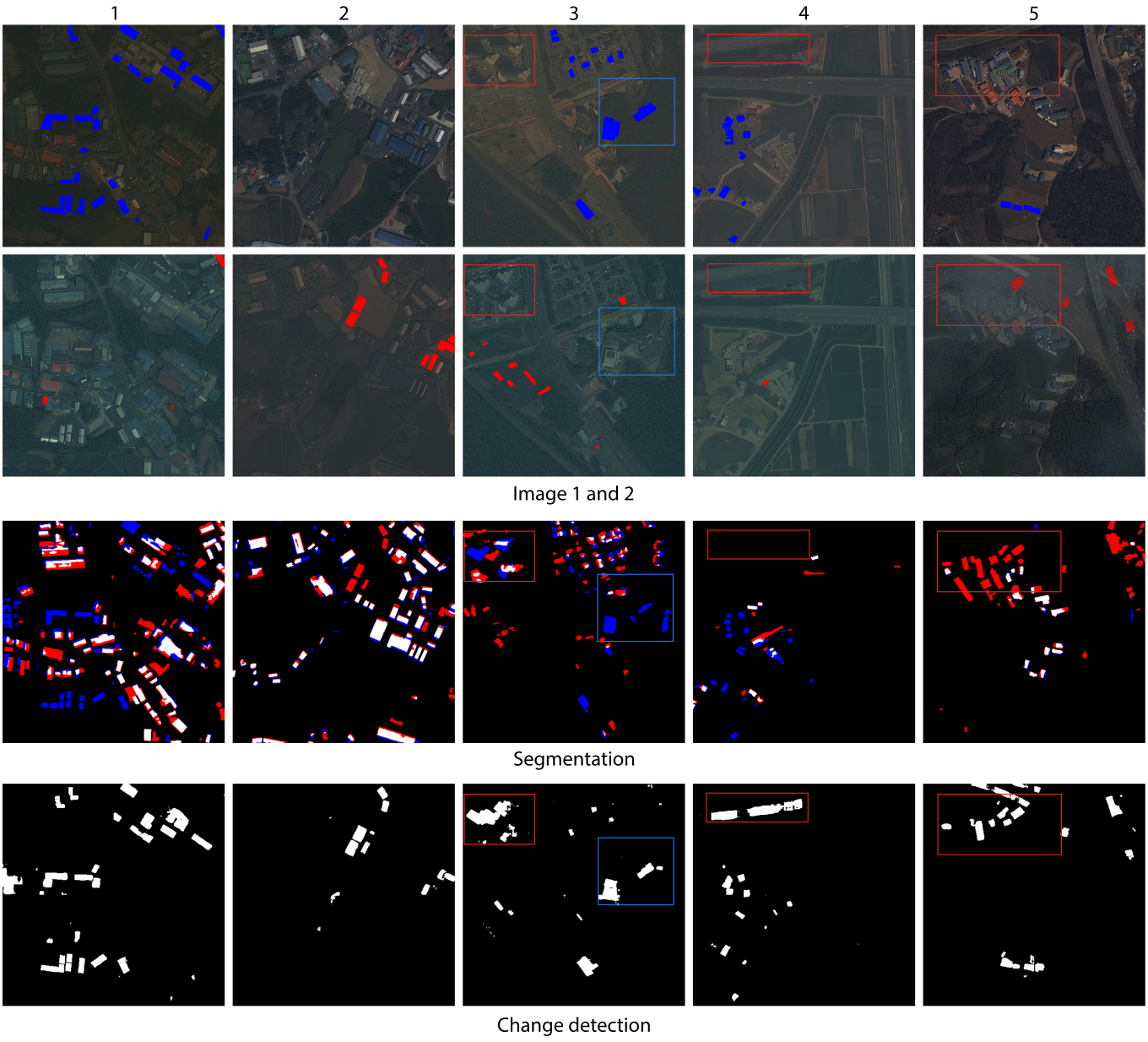


Fig. 5. The result of the S2Looking dataset on segmentation and change detection tasks. The first two rows present pairs of before and after images. Each image contains red and blue objects representing buildings that only exist in the other image at the same place. The segmentation row shows our I-DASM model results on both before and after images: White pixels are buildings that appear in both images, while blue and red pixels only appear in images 1 and 2, respectively. The last row is the outputs of our change detection model.

5, both segmentation and change detection models perform well on the are not covered by clouds. However, they fail completely in the opaque area.

Sometimes, the difference between two images is tough to classify into construction or non-construction groups. In sample 4, we see that the false positive pixels are even hard for people to determine which group they belong to, construction change or non-construction change. This also means that our change detection is robust to errors in the dataset labeling phase.

V. Conclusion

In this paper, we address the issue of distribution shifts in building segmentation tasks. We propose a framework for reducing the gap between different domains and improving the performance of the model on the test set without sacrificing its performance on the training set. Our method was evaluated using the Alabama dataset as the training set and the S2Looking dataset as the test set, and we showed that it could be used to improve the performance of change detection models on the S2Looking dataset. With the increasing use of satellite technology, change detection will play a vital role in various

industries, such as disaster forecasting, geographic monitoring, and tracking land use and urbanization. Our method offers a potential solution for improving change detection models using available data without the need for extensive manual labeling.

In our research, we tackled challenges such as domain shift, mismatched methodologies, and insufficient data. We ultimately arrived at the final architecture by redesigning the network and carefully tuning the parameters to improve performance. As a potential future improvement, we could combine histogram matching and DANN into a single model. Additionally, we could explore ways to use the segmentation masks further to improve the performance of the change detection model.

Acknowledgment

We would like to express our heartfelt gratitude to our mentor and professor for their invaluable support and guidance throughout our research journey. Their expertise and wisdom have been instrumental in shaping our research and providing us with valuable insights and feedback.

We are grateful for their timely feedback and encouragement, which helped us overcome challenges and stay motivated throughout the process. We could not have completed this research without their help and guidance, and we are proud to have had them as our mentors.

We would also like to thank our colleagues and peers for their valuable input and feedback on our research. Their support and collaboration have been invaluable in shaping our research and helping us to achieve our goals.

References

[1] Abolfazl Abdollahi et al. "Multi-Object Segmentation in Complex Urban Scenes from High-Resolution Remote Sensing Data". In: *Remote Sensing* 13.18 (2021). ISSN: 2072-4292. DOI: 10.3390/rs13183710. URL: https://www.mdpi.com/2072-4292/13/18/3710.

[2] Waleed Alsabhan and Turky Alotaiby. "Automatic Building Extraction on Satellite Images Using Unet and ResNet50". In: *Computational Intelligence and Neuroscience* 2022 (Feb. 2022), pp. 1–12. DOI: 10.1155/2022/5008854.

[3] Luca Bertinetto et al. *Fully-Convolutional Siamese Networks for Object Tracking*. 2016. DOI: 10.48550/ARXIV.1606.09549. URL: https://arxiv.org/abs/1606.09549.

[4] Cong Cao, Suzana Dragićević, and Songnian Li. "Land-Use Change Detection with Convolutional Neural Network Methods". In: *Environments* 6.2 (2019). ISSN: 2076-3298. DOI: 10.3390/environments6020025. URL: https://www.mdpi.com/2076-3298/6/2/25.

[5] Bodhiswatta Chatterjee and Charalambos Poullis. "On Building Classification from Remote Sensor Imagery Using Deep Neural Networks and the Relation Between Classification and Reconstruction Accuracy Using Border Localization as Proxy". In: *2019 16th Conference on Computer and Robot Vision (CRV)*. 2019, pp. 41–48. DOI: 10.1109/CRV.2019.00014.

[6] Hao Chen, Wenyuan Li, and Zhenwei Shi. "Adversarial Instance Augmentation for Building Change Detection in Remote Sensing Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–16. DOI: 10.1109/TGRS.2021.3066802.

[7] Hao Chen, Zipeng Qi, and Zhenwei Shi. "Remote Sensing Image Change Detection With Transformers". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–14. DOI: 10.1109/tgrs.2021.3095166. URL: https://doi.org/10.1109/TGRS.2021.3095166.

[8] Hao Chen and Zhenwei Shi. "A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection". In: *Remote Sensing* 12.10 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12101662. URL: https://www.mdpi.com/2072-4292/12/10/1662.

[9] Hao Chen and Zhenwei Shi. "A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection". In: *Remote Sensing* 12.10 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12101662. URL: https://www.mdpi.com/2072-4292/12/10/1662.

[10] Rodrigo Caye Daudt, Bertrand Le Saux, and Alexandre Boulch. *Fully Convolutional Siamese Networks for Change Detection*. 2018. DOI: 10.48550/ARXIV.1810.08462. URL: https://arxiv.org/abs/1810.08462.

[11] Sounak Dey et al. *SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification*. 2017. DOI: 10.48550/ARXIV.1707.02131. URL: https://arxiv.org/abs/1707.02131.

[12] David Freedman, Robert Pisani, and Roger Purves. "Statistics (international student edition)". In: *Pisani, R. Purves, 4th edn. WW Norton & Company, New York* (2007).

[13] Yaroslav Ganin et al. "Domain-Adversarial Training of Neural Networks". In: (2015). DOI: 10.48550/ARXIV.1505.07818. URL: https://arxiv.org/abs/1505.07818.

[14] Noel Gorelick et al. "Google Earth Engine: Planetary-scale geospatial analysis for everyone". In: *Remote Sensing of Environment* 202 (2017). Big Remotely Sensed Data: tools, applications and experiences, pp. 18–27. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2017.06.031. URL: https://www.sciencedirect.com/science/article/pii/S0034425717302900.

[15] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[16] Thorsten Hoeser and Claudia Kuenzer. "Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends". In: *Remote Sensing* 12 (May 2020). DOI: 10.3390/rs12101667.

[17] Jie Hu et al. *Squeeze-and-Excitation Networks*. 2017. DOI: 10.48550/ARXIV.1709.01507. URL: https://arxiv.org/abs/1709.01507.

[18] Vladimir I. Iglovikov et al. "TernausNetV2: Fully Convolutional Network for Instance Segmentation". In: *CoRR* abs/1806.00844 (2018). arXiv: 1806.00844. URL: http://arxiv.org/abs/1806.00844.

[19] Guoliang Kang et al. *Contrastive Adaptation Network for Unsupervised Domain Adaptation*. 2019. DOI: 10.48550/ARXIV.1901.00976. URL: https://arxiv.org/abs/1901.00976.

[20] H. Kuhn. "The Hungarian Method for the Assignment Problem". In: *Naval Research Logistic Quarterly* 2 (May 2012).

[21] Ming-Yu Liu and Oncel Tuzel. *Coupled Generative Adversarial Networks*. 2016. DOI: 10.48550/ARXIV.1606.07536. URL: https://arxiv.org/abs/1606.07536.

[22] Microsofts. *GlobalMLBuildingFootprints*. 2022. URL: https://github.com/microsoft/GlobalMLBuildingFootprints.

[23] Microsofts. *USBuildingFootprints*. 2018. URL: https://github.com/microsoft/USBuildingFootprints.

[24] Bipul Neupane, Teerayut Horanont, and Jagannath Aryal. "Deep Learning-Based Semantic Segmentation of Urban Features in Satellite Images: A Review and Meta-Analysis". In: *Remote Sensing* 13.4 (2021). ISSN: 2072-4292. DOI: 10.3390/rs13040808. URL: https://www.mdpi.com/2072-4292/13/4/808.

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *MICCAI*. 2015.

[26] Batuhan Sariturk et al. "FEATURE EXTRACTION FROM SATELLITE IMAGES USING SEGNET AND FULLY CONVOLUTIONAL NETWORKS (FCN)". In: *International Journal of Engineering and Geosciences* (Oct. 2020). DOI: 10.26833/ijeg.645426.

[27] Li Shen et al. "S2Looking: A Satellite Side-Looking Dataset for Building Change Detection". In: *Remote Sensing* 13.24 (Dec. 2021), p. 5094. DOI: 10.3390/rs13245094. URL: https://doi.org/10.3390/rs13245094.

[28] ASHBINDU SINGH. "Review Article Digital change detection techniques using remotely-sensed data". In: *International Journal of Remote Sensing* 10.6 (1989), pp. 989–1003. DOI: 10.1080/01431168908903939.

[29] David M. Szpakowski and Jennifer L. R. Jensen. "A Review of the Applications of Remote Sensing in Fire Ecology". In: *Remote Sensing* 11.22 (2019). ISSN: 2072-4292. DOI: 10.3390/rs11222638. URL: https://www.mdpi.com/2072-4292/11/22/2638.

[30] Onur Tasar et al. "ColorMapGAN: Unsupervised Domain Adaptation for Semantic Segmentation Using Color Mapping Generative Adversarial Networks". In: *IEEE Transactions on Geoscience and Remote Sensing* 58.10 (Oct. 2020), pp. 7178–7193. DOI: 10.1109/tgrs.2020.2980417. URL: https://doi.org/10.1109/TGRS.2020.2980417.

[31] Yanheng Wang et al. "Mask DeepLab: End-to-end image segmentation for change detection in high-resolution remote sensing images". In: *International Journal of Applied Earth Observation and Geoinformation* 104 (2021), p. 102582. ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2021.102582. URL: https://www.sciencedirect.com/science/article/pii/S0303243421002890.

[32] M. Weiss, F. Jacob, and G. Duveiller. "Remote sensing for agricultural applications: A meta-review". In: *Remote Sensing of Environment* 236 (2020), p. 111402. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2019.111402. URL: https://www.sciencedirect.com/science/article/pii/S0034425719304213.

[33] Garrett Wilson and Diane J. Cook. *A Survey of Unsupervised Deep Domain Adaptation*. 2018. DOI: 10.48550/ARXIV.1812.02849. URL: https://arxiv.org/abs/1812.02849.

[34] Chen Wu, Liangpei Zhang, and Bo Du. "Kernel Slow Feature Analysis for Scene Change Detection". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.4 (2017), pp. 2367–2384. DOI: 10.1109/TGRS.2016.2642125.

[35] Lin Wu et al. "A Segmentation Based Change Detection Method for High Resolution Remote Sensing Image". In: *Pattern Recognition*. Ed. by Shutao Li, Chenglin Liu, and Yaonan Wang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 314–324. ISBN: 978-3-662-45646-0.

[36] Yinghui Xiao and Qingming Zhan. "A review of remote sensing applications in urban planning and management in China". In: *2009 Joint Urban Remote Sensing Event*. 2009, pp. 1–5. DOI: 10.1109/URS.2009.5137653.

[37] Lu Xu et al. "The comparative study of three methods of remote sensing image change detection". In: *2009 17th International Conference on Geoinformatics*. 2009, pp. 1–4. DOI: 10.1109/GEOINFORMATICS.2009.5293490.

[38] Donggeun Yoo et al. *Pixel-Level Domain Transfer*. 2016. DOI: 10.48550/ARXIV.1603.07442. URL: https://arxiv.org/abs/1603.07442.