Background-Mixed Augmentation for Weakly Supervised Change Detection

Rui Huang¹, Ruofei Wang¹, Qing Guo² *, Jieda Wei¹, Yuxiang Zhang¹, Wei Fan¹, Yang Liu³,4
¹ School of Computer Science and Technology, Civil Aviation University of China, China
² Center for Frontier AI Research (CFAR), A*STAR, Singapore
³ Zhejiang Sci-Tech University, China
⁴ Nanyang Technological University, Singapore
{rhuang, yxzhang, wfan}@cauc.edu.cn, {rf_wang, jdwei1}@yeah.net, tsingqguo@ieee.org, yangliu@ntu.edu.sg

Abstract

Change detection (CD) is to decouple object changes (i.e., object missing or appearing) from background changes (i.e., environment variations) like light and season variations in two images captured in the same scene over a long time span, presenting critical applications in disaster management, urban development, etc. In particular, the endless patterns of background changes require detectors to have a high generalization against unseen environment variations, making this task significantly challenging. Recent deep learning-based methods develop novel network architectures or optimization strategies with paired-training examples, which do not handle the generalization issue explicitly and require huge manual pixel-level annotation efforts. In this work, for the first attempt in the CD community, we study the generalization issue of CD from the perspective of data augmentation and develop a novel weakly supervised training algorithm that only needs image-level labels. Different from general augmentation techniques for classification, we propose the background-mixed augmentation that is specifically designed for change detection by augmenting examples under the guidance of a set of background changing images and letting deep CD models see diverse environment variations. Moreover, we propose the augmented & real data consistency loss that encourages the generalization increase significantly. Our method as a general framework can enhance a wide range of existing deep learning-based detectors. We conduct extensive experiments in two public datasets and enhance four state-of-the-art methods, demonstrating the advantages of our method. We release the code at https://github.com/tsingqguo/bgmix.

Introduction

Change detection (CD) is to segment object changes (i.e., object missing or appearing) from background changes (i.e., environment variations) like light and season variations in two images captured in the same scene with a long time span (See examples shown in Fig. 1), presenting critical applications in the disaster management (Sublime and Kalinicheva 2019), urban development (Lee et al. 2021), resource monitoring, and utilization (Ye et al. 2021), and visual surveillance (Goyette et al. 2012), etc.

*Qing Guo is the corresponding author: tsingqguo@ieee.org. Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Figure 1: Comparison of three SOTA methods with/without our BGMix. Clearly, BGMix improves the three methods with significant F1 increases on both AICD and BCD datasets, presenting more accurate segmentation results.

Change detection is still challenging and far from real-world applications due to the complex and unknown environment variations. For example, in Fig. 1, there exist various differences on the lawn (e.g., color and density differences) between two images, which should be identified as background changes and not be segmented. Nevertheless, the state-of-the-art (SOTA) method WCDNet (Andermatt and Timofte 2021) focuses on the lawn variations while missing the desired object changes (i.e., the disappearance of a building). Other environment variations like season and illumination variations cause similar issues. The endless patterns of background changes require detectors to have a high generalization against unseen environment variations.

In recent years, a series of network architectures are designed with convolutional (Zhang et al. 2021; Zheng et al. 2021) and transformer layers (Hao Chen and Shi 2021; Ban-
dara and Patel (2022) to extract effective features to represent the changes. Although achieving impressive progress, they do not handle the generalization issue explicitly and usually require huge pixel-level annotation efforts. The weakly supervised strategy is an effective solution to alleviate the pixel-level annotation efforts (Andermatt and Timofte 2021; Wu, Du, and Zhang 2023; Ru et al. 2022). However, without pixel-level supervision, change detectors suffer from low detection accuracy due to complex background changes. The root reason for this dilemma is the training data does not cover diverse environment variations.

In this work, for the first attempt in the CD community, we study the generalization issue from the perspective of data augmentation and develop a novel weakly supervised algorithm that only needs image-level labels. Different from general augmentation methods for classification, we propose the background-mixed augmentation that is specifically designed for change detection by augmenting examples under the guidance of a set of background images and letting targeted deep models see diverse background changes. Moreover, we propose the augmented & real data consistency loss that encourages high generalization significantly. Our method as a general framework can enhance a wide range of existing deep learning-based detectors. As presented in Fig. 1, with our augmentation method, the performance of three change detectors is improved with large margins. Our contributions can be summarized as follows:

- We study the generalization issue of the CD from the data augmentation perspective and propose a novel weakly supervised change detection method denoted as BGMix.
- We propose a background-mixed augmentation that is costumed for change detection and augmented & real data consistency loss to encourage the generalization.
- We conduct extensive experiments in two public datasets and enhance four state-of-the-art methods, demonstrating the advantages of our method.

Related Work

Change detection methods. Conventional change detection methods are based on simple algebraic methods, such as gradient (Di Stefano, Matteo, and Mola 2003), change vector analysis (CVA) (Malila 1980) and thresholding (Rosin 1998). A simple strategy cannot deal with background change noises. More complex models are introduced such as Markovian data fusion (Moser, Angiati, and Serpico 2011) and dictionary learning (Gong et al. 2016a). To obtain a high-quality CD, (Feng et al. 2015) and (Stent et al. 2016) also propose image alignment and lighting correction before CD. With the development of deep learning, different deep models are introduced into CD, such as restricted Boltzmann machine (RBM) (Gong et al. 2016b), CNN (Ding et al. 2016), GAN (Kousuke, Kanji, and Takuma 2017; Gong et al. 2017), RCNN (Mou, Bruzzone, and Zhu 2019), LSTM (Lyu, Lu, and Mou 2016), and Transformer (Hao Chen and Shi 2021). Compared with RBM and GAN, it is easier to design variant CNN-based network architectures for CD. A representative network, ADCDnet (Huang et al. 2020) proposes to use absolute difference features of multiscale convolutional layers for CD. To extract more effective features, position- and channel-correlated attentions are incorporated into CD by Zhang et al. (Zhang et al. 2021). The current trend is using transformers for CDs. ChangeFormer (Bandara and Patel 2022) unifies hierarchically structured transformer encoder with Multi-Layer Perceptron (MLP) decoder in Siames network architecture to render multi-scale long-range details for CD. The success of the above deep change detectors is due in large part to a huge number of pixel-level labeled images. However, the existing CD datasets cannot contain all situations of environment changes, which limits the generalization ability of the recent change detectors.

Weakly supervised detection methods. The weakly supervised method is an effective way to conquer the needs of pixel-level data. Weakly Supervised Change Detection (WSCD) aims to detect pixel-level changes by using the image-level labels (Khan et al. 2017; Krähenbühl and Koltun 2011). Sakurada et al. (Sakurada, Shibuya, and Wang 2020) divide the semantic change detection to change detection and semantic segmentation and do not need the semantic change annotations. Andermatt et al. (Andermatt and Timofte 2021) design a custom remapping block and employ a CRF-RNN to refine the change mask. Kalita et al. (Kalita, Karatsiolis, and Kamilaris 2021) combine the PCA and K-means algorithm to predict changes from the image difference generated by a Siamese network. With the assumption that changed image pairs can be discriminated as unchanged image pairs by masking out change regions, Wu et al. (Wu, Du, and Zhang 2023) propose a generative adversarial network (GAN) to detect changes. Similar to fully supervised CD, WSCD also suffers from the low generalization issue. The root reason for this dilemma is that the training data cannot contain various environmental changes.

Data augmentation techniques. Data augmentation is an effective solution to improve the generalization capability of deep models. Cutout (DeVries and Taylor 2017) randomly masks out square regions of images to regularize the network training. Random erasing (Zhong et al. 2020) replaces the pixels with random intensities of a randomly selected rectangle region. Cutout or random erasing might result in information loss. Then, CutMix (Yun et al. 2019) cuts patches and pastes them among the training images. Later works pay more attention to mixing, such as MixUp (Huang, Zhang, and Zhang 2020), AugMix (Hendrycks et al. 2020), SnapMix (Huang, Wang, and Tao 2021), AlignMixup (Venkataramanan et al. 2022) etc. Compared with random cropping, SuperMix (Dabouei et al. 2021) and KeepAugment (Gong et al. 2021) preserve the informative regions (i.e. salient regions). AutoAugment (Cubuk et al. 2018) automatically searches augmentation policies in a predefined search space. Deeprepair (Yu et al. 2021) employs the style transformation to use some failure examples for augmentation guidance. There are also augmentation methods for other tasks like deraining (Guo et al. 2021) and night2day transformation (Fu et al. 2021). Existing data augmentation techniques are not suitable for change detection. A good data augmentation method for CD should not only simulate the object changes but also simulate various environment changes.
Problem Formulation and Motivation

Given two images \( I_1 \) and \( I_2 \) that are captured in the same scene with a long time span, the change detection task is to identify object changes in the two images. For example, when an object in \( I_1 \) disappears in \( I_2 \), a CD method is desired to output a binary map (i.e., \( C \)) where the missed object region is identified as one and other positions are assigned zero. Recent deep learning-based methods formulate this task as an image2image mapping task via a convolution neural network (CNN) (Lei et al. 2020), e.g.,

\[
C = \phi_\theta(I_1, I_2)
\]  

(1)

where \( \phi_\theta(\cdot) \) is the CNN and \( \theta \) denotes its parameters. The tensor \( C \) denotes the predicted changing mask. A straightforward implementation is to train the network via paired training examples, i.e., \( \mathcal{T}_{\text{train}} = \{< I_1, I_2, C_{\text{gt}} >\} \) via loss functions like binary cross-entropy, mean squared error (MSE), or \( L_1 \), where \( C_{\text{gt}} \) denotes the ground truth of the example. However, there are two limitations preventing such a solution for real-world applications: it requires laborious human efforts to label the pixel-wise ground truth, which can hardly construct large-scale datasets and the potential labeling noise might affect the model training. The training strategy does not consider the generalization to unseen environment changes (i.e., background variance) that have diverse patterns in different images.

An alternative learning method is the weak supervision based on the generative adversarial network (GAN) with image-level labels (Wu, Du, and Zhang 2023) that only indicates whether an image pair contains object changes or not. Although effective, existing weakly supervised change detection methods neglect the potential environment changes within a long time span, which might be caused by light variation, season variation, weather variation, etc. As a result, they easily generate obvious detection errors on unseen background variations. We take the recent weakly supervised change detection method as an example and use it to handle different image pairs. As shown in Fig. 2, the method can get accurate detection results when the background changes are minor (See the first case in Fig. 2) but produce obvious errors when the background changes have obvious color shifting (e.g., the second case.) Existing data augmentations (e.g., AugMix (Hendrycks* et al. 2020)) for image classification change the whole images at a time and present less effectiveness on background enrichment.

Methodology

Overview

To address the limitations, we propose a background-mixed augmentation (BGMix) to train a deep model (i.e., \( \phi_\theta(\cdot) \)) for change detection in a weakly supervised way. Before going to the main process, we first detail the required inputs:

- **Weakly supervised training set** \( \mathcal{T}_{\text{train}} \). We require a training dataset (i.e., \( \mathcal{T}_{\text{train}} = \{< I_1, I_2 >\}^T \)) with \( T \) examples and image-level labels. We name it a weakly supervised training dataset since we only know whether there exist object changes between \( I_1 \) and \( I_2 \) and do not require the pixel-wise annotations.

- **Background-guided set** \( \mathcal{B} \). We collect \( N \) image pairs where each image pair (e.g., \( < B_1, B_2 > \)) only contains background changes without any object changes. We denote the set as background-guided set, i.e., \( \mathcal{B} = \{< B_1, B_2 >\}^N \) (See Fig. 3 for three background changing images as examples). These background pairs have distinct appearances and are used to enrich the potential background changing patterns in the training examples.

- **Augmentation operation set** \( \mathcal{O} \). We set an operation set \( \mathcal{O} = \{o_1, \ldots, o_M\} \) that contains \( M \) augmentation operations. Each operation is used to transform the image pair \( < I_1, I_2 > \) to a novel one (i.e., \( < I_1^o, I_2^o > \)), e.g.,

\[
I_1^o, I_2^o = o_j(I_1, I_2, \alpha_j),
\]  

(2)

where \( \alpha_j \) denotes the operation-related parameters. For example, if we set the \( j \)th \( o_j(\cdot) \) as the translation operation, \( \alpha_j \) means the shifting distance and direction of the two images and the output images (i.e., \( I_1^o \) and \( I_2^o \)) are the shifted counterparts of the input images. Beyond the simple geometry augmentations like translation, we propose a specific operation (i.e., background-aware augmentation operation) for the CD task, which helps the CD models see different background changes.

Figure 2: Comparison of augmented examples between the AugMix and BGMix. AugMix will transform the whole image pixels randomly, while BGMix can effectively mix the background noise (i.e., cars, buildings) with change regions. We use a state-of-the-art method (i.e., WCDNet) to study the effect of BGMix, and change regions are pointed by the red arrows.
Intuitively, BGMix aims to utilize the background-guided set \( B \), the predicted changing mask \( C \), and the image pair with \( I_1 \) and \( I_2 \) to generate new image pairs with diverse background patterns. As a result, the targeted CNN can see different background interference, and then its generalization could be enhanced.

During the training process, we have a model \( \phi_B \) updated at the previous iteration and sample a training example (e.g., \( < I_1, I_2 > \)) from \( T_{\text{train}} \). Then, we can extract a changing mask by \( C = \phi_B(I_1, I_2) \). After that, we use the proposed augmentation strategy to generate an augmented image pair \( < \tilde{I}_1, \tilde{I}_2 > \) with the pipeline shown in Fig. 3.

Finally, we propose novel loss functions (i.e., augmented & real data consistency loss) to encourage the CNN to predict similar results across the original image pairs and background-augmented image pairs.

**Background-mixed Augmentation**

**Background-aware augmentation operation.** We augment a training example (e.g., \( < I_1, I_2 > \)) by enriching the background changes between the image pair. Instead of the common augmentations like translation or zooming, we use the collected background-guided set (i.e., \( B \)) to replace the original background of the input image pair according to the changing mask \( C \). Specifically, we first sample a background pair (i.e., \( < B_1, B_2 > \)) from the background-guided dataset (i.e., \( B \)) randomly, and then we transform the input image pair with

\[
I_1' = \phi_B(I_1, B_1, C), \quad I_2' = \phi_B(I_2, B_2, C),
\]

which can be detailed as

\[
I_1' = \text{Rep}(I_1, B_1, C) = I_1 \odot C + B_1 \odot (1 - C), \quad I_2' = \text{Rep}(I_2, B_2, C) = I_2 \odot C + B_2 \odot (1 - C),
\]

where \( \odot \) is the element-wise multiplication. The function \( \text{Rep}(I_1, B_1, C) \) is to replace the region in \( I_1 \) indicated by \( (1 - C) \) with the corresponding region in \( B_1 \). Note that, compared to common augmentation operations, the above method changes the background region of the image pair while maintaining the object changing region, leading to new image pairs with different background changes.

**Augmentation strategy (i.e., Line 1-11 in Algorithm 1).**

Inspired by recent augmentation methods for classification (Hendrycks* et al. 2020), we conduct multi-depth random augmentations on the image pair to guarantee the diversity of background variations. Specifically, we set \( K \) augmentation paths. For each patch, we first sample three operations for \( O \) (See Line 5 in Algorithm 1) and stack them to construct new operations (See Line 6 in Algorithm 1). Then, we randomly sample one operation from the newly constructed operations, which is used to augment the input image pair \( I_1, I_2 \) with the \( B \) and change mask \( C \) and get a new image pair. To conduct \( K \) times augmentation, we get \( K \) new image pairs that are mixed with randomly sampled weights (See Line 3 in Algorithm 1). Finally, the mixed image pair is further blended with the original input (See Line 9 in Algorithm 1) and get the final output, i.e., \( < \tilde{I}_1, \tilde{I}_2 > \).

**Augmented & Real Data Consistency Loss**

Intuitively, the CNN for change detection is desired to predict the same results between augmented image pairs (i.e., \( < \tilde{I}_1, \tilde{I}_2 > \)) with different backgrounds and the original image pairs (i.e., \( < I_1, I_2, l > \)). We propose the augmented & real data consistency loss to encourage this property explicitly, which contains five parts.

First, we encourage the perception similarity between <
Algorithm 1: Learning CD models via BGMix

Input: \( \phi_0(\cdot) \), Loss function \( L_{\text{con}} \). Training dataset
\( \mathcal{T}_{\text{train}} = \{ < I_1, I_2 > \} \). Background-guided set
\( B \), Operation set \( O = \{ \text{background-aware operation}, \text{rot, shear,trans, equalize} \} \), and
maximum epoch number (i.e., MaxIters).

Output: Trained CD model \( \phi_0(\cdot) \).

1 Function \( \text{BGMix}(B, I_1, I_2, C) \):
2 Initialize two empty maps \( \bar{I}_1, \bar{I}_2 \);
3 Sample mixing weights
\( (w_1, w_2, \ldots, w_K) \sim \text{Dirichlet} \);
4 for \( i = 1 \) to \( K \) do
5 Sample operations \( (o_1, o_2, o_3) \sim C \);
6 Combine via \( o_{12} = o_2o_1 \) and \( o_{123} = o_3o_2o_1 \);
7 Sample \( o \sim \{ o_1, o_{12}, o_{123} \} \);
8 \( \bar{I}_1 = w_i o(I_1, B, C) \), \( \bar{I}_2 = w_i o(I_2, B, C) \);
9 Sample a blending weight \( w_0 \sim \text{Beta} \);
10 return \( \{ I_1 = w_0 \bar{I}_1 + (1 - w_0) \bar{I}_2 | j \in \{1, 2\} \} \);
11 End function:
12 for \( t = 1 \) to MaxIters do
13 Sample an image pair via \( < I_1, I_2 > \sim \mathcal{T}_{\text{train}} \);
14 Perform \( C = \phi_0(I_1, I_2) \);
15 Generate an augmented image pair via
\( < \bar{I}_1, \bar{I}_2 > : = \text{BGMix}(B, I_1, I_2, C) \);
16 Sample a background pair \( < B_1, B_2 > \sim B \);
17 Generate \( < I'_1, I'_2 > = < \bar{I}_1, \bar{I}_2 >, < B'_1, B'_2 > \) &
\( < B'_1, B'_2 > \) via Eq. (7), and (8);
18 Perform \( C' = \phi_0(B_1, B_2) \);
19 Calculate loss function via Eq. (13):
20 Do back-propagation and update \( \phi_0 \).

\[ \begin{align*}
    I'_1, I'_2 > & < \bar{I}_1, \bar{I}_2 > \\
    & > \text{as the first loss function, i.e.,}
    \end{align*} \]

\[ \begin{align*}
    L_{\text{con}1} & = \Phi(I_1, I_2, I'_1, I'_2) \\
    & = 1 - \cos(\psi([I_1, I_2]), \psi([I'_1, I'_2])) \\
\end{align*} \]  \( \text{(6)} \)

where ‘\([\cdot]\)’ is the concatenation operation, ‘\(\psi(\cdot)\)’ means the
pre-trained VGG16 network for the perception feature extraction,
and ‘\(\cos(\cdot)\)’ is the cosine similarity function. The
hyper-parameter \( \lambda_1 \) is to weight the consistency loss.
The intuitive idea behind this is that the two image pairs
\( < I_1, I_2 > \) and \( < \bar{I}_1, \bar{I}_2 > \) should contain the same object
changes if the predicted mask \( C \) is accurate. For example,
if \( C \) mislabels the change pixels between \( I_1 \) and \( I_2 \) as
unchanged pixels, \( I_1, I_2 \) and \( \bar{I}_1, \bar{I}_2 \) would present distinct appearance
on the foreground change regions since the latter
image pair is produced according to the \( C \) (See Eq. (4)).

Second, to further allow the CNN to be robust to different
background changes, we randomly sample a background
pair \( < B_1, B_2 > \) from the set \( B \). Then, we use it to replace
the backgrounds of \( < I_1, I_2 > \) and get a new image pair by

\[ \begin{align*}
    I'_1 & = \text{Rep}(I_1, B_1, C), \quad I'_2 = \text{Rep}(I_2, B_2, C) \quad \text{(7)}
\end{align*} \]

where the function \( \text{Rep}(\cdot) \) is the same with one defined in
Eq. (4). Moreover, we can synthesize a new background pair
by pasting the background regions in the \( < I_1, I_2 > \) to the
corresponding positions in \( < B_1, B_2 > \). Then, we can get a
new background pair like Eq. (7), i.e.,

\[ \begin{align*}
    B'_1 & = \text{Rep}(B_1, I_1, C), \quad B'_2 = \text{Rep}(B_2, I_2, C) \quad \text{(8)}
\end{align*} \]

Intuitively, if \( C \) localizes object changes accurately, \( < I_1, I_2 > \) and
\( < I'_1, I'_2 > \) should present the same perception,
similar on \( < B_1, B_2 > \) & \( < B'_1, B'_2 > \), and we have

\[ \begin{align*}
    L_{\text{con}2} & = \Phi(I_1, I_2, I'_1, I'_2) + \Phi(B_1, B_2, B'_1, B'_2) \quad \text{(9)}
\end{align*} \]

where \( \Phi(\cdot) \) is the same as the function in Eq. (6).

Third, we encourage the background replacement while
utilizing the low area rate prior of object changes, that is,

\[ \begin{align*}
    L_{\text{con}3} & = \sum_{i=1}^{2} \left(1 - \text{SSIM}(I_i, I'_i) - L_1(I_i, I'_i)\right) \quad \text{(10)}
\end{align*} \]

where \( \text{SSIM}(\cdot) \) is the structure similarity loss. Minimizing
the above loss function leads to two results according to
the Eq. (7): The more background contexts from the background
pair are embedded and it benefits the augmentation. The
object changes in an image pair are usually small and take a
low area rate in the whole image (See Fig. 2). The proposed
loss function makes use of this prior to encouraging small
change areas in \( C \).

Fourth, we further employ two discriminators as adversarial
losses to encourage the CNN \( \phi_0(\cdot) \) to produce accurate
change masks. Intuitively, if the change mask \( C \) identifies
the change regions accurately, the discriminators cannot
distinguish \( < I'_1, I'_2 > \) from \( < I_1, I_2 > \), and \( < B'_1, B'_2 > \)
from \( < B_1, B_2 > \), respectively. Specifically, the loss function
could be represented as

\[ \begin{align*}
    L_{\text{con}4} & = \log(D_1(I_1, I_2)) + \log(1 - D_1(I'_1, I'_2)) \\
    & + \log(D_2(B_1, B_2)) + \log(1 - D_2(B'_1, B'_2)) \quad \text{(11)}
\end{align*} \]

where \( D_1 \) and \( D_2 \) are the two discriminators. It will be
minimized when the function \( \phi_0(\cdot) \) can detect the \( C \) accurately,
which means the synthesized image pairs in Eq. (7) and
Eq. (8) are close to realistic image pairs \( < I_1, I_2 > \) and
\( < B_1, B_2 > \) that discriminators cannot distinguish them.

Finally, to suppress the potential predicting errors caused by
background changes, we set a loss function to encourage
\( \phi_0(\cdot) \) to predict an all-zero mask when we take a background
pair as inputs, i.e.,

\[ \begin{align*}
    L_{\text{con}5} & = L_2(\phi_0(B_1, B_2), 0) \quad \text{(12)}
\end{align*} \]

where \( 0 \) is an all zero tensor. With the above loss functions,
we get the final loss function to optimize the CNN, i.e.,

\[ \begin{align*}
    L & = \lambda_1 L_{\text{con}1}(I_1, I_2, I'_1, I'_2) + L_{\text{con}1}(I_1, I_2, B_1, B_2), \\
    & + L_{\text{con}2}(I_1, I_2, B_1, B_2), \quad \text{(13)}
\end{align*} \]

where \( L_{\text{con}1}(I_1, I_2, B_1, B_2) \) is defined as, i.e.,

\[ \begin{align*}
    L_{\text{con}3} & = \lambda_2 L_{\text{con}2} + \lambda_3 L_{\text{con}3} + \lambda_4 L_{\text{con}4} + \lambda_5 L_{\text{con}5} \quad \text{(14)}
\end{align*} \]

and \( \{ \lambda_i \} \) are to balance different terms. The last term of
Eq. (13) is to utilize the \( L_{\text{con}3} \) on \( (I_1, I_2, B_1, B_2) \), that is,
we can conduct the same operations and loss functions from
Eq. (7) to Eq. (12) on the augmented image pair \( < I'_1, I'_2 > \)
by producing \( < I'_1, I'_2 > \), and \( < B'_1, B'_2 > \) via Eq. (7), and
(8), and calculating corresponding loss functions.
Learning Algorithm
Algorithm (i.e., Line 12-20 in Algorithm 1). At the $t$ iteration during training, we sample an image pair from the training dataset $T_{train}$ and perform the change detection via $\phi_{\theta}()$ (See Line 14 in Algorithm 1). Then, we generate an augmented image pair via the BGMix function. After that, we calculate the loss by defined functions (See Line 16-20 in Algorithm 1) and update the model parameters.

Implementation details. We set the number of augmentation paths $K$ to 4 and the augmentation operation set $O$ with eight kinds of operations, i.e., $O=\{\text{background-aware operation, auto contrast, equalize, posterize, rotate, solarize, shear, translate}\}$. The hyper-parameters $\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_4$ and $\lambda_5$ are set to 1, 1, 5, 1 and 0.01 for AICD, and 1, 1, 3, 1 and 0.01 for BCD, respectively. The proposed method is implemented with Pytorch (Paszke et al. 2019). All experiments in the subsequent section are conducted on a 24GB RTX3090 GPU. We use SGD to optimize the network parameters. The batch size, learning rate, and momentum are set to 4, 1e-4, and 0.5, respectively. The largest iteration number is 100k.

Experimental Results

Setup
Baselines. We use BGMix to enhance four change detection methods, i.e., AFA (Ru et al. 2022), FCD (Wu, Du, and Zhang 2023), WCDNet (Andermatt and Timofte 2021), and SG (Zhao, Shang, and Lu 2021), and compare with three SOTA data augmentation methods, i.e., CutMix (Yun et al. 2019), MixUp (Huang, Zhang, and Zhang 2020) and AugMix (Hendrycks* et al. 2020). FCD and WCDNet are two SOTA weakly supervised methods. AFA is a SOTA weakly supervised segmentation method. We modify the single-image input of AFA into double-image input to make it suitable for CD tasks. SG is a SOTA weakly supervised defocus blur detection method via dual adversarial discriminators.

We modify the architecture of SG for change detection and name this variant SGCD.

Datasets. We conduct the experiments on two widely used remote sensing CD datasets, i.e. AICD (Bourdias, Marraud, and Sahbi 2011) and BCD (Ji, Wei, and Lu 2018). AICD has 1000 aerial image pairs with the resolution of $800 \times 600$. We randomly select 900 image pairs as the training dataset and the rest for testing. For facilitating the training and inference, all images are cropped with $256 \times 256$ by sliding window manner without overlap except at the image boundaries. Following (Chen, Li, and Shi 2021), we paste the changes cropped from changing image pair onto the background image pairs to balance the ratio of the changing and background image pairs. After pasting, we obtain 8103 and 903 training and testing image pairs, respectively. BCD is constructed based on a high-resolution image pair with $32507 \times 15354$. We crop the image pairs having a resolution of $256 \times 256$ with a sliding window manner from the high-resolution image pair. We randomly select 90% from the cropped image pairs as the training image pairs, and the left is used as the testing set. Note that, we follow the setups of BYOL (Grill et al. 2020) and adopt random flip and color distortion to enrich the training examples of the two training datasets (i.e., AICD and BCD). After this operation, the ratio of changing image pairs to background images in each training dataset is about 1 : 1. Finally, we obtain 24309 and 2709 training and testing image pairs for AICD, and 15570 and 1728 training and testing image pairs for BCD, respectively. All compared data augmentation methods are used to further augment these training examples with their own augmentation strategies and loss functions.

Metrics. We adopt F1-score (F1), overall accuracy (OA), and Intersection over Union (IoU) for evaluation.
BiGMix also can increase the recall of the changes and highlight the real changes (See the results of change detectors can effectively suppress the background smaller than the changes in BCD. Trained with BiGMix, the effectiveness of our method.

0.624, respectively. Results of Table 1 manifest the effectiveness losses of \( \mathcal{L}_{\text{con}}^{1-5} \) results in different degrees of performance degradation on the two datasets. While using the proposed loss achieves the best performances on both datasets, which manifests the usefulness of our loss.

Ablation Study of Loss Functions

SGCD obtains better CD performances than other change detectors on both datasets. Thus, we use SGCD for the ablation study. We remove each term of the augmented & real data consistency loss and retain other losses to verify the effect of each loss. As shown in Table 2, SGCD collapses on the two datasets when removing \( \mathcal{L}_{\text{con2}} \) loss, which demonstrates that the qualities of the synthesized changing and background image pairs are critically important. Removing losses of \( \mathcal{L}_{\text{con1-5}} \) results in different degrees of performance degradation on the two datasets. While using the proposed loss achieves the best performances on both datasets, which manifests the usefulness of our loss.

Comparison of Data Augmentations

Table 1 shows the results of different WSCD methods with SOTA augmentation methods on AICD and BCD datasets. We see that: ❶ The compared data augmentation methods have different effects for different CD methods and datasets. Specifically, among the compared methods, MixUp achieves the highest F1 and IoU improvements for AFA and FCD on AICD and BCD, respectively. AugMix achieves the highest F1 and IoU improvements for FCD on AICD. ❷ Not all data augmentation methods can improve the performance of the change detectors. The F1 score of FCD with CutMix is 0.298 on AICD, which is lower than the F1 score (0.373) without data augmentation. On AICD and BCD, none of the state-of-the-art data augmentation methods improves the detection performance of WCDNet. ❸ Our BiGMix achieves the highest performance improvements for all the change detectors on both datasets. On AICD, BiGMix improves F1 scores of AFA, FCD, WCDNet and SGCD to 0.496, 0.470, 0.749 and 0.854, respectively. On BCD, BiGMix improves F1 scores of AFA, FCD, WCDNet and SGCD to 0.405, 0.371, 0.453 and 0.624, respectively. Results of Table 1 manifest the effectiveness of our method.

As shown in Fig. 4, the changes in AICD are much smaller than the changes in BCD. Trained with BiGMix, the change detectors can effectively suppress the background changes and highlight the real changes (See the results of AFA in Fig. 4). BiGMix also can increase the recall of the detected changes for FCD (See the 2nd-4th rows of Fig. 4) and WCDNet (See the 3rd and 4th rows of Fig. 4). As shown in the last two columns of Fig. 4, the details of the detected changes of SGCD trained with BiGMix are clearer than the results of SGCD without data augmentation.

Conclusion

In this paper, we study the generalization issue of change detection from data augmentation and develop a novel weakly supervised training algorithm. The proposed background-mixed augmentation is specifically designed for change detection by augmenting examples under the guidance of a set of background images and letting targeted deep models see diverse background variations. Moreover, we propose the augmented & real data consistency loss to encourage the generalization increase significantly. We use the proposed method as a general framework to enhance four SOTA change detectors. Extensive experimental results demonstrate the superiority of our method over other SOTA data augmentation methods on the change detection task. In the future, we can use recent adversarial attacks (Guo et al. 2020; Tian et al. 2021) to guide the augmentation and further enhance the detection accuracy.
Acknowledgment
The proposed research is supported by Scientific Research Project of Tianjin Educational Committee (Grant No. 20212ZD002), and A*STAR Centre for Frontier AI Research. It is also supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-019), Singapore National Cybersecurity R&D Program No. NRF2018NCR-NCR005-0001, National Satellite of Excellence in Trustworthy Software System No. NRF2018NCR-NSOE003-0001, NRF Investigatorship No. NRF-NRFI06-2020-0001. We gratefully acknowledge the support of NVIDIA AI Tech Center (NVAITC).

References


