Edge-Guided Parallel Network for VHR Remote Sensing Image Change Detection

Ye Zhu, Kaikai Lv, Yang Yu, and Wenjia Xu

***Abstract*—Change detection (CD) is an important research topic in the remote sensing field, and it has a wide range of applications, including resource monitoring, disaster assessment, urban planning, etc. Recently, deep learning (DL) has shown its advantages in CD. However, most existing DL-based methods cannot capture the complementary information between bitemporal and difference features. This article proposes an edge-guided parallel network (EGPNet) to solve this problem. First, our EGPNet extracts bitemporal and difference features simultaneously through a parallel encoding framework. During parallel encoding, we design a supplementary mechanism to enrich the difference features with bitemporal features. Second, we fuse bitemporal and difference features at each feature level to sufficiently exploit their complementarity. Finally, the edge-aware module and edge-guidance feature module are introduced to enhance the edge representation for improving blurred edges of detection results. Benefiting from the rich change-related information in difference features and detailed information in bitemporal features, our EGPNet can detect change regions entirely and accurately. Experimental results on the LEVIR-CD, SYSU-CD, and CDD datasets demonstrate that the proposed method outperforms several state-of-the-art approaches. Especially, our EGPNet can detect more precise and sharper edges than other methods.**

***Index Terms*—Change detection (CD), convolutional neural networks (CNNs), difference features, edge-guided network, remote sensing, two-stream architecture.**

I. Introduction

GIVEN a pair of coregistered images of the same region in different time phases, change detection (CD) is to identify where the changes have occurred. The change regions are assigned positive labels, whereas the unchanged regions are assigned negative labels. It is essentially a binary semantic segmentation task. Remote sensing satellite imaging technology advances by leaps and bounds, many remote sensing platforms, such as QuickBird, GeoEye, Worldview, and unmanned aerial vehicles (UAVs), can provide very high resolution (VHR) images [1]. These VHR images can capture detailed ground information, making it possible to observe our Earth from a closer perspective. CD, as one of the most important applications of remote sensing image interpretation, can be used in many fields, such as resource monitoring [2], disaster assessment [3], and urban planning [4]. According to the image analysis unit, traditional CD methods can be divided into two categories. The pixel-based CD (PBCD) method takes an image pixel as the fundamental unit of analysis [5], [6], [7]. The object-based CD (OBCD) method takes an image object as the fundamental unit of analysis to explore spatial context, texture, and shape information [8], [9], [10]. Although these methods need fewer samples for training and have strong interpretability, their accuracy is not very satisfactory due to the complexity of CD.

Recently, deep learning (DL) has dominated CD methods. Among them, some methods utilizing both bitemporal and difference features achieve better performance. Lei et al. [11] take difference features as input of the channel attention module to obtain attention weight for bitemporal features. Peng et al. [12] introduce a DE module to combine difference features in input space with the final features after decoding. Zhang et al. [13] fuse difference and bitemporal features based on the attention mechanism in a difference discrimination network. These works indicate the complementarity between bitemporal and difference features. Difference features can reflect changes explicitly, but they lack details. Even if bitemporal features contain detailed information, they cannot reflect change explicitly. Both bitemporal and difference features are equally crucial to CD. Capturing the complementary information between bitemporal and difference features is essential for CD. However, the difference features used by these methods are generated from the construction, not extraction. In other words, these methods lack an explicit extraction process of difference features. Following the spirit of the two-stream architecture [14], we propose a parallel encoding framework in our edge-guided parallel network (EGPNet) to extract these two kinds of features simultaneously and explicitly. To sufficiently leverage their complementarity, we fuse these two kinds of features at each feature level generating the fused features. Benefiting from the rich change-related information in difference features and detailed information in bitemporal features, our EGPNet can detect change regions entirely and accurately.

Besides, many state-of-the-art (SOTA) CD methods still suffer from blurred edges. As shown in Fig. 1, error detections are mainly focused on the edges. To solve this problem, we

Manuscript received 16 May 2023; revised 15 July 2023; accepted 11 August 2023. Date of publication 17 August 2023; date of current version 31 August 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62102129, Grant 61806071, and Grant 62276088, in part by the Natural Science Foundation of Hebei Province under Grant F2021202030, in part by the Natural Resources Science and Technology Plan Project of Hebei Province under Grant 454-0601-YBN-IBBM. *(Corresponding author: Yang Yu.)*

Ye Zhu, Kaikai Lv, and Yang Yu are with the School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China (e-mail: zhuye@hebut.edu.cn; 563167677@qq.com; yuyang@hebut.edu.cn).

Wenjia Xu is with the Data Center, Hebei Prospecting Institute of Hydrogeology and Engineering Geological (Hebei Remote Sensing Center), Shijiazhuang 050021, China (e-mail: 562153204@qq.com).

We open source the PyTorch implementation and the pretrained model at https://github.com/Lvkyky/EGPNet.

Digital Object Identifier 10.1109/JSTARS.2023.3306274

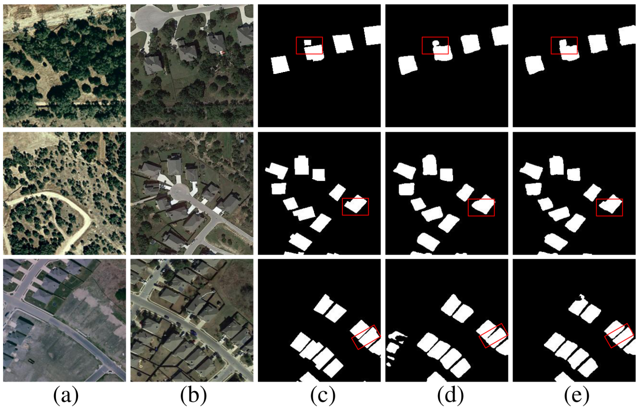

Fig. 1. Examples of edge blur. (a) T1 image. (b) T2 image. (c) Ground truth. (d) Detection results by BIT [15]. (e) Detection results by our EGPNet.

introduce an edge-aware module (EAM) and an edge-guidance feature module (EFM) [16]. EAM integrates the low-level local detailed information and high-level global location information to explore edge semantics under direct edge supervision. EFM can guide representation learning, enhancing edge representation. The contribution of our work can be summarized as follows.

1) A parallel encoding framework that can effectively extract bitemporal and difference features is proposed to explore feature complementarity.
2) We design a supplementary mechanism (SM), which can bridge the bitemporal encoder and difference encoder enriching the difference features with bitemporal features.
3) We introduce EAM and EFM to solve the edge blur problem of VHR remote sensing image CD.

The rest of this article is organized as follows. Section II reviews the related works. Section III gives the details of our proposed method. Experimental results and analysis are presented in Section IV, and finally, Section VI concludes this article.

II. Related Work

A. DL-Based CD

CD research has been largely driven by advances in semantic segmentation technology, which were often adapted to cope with the CD. A large family of CD methods is based on bitemporal features [17], [18], [19], [20], [21], [22]. They extract bitemporal features via the Siamese network, and the concatenation of bitemporal features is fed into the decoder to identify the changes. Daudt et al. [17] propose an FC-Siam-conc that adapts Unet [23] with Siamese architecture [24] for CD. Chen et al. [18] embed a nonlocal attention module into Siamese Unet to increase the detection capability of the model as well as the noise suppression capability. Fang et al. [19] adapt Unet++ [25] with Siamese architecture to retain shallow-layer information. Chen et al. [20] use two types of modality-independent structural relationships to solve the modal heterogeneity problem in unsupervised multimodal CD. Liu et al. [21] propose a multitask Siamese convolutional network combining the semantic information of the single bitemporal image. Chen et al. [22] use the semantic information of the single bitemporal image in a self-supervised learning framework to learn more discriminative features.

Other methods pay attention to difference features [15], [26], [27], [28], [29]. First, they extract bitemporal features through the Siamese network. Then, difference features are constructed by bitemporal features using recurrent neural network (RNN) or subtraction operation. Finally, difference features are used to recognize the changes. Chen et al. [26] integrate the merits of both CNN and RNN, CNN is used to extract bitemporal features, and RNN is used to generate difference features. Zhang et al. [27] design a differential pyramid to extract multilevel difference features explicitly, and then, the difference features are fed into Unet++ for further representation learning. Chen et al. [15] use a transformer to model space–time context in the token-based space to reduce pseudochange. Bandara et al. [28] adapt SegFormer [30] with Siamese architecture achieving higher performance than many models employing very large ConvNets. Chen et al. [29] design a structural relationship analysis framework in the Fourier domain to solve the modal heterogeneity problem of unsupervised multimodal CD. Some methods use the channel concatenation image as the initial input to extract difference features, Liu et al. [31] use depthwise separable convolution to extract difference features efficiently. Peng et al. [32] feed the channel concatenation of bitemporal images into Unet++ to extract difference feature for CD. In addition, metric-learning-based CD methods calculate the Euclidean distance pixelwise to generate the distance map [33], [34], [35]. They are also based on difference features. This article focuses on exploring the complementarity between bitemporal features and difference features.

B. Two-Stream Architecture

For a specific task, there is more than one type of information, which is usually heterogeneous and complementary. Thus, two-stream architecture is a natural choice for neural network design. For video action recognition, Simonyan et al. [14] propose a two-stream network composed of a spatial and a temporal network to integrate appearance and motion information. Zhou et al. [36] adopt faster R-CNN within a two-stream network for image manipulation detection. The RGB stream is to find tampering artifacts like substantial contrast differences and unnatural tampered boundaries. The noise stream detects the noise inconsistency between authentic and tampered regions. Zhang et al. [37] propose asymmetric two-stream architecture combining RGB information and depth information for saliency detection. Following the two-stream spirits, we design a novel parallel encoding framework to combine bitemporal and difference information for CD.

C. Edge-Guided Network

Edge cues are instrumental in many computer vision tasks, such as salient object detection [38], [39], [40], medical image segmentation [41], [42], etc. Usually, there is a subnetwork for edge detection. Edges generated through a Sobel operator or

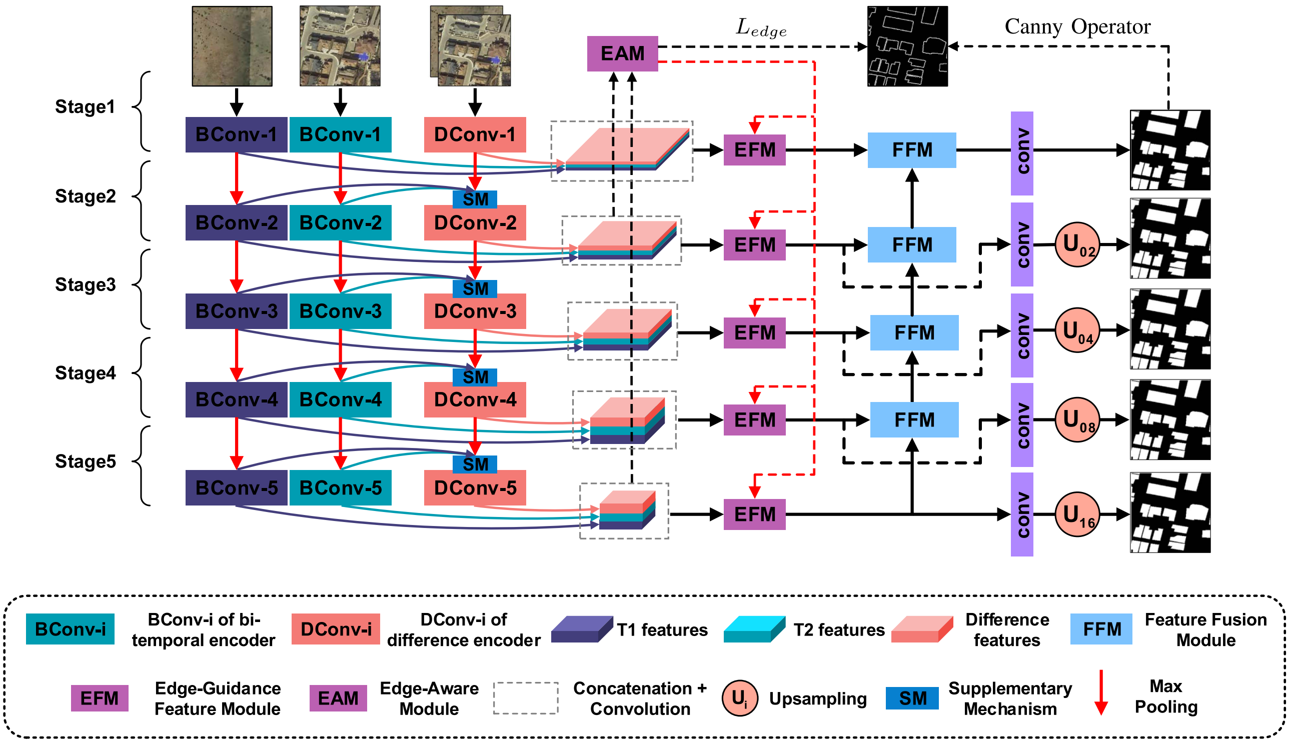

Fig. 2. Overall architecture of the proposed EGPNet, which is an encoder–decoder structure based on Unet. The encoder employs a *parallel encoding framework*, where BConv-i is the ith convolution block of the bitemporal encoder and DConv-i is the ith convolution block of the difference encoder. SM is used to enrich the difference feature flow with bitemporal feature flow. EAM integrates features at level 2 and level 5 to generate the edge map, which is injected into multilevel fused features through EFM for guiding their representation learning. The decoder employs FFM to combine low-level and high-level features progressively. In addition, we use 1×1 convolutions to produce change results at different feature levels and upsample them to 256×256 providing direct supervision for intermediate layers.

Canny operator will be involved in the calculation of edge loss. Then, the edge and no-edge features are integrated for the final detection. Thus, the network will be guided to pay more attention to edges making the edges more precise and sharper. However, research on edge cues is limited in the CD research community. Cheng et al. [43] adopt deformable convolution to achieve margin maximization clarifying the gap between changed and unchanged semantics. Bai et al. [44] propose an EGRCNN that incorporates both discriminative features and edge features to improve the edge quantity of CD results. Chen et al. [45] design an edge-guided transformer block for long-range context modeling and edge feature refinement. Xia et al. [46] propose an extra edge detection branch to guide change features with edge information. Different from EGRCNN [44] and EGDE-Net [45], which simply capture and fuse edge information at the end of the network, we introduce an EAM to explore edge semantics using selected features after encoding and design an EFM to inject edges into multilevel change features for guiding their representation learning.

III. Proposed Method

A. Overall Architecture

In Fig. 2, we illustrate the overall framework of our proposed EGPNet, encoder–decoder architecture. Different from conventional encoders, we propose a parallel encoder that is made up of a bitemporal encoder and a difference encoder. In the parallel encoder, we design an SM to enrich the difference feature flow using bitemporal feature flow. Next, the concatenation of bitemporal and difference features is fed into two convolution layers for sufficient semantic fusion to obtain the fused features. Then, we use EAM to generate edges that are injected into the fused features at each feature level through EFM for edge representation enhancement. During feature decoding, we progressively fuse different levels of feature maps and employ 1×1 convolution to map feature vectors to the desired number of classes. Five change results are produced at corresponding feature levels, and feature level 1 gives the best result.

B. Parallel Encoding Framework

1) Bitemporal Encoder: The bitemporal encoder adopts the Siamese network architecture as in [17] to obtain multilevel bitemporal features containing many details. Like the vanilla Unet [23], the bitemporal encoder includes five convolution stages. Each stage comprises one convolution block and one pooling layer. As shown in Fig. 3, the convolution block consists of 3×3 convolution, batch normalization, and Relu activation function. The first convolution layer is used to double the number of channels and the 2×2 max pooling layer to reduce the size of feature maps. These convolution blocks can be abbreviated as $\mathrm{BConv}-i$, where $i \in \{1, 2, 3, 4, 5\}$. In this

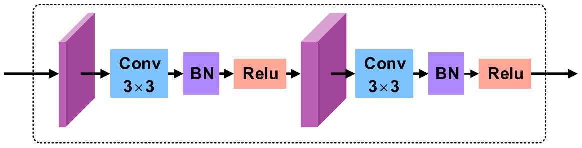

Fig. 3. Illustration of the convolution block.

TABLE I
DETAILS OF EACH CONVOLUTION BLOCK

BConv-i/DConv-i	kernel size	stride	padding	channel
1	3×3 3×3	1×1 1×1	1 1	32 32
2	3×3 3×3	1×1 1×1	1 1	64 64
3	3×3 3×3	1×1 1×1	1 1	128 128
4	3×3 3×3	1×1 1×1	1 1	256 256
5	3×3 3×3	1×1 1×1	1 1	512 512

article, the initial channel number is set to 32; Table I gives details of these convolution blocks. Given bitemporal images $I_1 \in \mathbb{R}^{C\times H\times W}, I_2 \in \mathbb{R}^{C\times H\times W}$, passing through the five convolution stages, respectively, we obtain the multilevel bitemporal features at the corresponding stages. We denote the five stages of bitemporal feature maps as f_{b1}^i, f_{b2}^i, where $i \in \{1, 2, 3, 4, 5\}$. The numerical superscript indicates the feature level, and the subscript represents the time phase.

2) Difference Encoder: Difference information is essential for CD because we can identify the changes directly from the difference information. Our idea is that difference features from extraction are superior to difference features from construction. Instead of using RNN or subtraction, we design an independent difference encoder for the representation learning of difference information. The difference encoder also includes five convolution stages consistent with the bitemporal encoder. We abbreviate these convolution blocks as $\mathrm{DConv} - i$, where $i \in \{1, 2, 3, 4, 5\}$, details are given in Table I. There is no direct input for the difference encoder. We stack I_1 and I_2 in the channel dimension producing $I_D \in \mathbb{R}^{2\,C\times H\times W}$, which can implicitly represent the difference information of input space. Then, we feed I_D into the difference encoder to extract multilevel difference features. These multilevel difference features are denoted as f_d^i, where $i \in \{1, 2, 3, 4, 5\}$. The numerical superscript indicates the feature level, and the subscript means the difference.

C. Supplementary Mechanism

It is considered that the semantic information contained in I_D is limited. It may be insufficient to consider only I_D. To extract semantic-rich difference features, we aim to enrich the flow of difference features with the flow of bitemporal features. As shown in Fig. 4, we design an SM that can construct difference features by bitemporal image features. Then, the constructed difference features are used to supplement the flow of difference features at each stage. This process can be formulated as

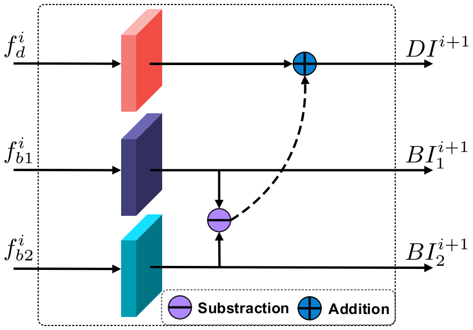

Fig. 4. Illustration of the SM. Where f_d^i represents the output of the difference encoder at the ith stage. f_{b1}^i and f_{b2}^i are the output of the bitemporal encoder at the ith stage. DI^{i+1} represents the input of the difference encoder at the $i+1$th stage. BI_1^{i+1} and BI_2^{i+1} are the input of the bitemporal encoder at the $i+1$th stage.

$$\mathrm{DI}^{i+1} = f_d^i + \left|f_{b1}^i - f_{b2}^i\right| \tag{1}$$

where DI^{i+1} represents the input of the difference encoder at the $i+1$th stage. f_d^i is the output of the difference encoder at the ith stage. f_{b1}^i and f_{b2}^i are the output of the bitemporal encoder at the ith stage. $||$ is absolute value operator. We construct difference features through subtraction and supplement original difference features using addition.

D. Bitemporal Difference Feature Fusion

To utilize change-related information in difference features and detailed information in bitemporal features. First, we directly concatenate bitemporal features and difference features in the channel dimension. Then, the concatenation of bitemporal and difference features is fed into two convolution layers for sufficient semantic fusion producing more powerful features. The fused features can locate the changes, especially their details, accurately. The fused features are denoted as f_f^i, where $i \in \{1, 2, 3, 4, 5\}$

$$f_f^i = \mathrm{F}_{\mathrm{conv3}}(\mathrm{F}_{\mathrm{conv3}}(\mathrm{concat}(f_{b1}^i, f_{b2}^i, f_d^i))) \tag{2}$$

where $\mathrm{F}_{\mathrm{conv3}}$ denotes the 3×3 convolution layer and concat represents feature channel concatenation.

E. Edge-Aware Module

The lack of using prior edge structure information leads to inaccurate detection results in the areas of building edges [44]. As shown in Fig. 5, we introduce an EAM to explore edge semantics under direct edge supervision. Low-level features contain rich edges but many nonchange-related edges. Thus, high-level features are needed to help locate change-related edges. f_f^2 and f_f^5 are selected to explore edge semantics. First, 1×1 convolution reduces the proportion of high-level features and upsampling to align spatial resolution. Then, the concatenation of f_f^2 and f_f^5 are passed through two 3×3 convolution layers to integrate semantic information further. Finally, 1×1

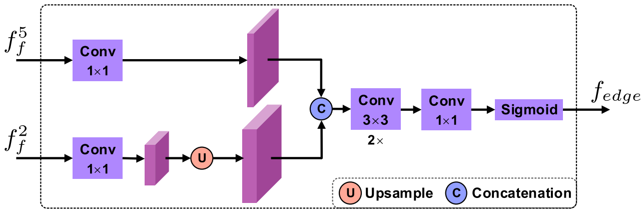

Fig. 5. Illustration of the EAM.

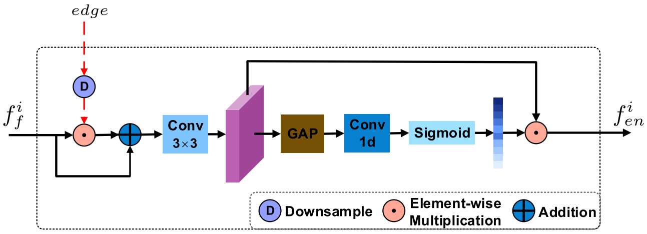

Fig. 6. Illustration of the EFM.

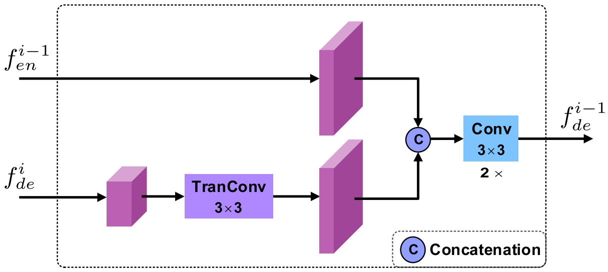

Fig. 7. Illustration of the decoding stage.

convolution followed by the Sigmoid function is used to produce the edge map denoted as f_{edge}.

F. Edge-Guidance Feature Module

EFM injects the edge map f_{edge} produced by EAM into the fused features f_f^i, which can guide the representation learning enhancing edge representation. As shown in Fig. 6, given the fused features f_f^i, where $i \in \{1,2,3,4,5\}$ and the edge map f_{edge}. First, we perform the elementwise multiplication between the downsampled edge map and the fused features at corresponding feature levels. Then, residual connection and 3×3 convolution layer are used for feature fusion. In this way, we can obtain the updated fused features f_{up}^i whose edges are enhanced

$$f_{\text{up}}^i = \text{F}_{\text{conv3}}\left(f_f^i \odot \text{D}(f_{\text{edge}}) \oplus f_f^i\right) \tag{3}$$

where D denotes downsampling. $\odot$ is elementwise multiplication and $\oplus$ is addition. Finally, we apply an efficient channel attention (ECA) module [47] to achieve further feature representation enhancement. ECA can capture local cross-channel interaction using 1D convolution. The enhanced features f_{en}^i can be denoted as

$$f_{\text{en}}^i = \sigma\left(F_{\text{1D}}^k\left(\text{GAP}\left(f_{\text{up}}^i\right)\right)\right) \odot f_{\text{up}}^i \tag{4}$$

where GAP represents global average pooling. F_{1D}^k is 1-D convolution whose kernel size is k. σ is the Sigmoid function. As described in [47], the kernel size k can be selected adaptively.

G. Progressive Feature Decoding

As shown in Fig. 7, in the top-down feature fusion module (FFM), we fuse deep-layer features with shallow-layer features progressively producing the final features f_{de}^i at different feature levels. Transposed convolution is used to align the channel number and the spatial resolution of feature maps. FFM can be formulated as

$$f_{\text{de}}^{i-1} = \text{F}_{\text{conv3}}\left(\text{F}_{\text{conv3}}\left(\text{concat}\left(f_{\text{en}}^{i-1}, \text{F}_{\text{trans}}\left(f_{\text{de}}^i\right)\right)\right)\right) \tag{5}$$

where F_{trans} refer to 3×3 transposed convolution layer. Concat denotes feature channel concatenation.

H. Loss Function

1) Change Supervision: For the CD task, the distribution of difficult and easy samples is unbalanced due to the influence of shadows, light, and seasonal changes [27]. Thus, the focal loss (FL) [48] that can focus on hard examples is adopted for change supervision.

$$\text{FL} = \begin{cases} -(1-p_t)^\gamma \log p_t, & y = 1 \\ -p_t^\gamma \log(1-p_t), & y = 0 \end{cases} \tag{6}$$

where p_t denotes the change probability. When $\gamma > 0$, the relative loss for easy examples will be reduced by paying more attention to hard examples. In this article, γ is set to 1. Besides, we adopt the deep supervision strategy to deal with the gradient vanishing problem, learning more discriminative features. As shown in Fig. 2, we upsample the intermediate features to 256×256 and employ 1×1 convolutions to produce CD results p_i, where $i \in \{1,2,3,4,5\}$ at each feature level. The change loss L_{change} can be defined as

$$L_{\text{change}} = \text{FL}(p_1, g) + \frac{1}{4}\sum_{i=2}^{5}\text{FL}(p_i, g) \tag{7}$$

where FL denotes focal loss. g is the change ground truth and p_i is the detection result at feature level i.

2) Change Edge Supervision: EAM produces an edge map f_{edge}. We employ the dice loss [49], which can solve the strong class imbalance problem for its supervision.

$$L_{\text{edge}} = \frac{2\sum_{x,y}\left(f_{\text{edge}}^{(x,y)} \times g_{\text{edge}}^{(x,y)}\right)}{\sum_{x,y}\left(\left(f_{\text{edge}}^{(x,y)}\right)^2 + (g_{\text{edge}}^{(x,y)})^2\right)} \tag{8}$$

where g_{edge} represents the ground truth of the edge, it is extracted from the change ground truth g through the Canny operator [50]. (x, y) indexes different pixels in f_{edge} or g_{edge}.

3) Overall Loss: Finally, the overall loss L_{total} is denoted as

$$L_{\text{total}} = \lambda L_{\text{edge}} + L_{\text{change}} \tag{9}$$

where L_{edge} is the edge loss and L_{change} is the change loss. λ control the contribution of L_{edge} in total loss. λ is set to 0.1 in this article.

IV. Experiment and Analysis

In this section, first, we will give the experimental setup, including the dataset, evaluation metrics, and implementation details. Next, we conduct comparative experiments to validate the performance of the proposed method on the LEVIR-CD [33], SYSU-CD, and CDD [34] datasets. Then, we design ablation experiments to validate the effectiveness of each part in our EGPNet. Finally, the network visualization is presented to understand our EGPNet intuitively.

A. Dataset

1) LEVIR-CD Dataset: LEVIR-CD [33] is a public CD dataset released by the Beijing University of Aeronautics and Astronautics. The changes are mainly about construction growth. It contains 637 VHR image pairs collected from Google Earth (GE). Its spatial resolution is 0.5 m per pixel, and the image size is 1024 × 1024. Due to the GPU memory limitation, these images are cropped into smaller image patches whose size is 256 × 256 following its original dataset split. Consequently, we can obtain 7120 pairs of image patches for training, 1024 for validation, and 2048 for testing, respectively.

2) SYSU-CD Dataset: SYSU-CD [34] is a challenging CD dataset released by Sun Yat-sen University. It covers many change types (e.g., suburban dilation, road expansion, and sea construction). It contains 20 000 pairs of labeled remote sensing images collected between 2007 and 2014 in Hong Kong. The size of each image is 256 × 256, and the spatial resolution is 0.5 m per pixel. There are 12 000 pairs of images for training, 4000 for validation, and 4000 for testing.

3) CDD Dataset: CDD [51] is a public CD dataset whose images are collected from GE. It contains 16 000 pairs of remote sensing images obtained from the same region in different seasons. It covers change objects of different sizes (e.g., cars, single trees, big constructions, and forest areas). The resolution of CDD is from 3 to 1 m per pixel, and the image size is 256 × 256. There are 10 000 pairs of images for training, 3000 for validation, and 3000 for testing.

B. Evaluation Metrics

To evaluate the performance of the proposed method, Precision, Recall, F1 score, intersection-of-union (IOU), and overall accuracy (OA), which are often used in binary classification tasks, are introduced. They are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{11}$$

$$\text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{12}$$

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

$$\text{OA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{14}$$

where TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives, and FN is the number of false negatives.

C. Implementation Details

We implement our EGPNet using the PyTorch DL library. We conduct all the experiments on a single NVIDIA GeForce RTX 3060 GPU. We initialize our EGPNet by KaiMing normalization [52]. During model training, we employ the Adam optimizer [53] for faster convergence. The initial learning rate is set to 1e^{-4}, and the linear decay strategy is adopted to adjust the learning rate. Due to GPU memory limitation, the batch size is set to 8, and the total epoch is set to 100 for both LEVIR-CD and SYSU-CD datasets. For the CDD dataset, the total epoch is set to 140.

D. Comparative Experiments

1) Comparative Methods: Several SOTA methods are selected for comparison, all implemented using code published by the original authors. These include FC-Siam-conc [17], FC-Siam-diff [17], IFNet [13], SNUNet [19], BIT [15], ISNet [43], ChangeFormer [28], EGRCNN [44], and EGCTNet [46]. FC-Siam-conc [17] is based on Unet [23] and concatenates the bitemporal features for skip connection at each layer. FC-Siam-diff [17] is also based on Unet and uses subtraction to construct difference features for skip connection. IFN [13] integrates difference and bitemporal features through the attention mechanism in a difference discrimination network. SNUNet [19] can retain fine low-level information through the dense connection between the encoder and decoder. BIT [15] uses a transformer to model the space–time context in a token-based space. ISNet [43] employs deformable convolution to achieve margin maximization. ChangeFormer [28] adapts SegFormer [30] with Siamese architecture to extract multilevel features with long-range dependency. EGRCNN [44] introduces a DAM to produce more discriminative features and a multilevel edge detection header to capture edge semantic information. EGCTNet [46] proposes an additional edge detection branch to improve edge accuracy.

2) Experiments on the LEVIR-CD Dataset: Table II reports the quantitative comparison results on the LEVIR-CD dataset. Our EGPNet outperforms other methods in terms of F1, IOU, and OA. The F1 score improves by 0.56% compared to the suboptimal method (ChangeFormer). The F1_{edge} score is not optimal because our edge guidance strategy focuses on how to use edges to guide the fused features, not on the edges themselves.

The results of the visual comparison are shown in Fig. 8. First, by combing the detailed information in the bitemporal features, our EGPNet can detect relatively intact change regions [e.g., Fig. 8 (1)] and small objects missed by other methods [e.g., Fig. 8 (3)]. Second, the difference encoder can explicitly capture difference information, which is useful for identifying

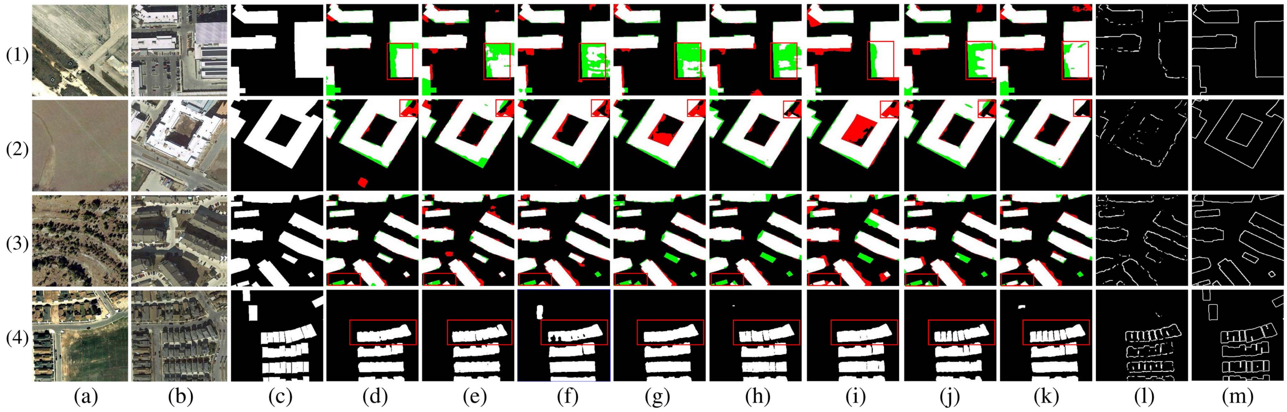

Fig. 8. Visual comparison with seven CD methods on the LEVIR-CD dataset. (a) T1 image. (b) T2 image. (c) GT. (d) IFN [34]. (e) SNUNet [19]. (f) BIT [15]. (g) ISNet [43]. (h) ChangeFormer [28]. (i) EGRCNN [44]. (j) EGCTNet [46]. (k) Ours (l) edges generated by EAM. (m) GT of edges. Colors: white for true positive, black for true negative, red for false positive, and green for false negative.

TABLE II
QUANTITATIVE COMPARISON RESULTS ON THE LEVIR-CD DATASET

Method	Precision	Recall	F1	F1$_{edge}$	IOU	OA
FC-Siam-conc	69.92	**94.70**	80.45	-	67.29	97.65
FC-Siam-diff	71.57	92.80	80.81	-	67.80	97.76
IFNet	**93.16**	87.45	90.22	-	82.18	99.03
SNUNet	91.03	89.45	90.23	-	82.20	99.01
BIT*	89.24	89.37	89.31	-	80.68	98.92
ISNet*	92.46	88.27	90.32	-	82.35	99.04
ChangeFormer*	92.05	88.80	90.40	-	82.48	99.04
EGRCNN	87.76	92.06	89.86	**41.80**	81.59	98.94
EGCTNet	90.57	88.88	89.72	40.10	81.36	98.96
EGPNet (ours)	92.03	89.93	**90.96**	40.66	**83.43**	**99.09**

* presents the results reported in the original article.
The bold values mean the best performance.

TABLE III
QUANTITATIVE COMPARISON RESULTS ON THE SYSU-CD DATASET

Method	Precision	Recall	F1	F1$_{edge}$	IOU	OA
FC-Siam-conc	81.15	70.74	75.59	-	60.76	89.22
FC-Siam-diff	**91.59**	49.79	64.51	-	47.61	87.08
IFNet	79.39	**79.30**	79.35	-	65.76	90.26
SNUNet	79.63	75.19	77.34	-	63.06	89.61
BIT	76.02	78.13	77.06	-	62.68	89.03
ISNet*	80.27	76.41	78.29	-	64.44	90.01
ChangeFormer	77.15	73.74	75.41	-	60.52	88.66
EGRCNN	77.46	81.39	79.37	**9.93**	65.80	90.03
EGCTNet	81.71	74.58	77.98	8.88	63.91	90.07
EGPNet (ours)	83.97	79.15	**81.49**	9.04	**68.76**	**91.52**

* presents the results reported in the original article.
The bold values mean the best performance.

pseudochanges caused by confounding factors [e.g., Fig. 8 (2)]. Finally, Fig. 8 (4) shows the great advantages of our model in detecting accurate edges. Other methods identify the new buildings as a whole change region, failing to detect the small gaps. Our EGPNet can detect the edges of each building, providing more detail about the change regions. The EGCTNet can also detect small gaps, but they are not as accurate as ours. This is because the edges generated by EAM are accurate enough and can guide the representation learning of the fused features. To display more types of changes, visual results of 1024 × 1024 images are given in Fig. 9.

3) Experiments on the SYSU-CD Dataset: Table III reports the quantitative comparison results on the SYSU-CD dataset. Our EGPNet outperforms other methods in terms of F1, IOU, and OA. In particular, the F1 score improves by 2.12% compared to the suboptimal method (EGRCNN) on this more challenging dataset, which indicates the robustness of our model. Our model can also perform well even when applied to complex change scenes.

TABLE IV
QUANTITATIVE COMPARISON RESULTS ON THE CDD DATASET

Method	Precision	Recall	F1	F1$_{edge}$	IOU	OA
FC-Siam-conc	89.66	56.10	69.02	-	52.69	93.89
FC-Siam-diff	88.44	54.25	67.25	-	50.66	93.59
IFNet	**97.60**	88.75	92.97	-	86.86	98.37
SNUNet	96.56	95.79	96.17	-	92.63	99.08
BIT	95.51	94.88	95.19	-	90.83	98.84
ISNet	95.15	93.48	94.31	-	89.23	98.63
ChangeFormer	95.45	95.40	95.42	-	91.25	98.89
EGRCNN	85.05	**97.52**	90.86	22.60	83.25	97.62
EGCTNet	94.11	91.95	93.02	20.11	86.95	98.33
EGPNet(ours)	97.10	96.37	**96.73**	**25.35**	**93.67**	**99.21**

* presents the results reported in the original article.
The bold values mean the best performance.

Fig. 10 shows the visual comparison results on the SYSU-CD dataset. It can be seen that the proposed method achieves satisfactory performance. First, our proposed method can detect accurate and sharp edges. The results detected by our EPGNet have few error detections around the edges [e.g., Fig. 10 (1) and (2)]. Second, our EPGNet is better at avoiding false detections [e.g., Fig. 10 (3) and (4)]. Taking Fig. 10 (4) as an example, other methods misidentify the motorway as a change region due to illumination interference. Our proposed method achieves better discrimination results because the difference encoder can efficiently extract the difference features associated with the changes of interest, eliminating interfering factors.

4) Experiments on the CDD Dataset: Table IV reports the quantitative comparison results on the CDD dataset. Our EGPNet outperforms other methods in terms of F1, IOU, OA, and F1$_{edge}$. The F1 score improves by 0.56% compared to the suboptimal method (SNUNet). Fig. 11 shows the visual comparison results on the CDD dataset. Our EGPNet can detect intricate change scenarios completely and accurately [e.g., Fig. 11 (1)

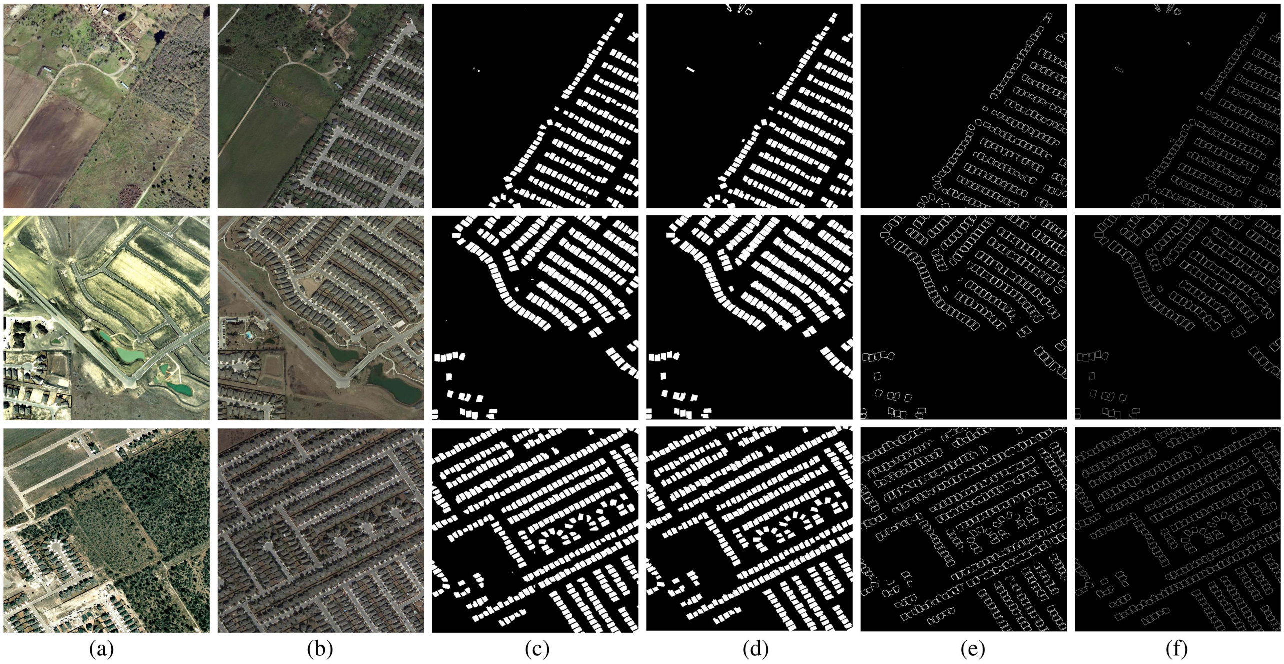

Fig. 9. Visual results of 1024 × 1024 images on the LEVIR-CD dataset. (a) T1 image. (b) T2 image. (c) EGPNet. (d) GT. (e) Edges generated by EAM. (f) GT of edges.

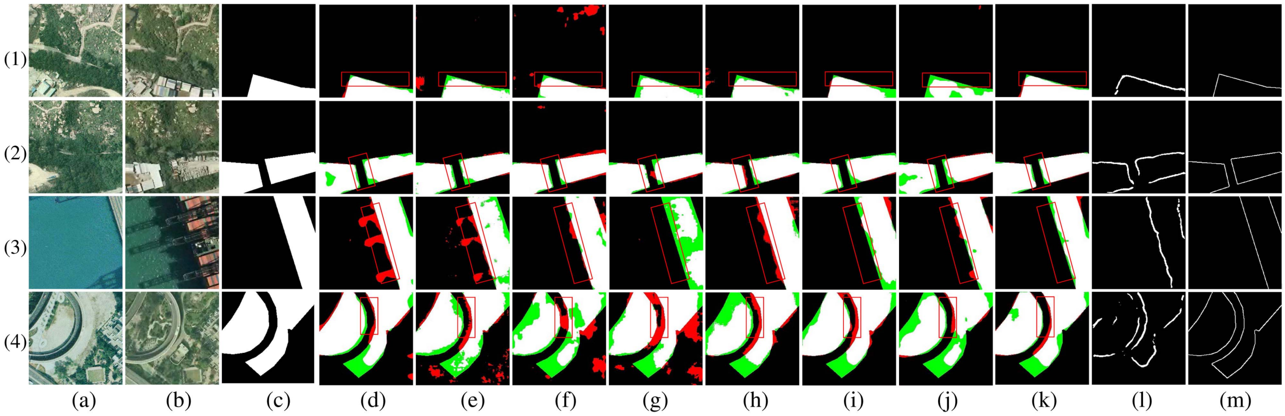

Fig. 10. Visual comparison with seven CD methods on the SYSU-CD dataset. (a) T1 image. (b) T2 image. (c) GT. (d) IFN [34]. (e) SNUNet [19]. (f) BIT [15]. (g) ISNet [43]. (h) ChangeFormer [28]. (i) EGRCNN [44]. (j) EGCTNet [46]. (k) Ours. (l) Edges generated by EAM. (m) GT of edges. Colors: White for true positive, black for true negative, red for false positive, and green for false negative.

and (4)] because the encoder can extract semantic-rich features with a parallel encoding framework. For change objects with regular shapes [e.g., Fig. 11 (2)], our method can restore the real shape of the objects accurately with the help of the edge guidance strategy. In particular, Fig. 11 (3) shows the great advantages of the proposed method in capturing small details.

E. Model Efficiency Analysis

For a comprehensive comparison with other SOTA methods, we implement our EGPNet with different model capacities (the initial number of channels is set to 8/16/24/32/40). We test all methods on a server equipped with an E5-1650 CPU and RTX 3060 GPU and report the number of parameters (Params), floating point operations per second (FLOPs), F1 score, and IOU score of different methods on the LEVIR-CD and SYSU-CD datasets. As shown in Table V, the F1 score of EGPNet-8 reaches 77.53% on the SYSU dataset, outperforming other methods that also use a light backbone (e.g., FC-Siam-conc, FC-Siam-diff, and BIT). EGPNet-16 has fewer parameters and lower computational complexity but achieves a higher F1 score (78.77%) on the SYSU dataset compared to (ISNet, SNUNet, ChangeFormer), demonstrating the efficiency of our proposed method. As the initial number of channels increases, EGPNet-32 achieves the optimal and best performance on both the LEVIR-CD and SYSU-CD datasets. The EGPNet-40 may suffer from the overfitting problem leading to a decrease in accuracy.

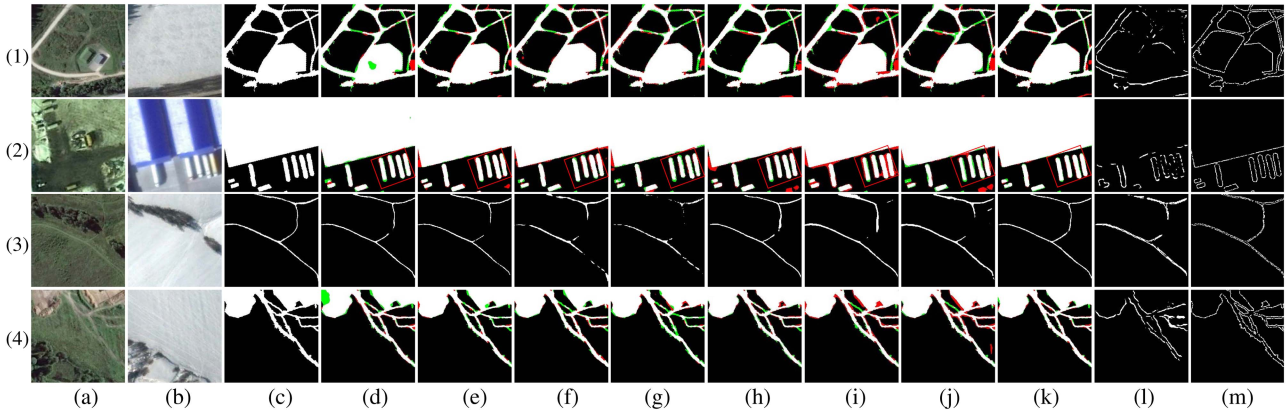

Fig. 11. Visual comparison with seven CD methods on the CDD dataset. (a) T1 image. (b) T2 image. (c) GT. (d) IFN [34]. (e) SNUNet [19]. (f) BIT [15]. (g) ISNet [43]. (h) ChangeFormer [28]. (i) EGRCNN [44]. (j) EGCTNet [46]. (k) Ours. (l) Edges generated by EAM. (m) GT of edges. Colors: White for true positive, black for true negative, red for false positive, and green for false negative.

TABLE V
MODEL EFFICIENCY COMPARISON RESULTS ON THE SYSU-CD DATASET AND THE LEVIR-CD DATASET

Method	Params (M)	FLOPS (G)	LEVIR-CD		SYSU-CD	
			F1	IOU	F1	IOU
FC-Siam-conc	1.55	5.33	80.45	67.29	75.59	60.76
FC-Siam-diff	1.35	4.73	80.81	67.80	64.51	47.61
IFNet	35.73	82.26	90.22	82.18	79.35	65.76
SNUNet	12.03	54.83	90.23	82.20	77.34	63.06
BIT	3.04	8.75	89.31	80.68	77.06	62.68
ISNet	34.55	21.61	90.32	82.35	78.29	64.44
ChangeFormer	41.03	202.79	90.40	82.48	75.41	60.52
EGRCNN	9.63	17.64	89.86	81.59	79.37	65.80
EGCTNet	106.13	38.47	89.72	81.63	77.98	63.91
EGPNet-8	2.78	4.92	88.27	79.00	77.53	63.30
EGPNet-16	11.09	19.45	89.82	81.52	78.77	64.98
EGPNet-24	24.93	43.59	89.84	81.55	78.76	64.97
EGPNet-32	44.32	77.33	**90.96**	**83.43**	**81.49**	**68.76**
EGPNet-40	69.24	120.69	90.63	82.86	81.21	68.37

The bold values mean the best performance.

TABLE VI
EFFECT OF DIFFERENT INPUT FOR DIFFERENCE ENCODER

Method	LEVIR-CD		SYSU-CD	
	F1	IOU	F1	IOU
DNet-dici	88.93	80.07	77.62	63.43
DNet-di	74.12	58.89	71.44	55.57
DNet-ci	**89.00**	**80.19**	**78.01**	**63.94**

The bold values mean the best performance.

F. Ablation Experiments

In this part, we perform extensive ablation studies on the LEVIR-CD and SYSU-CD datasets to validate the effectiveness of the parallel encoding framework, SM, and the edge guidance strategy. The following models are set for comparison.

1) *BNet:* Our base model using single bitemporal features.
2) *DNet-di:* Our base model using single difference features. It takes the differential image of the T1 and T2 images as input to the difference encoder.
3) *DNet-ci:* Our base model using single difference features. It takes the channel concatenation of the T1 and T2 images as input to the difference encoder.
4) *DNet-dici:* The combination of DNet-di and DNet-ci.
5) *ParalNet-di:* Our parallel model with an SM (the combination of BNet and DNet-di).
6) *ParalNet-ci:* Our parallel model with an SM (the combination of BNet and DNet-ci).
7) *EGPNet:* ParalNet-ci + edge guidance strategy.

1) Effect of Different Input for Difference Encoder: In order to find the optimal input to the difference encoder, we try different input forms. These are difference input (DI), concatenation input (CI), and "DICI." DI is generated by subtraction between bitemporal images. We stack bitemporal images in the channel dimension to generate CI. "DICI" is the concatenation of DI and CI in the channel dimension. Three models, DNet-di, DNet-ci, and DNet-dici, are set for comparison. As shown in Table VI, the experiments show that CI is the best choice for the input of the difference encoder. Images in the input space have much noise, and subtraction will pass the noise of the bitemporal images to differential image amplifying noise. Therefore, the differential image is inappropriate for the input of the difference encoder.

2) Effect of Different Strategies for SM: In order to find the optimal strategy for SM, using ParalNet-ci as the base model, we consider four different strategies, namely, none supplement (NS), concatenation supplement (CS), multiplication supplement (MS), and addition supplement (AS). As shown in Fig. 12, there is no interaction between the two feature flows in NS. In CS, we use concatenation to supplement the difference feature flow where a 1×1 convolution layer is used to adjust the number of channels to fit DI^{i+1}. In MS, we use elementwise multiplication to supplement the difference feature flow. In AS, we use addition to supplement the difference feature flow. As shown in Table VII, the experimental results show that AS strategy gives the best result.

3) Ablation on Parallel Encoding Framework: As shown in Table VIII, the parallel encoding framework brings consistent improvements in the F1 score when combing different difference encoder input forms on the two datasets. On the SYSU-CD dataset, improvement for the F1 score is significant. The combination of BNet and DNet-ci improves the F1 score by 1.8% compared to DNet-ci. On the LEVIR-CD dataset, the combination of

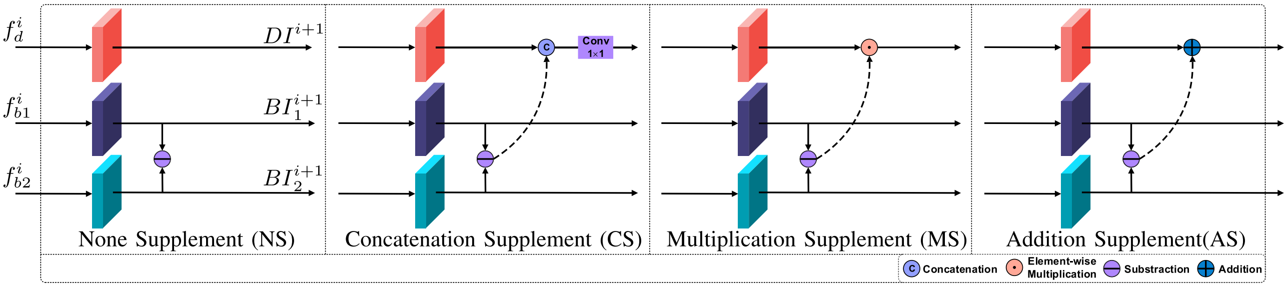

Fig. 12. Illustration of different strategies for SM where f_d^i represents the output of the difference encoder at the ith stage. f_{b1}^i and f_{b2}^i are the output of the bitemporal encoder at the ith stage. DI^{i+1} represents the input of the difference encoder at the $i+1$th stage. BI_1^{i+1} and BI_2^{i+1} are the input of the bitemporal encoder at the $i+1$th stage.

TABLE VII
EFFECT OF DIFFERENT STRATEGIES FOR SM

Method	LEVIR-CD		SYSU-CD	
	F1	IOU	F1	IOU
NS	90.17	82.10	79.75	66.32
MS	90.28	82.28	79.45	65.91
CS	90.28	82.28	79.28	65.67
AS	**90.34**	**82.37**	**79.81**	**66.40**

The bold values mean the best performance.

TABLE VIII
ABLATION ON PARALLEL ENCODING FRAMEWORK

Method	LEVIR-CD		SYSU-CD	
	F1	IOU	F1	IOU
BNet	90.24	82.22	77.32	63.03
DNet-ci	89.00	80.19	78.01	63.94
DNet-di	74.12	58.89	71.44	55.57
ParalNet-ci	90.34	82.37	**79.81**	**66.40**
ParalNet-di	**90.42**	**82.51**	78.85	65.08

The bold values mean the best performance.

TABLE IX
ABLATION ON EDGE GUIDANCE

Method	LEVIR-CD		SYSU-CD	
	F1	IOU	F1	IOU
ParalNet-ci	90.34	82.37	79.81	66.40
EGPNet	**90.96**	**83.43**	**81.49**	**68.76**

The bold values mean the best performance.

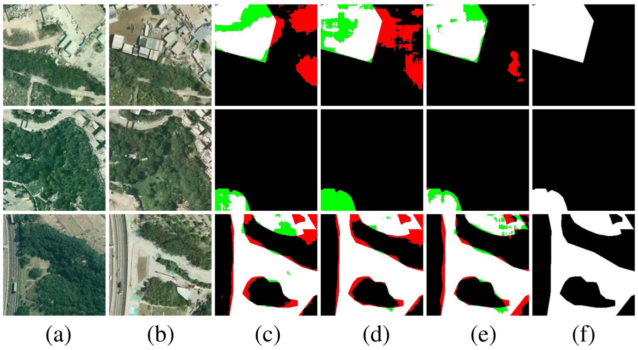

Fig. 13. Visual comparison for parallel encoding ablation on the SYSU-CD dataset. (a) T1 image. (b) T2 image. (c) BNet. (d) DNet-ci. (e) ParalNet-ci. (f) GT. Colors: White for true positive, black for true negative, red for false positive, and green for false negative.

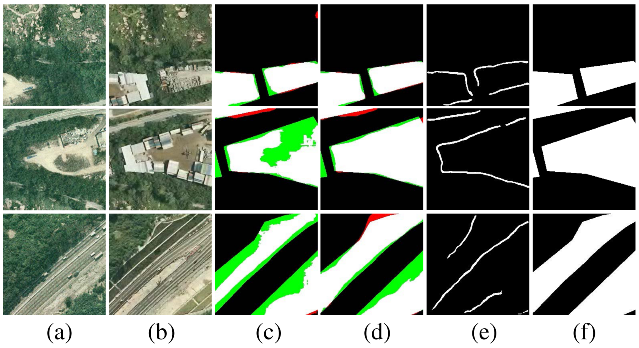

Fig. 14. Visual comparison for edge guidance ablation on the SYSU-CD dataset. (a) T1 image. (b) T2 image. (c) ParalNet-ci. (d) EGPNet. (e) Edges generated by EAM. (f) GT. Colors: White for true positive, black for true negative, red for false positive, and green for false negative.

BNet and DNet-di improves the F1 score by 0.18% compared to BNet. These results indicate the vital importance of the parallel encoding strategy, which can explore the complementary information between bitemporal and difference features. As shown in Fig. 13, models using single features have many false positives and false negatives [e.g., Fig. 13(c) and (d)]. Benefiting from the feature complementarity, ParalNet-ci can detect change regions entirely and accurately [e.g., Fig. 13(e)].

4) Ablation on Edge Guidance: Table IX shows consistent and significant improvements in F1 score on the LEVIR-CD and SYSU-CD datasets when EAM and EFM are added to ParalNet-ci. The F1 score improves by 0.62% and 1.68% on the two datasets, respectively. This indicates that the introduced edge guidance strategy can guide the representation learning of the fused features, leading to accurate edge detection results with low computational costs. As shown in Fig. 14, the edge guidance strategy can help correct the edge errors (see the first row). On the other hand, the edge guidance strategy can help locate the internal change regions, leading to more intact results (see the second and third rows). Besides, we embed EAM and EFM into

TABLE X
ABLATION ON EDGE GUIDANCE

Method	LEVIR-CD		SYSU-CD	
	F1	IOU	F1	IOU
FC-Siam-diff	80.45	67.29	64.51	47.61
FC-Siam-diff+edge	**84.27**	**72.82**	**74.42**	**59.26**

The bold values mean the best performance.

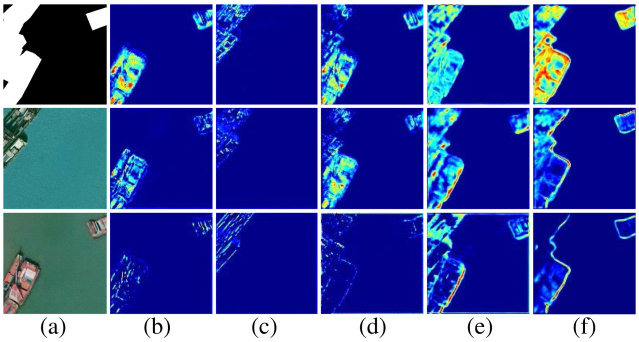

Fig. 15. Example of network visualization on the SYSU-CD dataset. (a) T1 image, T2 image, and the change ground truth. (b) T1 features. (c) T2 features. (d) Difference features. (e) Fused features. (f) Features after EFM.

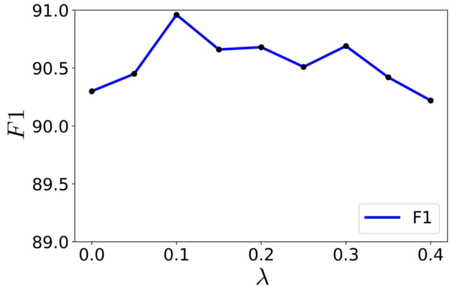

Fig. 16. Influence of the values of λ on the F1 score.

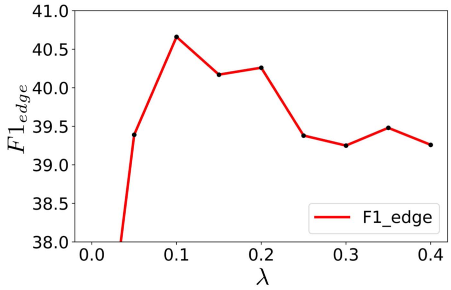

Fig. 17. Influence of the values of λ on the $\mathrm{F1}_{\mathrm{edge}}$ score.

the commonly used model FC-Siam-diff. As shown in Table X, the F1 score improves significantly, demonstrating the generality of the introduced edge guidance strategy.

G. Network Visualization

To understand our EGPNet intuitively, we visualize the activation maps at feature level 2. Given the bitemporal images, the bitemporal encoder produces bitemporal features f_{b1}^2, f_{b2}^2, and the difference encoder produces difference features f_d^2 at feature level 2. Then, we integrate $f_{b1}^2, f_{b2}^2, f_d^2$ to produce the fused features f_f^2. Finally, f_f^2 is passed through EFM for edge representation enhancement, producing f_{en}^2. Three representative activation maps are selected for visualization from $f_{b1}^2, f_{b2}^2, f_d^2$, and f_{en}^2, respectively. Fig. 15(b) and (c) shows bitemporal features. They can reflect details in bitemporal images but cannot explicitly reflect the changes. Fig. 15(d) shows difference features. They can reflect the main difference between bitemporal images but lack many details, which will cause the missed detection problem. Fig. 15(e) shows the fused features. A relatively complete and accurate picture of the difference can be obtained by combining the merits of bitemporal and difference features. This demonstrates the effectiveness of the parallel encoding framework. Fig. 15(f) shows features after EFM. From Fig. 15(e) and (f), we can see that the features after EFM have stronger and sharper edges. The edge guidance strategy improves the edge representation significantly.

V. Discussion

In this section, we perform extensive experiments on the LEVIR-CD dataset to find the optimal value for λ and discuss the effect of different λ values. λ that controls the proportion of edge loss in total loss has a great influence on the performance of the proposed method. As shown in Figs. 16 and 17, smaller edge loss causes the network to pay too much attention to internal regions resulting in lower $\mathrm{F1}_{\mathrm{edge}}$ and F1 scores. Larger edge loss causes the network to pay too much attention to edges, which is also detrimental to the performance of the proposed method. 0.1 achieves the optimal balance. Under this circumstance, the generated edges are accurate and can well guide the representation learning of the fused features.

Benefiting from the proposed parallel encoding framework and edge guidance strategy, our EGPNet achieves higher accuracy than several SOTA methods. However, our work is based on the commonly used Unet [17] network, which is not the latest semantic segmentation model. Recently, VIT (Vision Transformer) [54] has shown advantages in CD [15], [28]. Our future work is to apply VIT models to extract bitemporal and difference features effectively.

VI. Conclusion

In this article, we propose an EGPNet for VHR remote sensing image CD. To utilize detailed information in bitemporal features and change-related information in difference features, we propose a parallel encoding framework in which we design an SM to enrich the difference feature flow with bitemporal feature flow. Benefiting from the feature complementarity, the EGPNet can detect the change regions completely, especially their details, more accurately. To enhance the edge representation, we introduce an edge guidance strategy composed of EAM and EFM. Our proposed network outperforms many SOTA methods on the LEVIR-CD, SYSU-CD, and CDD datasets, and the results detected by our EGPNet have more precise and sharper edges.

References

[1] J.-S. Zhang, C.-Y. He, Y.-Z. Pan, and J. Li, "The high spatial resolution RS image classification based on SVM method with the multi-source data," *J. Remote Sens.*, vol. 10, no. 1, 2006, Art. no. 49.

[2] K. S. Willis, "Remote sensing change detection for ecological monitoring in United States protected areas," *Biol. Conservation*, vol. 182, pp. 233–242, 2015.

[3] D. Qin et al., "MSIM: A change detection framework for damage assessment in natural disasters," *Expert Syst. Appl.*, vol. 97, pp. 372–383, 2018.

[4] I. R. Hegazy and M. R. Kaloop, "Monitoring urban growth and land use change detection with GIS and remote sensing techniques in Daqahlia governorate Egypt," *Int. J. Sustain. Built Environ.*, vol. 4, no. 1, pp. 117–124, 2015.

[5] N. Quarmby and J. Cushnie, "Monitoring urban land cover changes at the urban fringe from spot HRV imagery in South-East England," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 953–963, 1989.

[6] E. H. Wilson and S. A. Sader, "Detection of forest harvest type using multiple dates of landsat TM imagery," *Remote Sens. Environ.*, vol. 80, no. 3, pp. 385–396, 2002.

[7] F. Yuan, K. E. Sawaya, B. C. Loeffelholz, and M. E. Bauer, "Land cover classification and change analysis of the twin cities (Minnesota) metropolitan area by multitemporal landsat remote sensing," *Remote Sens. Environ.*, vol. 98, no. 2/3, pp. 317–328, 2005.

[8] O. Miller, A. Pikaz, and A. Averbuch, "Objects based change detection in a pair of gray-level images," *Pattern Recognit.*, vol. 38, no. 11, pp. 1976–1992, 2005.

[9] A. Lefebvre, T. Corpetti, and L. Hubert-Moy, "Object-oriented approach and texture analysis for change detection in very high resolution images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2008, vol. 4, pp. IV-663–IV-666.

[10] O. Hall and G. J. Hay, "A multiscale object-specific approach to digital change detection," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 4, no. 4, pp. 311–327, 2003.

[11] T. Lei et al., "Difference enhancement and spatial–spectral nonlocal network for change detection in VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Dec. 10, 2021, Art. no. 4507013.

[12] X. Peng, R. Zhong, Z. Li, and Q. Li, "Optical remote sensing image change detection based on attention mechanism and image difference," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7296–7307, Sep. 2021.

[13] C. Zhang et al., "A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images," *ISPRS J. Photogrammetry Remote Sens.*, vol. 166, pp. 183–200, 2020.

[14] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 568–576.

[15] H. Chen, Z. Qi, and Z. Shi, "Remote sensing image change detection with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jul. 20, 2021, Art. no. 5920416.

[16] Y. Sun, S. Wang, C. Chen, and T.-Z. Xiang, "Boundary-guided camouflaged object detection," in *Proc. 31th Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 1335–1341, doi: 10.24963/ijcai.2022/186.

[17] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional Siamese networks for change detection," in *Proc. IEEE Int. Conf. Image Process.*, 2018, pp. 4063–4067.

[18] T. Chen, Z. Lu, Y. Yang, Y. Zhang, B. Du, and A. Plaza, "A Siamese network based U-Net for change detection in high resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 2357–2369, Mar. 11, 2022.

[19] S. Fang, K. Li, J. Shao, and Z. Li, "SNUNet-CD: A densely connected Siamese network for change detection of VHR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Feb. 17, 2021, Art. no. 8007805.

[20] H. Chen, N. Yokoya, C. Wu, and B. Du, "Unsupervised multimodal change detection based on structural relationship graph representation learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Dec. 14, 2022, Art. no. 5635318.

[21] Y. Liu, C. Pang, Z. Zhan, X. Zhang, and X. Yang, "Building change detection for remote sensing images using a dual-task constrained deep Siamese convolutional network model," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 5, pp. 811–815, May 2021.

[22] H. Chen, W. Li, S. Chen, and Z. Shi, "Semantic-aware dense representation learning for remote sensing image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Aug. 30, 2021, Art. no. 5630018.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.

[24] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change detection based on deep Siamese convolutional network for optical aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1845–1849, Oct. 2017.

[25] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 1856–1867, Jun. 2020.

[26] H. Chen, C. Wu, B. Du, L. Zhang, and L. Wang, "Change detection in multisource VHR images via deep Siamese convolutional multiple-layers recurrent neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2848–2864, Apr. 2020.

[27] X. Zhang et al., "DifUNet++: A satellite images change detection network based on UNet and differential pyramid," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Jan. 22, 2021, Art. no. 8006605.

[28] W. G. C. Bandara and V. M. Patel, "A transformer-based Siamese network for change detection," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2022, pp. 207–210.

[29] H. Chen, N. Yokoya, and M. Chini, "Fourier domain structural relationship analysis for unsupervised multimodal change detection," *ISPRS J. Photogrammetry Remote Sens.*, vol. 198, pp. 99–114, 2023.

[30] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 12077–12090.

[31] R. Liu, D. Jiang, L. Zhang, and Z. Zhang, "Deep depthwise separable convolutional network for change detection in optical aerial images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1109–1118, Mar. 16, 2020.

[32] D. Peng, Y. Zhang, and H. Guan, "End-to-end change detection for high resolution satellite images using improved UNet," *Remote Sens.*, vol. 11, no. 11, 2019, Art. no. 1382.

[33] H. Chen and Z. Shi, "A spatial-temporal attention-based method and a new dataset for remote sensing image change detection," *Remote Sens.*, vol. 12, no. 10, 2020, Art. no. 1662.

[34] Q. Shi, M. Liu, S. Li, X. Liu, F. Wang, and L. Zhang, "A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jun. 29, 2021, Art. no. 5604816.

[35] J. Chen et al., "DASNet: Dual attentive fully convolutional Siamese networks for change detection in high-resolution satellite images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1194–1206, Nov. 13, 2020.

[36] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1053–1061.

[37] M. Zhang, S. X. Fei, J. Liu, S. Xu, Y. Piao, and H. Lu, "Asymmetric two-stream architecture for accurate RGB-D saliency detection," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 374–390.

[38] J.-X. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "EGNet: Edge guidance network for salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8779–8788.

[39] W. Guan, T. Wang, J. Qi, L. Zhang, and H. Lu, "Edge-aware convolution neural network based salient object detection," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 114–118, Jan. 2019.

[40] J. Zhang, Y. Dai, F. Porikli, and M. He, "Deep edge-aware saliency detection," 2017, *arXiv:1708.04366*.

[41] Z. Zhang, H. Fu, H. Dai, J. Shen, Y. Pang, and L. Shao, "ET-Net: A generic edge-attention guidance network for medical image segmentation," in *Proc. 22nd Int. Conf. Med. Image Comput. Comput. Assist. Intervention*, 2019, pp. 442–450.

[42] S. Shamim, M. J. Awan, A. M. Zain, U. Naseem, M. A. Mohammed, and B. Garcia-Zapirain, "Automatic COVID-19 lung infection segmentation through modified Unet model," *J. Healthcare Eng.*, vol. 2022, 2022, Art. no. 6566982.

[43] G. Cheng, G. Wang, and J. Han, "ISNet: Towards improving separability for remote sensing image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, May 11, 2022, Art. no. 5623811.

[44] B. Bai, W. Fu, T. Lu, and S. Li, "Edge-guided recurrent convolutional neural network for multitemporal remote sensing image building change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5610613.

[45] Z. Chen et al., "EGDE-Net: A building change detection method for high-resolution remote sensing imagery based on edge guidance and differential enhancement," *ISPRS J. Photogrammetry Remote Sens.*, vol. 191, pp. 203–222, 2022.

[46] L. Xia, J. Chen, J. Luo, J. Zhang, D. Yang, and Z. Shen, "Building change detection based on an edge-guided convolutional neural network combined with a transformer," *Remote Sens.*, vol. 14, no. 18, 2022, Art. no. 4524.

[47] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11534–11542.

[48] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[49] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.

[50] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[51] M. Lebedev, Y. V. Vizilter, O. Vygolov, V. A. Knyaz, and A. Y. Rubis, "Change detection in remote sensing images using conditional adversarial networks," *Int. Arch. Photogrammetry, Remote Sens., Spatial Inf. Sci.*, vol. 42, pp. 565–571, 2018.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[53] D. Kinga et al., "A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, vol. 5, 2015, Art. no. 6.

[54] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021.

Ye Zhu received the B.E. degree in electronic information science and engineering from the Shandong University of Science and Technology, Qingdao, China, in 2011, and the Ph.D. degree in computer application technology from the College of Computer Science and Technology, Jilin University, Changchun, China, in 2017.

She is currently an Associate Professor with the School of Artificial Intelligence, Hebei University of Technology, Tianjin, China. Her current research interests include remote sensing, artificial intelligence, and multimedia forensics.

Kaikai Lv received the B.E. degree in mechanical engineering from Henan Polytechnic University, Jiaozuo, China, in 2016. He is currently working toward the M.S. degree in computer technology with the School of Artificial Intelligence, Hebei University of Technology, Tianjin, China.

His current research interests include remote sensing and image processing.

Yang Yu received the Ph.D. degree in microelectronics and solid-state electronics from the Hebei University of Technology, Tianjin, China, in 2012.

He is currently an Associate Professor with the School of Artificial Intelligence, Hebei University of Technology. His current research interests include remote sensing and image processing.

Wenjia Xu received the B.S. degree in environmental science from Beijing Forestry University, Beijing, China, in 2005, and the M.S. degree in ecology from the Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing, in 2008.

She is currently a Senior Engineer with the Hebei Prospecting Institute of Hydrogeology and Engineering Geological (Hebei Remote Sensing Center), Shijiazhuang, China. Her current research interests include remote sensing Big Data and artificial intelligence remote sensing.