# ADAAUG: An Adaptive Data Augmentation Method for Change Detection

Rui Huang[1], Jieda Wei[1], Sihua Gao[1], Zongyu Guo[1], Yan Xing[1*], Weifeng Xu[2], Qing Guo[3]
[1]*Civil Aviation University of China, China*
[2]*North China Electric Power University, China*
[3]*Agency for Science, Technology and Research, Singapore*
{rhuang, shgao, yxing}@cauc.edu.cn, {jdwei1,zongyuguo}@yeah.net, weifengxu@163.com, tsingguo@ieee.org

***Abstract*—Current data augmentation methods for change detection utilize fixed strategies to produce image samples without considering the specificity of the change image pairs. In this paper, we propose ADAAUG, a new reinforcement learning framework that adaptively augments the change image pairs. An actor selects the best augmentation operation from the operation set according to the image pair. The augmented image pairs make the change detector easier to learn the optimal parameters and improve the final detection performance. Besides, we propose a mask guided mixing customed for change detection data augmentation, which mixes the change regions of the current training sample based on the prediction results and labels to generate high-quality samples with more positive samples. The extensive experiments on the standard benchmarks show that ADAAUG achieves favorable performance compared to SOTA data augmentation methods.**

***Index Terms*—Change detection, data augmentation, reinforcement learning**

## I. INTRODUCTION

Data augmentation is a widely used strategy to increase the diversity of training data, which improves the model generalization for image classification [1]–[4] and object detection [5]–[7].

Unlike image classification and object detection, change detection identifies object changes from background changes in two images captured in the same scene over a long time span. Directly utilizing data augmentation methods for image classification or object detection for change detection fails to capture the specificity of the change image pairs, such as temporal consistency, diverse patterns of background changes, and adaptive augmentation operations selection. Recent works study foreground [8] and background diversity [9] for change detection. Specifically, Xing et al., [8] propose a novel mask mix data augmentation method to generate image pairs to help the neural network capture the foreground changes. Huang et al., [9] propose the background-mixed augmentation by augmenting examples under the guidance of a set of background changing images to let change detectors see diverse environment variations. However, to the best of our knowledge, there is no data augmentation method for change detection considering adaptively selecting augmentation operations for image pairs.

In the image classification area, reinforcement learning-based data augmentations were studied to search for optimal augmentation policy. AutoAugment [10] uses reinforcement learning to search for an optimal augmentation policy in a search space and achieve good performance. However, the search process of AutoAugment takes a large amount of training time. For this reason, some advanced automatic augmentation methods [5], [11], [12] based on reinforcement learning try to reduce the search time. A representative work, Patch AutoAugment [11] employs multi-agent reinforcement learning to search for an optimal data augmentation policy for the patch level, obtaining remarkable gains over AutoAugment. Although reinforcement learning has been successful in data augmentation for image classification and object detection, its application in change detection remains a challenge. Specifically, change detection environments contain complex change information and background noise, and existing reinforcement learning-based data augmentation methods focus only on common image transformations and fail to attend to the diversity of change targets.

In this paper, we propose an adaptive data augmentation method for change detection, dubbed as ADAAUG, a new reinforcement learning framework for change detection data augmentation. An actor chooses an optimal data augmentation operation according to the state of the input image pair. Then the augmented image pairs are used to update the change detector. Since the actor always chooses the best operation, the augmented image pairs make the change detector easier to learn the optimal parameters and improve the final detection performance. We use the Advantage Actor-Critic algorithm [13] to train our actor. Besides, we propose a mask guided mixing data augmentation customed for change detection, which mixes the change regions of the current training sample based on the prediction results and labels to generate high-quality image pairs with more positive samples. We have conducted various experiments on the standard benchmarks with several different detectors. The experimental results demonstrate that ADAAUG achieves favorable performance compared to SOTA data augmentation methods. Our contributions can be summarized as follows:

- We propose a reinforcement learning based change detection data augmentation to adaptively choose the best augmentation operation for each image pair, which can improve the generalization ability of change detectors.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00026

- We propose a mask guided mixing data augmentation customed for change detection. The augmented image pairs have more positive samples to facilitate the change detector's training.
- We conduct extensive experiments in two public datasets with three different change detectors. The experimental results demonstrate that our method achieves favorable performance against SOTA data augmentation methods.

## II. Preliminaries

Reinforcement learning (RL) models the policy-choose problem as a Markov decision process (MDP) with $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, T)$. During training, an agent interacts with the environment by executing actions and receiving rewards. At each time step $t$, the agent observes the current state $s_t \in \mathcal{S}$ and executes the action $a_t \in \mathcal{A}$ through the policy $\pi(a_t|s_t)$, where $\pi(a_t|s_t)$ is a mapping from state $s_t$ to action $a_t$. The agent obtains a reward $r_t : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ to evaluate the goodness of the action $a_t$ and transitions to the next state $s_{t+1}$ with a state transition function denoted as $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$. This exploration process continues until the agent reaches the final state. The long-term discounted reward $R_t = \sum_{k=0}^{T} \gamma^k r_{t+k}$ is accumulated with the discount factor $\gamma \in (0, 1]$ in a time horizon $T$. The agent optimizes policy $\pi^*$ to maximize reward $R_t$ by constantly trying and learning. In this paper, we adopt RL to adaptively choose the optimal data augmentation policy for each image pair to improve the performance of the change detection tasks.

## III. Method

As shown in Fig. 1, given an image pair $\langle \mathbf{I}^1, \mathbf{I}^2 \rangle$, we extract deep features as the state. The actor network selects an augmentation operation $op(\cdot)$ for the image pair based on the state. Then, we train a change detector with the augmented image pair $\langle \tilde{\mathbf{I}}^1, \tilde{\mathbf{I}}^2 \rangle$. We use the feedback from the change detector on the policy evaluation dataset $\mathcal{T}$ as a reward to update the policy network.

### A. Augmentation operation set $\mathcal{O}$

Due to the lack of a data augmentation method for change detection, we first introduce a novel data augmentation method, dubbed as *Mask-Guided Mixing* (MGM). MGM uses the prediction $\mathbf{Y}_i$, the ground truth map $\mathbf{M}_i$ of the image pair $\langle \mathbf{I}_i^1, \mathbf{I}_i^2 \rangle$ and the ground truth map of the $k$-th change object $\mathbf{M}_j^k$ in the $j$-th image pair $\langle \mathbf{I}_j^1, \mathbf{I}_j^2 \rangle$, to generate candidate augmentation position $P_c$ by

$$P_c^k = \Gamma(\mathbf{Y}_i \vee \mathbf{M}_i, \mathbf{M}_j^k), \tag{1}$$

where $\vee$ is the logical or operation, $P_c^k = (P_{cx}^k, P_{cy}^k)$ represent the coordinates of the x and y axes, respectively. $\Gamma(A, B)$ is the change position generating function, seeking the position in A to past the change object of B. Specifically, we search the background region of A to find a position can paste the change object of B.

According to $P_c$, the change region of $\langle \mathbf{I}_j^1, \mathbf{I}_j^2 \rangle$ is pasted onto $\langle \mathbf{I}_i^1, \mathbf{I}_i^2 \rangle$ to generate the augmented sample $\langle \tilde{\mathbf{I}}^1, \tilde{\mathbf{I}}^2 \rangle$. Specifically, We first compute the affine transformation matrix $\mathbf{T}_k$ at the corresponding position of $k$-th change object:

$$\mathbf{T}_k = \begin{bmatrix} P_{cx}^k - X(\mathbf{M}_j^k) \\ P_{cy}^k - Y(\mathbf{M}_j^k) \end{bmatrix}, \tag{2}$$

where $X(\cdot)$ and $Y(\cdot)$ calculate the center coordinate of the change object in the $x$ and $y$ axes, respectively. Then, we use $\mathbf{T}_k$ to blend the change objects of images pairs $\langle \mathbf{I}_j^1, \mathbf{I}_j^2 \rangle$ and $\langle \mathbf{I}_i^1, \mathbf{I}_i^2 \rangle$ to get the augmented image pair:

$$\begin{aligned} \tilde{\mathbf{I}}^1 &= \sum_{k=1}^{K} \mathbf{M}_j^k \mathbf{T}_k \odot \left(\delta \mathbf{I}_j^1 + (1-\delta)\mathbf{I}_j^2\right) + (1 - \mathbf{M}_j^k \mathbf{T}_k) \odot \mathbf{I}_i^1, \\ \tilde{\mathbf{I}}^2 &= \sum_{k=1}^{K} \mathbf{M}_j^k \mathbf{T}_k \odot \left(\delta \mathbf{I}_j^2 + (1-\delta)\mathbf{I}_j^1\right) + (1 - \mathbf{M}_j^k \mathbf{T}_k) \odot \mathbf{I}_i^2, \end{aligned} \tag{3}$$

where $\delta$ is randomly selected from $\{0, 1\}$ with uniform probability. Note that, we return $\langle \tilde{\mathbf{I}}^2, \tilde{\mathbf{I}}^1 \rangle$ with probability 0.5.

Finally, our operation set $\mathcal{O}$ consists of MGM, AugMix [1], MUM [6], CutMix [7], GridMask [14], MixUp [15], TokenMix [16], without augmentation (W/o Aug). Let $\mathcal{O} = \{op_1, \ldots, op_M\}$ be the set of all available operations. We use $op(\cdot)$ to denote the transformation of image pair $\langle \mathbf{I}^1, \mathbf{I}^2 \rangle$ to a novel one (i.e.,$\langle \tilde{\mathbf{I}}^1, \tilde{\mathbf{I}}^2 \rangle$), as

$$\langle \tilde{\mathbf{I}}^1, \tilde{\mathbf{I}}^2 \rangle = op(\mathbf{I}^1, \mathbf{I}^2). \tag{4}$$

### B. Policy evaluation set $\mathcal{T}$

Prior works [11], [12] directly use the feedback of the target model from training samples to assess the strategy. However, due to the augmentation operation that might change the label of the image pair, using a training dataset for evaluation cannot accurately evaluate the policy. Thus, we propose to use a fixed validation dataset as a policy evaluation set $\mathcal{T} = \{\langle \mathbf{I}_k^1, \mathbf{I}_k^2, \mathbf{M}_k \rangle\}_{k=1}^{K}$. Note that each $\mathbf{M}_k$ should have at least one change region.

### C. Policy Modeling

The actor network chooses "optimal" augmentation operation for each image pair of an batch $\{\langle \mathbf{I}_i^1, \mathbf{I}_i^2, \mathbf{M}_i \rangle\}_{i=1}^{B}$). Then the critic network evaluates the effectiveness of the actor network. We detail the state, action, and reward for the augmentation policy as below.

**State**. Since we conduct data augmentation for change detection, we use the absolute difference features of an image pair as the state of RL. The basic feature extraction network consists of three convolutional layers. We adopt ReLU as a nonlinear activation function for the first two convolutional layers. The (kernel size, stride, and output channel number) of these three convolutional layers are set to $(8, 4, 32)$, $(4, 2, 64)$, and $(3, 1, 64)$, respectively. The state $s_t = \phi_e(abs(\mathbf{I}^1 - \mathbf{I}^2))$, where $\phi_e(\cdot)$ denotes a feature extractor, $abs(\cdot)$ denotes absolute difference.

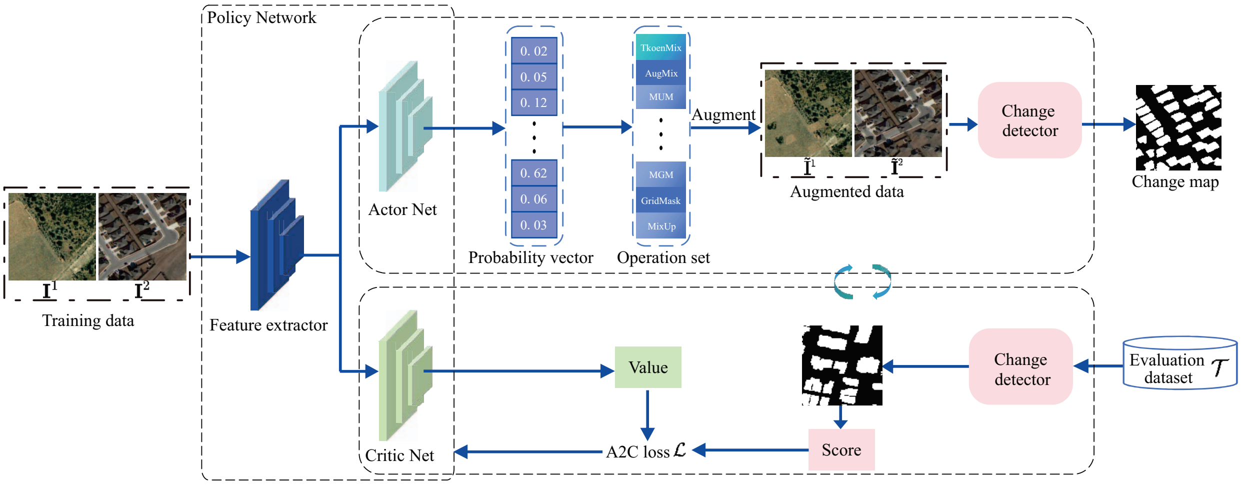


Fig. 1. The proposed Adaptive Augmentation (AdaAug) framework.

**Action**. The actor network is a fully connected network with 2 layers. The first layer has 256 neurons. The second layer has 8 neurons (i.e., the number of augmentation operations). We use augmentation operation set $\mathcal{O}$ as the action space $\mathcal{A}$. Given a state $s_t$ and action space $\mathcal{A}$, the actor chooses an action $a_t \in \mathcal{A}$ to augment the training sample.

**Reward**. The reward reflects the good and or bad of the action. We use the quantitative values before and after updating the change detector. Let $\psi(\cdot, \mathbf{W}_{\text{bef}})$ and $\psi(\cdot, \mathbf{W}_{\text{aft}})$ denote a change detector with parameters before and after updating with augmentation operations on image pairs in a batch. The reward $r$ is defined as

$$r = S(\psi(\langle \mathbf{I}^1, \mathbf{I}^2 \rangle, \mathbf{W}_{\text{aft}}) - S(\psi(\langle \mathbf{I}^1, \mathbf{I}^2 \rangle, \mathbf{W}_{\text{bef}})), \tag{5}$$

where $S(\cdot)$ represents a score function.

### D. Policy Learning

We adopt the Advantage Actor-Critic (A2C) algorithm [13] to learn the augmentation policy. The agent has an actor to learn the discrete policy $\pi(a|s)$. And the critic network of the agent estimates the state's value $V^{\pi}(s)$. We build the critic network with two fully connected layers network. The first layer has 256 neurons. And the second layer has 1 neuron. We model the action-value Q function as

$$Q^{\pi}(s, a) = \mathbb{E}[R_t | s, a], \tag{6}$$

where $R_t = \sum_{l=0}^{T} \gamma^l r_{t+l}$ is the long-term discounted reward. The advantage function of the augmentation policy is given as

$$A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s). \tag{7}$$

Let $\mathbf{\Theta}_{\text{critic}}$ denote the critic network parameters. The loss to update $\mathbf{\Theta}_{\text{critic}}$ is defined as square value of advantage function by

$$\mathcal{L}(\mathbf{\Theta}_{\text{critic}}) = \frac{1}{2}(A^{\pi}(s, a))^2. \tag{8}$$

Let $\mathbf{\Theta}_{\text{actor}}$ denote the actor network parameters. The loss to update $\mathbf{\Theta}_{\text{actor}}$ is defined as

$$\mathcal{L}(\mathbf{\Theta}_{\text{actor}}) = -log\pi_{\mathbf{\Theta}_{\text{actor}}}(a|s)A^{\pi}(s, a). \tag{9}$$

To encourage exploration and prevent the policy from converging too quickly to suboptimal solutions, we add an entropy term to the policy loss, Equation (9) is rewritten as

$$\mathcal{L}(\mathbf{\Theta}_{\text{actor}}) = -log\pi_{\mathbf{\Theta}_{\text{actor}}}(a|s)A^{\pi}(s, a) + \beta\mathcal{H}(\pi(s)), \tag{10}$$

where $\beta$ is the hyperparameter. The entropy term $\mathcal{H}(\pi(s))$ is defined as

$$\mathcal{H}(\pi(s)) = -\sum_{i=1}^{M} \pi(a_i|s) log(\pi(a_i|s)), \tag{11}$$

where $M$ is the number of augmentation operations and $a_i$ represents augmentation operation $op_i$. The overall loss function $\mathcal{L}$ is a weighted sum of policy loss and critic loss as

$$\mathcal{L} = \lambda_1 \mathcal{L}(\omega) + \lambda_2 \mathcal{L}(\theta), \tag{12}$$

where the hyper-parameters $\lambda_1 = \lambda_2 = 1$.

### E. Implementation details

We use Adam optimizer with a learning rate of 1e-3 to learn the policy model. Following [11], we set the time horizon $T = 1$. The change detectors are trained with batch size of 8 and epoch of 200. Other super-parameters for training change detectors are set as the default. Before training our policy model, we warm up each change detector by training change detector $n \in [3, 9]$ epochs on BCD and LEVIR datasets.

## IV. Experiment

### A. Setup

**Datasets.** We conduct extensive experiments on two representative remote sensing change detection datasets (i.e., LEVIR and BCD) to verify the practical performance of the proposed ADAAUG. LEVIR [17] is a large-scale and very high resolution change detection dataset. It has 637 pairs of images with $1024 \times 1024$ pixels. We crop $256 \times 256$ image patches in a non-overlap manner. The image patches are randomly partitioned into training and testing sets with 8144 and 2048 image pairs, respectively. BCD [18] has a pair of aerial images with a spatial size of $32,507 \times 15,354$ of resolution with 0.2 m/pixel. We crop $256 \times 256$ image patches in a non-overlap manner. We randomly partition the image patches into 6858 training image patches and 762 testing image patches.

TABLE I
QUANTITATIVE COMPARISON OF THE CHANGE DETECTORS WITH DIFFERENT DATA AUGMENTATION METHODS ON BCD AND LEVIR. THE **BOLD** VALUE IS THE BEST.

| Model | Method | BCD | | | | LEVIR | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $F1\uparrow$ | $IoU\uparrow$ | $OA\uparrow$ | $k\uparrow$ | $F1\uparrow$ | $IoU\uparrow$ | $OA\uparrow$ | $k\uparrow$ |
| DTCDSCN [19] | w/o Aug | 87.20 | 77.31 | 99.03 | 86.70 | 88.90 | 80.02 | 98.89 | 88.32 |
| | MixUp | 85.44 | 74.58 | 98.94 | 84.89 | 87.98 | 78.53 | 98.84 | 87.37 |
| | CutMix | 89.13 | 80.40 | 99.20 | 88.73 | 90.25 | 82.23 | 99.03 | 89.73 |
| | AugMix | 88.67 | 79.62 | 99.18 | 88.24 | 89.08 | 80.32 | 98.90 | 88.50 |
| | MUM | 86.80 | 76.67 | 99.02 | 86.28 | 89.64 | 81.22 | 98.96 | 89.09 |
| | TokenMix | 85.84 | 75.19 | 98.89 | 85.26 | 88.09 | 78.72 | 98.83 | 87.48 |
| | GridMask | 88.45 | 79.29 | 99.14 | 88.01 | 89.65 | 81.25 | 98.95 | 89.10 |
| | MaskMix | 89.65 | 81.24 | 99.25 | 89.26 | 89.28 | 80.63 | 98.94 | 88.72 |
| | MGM(Ours) | 92.20 | 85.54 | 99.45 | 91.92 | 90.38 | 82.44 | 99.04 | 89.87 |
| | AdaAug(Ours) | **93.17** | **87.21** | **99.49** | **92.90** | **91.13** | **83.70** | **99.11** | **90.66** |
| DMINet [20] | w/o Aug | 89.52 | 81.03 | 99.21 | 89.11 | 90.04 | 81.89 | 99.00 | 89.52 |
| | MixUp | 86.95 | 76.92 | 99.04 | 86.46 | 89.82 | 81.52 | 98.97 | 89.28 |
| | CutMix | 91.27 | 83.94 | 99.36 | 90.94 | 90.57 | 82.76 | 99.06 | 90.07 |
| | AugMix | 91.19 | 83.81 | 99.39 | 90.86 | 90.36 | 82.41 | 99.04 | 89.85 |
| | MUM | 89.86 | 81.58 | 99.25 | 89.47 | 90.54 | 82.71 | 99.06 | 90.04 |
| | TokenMix | 89.78 | 81.45 | 99.26 | 89.39 | 88.81 | 79.87 | 98.94 | 88.25 |
| | GridMask | 90.83 | 83.20 | 99.32 | 90.48 | 90.41 | 82.50 | 99.06 | 89.91 |
| | MaskMix | 91.87 | 84.96 | 99.41 | 91.56 | 90.74 | 83.04 | 99.06 | 90.24 |
| | MGM(Ours) | 92.77 | 86.52 | 94.47 | 92.50 | 90.75 | 83.07 | 99.07 | 90.26 |
| | AdaAug(Ours) | **93.55** | **87.89** | **99.52** | **93.31** | **91.24** | **83.89** | **99.12** | **90.78** |
| BIT [21] (Transformer) | w/o Aug | 90.38 | 82.45 | 99.28 | 90.01 | 89.17 | 80.46 | 98.93 | 88.61 |
| | MixUp | 84.87 | 73.72 | 98.89 | 84.29 | 86.68 | 76.49 | 98.73 | 86.02 |
| | CutMix | 89.84 | 81.56 | 99.23 | 89.44 | 89.96 | 81.75 | 99.00 | 89.44 |
| | AugMix | 91.57 | 84.44 | 99.38 | 91.25 | 89.74 | 81.39 | 98.98 | 89.20 |
| | MUM | 89.40 | 80.83 | 99.21 | 88.99 | 89.88 | 81.62 | 98.99 | 89.35 |
| | TokenMix | 87.30 | 77.46 | 99.06 | 86.81 | 88.73 | 79.74 | 98.91 | 88.16 |
| | GridMask | 91.32 | 84.02 | 99.36 | 90.98 | 90.05 | 81.90 | 99.00 | 89.52 |
| | MaskMix | 91.49 | 84.32 | 99.39 | 91.18 | 89.78 | 91.46 | 98.99 | 89.25 |
| | MGM(Ours) | 92.25 | 85.62 | 99.44 | 91.96 | 90.49 | 82.63 | 99.05 | 89.99 |
| | AdaAug(Ours) | **93.24** | **87.33** | **99.50** | **92.98** | **90.83** | **83.19** | **99.07** | **90.34** |

**Criteria.** We adopt F1-score (F1), overall accuracy (OA), Intersection over Union (IoU) and Kappa coefficient ($k$) for evaluation.

### B. Comparison of Data Augmentations

Table I reports the performance of different change detection methods change detectors with different data augmentation methods on BCD and LEVIR datasets. We can find that: (1) Not all data augmentation methods can improve the performance of change detection models. For example, on the LEVIR dataset, the performance of DTCDSCN, DMINet, and BIT with the MixUp and TokenMix is lower than the performance without data augmentation; (2) Our MGM outperforms all other compared augmentation methods on both datasets with large margins. Compared with the results of DTCDSCN without data augmentation, DTCDSCN with MGM achieves 5.73% relative F1 improvement. On LEVIR, BIT with MGM achieves 1.48% relative F1 improvement over BIT without data augmentation. (3) ADAAUG promotes the performance of different change detectors with large margins on both BCD and LEVIR datasets. On BCD, ADAAUG improves the F1 of DTCDSCN, DMINet, and BIT to 93.17%, 93.55%, and 93.24%, respectively. On LEVIR, ADAAUG improves the F1 of DTCDSCN, DMINet, and BIT to 91.13%, 91.24%, and 90.83%, respectively. The quantitative results on both datasets demonstrate the effectiveness of our proposed ADAAUG.

Table II shows the running time (Average time per epoch) of DMINet with different data augmentation methods on LEVIR. We can find that AugMix takes the longest time (598.11s) due to performing multiple image transformations. Other data augmentation methods (e.g., MixUp, CutMix, MUM, GridMask, TokenMix, and MaskMix) achieve lower time spent than ADAAUG (504.97s) by a single algebraic operation. However, they do not fully explore the contribution of different data augmentation methods to the change detector, resulting in achieving a lower performance than ADAAUG.

TABLE II
TIME COMPARISON (AVERAGE TIME PER EPOCH) OF THE DMINET WITH DIFFERENT DATA AUGMENTATION METHODS ON LEVIR.

| Method | MixUp | CutMix | MUM | GridMask | TokenMix | MaskMix | AugMix | AdaAug |
|---|---|---|---|---|---|---|---|---|
| Time(s) | 250.29 | 242.80 | 237.54 | 254.64 | 227.36 | 305.51 | 598.11 | 504.97 |

### C. Result visualization

*1) Change detection result visualization:* The visualization results of using AADAUG on the two datasets are shown in Fig.2. On LEVIR, change detectors trained with ADAAUG can detect tiny change targets and obtain integral change regions. As demonstrated in the 1st row of Fig.2, the results of different change detectors trained with ADAAUG can detect the tiny change buildings at the left-bottom of the images. In the 3rd and 5th rows, the results of different change detectors trained with ADAAUG are more accurate than the results without ADAAUG.

*2) Selected policy visualization:* In Fig.3, we plot the curves of the proportion of different augmentation operations selected by ADAAUG with DMINet on LEVIR during the training process. We can find that (1) in the whole training stage, no augmentation operation takes a lower proportion; (2) MGM, MUM, and GridMask have higher proportions been selected by ADAAUG; (3) with the learning of the change detector, ADAAUG selects the single data augmentation operation's proportion is less than 35%. MGM, CutMix, MUM, and GridMask are alternately selected. This demonstrates that ADAAUG can adaptively select data augmentation operations for different image pairs.

### D. Ablation Study

*1) Ablation study of augmentation operation selection policy:* To verify the effectiveness of ADAAUG, we test two additional augmentation operation selection policies, random selection and uniform selection. Specifically, random selection randomly selects an augmentation operation for each image pair. While uniform selection selects an augmentation operation for each image pair with the same probability. As shown in Table III, the performance of the change detection

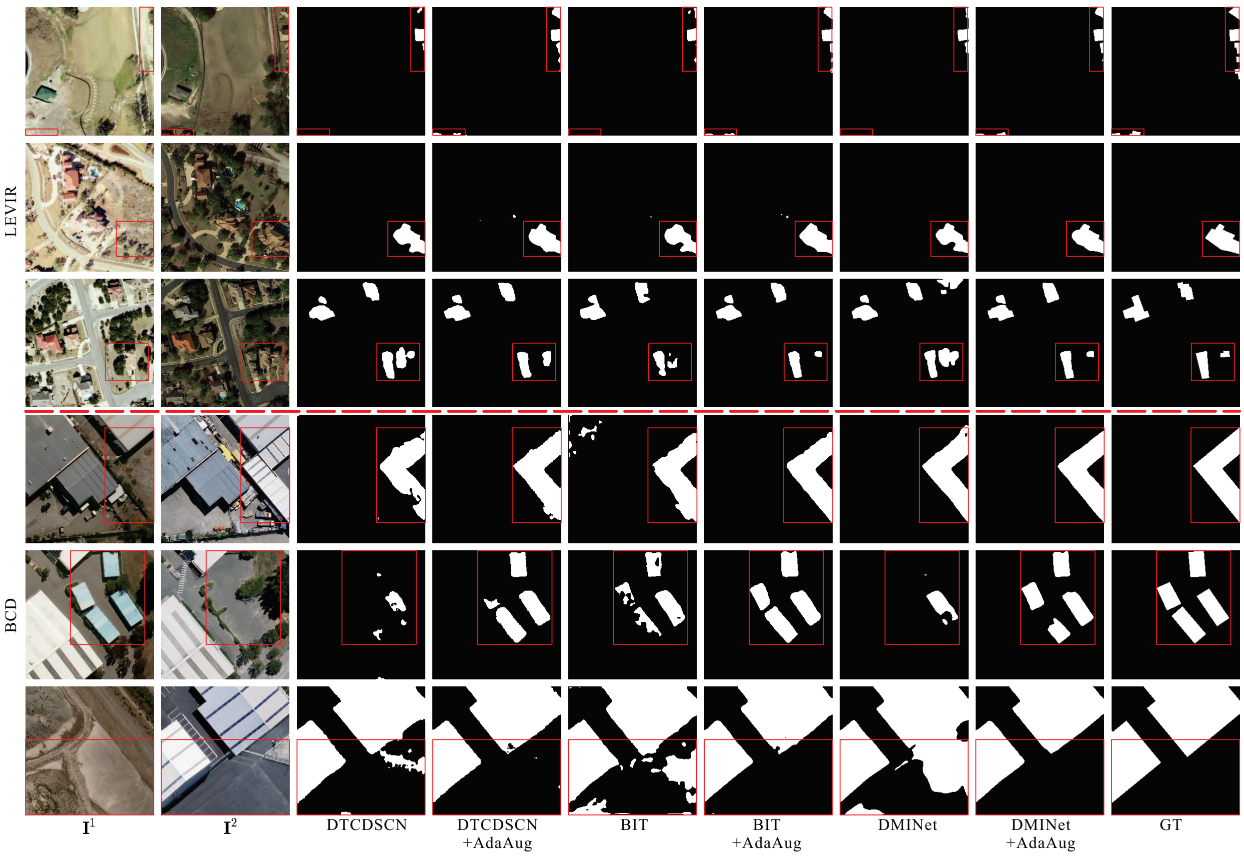


Fig. 2. Visual comparison of different change detection models with (+ADAAUG) and without ADAAUG on LEVIR and BCD.

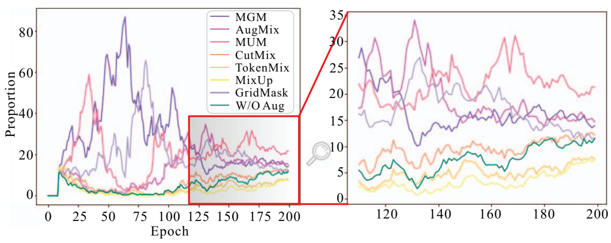


Fig. 3. The proportion of different augmentation operations during training.

TABLE III
PERFORMANCE OF DMINET WITH DIFFERENT AUGMENTATION OPERATION SELECTION POLICIES ON BCD AND LEVIR. THE **BOLD** VALUE IS THE BEST.

| Method | BCD | | | | LEVIR | | | |
|---|---|---|---|---|---|---|---|---|
| | $F1\uparrow$ | $IoU\uparrow$ | $OA\uparrow$ | $k\uparrow$ | $F1\uparrow$ | $IoU\uparrow$ | $OA\uparrow$ | $k\uparrow$ |
| Random | 92.81 | 86.59 | 99.48 | 92.54 | 91.03 | 83.54 | 99.10 | 90.56 |
| Uniform | 92.90 | 86.73 | 99.48 | 92.62 | 91.18 | 83.78 | **99.12** | 90.71 |
| AdaAug | **93.55** | **87.89** | **99.52** | **93.31** | **91.24** | **83.89** | **99.12** | **90.78** |

on both datasets decreases when using the random and uniform selection policies. It demonstrates that ADAAUG with meaningful guidance significantly surpasses the random and uniform selection policies.

*2) Ablation study of the number of augmentation operations:* We explore the effect of the number of augmentation operations in $\mathcal{O}$ on the change detection performance. We randomly select $m$ operations from $\mathcal{O}$ to form a new operation set $\hat{\mathcal{O}}$ for the ablation study. Specifically, we set the number of $m$ to 4 and 6. To prevent the impact of random selection, we conduct 3 experiments on each quantity $m$ and take the average results as the final results. As shown in Table IV, change detector (i.e., DMINet) with a small number of policies perform poorly in the real world. $m = 8$ achieves the best performance on both BCD and LEVIR datasets. Thus, we fix $m$ to 8 in our experiment.

*3) The effect of warm-up epoch number:* Directly using ADAAUG in the training of a change detector makes the change detector hard to converge. Thus, we adopt a warm-up strategy in our experiment to provide better feedback on the quality of the augmentation policy. In the warm-up phase, we train the change detector with original data. As shown in Table V, we observe the performance of DMINet during the warm-up stages of 3, 6, and 9 epochs. We can find that

TABLE IV
ABLATION STUDY OF THE NUMBER OF AUGMENTATION OPERATIONS. THE **BOLD** VALUE IS THE BEST.

| Number | BCD | | | | LEVIR | | | |
|---|---|---|---|---|---|---|---|---|
| | $F1\uparrow$ | $IoU\uparrow$ | $OA\uparrow$ | $k\uparrow$ | $F1\uparrow$ | $IoU\uparrow$ | $OA\uparrow$ | $k\uparrow$ |
| 4 | 92.99 | 86.91 | 99.48 | 92.73 | 91.09 | 83.64 | 99.10 | 90.62 |
| 6 | 93.50 | 87.79 | **99.52** | 93.25 | 90.46 | 82.58 | 99.05 | 89.95 |
| 8 | **93.55** | **87.89** | **99.52** | **93.31** | **91.24** | **83.89** | **99.12** | **90.78** |

different warm-up epochs have different impacts on the change detection performances. In particular, the best performance is achieved with epochs of 6 and 9 on BCD and LEVIR, respectively.

TABLE V
RESULTS OF ADAAUG WITH DIFFERENT WARM-UP EPOCHS. THE **BOLD** VALUE IS THE BEST.

| Epoch | BCD | | | | LEVIR | | | |
|---|---|---|---|---|---|---|---|---|
| | $F1\uparrow$ | $IoU\uparrow$ | $OA\uparrow$ | $k\uparrow$ | $F1\uparrow$ | $IoU\uparrow$ | $OA\uparrow$ | $k\uparrow$ |
| 3 | 92.33 | 85.76 | 99.44 | 92.04 | 90.99 | 83.48 | 99.11 | 90.52 |
| 6 | **93.55** | **87.89** | **99.52** | **93.31** | 91.18 | 83.79 | **99.12** | 90.71 |
| 9 | 93.07 | 87.03 | 99.49 | 92.80 | **91.24** | **83.89** | **99.12** | **90.78** |

## V. CONCLUSION

We have proposed a novel adaptive data augmentation method, ADAAUG, for change detection. We formulate the data augmentation as reinforcement learning by treating data augmentation operation selection as action selection according to the state. Thus, our method can choose an optimal augmentation operation for the image pair. The augmented image pairs make the change detector easier to learn the optimal parameters to improve the final detection performances. In addition, we have proposed a mask guided mixing for change detection data augmentation. It mixes the change regions of the current training sample based on the prediction results and labels to generate high-quality samples with more positive samples. Extensive experimental and visualization results demonstrate the superiority of our method over other data augmentation methods on the change detection task.

## REFERENCES


[1] Hendrycks D, Mu N, Cubuk E D, et al. Augmix: A simple method to improve robustness and uncertainty under data shift[C]//International conference on learning representations. 2020, 1(2): 5.

[2] Dabouei A, Soleymani S, Taherkhani F, et al. Supermix: Supervising the mixing data augmentation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 13794-13803.

[3] Gong C, Wang D, Li M, et al. Keepaugment: A simple information-preserving data augmentation approach[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 1055-1064.

[4] Zhong Z, Zheng L, Kang G, et al. Random erasing data augmentation[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 13001-13008.

[5] Zoph B, Cubuk E D, Ghiasi G, et al. Learning data augmentation strategies for object detection[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16. Springer International Publishing, 2020: 566-583.

[6] Kim J M, Jang J Y, Seo S, et al. Mum: Mix image tiles and unmix feature tiles for semi-supervised object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 14512-14521.

[7] Yun S, Han D, Oh S J, et al. Cutmix: Regularization strategy to train strong classifiers with localizable features[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6023-6032.

[8] Xing Yan, Wei Jieda, Wang Ruofei, et al. MaskMix: mask mixing augmentation method for change detection [J]. Application Research of Computers, 2023, 40 (12): 3834-3840,3847.

[9] Huang R, Wang R, Guo Q, et al. Background-mixed augmentation for weakly supervised change detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2023, 37(7): 7919-7927.

[10] Cubuk E D, Zoph B, Mane D, et al. Autoaugment: Learning augmentation strategies from data[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 113-123.

[11] Lin S, Yu T, Feng R, et al. Local patch autoaugment with multi-agent collaboration[J]. IEEE Transactions on Multimedia, 2023.

[12] Lin S, Zhang Z, Li X, et al. Selectaugment: Hierarchical deterministic sample selection for data augmentation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2023, 37(2): 1604-1612.

[13] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International conference on machine learning. PMLR, 2016: 1928-1937.

[14] Chen P, Liu S, Zhao H, et al. Gridmask data augmentation[J]. arXiv preprint arXiv:2001.04086, 2020.

[15] Zhang H, Cisse M, Dauphin Y N, et al. mixup: Beyond empirical risk minimization[C]//International Conference on Learning Representations. 2018.

[16] Liu J, Liu B, Zhou H, et al. Tokenmix: Rethinking image mixing for data augmentation in vision transformers[C]//European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022: 455-471.

[17] Chen H, Shi Z. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection[J]. Remote Sensing, 2020, 12(10): 1662.

[18] Ji S, Wei S, Lu M. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set[J]. IEEE Trans on Geoscience and Remote Sensing, 2018, 57(1): 574-586.

[19] Liu Y, Pang C, Zhan Z, et al. Building change detection for remote sensing images using a dual-task constrained deep siamese convolutional network model[J]. IEEE Geoscience and Remote Sensing Letters, 2020, 18(5): 811-815.

[20] Feng Y, Jiang J, Xu H, et al. Change detection on remote sensing images using dual-branch multilevel intertemporal network[J]. IEEE Transactions on Geoscience and Remote Sensing, 2023, 61: 1-15.

[21] Chen H, Qi Z, Shi Z. Remote sensing image change detection with transformers[J]. IEEE Trans on Geoscience and Remote Sensing, 2021, 60: 1-14.