Embedding Domain-Invariant Building Segmentation Information On Change Detection Model

Minh-Khoa Le[1,2], Dinh Phong Vo, Bac Le[1,2*]

[1]Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam.
[2]Vietnam National University, Ho Chi Minh City, Vietnam.

*Corresponding author(s). E-mail(s): lhbac@fit.hcmus.edu.vn;
Contributing authors: 18120415@student.hcmus.edu.vn; dpvo@tuta.io;

Abstract

The use of remote sensing data in combination with deep learning methods has resulted in great successes on tasks such as building segmentation and change detection. While some studies have demonstrated that supervised deep learning approaches can outperform traditional methods, which use algebra or transformation techniques with hand-crafted features, these methods are often limited to small datasets and 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 adaptation segmentation model for building segmentation and apply it to the problem of change detection. In addition, empirical evaluations show that our proposed methods can outperform some competitive baselines on popular benchmark datasets.

Keywords: deep learning, remote sensing, change detection, building segmentation

1 Introduction

Satellite imagery is a rich source of information to solve many humanitarian issues[21], [23]. As the number of satellites has increased, the amount of remote sensing data available has also grown rapidly [15], [16] providing a wealth of valuable information

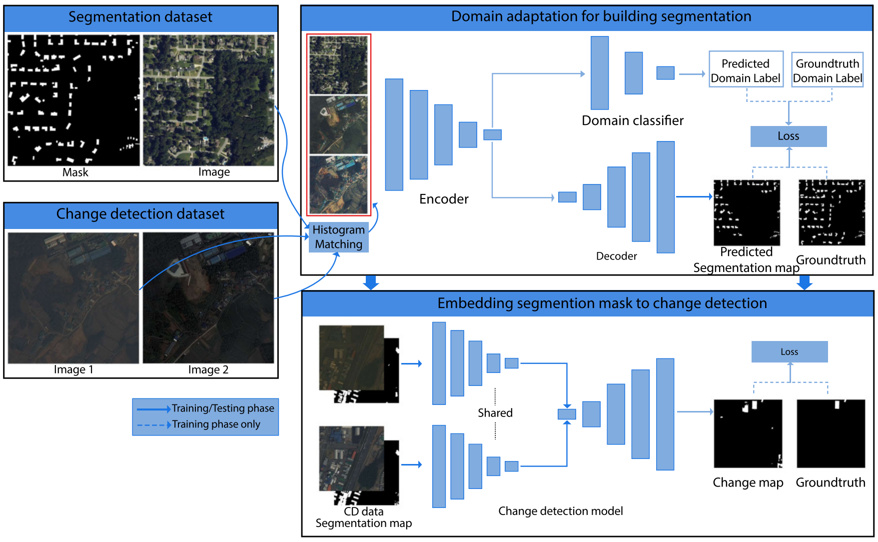


Fig. 1: Overview of our proposed method including two parts: domain adaptation for building segmentation and embedding segmentation mask to change detection. In the first part, after being applied histogram matching algorithm, images from segmentation and change detection datasets are fed into our domain adaptation segmentation model to extract segmentation masks and predict image domains. In the training phase, losses are calculated to optimize the model. In the second part, segmentation information is used further with images from the change detection dataset to predict the change map. In the training phase, we use a predicted map and ground truth to optimize our model.

that can be applied to a variety of human problems [11]. As a result, the goal of quickly detecting or calculating insights from satellite images becomes possible, realizing the desire to identify or assess damage from calamities quickly. The vast amount of remote sensing images also 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.

Building segmentation and change detection are two crucial problems in remote sensing. Building segmentation is a classification task that extracts building from a satellite image [17]. Change detection is the process of identifying objects' differences in the same geographical location at different times [20]. They are widely used in many applications, such as agricultural monitoring [11].

Due to the explosive growth in the quantity and quality of satellites, deep learning models, which result in an abundant source of high-quality satellite images, have shown outstanding performance in remote sensing tasks [4], [6]. However, supervised deep learning power heavily relies on hand-labeling datasets, which is resources consuming [5]. Moreover, distribution shifts between remote sensing datasets cause the model's poor performance across multiple datasets.

An important characteristic of satellite images is the off-nadir angle, the angle between the satellite and the target at the time of collection. Satellites typically capture images at large off-nadir angles to collect more data. However, this results in poor geometric consistency for tall objects when multiple off-nadir angles are introduced. Among few datasets that consist of multiple-angle images, Shen et al. have introduced a new change detection dataset called S2Looking [19], which includes off-nadir angles ranging from $-35°$ to $35°$.

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 [13]. Researchers have improved their model's performance by embedding segmentation masks or network architecture in change detection [22], [26].

Our contributions. To summarize, we make the following contributions.

1. We propose Domain Adaptation Segmentation Model (DASM) for building segmentation. We also apply many pre-processing methods to achieve better evaluation scores on unseen data with distribution shifts.
2. We embed building segmentation information into input data to improve the change detection baseline.
3. We conduct experiments to demonstrate our methods against baselines and SOTA on popular remote sensing datasets, where our methods show better F1-Score than some previous works.

2 Related Work

2.1 Building segmentation for satellite imagery

Building segmentation is a binary segmentation task that aims to extract buildings from satellite images. Many researchers have designed their networks based on UNet architecture because of its excellent performance on the generic image segmentation task[12]. Iglovikov et al. introduce the TernausNetV2 [13] with the improvement from UNet by using WideResNet-38 network with in-place activated batch normalization as the encoder, showing superior performance in the building segmentation task. Nevertheless, those methods suffered drastically from image contrast, brightness, and off-nadir angles [11].

2.2 Change detection

Change detection became possible after the existence of satellite images. In the past, image differencing was the primary technique [20]. However, it easily suffers from changes that are not in favour. For example, sessional changes can create differences in color at different times, dramatically decreasing the image differencing method performance. Recently, deep learning methods have been developed with improved accuracy since the rise of Convolutional Neural Networks (CNNs) [4], [6]. Some of them are post-classification methods whose structures contain two classifiers to classify the bi-temporal images [2], [25]. Then, the change map can be obtained by comparing classification maps.

Many attempts have been 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 UNet, and Siamese Net [6]. However, deep-learning-based methods require a large amount of computing resources, which are not always available. Moreover, the performance of those models decreases vastly when heterogeneous data are introduced.

2.3 Domain adaptation

One of the most important problems of machine learning is domain shift. If a machine learning model is trained with a distribution - or domain of data, we can not expect it to perform well on other domains. As a result, the field domain adaptation was born to improve the model on a target domain using the knowledge from another source domain.

One of the main domain adaptation methods is domain-invariant feature learning. It focuses on aligning source and target domains by producing domain-invariant feature representations [24]. In particular, a network can produce domain-invariant feature representations if the output feature follows the same distribution, whether input comes from the source or target domain.

Adversarial-based domain adaptation method aims to minimize differences between domains via an adversarial objective. A method called Domain Adversarial Training of Neural Networks (DANN) uses domain confusion loss to confuse high-level layers about the input's domain [9]. 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 representations and perform well on the label prediction task. On the other hand, those methods are partly invariant to input domains that we need to retrain our models when a new data domain comes. Additionally, domain adaptation methods do not perform well as traditional supervised manner [9].

3 Proposed Methods

Our idea is to use segmentation masks generated from our segmentation model as an additional information channel of the input images, making our change detection model more aware of building entities in the image. An overview of the method is shown in Fig. 1.

In the first step, we use a UNet [1] architecture as the segmentation model. We got training and test dataset which are from different settings in satellite, space and time. As a result, our two datasets come from two different domains, which may make the model perform poorly. Therefore, we introduce the domain adaptation technique in the building segmentation task to address this problem. Moreover, pre-processing methods are applied to improve the model, including per-image normalization and histogram matching.

3.1 Domain adaptation for building segmentation

3.1.1 Baseline segmentation model

A baseline model inspired by TernausNetV2 [13] is chosen to solve domain adaptation building segmentation. Concretely, the model has an encoder-decoder architecture, and the backbone is ImageNet pre-trained ResNet-50 [10]. ResNet has been proposed and is well-known for the ease of training networks substantially deeper than those used previously [10]. Moreover, skip connections are added at each convolution block, allowing features from the input image and encoders to flow directly to high-level feature maps, improving localization accuracy and speeding up the training process [18].

3.1.2 Domain adaptation segmentation model

On the baseline segmentation model, we apply the Domain-Adversarial Neural Networks (DANN) technique [9] 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 simple neural network with dropout layers - binary domain classifier is added to classify images from different domains (the domain classifier in Fig. 1). The classifier's task is to output a value close to 1 when the input is from the training dataset domain and 0 when the input is from the test dataset domain.

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 are 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 are defined as:

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

λ is a hyperparameter controlling the importance of the domain classification loss. We can minimize function 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 classification loss from segmentation loss, denoting that we aim to confuse the decoder module about the image domains by maximizing the domain classification loss. We expect the subtraction makes the loss converge to a saddle point, where domain classification loss is maximized while minimizing the segmentation loss. Such subtraction can be implemented easily by a gradient reversal layer [9]. 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 as the learning rate. Converging to the saddle point, the encoder fails to distinguish domains. As a result, the encoder generates domain-invariant features.

3.1.3 Pre-processing methods

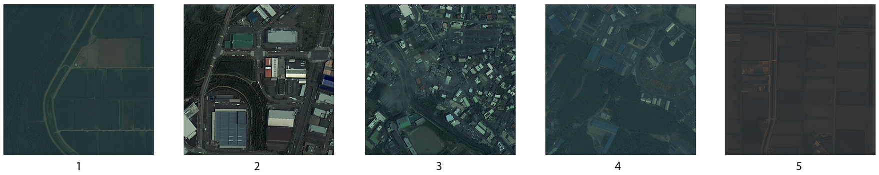
We call the combination of DASM and pre-processing methods as Improved Domain-Adaptation Segmentation Model (I-DASM). The structure of our model has illustrated in Fig. 1.

Per-image normalization

Image normalization is the process of bringing images with different ranges of pixel intensity into the same range. As a result, images from different domains share the same contrast.

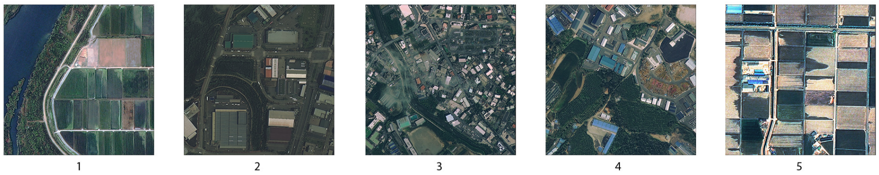
$$\hat{x_i} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{7}$$

In equation 7, μ, σ is the mean and standard deviation of the 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.



(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.

Histogram matching

Histogram matching is the process of adjusting gamma and saturation to correct colors to produce an output image that shares an equivalent color distribution with the input image. Each input for the histogram matching contains a pair of images, a grayscale source image S that needed to be applied histogram matching and a grayscale reference image R. For an image T, we have a probability density function p^T that can be computed from its color histogram:

$$p^T(a_i) = \frac{n_i^T}{n^T} \tag{8}$$

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.

$$F^T(a_j) = \sum_{i=0}^{j} p^T(a_i), \quad j = 0, ..., G - 1 \tag{9}$$

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 various images from the training dataset as reference images. Each source image is compared with the collection, and select the most similar to it. To compare two images, we use Pearson's Correlation Coefficient [8] as the similarity metric. The higher the metric is, the more similar they are. The metric Pearson's Correlation Coefficient s between two histograms H_1 and H_2 can be calculated as

$$s(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 \sum_I (H_2(I) - \bar{H}_2)^2}} \tag{10}$$

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

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

3.2 Embedding Segmentation to Change Detection

3.2.1 Baseline change detection model

We use the baseline Siamese UNet [18] architecture with ResNet-50 [10] as the backbone. The network employs an encoder-decoder design based on FC-Siam-Diff [6]. The encoding layers of the network are separated into two streams of similar structure with shared weights as in a traditional Siamese network [7] 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, gets the absolute value of their subtraction before feeding them to the decoders. It makes the network simpler and helps it learn to compare the differences between images [6].

3.2.2 Improved Change Detection Model (ICDM)

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 probability building segmentation mask (Fig. 1). Concretely, the segmentation mask is the output of the I-DASM.

4 Experiments

We conduct experiments to evaluate our methods against baselines and existing methods:

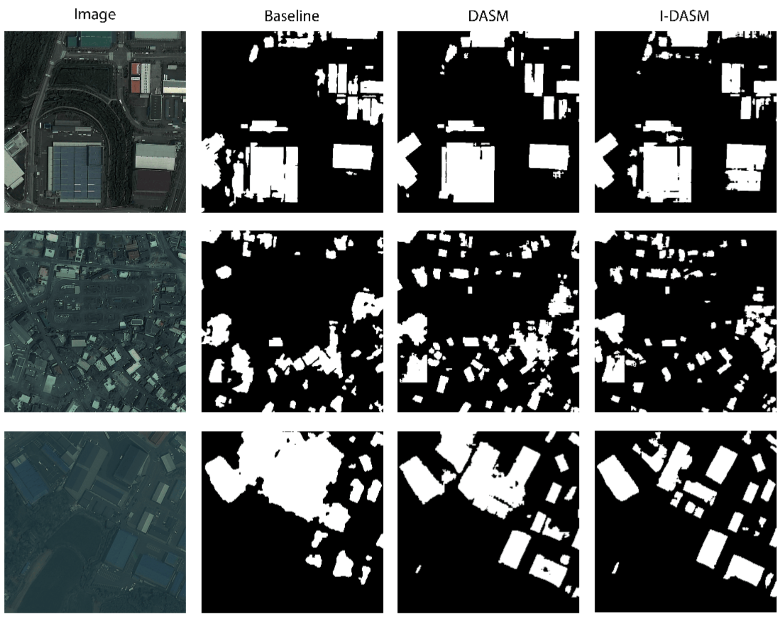


Fig. 3: Images from S2Looking dataset and segmentation results of models. Each row contains a sample from the dataset and outputs from models.

1. Performance of our proposed method against baseline in segmentation task on Alabama dataset 4.1 and change detection task on S2Looking dataset [19].
2. Comparing our model with many existing methods and baseline on segmentation task using S2Looking dataset.

Our code and instruction is available at https://github.com/ktncktnc/ResUnet/

4.1 Experiments details

We collected Alabama satellite images from Bing Maps, combined with US Building Footprint [15] to create a building segmentation dataset called Alabama dataset. 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.

There are no ground-truth segmentation labels in the S2Looking dataset, so we use change detection mask as the label to compare segmentation models. With a pair of input RGB images from the change detection dataset, segmentation models result in 2 masks. We extract buildings from segmentation masks into a set of objects. And then using the Hungarian algorithm [14] 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. Evaluation metrics on both tasks are Precision, Recall, and F1-Score.

During training, we use the Adam optimizer with a learning rate of 0.01. The learning rate is reduced by a factor of ten every ten steps. Our objective function

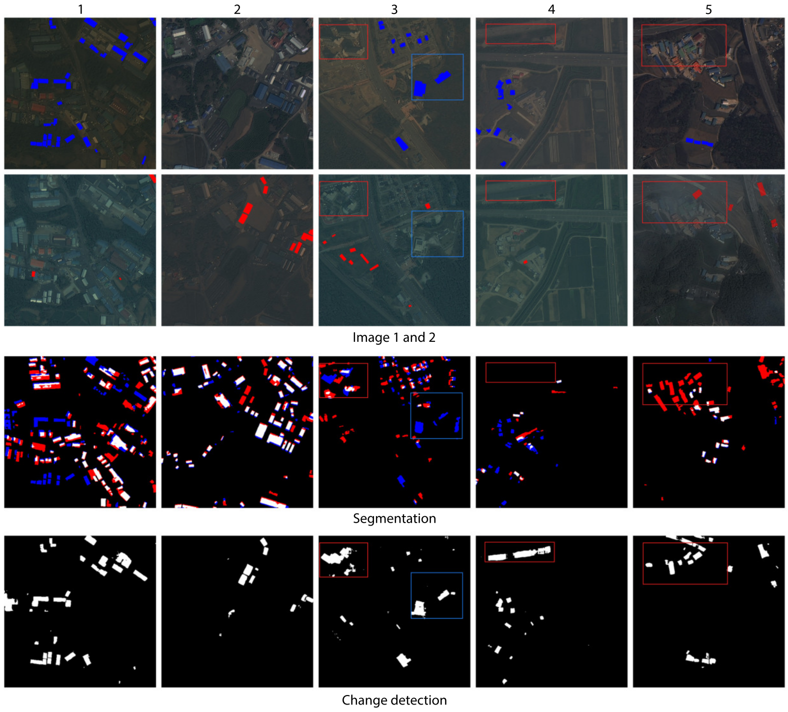


Fig. 4: Result of S2Looking dataset on segmentation and change detection tasks. First two rows present pairs of before and after images. Each image contains red and blue objects representing buildings that only exist in the other image at the same place. The segmentation row shows our I-DASM model results on both before and after images: White pixels are buildings that appear in both images, and blue and red pixels only appear in images 1 and 2, respectively. The last row is the outputs of our change detection model.

combines Cross-Entropy Loss and Dice Loss. For the change detection task, we randomly select six 512×512 parts from each pair of images and the change detection mask using random cropping to address class imbalance and prevent overfitting. Our pre-processing phase includes an augmentation pipeline with random shift, scale, rotation, RGB shift, brightness, and contrast to enhance model robustness. Each method is trained for 20 epochs with these settings.

4.2 Experiments on segmentation models

In table 1, we show the F1-Score of our models on two different tasks: segmentation in the Alabama dataset and change detection in the S2Looking [19] dataset. Although the baseline shows excellent performance on the segmentation task - at 0.8421, it is not good at the change detection in S2Looking [19] with a deficient F1-Score. Our DASM improves the change detection F1-Score while preserving a high segmentation

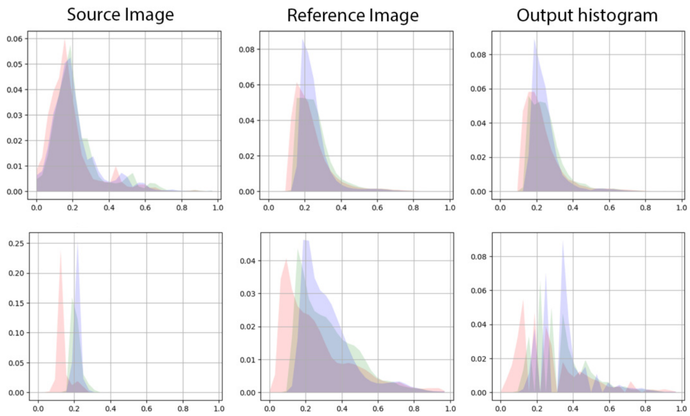


Fig. 5: Histograms of source, reference, and output images. Each row presents a sample from the dataset. Each graph has red, green, and blue histograms corresponding to the red, green, and blue channels.

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.

Table 1: Result for the evaluated segmentation methods

Method	**Alabama F1-Score**	**S2Looking F1-Score**
Baseline (Section 3.1.1)	0.8412	0.1520
DASM (Section 3.1.2)	0.8020	0.1950
I-DASM (Section 3.1.3)	0.8336	0.2084

Fig. 3 shows that the baseline easily suffers from the color differences between source and target domains. When buildings and backgrounds are overlapped, it fails to distinguish and classifies background pixels as building pixels. The problem becomes more severe when the color distribution of the input image is unusual (Last row of Fig. 3). On the other hand, DASM shows better performance on the test set. It better separates buildings and backgrounds and recognizes the border between them. Furthermore, I-DASM improves the result on the target domain (S2Looking dataset) and the source domain (Alabama dataset). Thanks to the domain adaptation component, the encoder can extract features from input partly invariant to input domains.

With low-contrast source images from S2Looking and reference images from the Abalama, histogram matching results in images with the same color distribution as reference images, better brightness and contrast. The first sample in Fig. 5 shows that the output histogram resembles the reference one. This is the case for the first 4 samples in Fig. 2, where output and reference images share the same brightness and contrast, minimizing the domain difference.

Table 2: Result of our change detection methods and other methods on S2Looking.

Method	**Precision**	**Recall**	**F1-Score**
FC-EF [6]	**0.8136**	0.0895	0.1613
FC-Siam-Diff [6]	**0.8329**	0.1576	0.2650
STANet-Base [3]	0.2575	**0.5629**	0.3534
CDNet [4]	0.6748	0.5493	**0.6056**
I-DASM (Section 3.1.3)	0.1393	0.4133	0.2084
Baseline (Section 3.2.2)	0.5731	0.5514	0.5621
ICDM (Section 3.2.2)	0.5454	**0.6051**	**0.5737**

In the second row of Fig. 5, the algorithm results in significant deviations between some neighboring bins in the histogram when the difference between source and reference images is vast. In sample 5 of Fig. 2), source and reference images are far different, so that the algorithm results in a high contrast and brightness output image. Moreover, histogram matching does not give attention to the image's context and even tries to match the building colors with landscape colors. Additionally, histogram matching fails with images containing big clouds, skewing the color distribution.

4.3 Experiments on change detection models

In table 2, we observed that our baseline achieved F1-Score, at 0.5621, higher than some previous methods, such as FC-Siam-Diff [6]. By providing the baseline building information, our method is improved by over 1%, getting a higher F1-Score at 0.5737. Our methods archive higher F1-Scores than other methods, such as FC-Siam-Diff [6], and STANet-Base [3]. For S2Looking [19] dataset, CDNet [4] performs better than our model, at an F1-Score of 0.6056, around 3% higher than our object-level change detection model.

Fig. 4 shows our ICDM 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. However, ICDM does not improve much compared to the baseline. We hypothesize that during the training phase, the baseline has also learned the features of buildings, and it has the ability to recognize differences between buildings and backgrounds. As a result, incorporating segmentation information is not helpful for our baseline model.

ICDM does not solve the current problems of the baseline. It still suffers from the off-nadir angle. Although there is no change in buildings, the two images' off-nadir angles vary. Our model also fails to recognize buildings when there are clouds in the image. In sample 5 in Fig. 4, both segmentation and change detection models perform well in clear places but fail completely in the opaque area.

5 Conclusion

In this paper, we proposed method I-DASM to address the issue of distribution shifts in building segmentation tasks. By incorporating pre-processing methods with domain-adaptation neural network, our model is able to reduce the gap between different domains and improve the performance of the model on the test set without sacrificing its performance on the training set. Furthermore, we also offer a potential solution for improving change detection models using available data without the need for extensive manual labelling.

On the other hand, our article has some limitations. First, we have not explained the underlying theory of DANN and how it works. Second, the histogram matching algorithm is still insufficient and results in poor performance in many cases. Due to the naive approach without concern the content of the image, the algorithm fails when there are abnormal objects (e.g. big clouds). Lastly, embedding segmentation masks as the fourth channel of input of change detection does not improve the model.

Acknowledgment

This research is supported by research funding from the Faculty of Information Technology, University of Science, Vietnam National University - Ho Chi Minh City.
Authors' contribution:

- Minh-Khoa Le: Conceptualization, Methodology, Software, Writing – original draft & editing.
- Dinh - Phong Vo: Methodology, Formal analysis, Visualization, Writing – review & editing.
- Bac Le: Conceptualization, Methodology, Supervision, Review, Validation Declarations.

Data availability. The datasets generated and/or analysed during the current study are available in the GitHub repository, https://github.com/ktncktnc/ResUnet.
Code Availability. The code implemented during the current study are available in the GitHub repository, https://github.com/ktncktnc/ResUnet.
Declarations

- Informed consent. Informed consent was obtained from all of the subjects involved in this study.
- Competing interests. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

References

[1] Alsabhan W, Alotaiby T (2022) Automatic building extraction on satellite images using unet and resnet50. Computational Intelligence and Neuroscience 2022:1–12

[2] Cao C, Dragićević S, Li S (2019) Land-use change detection with convolutional neural network methods. Environments 6(2)

[3] Chen H, Shi Z (2020) A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. Remote Sensing 12(10)

[4] Chen H, Li W, Shi Z (2022) Adversarial instance augmentation for building change detection in remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 60:1–16

[5] Chen H, Qi Z, Shi Z (2022) Remote sensing image change detection with transformers. IEEE Transactions on Geoscience and Remote Sensing 60:1–14

[6] Daudt RC, Saux BL, Boulch A (2018) Fully convolutional siamese networks for change detection

[7] Dey S, Dutta A, Toledo JI, et al (2017) Signet: Convolutional siamese network for writer independent offline signature verification

[8] Freedman D, Pisani R, Purves R (2007) Statistics (international student edition). Pisani, R Purves, 4th edn WW Norton & Company, New York

[9] Ganin Y, Ustinova E, Ajakan H, et al (2016) Domain-adversarial training of neural networks

[10] He K, Zhang X, Ren S, et al (2015) Deep residual learning for image recognition

[11] Hoeser T, Kuenzer C (2020) Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends. Remote Sensing 12

[12] Iglovikov V, Shvets A (2018) Ternausnet: U-net with VGG11 encoder pre-trained on imagenet for image segmentation. CoRR 1801.05746

[13] Iglovikov VI, Seferbekov SS, Buslaev AV, et al (2018) Ternausnetv2: Fully convolutional network for instance segmentation. CoRR 1806.00844

[14] Kuhn H (2012) The hungarian method for the assignment problem. Naval Research Logistic Quarterly 2

[15] Microsofts (2018) United states microsoft building footprints

[16] Microsofts (2022) Global microsoft building footprints

[17] Neupane B, Horanont T, Aryal J (2021) Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis. Remote Sensing 13(4)

[18] Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: MICCAI

[19] Shen L, Lu Y, Chen H, et al (2021) S2looking: A satellite side-looking dataset for building change detection. Remote Sensing 13(24):5094

[20] SINGH A (1989) Review article digital change detection techniques using remotely-sensed data. International Journal of Remote Sensing 10(6):989–1003

[21] Szpakowski DM, Jensen JLR (2019) A review of the applications of remote sensing in fire ecology. Remote Sensing (22)

[22] Wang Y, Gao L, Hong D, et al (2021) Mask deeplab: End-to-end image segmentation for change detection in high-resolution remote sensing images. International Journal of Applied Earth Observation and Geoinformation 104:102582

[23] Weiss M, Jacob F, Duveiller G (2020) Remote sensing for agricultural applications: A meta-review. Remote Sensing of Environment p 111402

[24] Wilson G, Cook DJ (2018) A survey of unsupervised deep domain adaptation

[25] Wu C, Zhang L, Du B (2017) Kernel slow feature analysis for scene change detection. IEEE Transactions on Geoscience and Remote Sensing (4):2367–2384

[26] Wu L, Zhang Z, Wang Y, et al (2014) A segmentation based change detection method for high resolution remote sensing image. In: Li S, Liu C, Wang Y (eds) Pattern Recognition. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 314–324