Received: 15 June 2021 | Revised: 6 December 2021 | Accepted: 8 December 2021

IET Signal Processing

DOI: 10.1049/sil2.12095

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Selecting change image for efficient change detection

Rui Huang[1] | Ruofei Wang[1] | Yuxiang Zhang[1] | Yan Xing[1] | Wei Fan[1] | Kai Leung Yung[2]

[1]College of Computer Science and Technology, Civil Aviation University of China, Tianjin, China

[2]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Correspondence

Yuxiang Zhang, College of Computer Science and Technology, Civil Aviation University of China, Tianjin, 300 300, China.
Email: yxzhang@cauc.edu.cn

Funding information

Natural Science Foundation of Tianjin City

Abstract

Change detection (CD) is a fundamental problem that aims at detecting changed objects from two observations. Previous CNN-based CD methods detect changes through multi-scale deep convolutional features extracted from two images. However, we find that change always occurs in the 'Query' image for fixed cameras. This condition means that changes can be detected in advance from a single image with a coarse change. In this paper, we propose an efficient CD method to detect precise changes from the change image. First, a change image selector is designed to identify the image containing changes. Second, a coarse change prior map generator is proposed to generate coarse change prior to indicate the position of changes. Then, we introduce a simple multi-scale CD module to refine the coarse change detection. As only one image is used in the multi-scale CD module, our method is more efficient in training and testing than other compared methods. Numerous experiments have been conducted to analyse the effectiveness of the proposed method. Experimental results show that the proposed method achieves superior detection performance and higher speed than other compared CD methods.

KEYWORDS

change detection, change image selector, efficient change detection, multi-scale change detection

1 | INTRODUCTION

Change detection (CD) aims at finding the difference between two observations in the same place with a time span, which has wide applications in urban development [1], disaster assessment [2], resource monitoring and utilisation [3], and security and military operations [4, 5].

In general, the last observation is called as reference image **X** and the current observation is query image **Y**. Previous CD methods [6–10] design various models to detect changes by using both **X** and **Y**. Feng et al. [6] proposed an iterative optimisation CD method by modelling the CD as iteration of camera pose alignment, lighting correction and low-rank from two-time observations. Alcantarilla et al. [8] combined **X** and **Y** into a six-channel image and employed a fully convolutional network [11] to generate changes. However, as shown in the top three rows of Figure 1, in most real-world scenes, the change always occurs in a single image. Detecting changes from the image pair needs to extract features from both of the observed images, which might waste computational resources. As shown in the bottom two rows of Figure 1, ADCDnet, an image-pair-based CD method, obtains imperfect CD results because of the large camera view difference between the first image pair and the strong shadows occurred in the unchanged image of the second image pair. Due to the long time span of two observations, there might be camera pose misalignment, lighting differences and other influences between the two observations. Detecting changes from the image pair might affect the precision of the detection results.

In this study, we aim to solve a novel problem that involves detecting the change from the change image. To this end, we first propose a change image selector (CIS) to identify the image containing changes. The CIS is formulated as a binary image classifier based on spatial pyramid pooling (SPP) [12] with the combined features of both observed images. Then, we present a coarse change prior map generator (CCPMG) to generate a coarse change prior map $\mathbf{M_p}$ with absolute difference of features from two images to indicate the position of change. We refine $\mathbf{M_p}$ by using a multi-scale CD module with

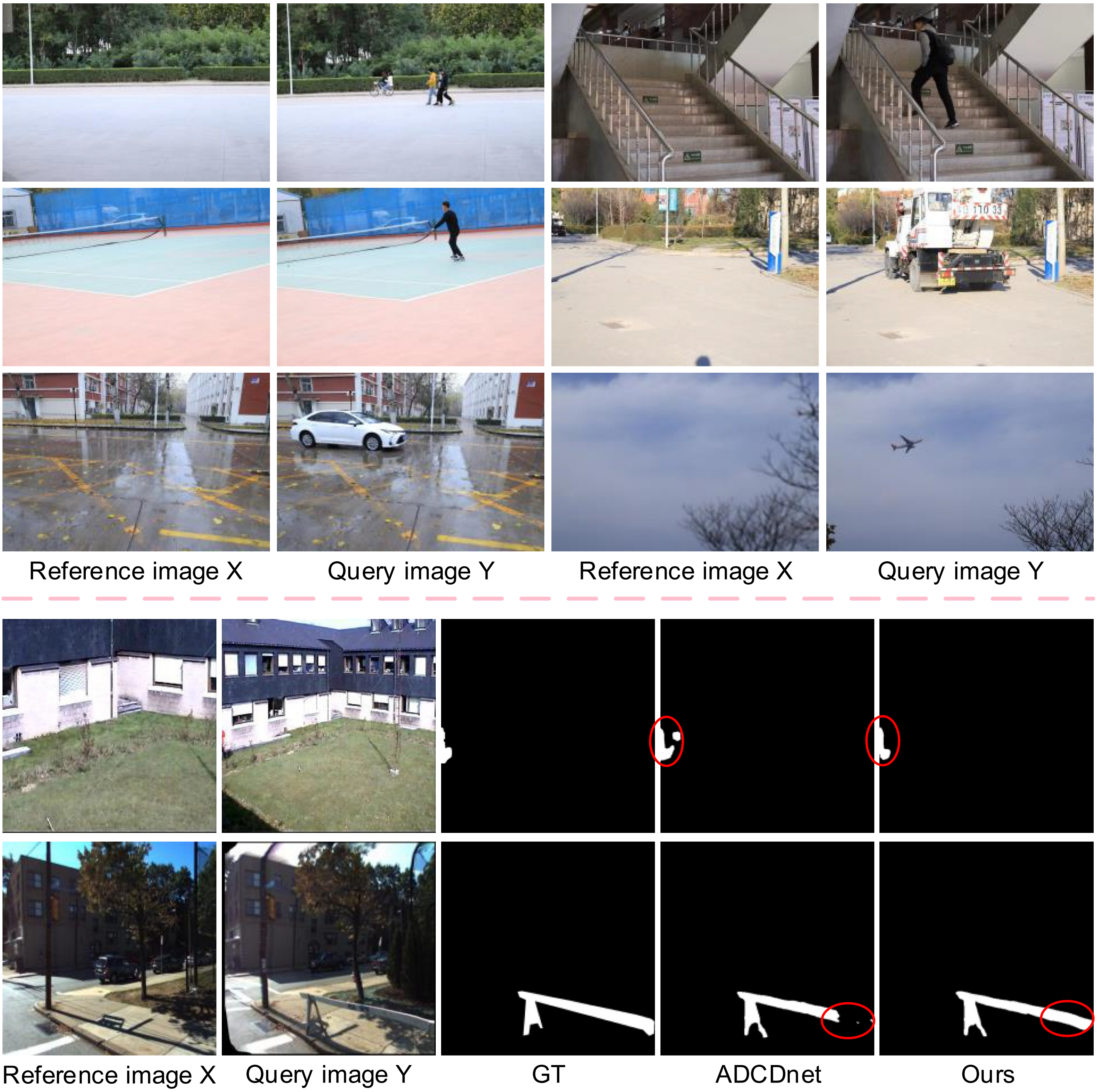

FIGURE 1 Motivation of the proposed method. The top rows show examples of real scenes for CD. Change always occurs in the query image. The bottom rows show the CD results of an image-pair-based CD method (e.g. ADCDnet) and our proposed single-image-based method

the features extracted from the selected change image. Without combining the features from the unchange image, the proposed method potentially obtains the following merits: 1) it does not need to learn convolutional layers for fusing the features of two images, 2) it can alleviate the bad affections caused by the unchanged image, and 3) it is highly efficient. Our main contributions can be summarised as follows:

- We propose to detect the change from the image containing change. To this end, we propose a change image selector to identify the image containing change and a coarse change prior map generator to generate change cues.
- With change cues, our method only uses the features of the change image in the multi-scale CD module and refines the features in a bottom-up manner. Our method is highly efficient without using the features of both images.
- Although we only use change image features in the multi-scale CD module, our method can achieve better results than those CD methods using both images in the whole process.

2 | RELATED WORK

2.1 | Traditional CD methods

Traditional CD methods employ handcrafted features and well-designed algorithms to generate pixel-level difference. Intensity difference of image pairs is used for early CD methods [13]; however, image intensity is easily affected by camera pose variation and lighting difference, which results in unstable CD results. Malila et al. [14] proposed change vector analysis (CVA) to detect forest change by thresholding on the change vector that was described by the magnitude and direction of change. Chen et al. [15] presented a double-window flexible pace search (DFPS) to determine reasonable thresholds of change magnitude and change direction. More complex methods such as principal component analysis (PCA), classification, and optimisation are introduced into CD to generate high-quality results. Li et al. [16] stacked two multi-band images and compressed them with PCA to form a PCA image for later interactive supervised classification to monitor rapid changes of land use and urban expansion. Gong et al. [17] proposed using a fuzzy clustering algorithm to classify changed and unchanged regions in the fused difference image. Feng et al. [6] proposed an iteration optimisation change detection by modelling the changes as iteration of camera pose alignment, lighting correction, and low-rank from two observations. Stent et al. [18] proposed using the generalised Patch-Match correspondence algorithm to align images and using the thin plate spline model to estimate the illumination variation to conquer the camera pose variation and lighting difference between reference and query images. Gharbia et al. [19] used the log ratio of two images after being registered with Scale-Invariant Feature Transform (SIFT) to detect the change. For stable monitoring scenes, the change can be detected by background subtraction [20–23].

Traditional CD methods do not need many training images and are easy to implement; however, they suffer from camera pose variation and lighting difference. Thus, how to extract effective features, set thresholds and train robust classifiers are essential to obtain promising CD results.

2.2 | Deep learning based CD methods

CNN improves the performance of various computer vision tasks by simultaneously learning feature extractors and classifiers. Most of the state-of-the-art CD methods are designed with the CNN to avoid the tediousness of designing features and learning classifiers. Gong et al. [24] created a deep neural network by stacking a restricted Boltzmann machine for detecting the change in the Synthetic Aperture Radar (SAR) images. Sakurada et al. [25] used the CNN to extract features from the image grid of image pairs and calculated the distance of each grid to form a dissimilarity map. Huang et al. [7] proposed a camera pose correction network and a fine-grained CD network to detect the fine-grained change of high-value scenes. Mou et al. [9] proposed a recurrent CNN to find the change in the earth's surface. Jing et al. [26] proposed a tri-Siamese-LSTM to detect the change in remote sensing images with very high resolution. Zhang et al. [27] proposed an ensemble CNN to obtain change detection by reducing the discriminative distance of unchanged samples and enlarging the discriminative distance of changed samples. Huang et al. [28] study different fluid pyramid integration networks for CD. Deep learning (DL) based CD can extract effective features from images and conquer the camera pose variation and lighting difference as well as obtain promising CD results.

Besides CD, RGB-D salient detection [29, 30], co-salient detection [31, 32] and video object segmentation [33, 34] also utilise multiple images to achieve high-detection performance by using additional information. As the depth map only provides additional information for RGB-D salient detection, one can distill the depth branch by making models potentially learn the depth map from the RGB image. Piao et al. [35] proposed to use a knowledge distiller to transfer the depth knowledge from the depth stream to the RGB stream. Unlike RGB-D salient detection, the change can be found merely by comparing two images, which makes us unable to use the change image for CD. As discussed in Section 1, in most real-world applications, the change always occurs in a single image. If we know the coarse position of the change on the change image, we can generate precise CD results from the change image. In this paper, we propose an efficient CD method by selecting the change image and coarse change prior map generation to detect the change, which can alleviate the bad effects of the unchanged image.

3 | METHOD

To detect the change from the change image, we need to identify the change image and generate a coarse change prior map first. We use very simple network architectures for the change image selector and coarse change prior map generator. Then, we design a multi-scale change detection module to detect change from the change image. In the following section, we introduce the basic feature extraction, change image selector, coarse change prior map generator, multi-scale change detection, and training policies of the proposed method. The Figure 2 shows our proposed network architecture.

3.1 | Basic feature extraction

We can use any state-of-the-art image classification network as our feature extraction backbone. Without loss of generality, we adopt VGG16 [36] as the basic feature extraction network. It has five convolutional modules labelled Conv1, …, Conv5. The output of the last convolutional layer in each convolutional module is our basic feature. We use $\mathbf{F}_{\mathrm{X}}^{\mathrm{Conv}_1}$ and $\mathbf{F}_{\mathrm{Y}}^{\mathrm{Conv}_1}$ to denote the features of the first convolutional module that are extracted from $\mathbf{X}$ and $\mathbf{Y}$, respectively. To make the proposed model take less time in selecting the change image and generating coarse change prior map, we only use $\mathbf{F}_{\mathrm{X}}^{\mathrm{Conv}_1}$ and $\mathbf{F}_{\mathrm{Y}}^{\mathrm{Conv}_1}$ for the CIS and CCPMG. After obtaining the change image, all subsequent features $\mathbf{F}_{\mathrm{C}}^{\mathrm{Conv}_i}$, $i = 1, 2, \ldots, 5$ are extracted from the change image.

To increase the feature representation ability, we fuse the features, such as the method proposed in UNet [37], which shows superior feature extraction ability for semantic segmentation [38, 39] and change detection [10, 28]. The fused convolutional features of the $(i + 1)$-th layer are concatenated with those convolutional features of the ith layer. Thus, the features of low layers can encode high-level semantic information. The ith fused convolutional feature can be denoted as $\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc}_i}$, which can be computed by

$$\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc}_i} = \psi\left(cat\left(\mathbf{F}_{\mathrm{C}}^{\mathrm{Conv}_i}, \mathbf{F}_{\mathrm{C}}^{\mathrm{Enc}_{i+1}}\right)\right), s.t. \quad i = 1, \ldots, 4 \qquad (1)$$

where $cat(\cdot)$ denotes concatenate operation and $\psi(\cdot)$ includes three convolutional blocks, each of which is formed by a convolutional layer, a batch normalisation layer and a ReLU layer. We upsample the features to twice the size for later fusion. $\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc}_5}$ only uses $\mathbf{F}_{\mathrm{C}}^{\mathrm{Conv}_5}$, that is $\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc}_5} = \psi\left(\mathbf{F}_{\mathrm{C}}^{\mathrm{Conv}_5}\right)$.

3.2 | Change image selector

Identifying the change image is important for the proposed method. We concatenate $\mathbf{F}_{\mathrm{X}}^{\mathrm{Conv}_1}$ and $\mathbf{F}_{\mathrm{Y}}^{\mathrm{Conv}_1}$ together to generate features for a three-layer fully connected network. However, the concatenated convolutional features of variant-sized images have different resolutions, which are not flexible. We adopt spatial pyramid pooling (SPP) [12] to generate a fixed-length feature vector. Figure 3 shows the network architecture of CIS. The first layer is the input layer that takes a 1344 dimensional feature vector generated by SPP as input. The second layer is a hidden layer that has 128 neurons. The last layer consists of two neurons to compute the probability of selecting **X** or **Y** to be the change image. If the first probability is larger than the second, then **X** is selected as the change image. Otherwise, **Y** is selected as the change image. We use 0 to denote that **X** is selected and 1 to denote that **Y** is selected in Figure 2.

3.3 | Coarse change prior map generator

Previous CD methods use image difference and threshold to generate the change map. However, camera pose variation and lighting difference result in poor CD results. Liu et. al [40] showed that a high-layer convolutional feature can extract semantic information. However, extracting features in the higher layer from two images requires high computation cost, which affects the efficiency of the detector. As shown in Figure 4, CCPMG is a simple Siamese network with three convolutional blocks, which reduce the resolution of $\mathbf{F}_{\mathrm{X}}^{\mathrm{Conv}_1}$ and $\mathbf{F}_{\mathrm{Y}}^{\mathrm{Conv}_1}$ to 80 × 80, 40 × 40 and 20 × 20 to capture multi-scale information of the changed object. Then, the features at the corresponding levels are subtracted and convolved to generate absolute difference features $\mathbf{F}_{\mathrm{AD}}^{\mathrm{i}}$, where $i = 1, 2, 3$. The features are resized and concatenated to generate $\mathbf{F}_{\mathrm{AD}}^{\mathrm{Fusion}}$ by

$$\mathbf{F}_{\mathrm{AD}}^{\mathrm{Fusion}} = Conv\left(cat\left(Bi\left(\mathbf{F}_{\mathrm{AD}}^{1}\right), \mathbf{F}_{\mathrm{AD}}^{2}, Bi\left(\mathbf{F}_{\mathrm{AD}}^{3}\right)\right), 512\right), \qquad (2)$$

where $Bi(\cdot)$ is a bilinear sampling operation and $Conv(\cdot, 512)$ denotes a convolutional layer that outputs 512 feature maps. $\mathbf{M}_{\mathbf{p}}$ can be generated as

$$\mathbf{M}_{\mathbf{p}} = Conv\left(\mathbf{F}_{\mathrm{AD}}^{\mathrm{Fusion}}, 2\right), \qquad (3)$$

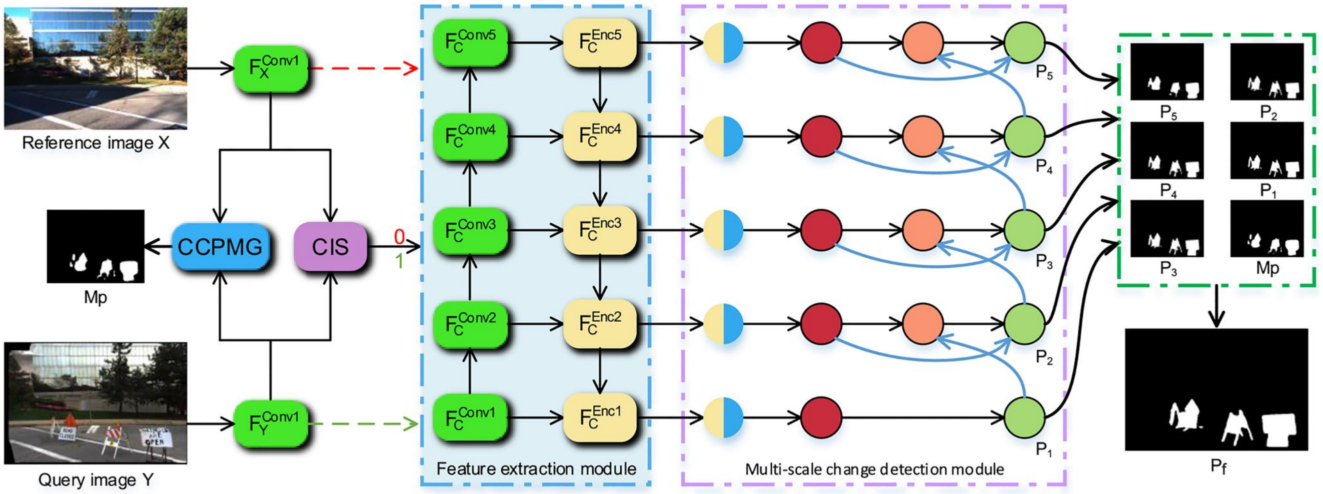

FIGURE 2 The network architecture of the proposed change detection method. CCPMG is the coarse change prior map generator. $\mathbf{M}_{\mathbf{p}}$ is the coarse change prior map. CIS denotes the change image selector. 0 denotes that **X** is the change image. 1 denotes that **Y** is the change image. The circle with yellow and blue colours means concatenation of $\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc}_i}$ and $\mathbf{F}_{\mathrm{AD}}^{\mathrm{Fusion}}$. The multi-scale predictions will be concatenated with the M_p to generate the final prediction P_f. We omit the supervise signal to make the framework clear

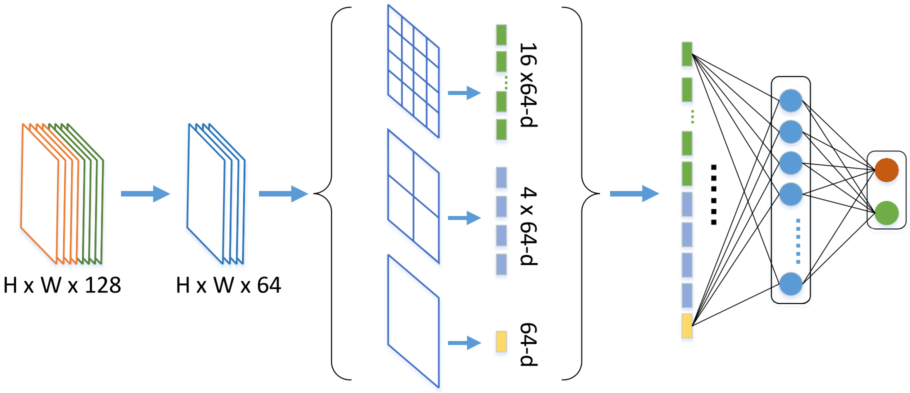

FIGURE 3 The network architecture of CIS. Note that two colours of the input feature maps represent the features extracted from two images

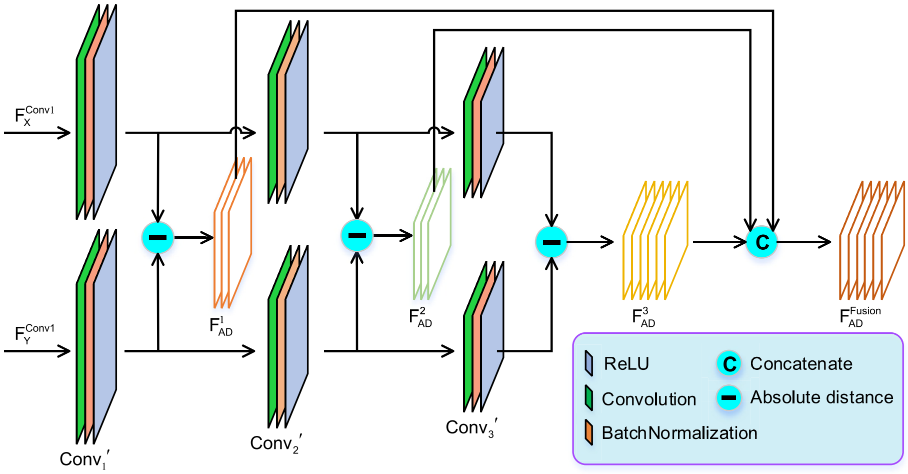

FIGURE 4 The network architecture of CCPMG

where $Conv(\cdot, 2)$ is a convolutional layer to generate a two-channel map wherein one channel map is the changed probability map. The other is the unchanged probability map. As shown in Figure 5, we can find that $\mathbf{M_p}$ is capable of locating the changed objects. However, the detected changed objects are too coarse for their details to be detected. Note that $\mathbf{M_p}$ is only used to calculate the loss of CCPMG. We use $\mathbf{F}_{\mathrm{AD}}^{\mathrm{Fusion}}$ in later multi-scale change detection, which has more information of changes than $\mathbf{M_p}$ as shown in Figure 10.

3.4 | Multi-scale change detection

$\mathbf{M_p}$ only describes the coarse position of the change, which needs further refinement. We resize feature $\mathbf{F}_{\mathrm{AD}}^{\mathrm{Fusion}}$ according to the size of $\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc_i}}$ and fuse them with $\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc_i}}$ through convolutional layers. The resulting features encode the information of the changed object. The features are called as $\mathbf{F}_i$. We propose a multi-scale change detection module with bottom-up feature enhancement. The feature of bottom-up fusion can be computed as

$$\mathbf{F}_i' = \phi(cat(\phi(cat(\phi(\mathbf{F}_i), Dn(\mathbf{F}_{i-1}'))), \phi(\mathbf{F}_i))) \tag{4}$$

where $i = 2, \ldots, 5$, $Dn(\cdot)$ denotes down-sampling operation, $\phi(\cdot)$ denotes operation of convolution, Batch Normalisation and ReLU. Note that $\mathbf{F}_1' = \phi(\mathbf{F}_1)$. We generate prediction from $\mathbf{F}_i'$ by

$$\begin{aligned} \mathbf{P}_i &= Conv(\mathbf{F}_i', 2), \\ \mathbf{P}_f &= Conv(cat(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5, \mathbf{M_p}), 2), \end{aligned} \tag{5}$$

where $i = 1, \ldots, 5$. We add supervisions on $\mathbf{P}_i$ and $\mathbf{P}_f$ for fast convergence. $\mathbf{P}_f$ is used as the final change detection result.

3.5 | Training policies

We use cross entropy and binary cross-entropy losses for CIS and change detection, respectively. The training and testing of the proposed network are conducted on the PyTorch [41] platform with Nvidia 2080Ti. The basic learning rate of

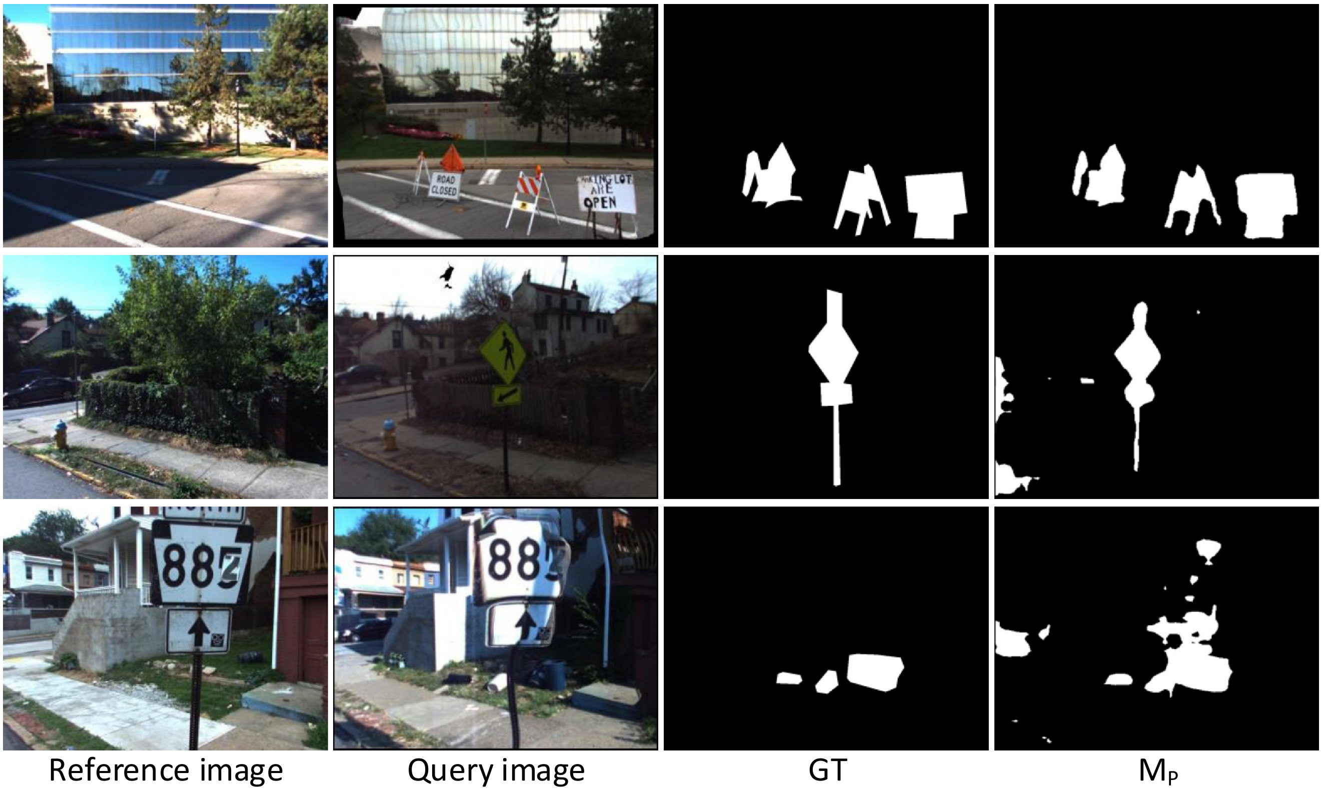

FIGURE 5 Examples of generated coarse change prior map $\mathbf{M_P}$

the entire network is set to 1e-3. We set the batch size to 4. The parameters are updated by the Adam algorithm with the momentum of 0.9 and the weight decay of 0.999. All models are trained with 20 epochs to alleviate overfitting.

4 | EXPERIMENT

4.1 | Setup

Baselines We compare the proposed method with five CD methods, namely MFCNET [42], FCN [11], ARPPNET [43], SEU-Net2 [44] and ADCDnet [10].

MFCNET [42] combines MatchNet [45] and the fully convolutional network [11] to identify temporal changes in multiple images. MatchNet [45] is used to yield high-level features from a pair of images, whereas the fully convolutional network [11] is used to encode and decode for the image pair.

FCN [11] takes the stacked image pair as the input of a fully convolutional network to detect the change.

ARPPNET [43] proposes a spatiotemporal convolutional network with retrospective convolution to detect the change between the current frame and frames from historical observation.

SEU-Net2 [44] is a Siamese encoder U-Net used to encode the foreground and background to find the difference. Then, the decoder interprets the difference to the binary mask. In the following experiments, SEU-Net2 shows the best performance among the compared methods used in this paper.

ADCDnet [10] fuses the absolute difference of multi-scale deep convolutional features of image pairs to generate CD results.

All compared methods are implemented with PyTorch [41] and trained with the same super parameters on the same training set.

Datasets We use the object-level CD benchmark datasets, which is VL-CMU-CD [8] and CDnet [46], in our experiment. Most changes of these datasets occur in a single image, which coincides with the idea of the proposed method.

VL-CMU-CD [8] has 1362 image pairs of street views with a long time span and various changes. The reference and query images of VL-CMU-CD have different shooting angles, shades, weather, and seasonal changes. The resolution of the images in this dataset is 1024 × 768. We randomly select 80% image pairs for training and the left 20%, for testing. To expand the data, we randomly crop five image pairs from each sampled image pair to generate a training set with 4001 image pairs.

CDnet [46] has 31 videos that are captured in different scenarios. The maximum resolution of the video frames is 720 × 480. The videos that have small ROI regions are removed. We select the image that has no changed objects

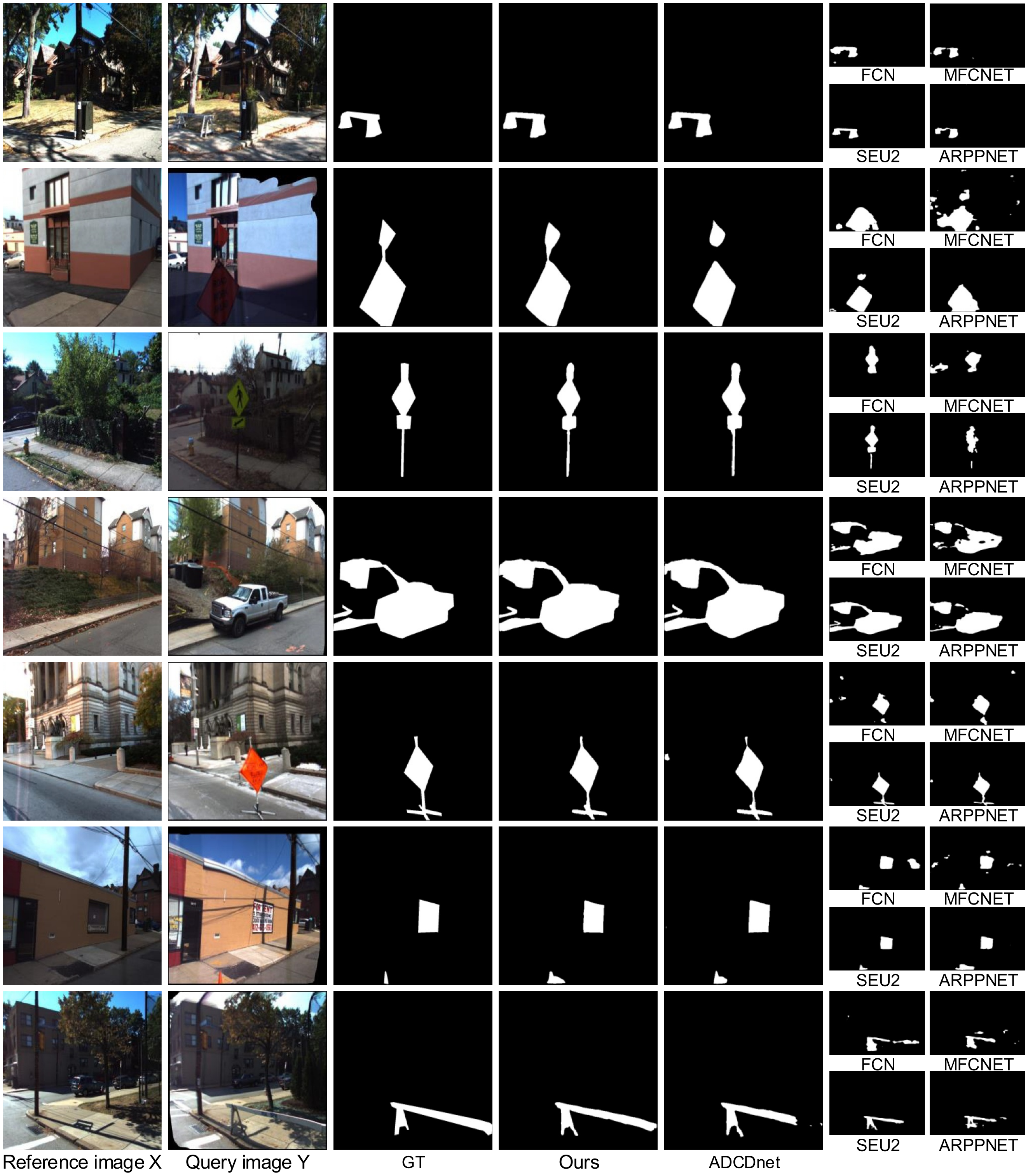

FIGURE 6 The CD results of different change detectors on VL-CMU-CD

as reference image **X**, and the image that has changed as query image **Y**. We randomly select 80% image pairs of each scene to construct the training set. The left image pairs are used as the testing set. Finally, we obtain a training set with 40,148 image pairs and a testing set with 10,061 image pairs.

To reduce the training time and obtain accurate detection results, we resize the images to 320 × 320 for training and 480 × 480 for testing. Experimental results show that the larger the image, the more detailed information is retained.

To train the CIS, we manually label the change image on the two benchmark datasets. The image containing change is labelled as ‘0’. The image without change is labelled as ‘1’. If both of the images contain change, then we label the image with the large change object as the change image. If **X** is a change image, the label of the output of CIS is ‘0’. If **Y** is a change image, the label of the output of CIS is ‘1’.

Criteria We use F1-measure (F1), precision (Pre), recall (Re), specific (Sp), false positive ratio (FPR), false negative ratio (FNR) and percentage of wrong classification (PWC) to evaluate different change detection methods. Among these criteria, F1 is the harmonic mean of precision and recall, which is the most important criterion.

4.2 | Results and analysis

4.2.1 | Results on VL-CMU-CD

Some CD results of different change detectors are shown in Figure 6. Our method is capable of detecting changed objects in various scenarios. In the second row of Figure 6, only the proposed method detects the pole, which demonstrates that our method is good at processing the details of the changed object. In the last row of Figure 6, our method detects the entire changed object while other methods only detect parts of the changed object. Detecting from the change image may alleviate introducing interference from another image, thereby improving the CD results.

Table 1 shows the quantitative results of the different change detectors on VL-CMU-CD. The F1 values of ADCDnet and SEU-Net2 are 0.9342 and 0.9251, respectively. They are the only two methods among the compared CD methods whose F1 values are higher than 0.9. The F1 value of the proposed method is 0.9399, which is the highest one on VL-CMU-CD. Compared with ADCDnet and SEU-Net2, the proposed method achieves 0.61% and 1.60% relative F1 value improvements, respectively. Besides the F1 value, other criteria of the proposed method are better than those of compared methods.

TABLE 1 Quantitative comparison of different change detection methods on VL-CMU-CD. The first three values for each metric are marked red, green and blue, respectively

Method	F1 ↑	Re ↑	Pre ↑	Sp ↑	FPR ↓	FNR ↓	PWC ↓
MFCNET	0.7922	0.7854	0.8286	0.9879	0.0121	0.1714	2.1815
FCN	0.7947	0.7806	0.8504	0.9878	0.0122	0.1496	2.1508
ARPPNET	0.8576	0.8816	0.8554	0.9932	0.0068	0.1446	1.5913
SEU-Net2	0.9251	0.9287	0.9308	0.9962	0.0038	0.0692	0.8557
ADCDnet	0.9342	0.9311	0.9434	0.9960	0.0040	0.0566	0.7818
Ours	0.9399	0.9357	0.9473	0.9963	0.0037	0.0527	0.6531

4.2.2 | Results on CDnet

Figure 7 shows the CD results of various CD methods on CDnet. We can find that the proposed method can detect the changed object under different scenes. Compared with other CD methods, the proposed method has two merits. The first merit is that the detected changed objects are more complete than that of other compared methods, such as the bicycle in the second row and the person in the fifth row of Figure 7. The second merit is that the detected results of the proposed method are clearer in the background as shown in the third and seventh rows of Figure 7.

Table 2 shows the quantitative results of the various change detectors on CDnet. The F1 value of ADCDnet is 0.8621, which is the only method among the compared CD methods whose F1 values are higher than 0.85. The F1 value of the proposed method is 0.8826, which is the highest one on CDnet. Compared with ADCDnet, the proposed method achieves 2.4% relative F1 value improvements. Besides the F1 value, the other criteria of the proposed method are also increased.

4.2.3 | Running time

We test the running time of various CD methods on a 480 × 480 image pair. MFCNET is the fastest method, which takes 0.0171s. ADCDnet takes 0.1020s to obtain the best performance. Our method takes 0.0986s to process an image pair. Figure 8 shows the scatter of F1 value and running time. Although the proposed method and ADCDnet are slower than other compared CD methods, they achieve better performance.

4.2.4 | Failure cases

The basic assumption of the proposed method is that the change occurs on a single image. However, as shown in the first row of Figure 9, in certain scenes, changes occur on both of the two images. The proposed method may fail to detect the change. Besides, our method faces difficulty in detecting a small change that has similar appearance with the background caused by the intense light irradiation, such as the scene shown in the second row of Figure 9. In the third row of Figure 9, our method only detects a small part of the changed object.

5 | ABLATION STUDY

We conduct all ablation studies on VL-CMU-CD because it has a relatively smaller training set than CDnet.

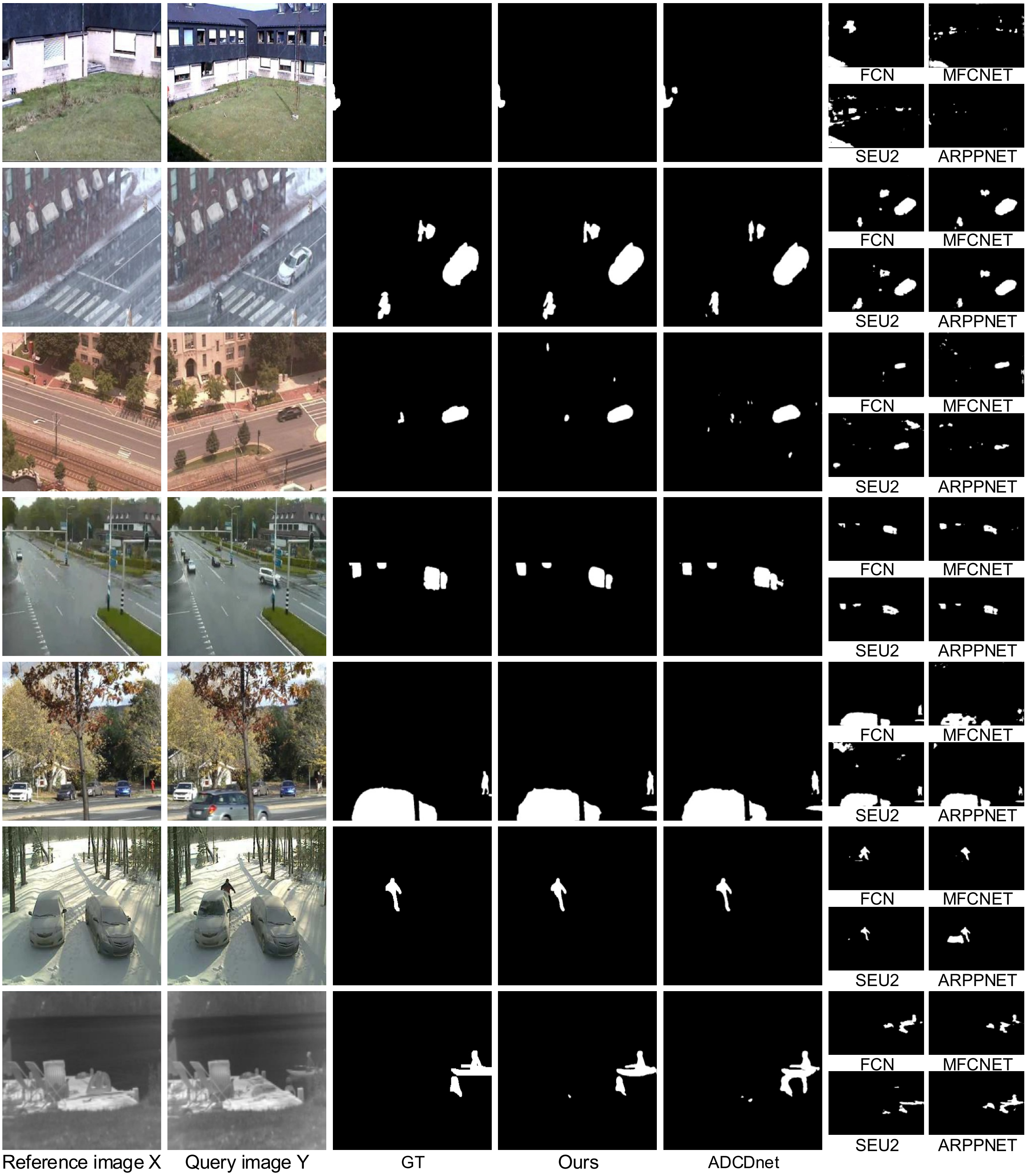

FIGURE 7 The CD results of different change detectors on the CDnet dataset

5.1 | Performance of CIS using different convolutional features

Selecting the change image correctly and efficiently is essential for the proposed method. We report the classification accuracy and running time of CIS using features from different convolutional layers that are extracted from images with the size of 320 × 320 in Table 3. Except $Conv_1$, the classification accuracy of CIS with $Conv_{2,3,4,5}$ is 100%. Although the accuracy of using the features from $Conv_1$ is merely 99.4%, the running time is 6.3 ms, which is the fastest. Thus, we use the features extracted from $Conv_1$ for efficiency.

5.2 | Effect of size of $\mathbf{F}_{\mathrm{AD}}^{\mathrm{Fusion}}$

In CCPMG, we use a simple Siamese network with three convolutional blocks to reduce the resolution of $\mathbf{F}_{\mathrm{X}}^{\mathrm{Conv}_1}$ and $\mathbf{F}_{\mathrm{Y}}^{\mathrm{Conv}_1}$ to 80 × 80, 40 × 40 and 20 × 20. Thus the sizes of $\mathbf{F}_{\mathrm{AD}}^{1}$, $\mathbf{F}_{\mathrm{AD}}^{2}$ and $\mathbf{F}_{\mathrm{AD}}^{3}$ are 80 × 80, 40 × 40 and 20 × 20, respectively. We can resize $\mathbf{F}_{\mathrm{AD}}^{1,2,3}$ to different sizes and generate a coarse change prior map. We resize these features into 20 × 20, 40 × 40, 80 × 80 and 160 × 160 and generate $\mathbf{M}_{\mathbf{p}}$. Table 4 shows the quantitative results of using different feature sizes in CCPMG. We can find that using 40 × 40 achieves the best performance. Thus, we adopt a feature size of 40 × 40 for CCPMG in our experiment.

5.3 | Visualisation of $\mathbf{F}_{\mathrm{AD}}^{\mathrm{Fusion}}$

We show some examples of the heat maps and feature maps of learnt F_{AD}^{fusion} in Figure 10. We can find that the response values of the heat maps of the changes are higher than that of the backgrounds. From the feature maps of F_{AD}^{fusion}, we can find that parts of feature maps have high values on changes and other feature maps have high values on backgrounds. That is to say, the feature maps of F_{AD}^{fusion} already have the ability of localising the changes in each image pair. Thus, it is not necessary to incorporate M_p in our multi-scale change detection module.

TABLE 2 Quantitative comparison of different change detection methods on CDnet. The first three values for each metric are marked red, green and blue, respectively

Method	F1 ↑	Re ↑	Pre ↑	Sp ↑	FPR ↓	FNR ↓	PWC ↓
MFCNET	0.7416	0.7784	0.7517	0.9926	0.0074	0.2483	1.4072
FCN	0.7933	0.8286	0.7951	0.9951	0.0049	0.2049	1.1034
ARPPNET	0.8339	0.8354	0.8656	0.9956	0.0044	0.1344	0.8178
SEU-Net2	0.8103	0.7923	0.8818	0.9907	0.0093	0.1182	1.1468
ADCDnet	0.8621	0.8898	0.8554	0.9965	0.0035	0.1446	0.6587
Ours	0.8826	0.8838	0.9017	0.9968	0.0032	0.0983	0.5188

5.4 | Effectiveness of $\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc}_i}$

We fuse the features in a top-down manner in the basic feature extraction part. The generated feature is denoted as $\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc}_i}$. We directly use the original features of VGG16 to demonstrate the effectiveness of this additional operation. Table 5 shows the performance of different experiment setups. Without using $\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc}_i}$, the F1 value of our method drops to 0.9368, which demonstrates that $\mathbf{F}_{\mathrm{C}}^{\mathrm{Enc}_i}$ is useful for the proposed CD method.

5.5 | Importance of bottom-up feature enhancement

Multi-scale change detection module adopts bottom-up feature enhancement. We conduct experiments on the proposed network by removing bottom-up feature enhancement in the multi-scale change detection module. As shown in Table 6, the F1 value of removing bottom-up feature enhancement for the proposed method is 0.9342, which is lower than the full version of the proposed method. This result demonstrates that the bottom-up feature enhancement can improve the feature representational ability and lead to better CD performance.

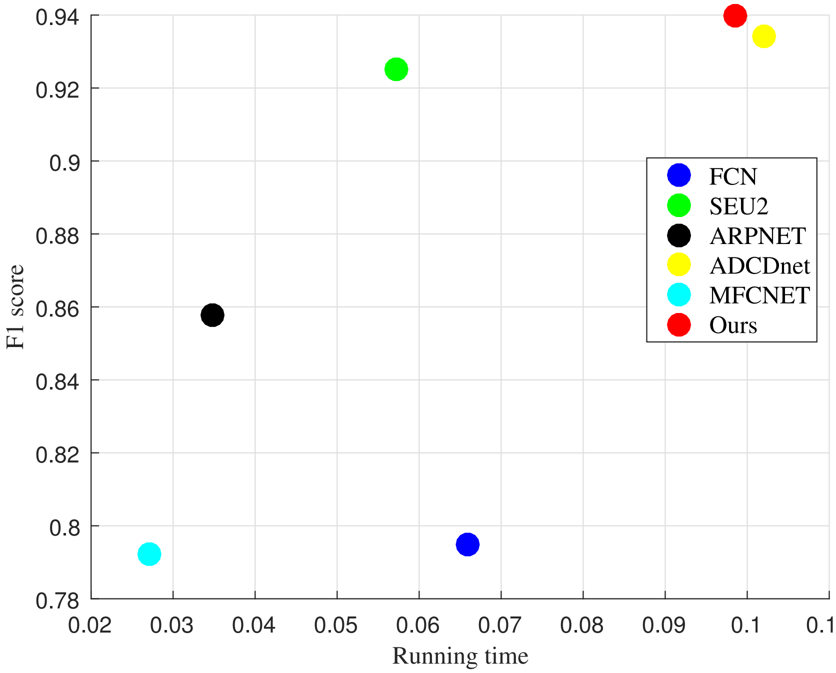

FIGURE 8 F1 value and running time trade-off for different change detectors

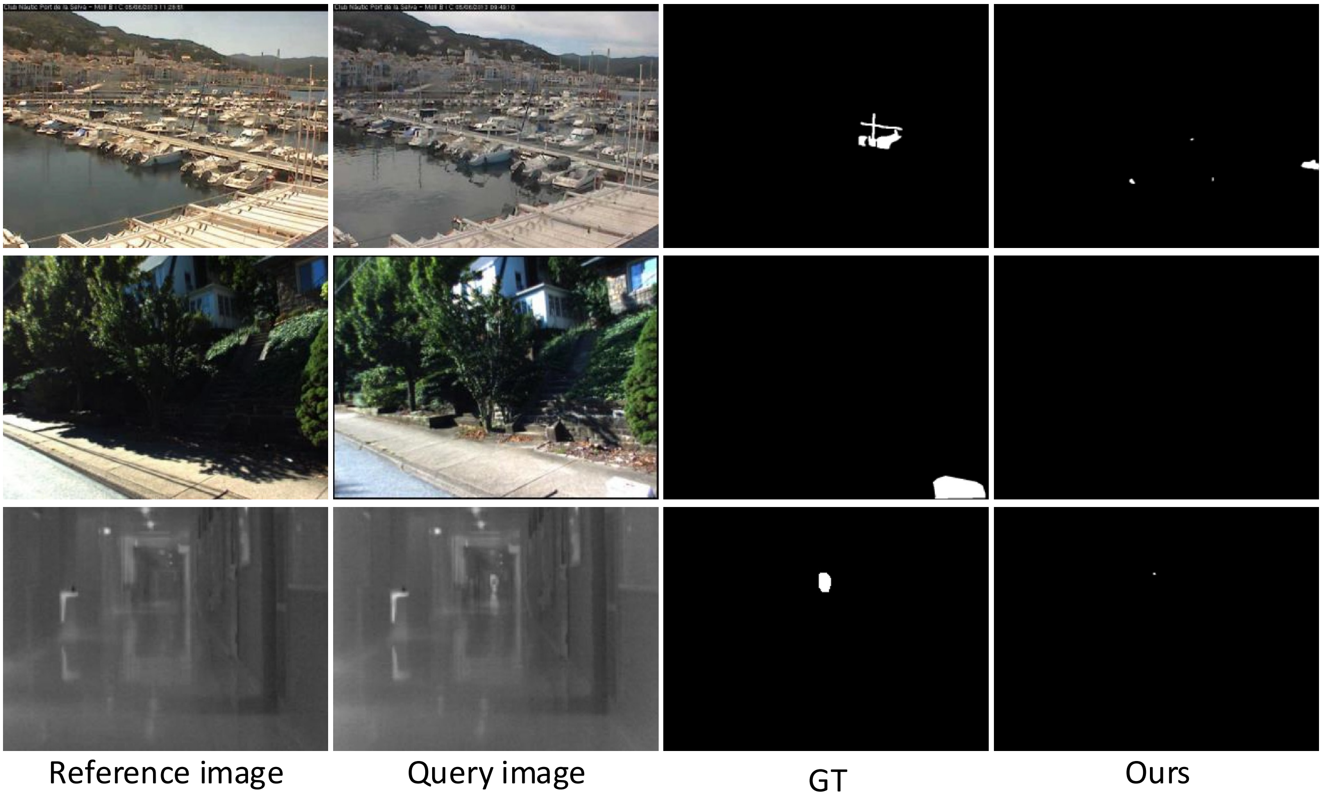

FIGURE 9 Fail cases of the proposed method

TABLE 3 The accuracy and processing time of CIS using features of Conv1-5 of VGG16

	Conv$_1$	*Conv*$_2$	*Conv*$_3$	*Conv*$_4$	*Conv*$_5$
Accuracy	99.4%	100%	100%	100%	100%
Time (ms)	6.3	6.9	7.9	8.6	9.5

TABLE 4 Different feature sizes for CCPMG. The **bold** is the best

Size	F1 ↑	Re ↑	Pre ↑	Sp ↑	FPR ↓	FNR ↓	PWC ↓
20	0.8935	**0.9048**	0.8927	**0.9950**	**0.0050**	0.1073	1.0124
40	**0.9086**	0.9022	0.9268	0.9943	0.0057	0.0732	**0.9464**
80	0.9074	0.8959	**0.9325**	0.9941	0.0059	**0.0675**	0.9576
160	0.9043	0.9047	0.9156	0.9947	0.0053	0.0844	1.0369

TABLE 5 The performance of the proposed method without $\mathbf{F}_{C}^{\mathrm{Enc}_i}$ (w/o Enc) in the basic feature extraction part. The **bold** is the best

	F1 ↑	Re ↑	Pre ↑	Sp ↑	FPR ↓	FNR ↓	PWC ↓
w/o Enc	0.9368	0.9295	**0.9480**	0.9959	0.0041	**0.0520**	0.6842
Ours	**0.9399**	**0.9357**	0.9473	**0.9963**	**0.0037**	0.0527	**0.6531**

TABLE 6 The performance of the proposed method without using bottom-up feature enhancement. The **bold** is the best

	F1 ↑	Re ↑	Pre ↑	Sp ↑	FPR ↓	FNR ↓	PWC ↓
w/o bottom-up	0.9342	0.9337	0.9391	0.9961	0.0039	0.0609	0.7077
Ours	**0.9399**	**0.9357**	**0.9473**	**0.9963**	**0.0037**	**0.0527**	**0.6531**

6 | CONCLUSION

We have proposed an efficient change detection method with the change image. The proposed method contains the change image selector, coarse change prior map generator, and multi-scale change detection. We can easily obtain the change image and coarse position of the change with simple network architectures. Then, we treat CD as a coarse change map refinement process by a multi-scale CD module with the features extracted from the change image. Without using the features of both images in the multi-scale CD module, our method can alleviate the bad affections caused by the unchanged image. The resulting network is highly efficient in achieving high detection performance. We have conducted numerous experiments on VL-CMU-CD and CDnet datasets to analyse our method. In our future work, we will investigate a more complex problem where the change occurs on both images.

ACKNOWLEDGMENT

The authors would like to thank the reviewers and editors for their hard work and constructive suggestions. This work was

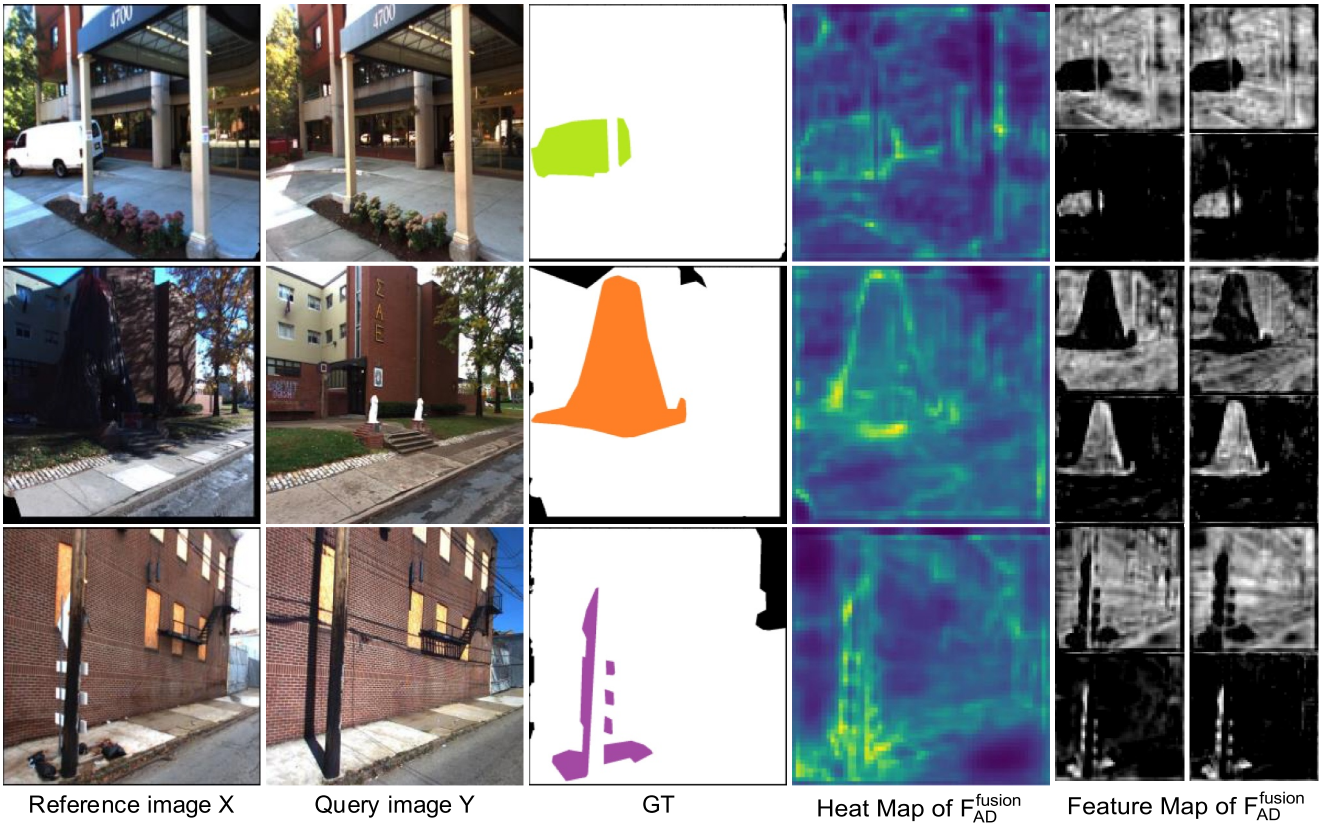
Reference image X
Query image Y
GT
Heat Map of F_{AD}^{fusion}
Feature Map of F_{AD}^{fusion}

FIGURE 10 Examples of the heat maps and feature maps of learnt F_{AD}^{fusion}

supported by the Natural Science Foundation of Tianjin under Grant No. 20JCQNJC00720.

CONFLICT OF INTEREST

We declare that we have no financial and personal relationships with other people or organisations that can inappropriately influence our work, and there is no professional or other personal interests of any nature or kind in any product, service and/or company that could be constructed as influencing the position presented in, or the review of, the manuscript entitled.

DATA AVAILABILITY STATEMENT

Data openly available in a public repository that issues datasets with DOIs.

ORCID

Rui Huang https://orcid.org/0000-0002-3343-066X

REFERENCES

1. Velastin, S.A., Orwell, J., Buch, N.: A review of computer vision techniques for the analysis of urban traffic. IEEE Trans. Intell. Transport. Syst. 12(3), 920–939 (2011)
2. Brunner, D., Bruzzone, L, Lemoine, G.: Change detection for earthquake damage assessment in built-up areas using very high resolution optical and sar imagery. In: 2010 IEEE International Geoscience and Remote Sensing Symposium, IEEE, pp. 3210–3213. (2010)
3. Khan, S.H., et al.: Forest change detection in incomplete satellite images with deep neural networks. IEEE Trans. Geosci. Rem. Sens. 55(9), 5407–5423 (2017)
4. Liu, H., Chen, S., Kubota, N.: Intelligent video systems and analytics: a survey. IEEE Trans. Ind. Inf. 9(3), 1222–1233 (2013)
5. Liang, D., et al.: Adaptive local spatial modeling for online change detection under abrupt dynamic background. In: 2017 IEEE International Conference on Image Processing (ICIP), IEEE, pp. 2020–2024. (2017)
6. Feng, W., et al.: Fine-grained change detection of misaligned scenes with varied illuminations. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1260–1268. (2015)
7. Huang, R., et al.: Learning to detect fine-grained change under variant imaging conditions. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 2916–2924. (2017)
8. Alcantarilla, P.F., et al.: Street-view change detection with deconvolutional networks. Aut. Robots. 42(7), 1301–1322. (2018)
9. Mou, L., Bruzzone, L., Zhu, X.X.: Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. IEEE Trans. Geosci. Rem. Sens., 1–12 (2018)
10. Huang, R., et al.: Change detection with absolute difference of multiscale deep features. Neurocomputing. 418, 102–113. (2020)
11. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440. (2015)
12. He, K., et al.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. 37(9), 1904–16 (2014)
13. Muchoney, M.D., Haack, N.B.: Change detection for monitoring forest defoliation. Photogramm. Eng. Rem. Sens. 60(10), 1243–1251 (1994)
14. Malila, W.A.: Change vector analysis: an approach for detecting forest changes with landsat. In: LARS symposia, p. 385. (1980)
15. Chen, J., et al.: Land-use/land-cover change detection using improved change-vector analysis. Photogramm. Eng. Rem. Sens. 69(4), 369–379 (2003)

16. Li, X., Yeh, A.G.O.: Principal component analysis of stacked multi-temporal images for the monitoring of rapid urban expansion in the pearl river delta. Int. J. Rem. Sens. 19(8), 1501–1518 (1998)
17. Gong, M., Zhou, Z., Ma, J.: Change detection in synthetic aperture radar images based on image fusion and fuzzy clustering. IEEE Trans. Image Process. 21(4), 2141–2151 (2012)
18. Stent, S., et al.: Precise deterministic change detection for smooth surfaces. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, pp. 1–9. (2016)
19. Gharbia, A.Y.A., et al.: Registration-based change detection for sar images. NRIAG J Astron. Geophy. 9(1), 106–115 (2020)
20. Zheng, W, Wang, K, Wang, F.Y.: A novel background subtraction algorithm based on parallel vision and bayesian gans. Neurocomputing. 394 (2019)
21. Tezcan, O., et al.: A fully-convolutional neural network for background subtraction of unseen videos. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2774–2783. (2020)
22. Isik, S., et al.: Swcd: a sliding window and self-regulated learning-based background updating method for change detection in videos. J. Electron. Imag. 27(2), 1 (2018)
23. Lee, S.-h., et al.: Wisenetmd: motion detection using dynamic background region analysis. Symmetry. 11(5), 621 (2019). https://doi.org/10.3390/sym11050621. https://www.mdpi.com/2073-8994/11/5/621
24. Gong, M., et al.: Change detection in synthetic aperture radar images based on deep neural networks. IEEE Transact. Neural Networks Learn. Syst. 27(1), 125–138 (2017)
25. Sakurada, K., Okatani, T.: Change detection from a street image pair using cnn features and superpixel segmentation. BMVC. 61, 1–12 (2015)
26. Jing, R., et al.: Object-based change detection for vhr remote sensing images based on a trisiamese-lstm. Int. J. Rem. Sens. 41(16), 6209–6231 (2020)
27. Zhang, X., et al.: Change detection in very high-resolution images based on ensemble cnns. Int. J. Rem. Sens. 41(12), 4757–4779 (2020)
28. Huang, R., et al.: Change detection with cross enhancement of high-and low-level change-related features. IET Image Process. 15(13), 3380–3391 (2021)
29. Huang, R., Xing, Y., Zou, Y.: Triple-complementary network for rgb-d salient object detection. IEEE Signal Process. Lett. 27, 775–779 (2020)
30. Huang, R., Xing, Y., Wang, Z.: Rgb-d salient object detection by a cnn with multiple layers fusion. IEEE Signal Process. Lett. 26(4), 552–556 (2019)
31. Cong, R., et al.: An iterative co-saliency framework for rgbd images. IEEE Trans. Cybern. 49(1), 233–246 (2017)
32. Huang, R., et al.: Exemplar-based image saliency and co-saliency detection. Neurocomputing. 371, 147–157 (2020)
33. Oh, S.W., et al.: Video object segmentation using space-time memory networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9226–9235. (2019)
34. Cheng, H.K., Tai, Y.W., Tang, C.K.: Rethinking space-time networks with improved memory coverage for efficient video object segmentation. arXiv preprint arXiv:2106.05210 (2021)
35. Piao, Y, et al.: A2dele: Adaptive and attentive depth distiller for efficient rgb-d salient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9060–9069. (2020)
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer, pp. 234–241. (2015)
38. Fu, J., et al.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3146–3154. (2019)
39. Huang, Z., et al.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 603–612. (2019)
40. Liu, W., et al.: High-level semantic feature detection: A new perspective for pedestrian detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5187–5196. (2019)
41. Paszke, A., et al.: Automatic differentiation in pytorch (2017)
42. Chen, Y., Ouyang, X., Agam, G.: Mfcnet: End-to-end approach for change detection in images. In: 2018 25th IEEE International Conference on Image Processing (ICIP), IEEE, pp. 4008–4012. (2018)
43. Chen, C., Zhang, S., Du, C.: Learning to detect instantaneous changes with retrospective convolution and static sample synthesis. arXiv preprint arXiv:1811.08138 (2018)
44. Santana, M.C.S., et al.: A novel siamese-based approach for scene change detection with applications to obstructed routes in hazardous environments. IEEE Intell. Syst. 35(1), 44–53 (2020). https://doi.org/10.1109/MIS.2019.2949984
45. Han, X., et al.: Matchnet: Unifying feature and metric learning for patch-based matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3279–3286. (2015)
46. Goyette, N., et al.: Changedetection. net: A new change detection benchmark dataset. In: 2012 IEEE computer society conference on computer vision and pattern recognition workshops, IEEE, pp. 1–8. (2012)

How to cite this article: Huang, R., et al.: Selecting change image for efficient change detection. IET Signal Process. 16(3), 327–339 (2022). https://doi.org/10.1049/sil2.12095