Multiscale Diff-Changed Feature Fusion Network for Hyperspectral Image Change Detection

Fulin Luo, *Senior Member, IEEE*, Tianyuan Zhou, Jiamin Liu, Tan Guo, *Member, IEEE*, Xiuwen Gong, and Jinchang Ren, *Senior Member, IEEE*

***Abstract*— For hyperspectral image (HSI) change detection (CD), multiscale features are usually used to construct the detection models. However, the existing studies only consider the multiscale features containing changed and unchanged components, which is difficult to represent the subtle changes between bitemporal HSIs in each scale. To address this problem, we propose a multiscale diff-changed feature fusion network (MSDFFN) for HSI CD, which improves the ability of feature representation by learning the refined change components between bitemporal HSIs under different scales. In this network, a temporal feature encoder–decoder subnetwork, which combines a reduced inception (RI) module and a cross-layer attention module to highlight the significant features, is designed to extract the temporal features of HSIs. A bidirectional diff-changed feature representation (BDFR) module is proposed to learn the fine changed features of bitemporal HSIs at various scales to enhance the discriminative performance of the subtle change. A multiscale attention fusion (MSAF) module is developed to adaptively fuse the changed features of various scales. The proposed method can not only discover the subtle change in bitemporal HSIs but also improve the discriminating power for HSI CD. Experimental results on three HSI datasets show that MSDFFN outperforms a few state-of-the-art methods.**

***Index Terms*— Attention fusion, change detection (CD), convolutional encoder–decoder network, hyperspectral image (HSI), multiscale features.**

I. Introduction

CHANGE detection (CD) based on remote sensing data is an important technology to detect the changes in the Earth's surface and has a wide range of applications in urban planning, environmental monitoring, agriculture investigation, disaster assessment, and map revision [1], [2]. Remote sensing CD is to detect the land-cover changes in images at the same area under two different time periods. It is the response of pixelwise characteristic change to land-cover changes. With the advancement of imaging spectroscopy, bitemporal hyperspectral images (HSIs) have been widely used for CD [3]. HSI data have the advantage of providing continuous and detailed spectral features in a large spectral range, with the characteristics of "image-spectrum merging" [4]. This characteristic is beneficial to discriminate the changed regions between two images [5]. Recently, researchers have come up with many methods to extract useful features for HSI CD [6].

In early research, the CD methods can be categorized as algebra-based methods, transformation-based methods, and classification-based methods. The algebra-based methods mainly include image difference, image ratio, image regression, absolute distance (AD), change vector analysis (CVA) [7], [8], etc. The most representative CVA is the subtraction of two temporal images to get spectral change vectors, where the magnitude and direction of change vectors show the degree of variation, and then the change vectors can be classified by a threshold. CVA ignores the similarity between adjacent pixels; later, Thonfeld et al. [9] proposed robust CVA (RCVA) which considers the influence of neighborhood pixels. The accuracy of the radiation and geometric correction has an important impact on the results of the algebra-based methods. The transformation-based methods project HSIs into another feature space to represent the changed pixels or regions. Among them, principal component analysis (PCA) [10] is the most widely used data dimensionality reduction algorithm which maps data to the direction with the largest variance [11]. PCA with k-means (PCAKM) [12] uses PCA to generate low-dimensional features and perform k-means clustering on the reduced features to obtain change results. Multivariate change detection (MAD) [13] uses canonical correlation analysis (CCA) [14] to maximize the correlation between the features of multitemporal images. Nielsen [15] proposed an iteratively reweighted MAD (IRMAD) method which conducts the weighted iteration according to the chi-square distance. Slow feature analysis (SFA) [16] extracts the most temporally invariant component from the bitemporal images to transform the data into a new feature space. In the classification-based methods, the post classification method first learns and classifies the bitemporary images, respectively, and then compares and analyzes the changes. The direct classification method is to combine the bitemporal images together, and then a classifier is used to find the changing categories. The typical classifiers are K-nearest neighbors (KNNs) [17], support vector machine

Manuscript received 18 October 2022; revised 14 December 2022; accepted 27 January 2023. Date of publication 31 January 2023; date of current version 9 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62071340 and Grant 62201109 and in part by the Natural Science Foundation of Chongqing under Grant CSTB2022NSCQ-MSX0452. *(Corresponding author: Jiamin Liu.)*

Fulin Luo and Tianyuan Zhou are with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: luoflyn@163.com; zhoutianyuan1016@163.com).

Jiamin Liu is with the Key Laboratory of Optoelectronic Technology and Systems of the Education Ministry of China, Chongqing University, Chongqing 400044, China (e-mail: liujm@cqu.edu.cn).

Tan Guo is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: guot@cqupt.edu.cn).

Xiuwen Gong is with the Faculty of Engineering, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: xiuwen.gong@sydney.edu.au).

Jinchang Ren is with the National Subsea Centre, Robert Gordon University, AB10 7AQ Aberdeen, U.K. (e-mail: jinchang.ren@ieee.org).

Digital Object Identifier 10.1109/TGRS.2023.3241097

(SVM) [18], etc. Conventional CD methods are often based on the spectral difference between the corresponding bands to measure the degree of change. They do not take into account the correlation between bands and cannot fully exploit the intrinsic characteristics of complex HSIs.

Recently, convolutional neural network (CNN) has become a research focus in CD [19] because of its stronger adaptive feature extraction ability. For example, Saha et al. [20] proposed a context-based deep CVA (DCVA), and it performs semantic segmentation on a single image with the same pretrained CNN to obtain a coherent depth feature supervector. Du et al. [21] used SFA to learn slowly changing features and enhanced the discrimination of changed pixels. These methods use CNN to automatically extract abstract features, and then process features by the traditional methods. There are also many methods that use CNN directly to find change pixels [22]. Considering the characteristics of HSIs, Wang et al. [23] proposed a general end-to-end 2-D CNN (GETNET), which performs spectral unmixing on the input HSIs to obtain a mixed affinity matrix, and then uses CNN to mine the feature information. Lin et al. [24] proposed a bilinear CNN (BCNN), and it finds the relationship between bitemporal feature maps by designing a combined bilinear feature. Mou et al. [25] and Chen et al. [26] proposed two networks named recurrent CNN (ReCNN) and Siamese convolutional recurrent neural network (SiamCRNN), and they use recurrent neural network (RNN) and long-short term memory (LSTM) [27] to find the spatio-temporal relationship of bitemporal HSIs, respectively. Among them, RNN is able to extract temporal features based on the cyclic hidden state of the previous time. And LSTM, as a special RNN structure, can effectively overcome the problem of gradient disappearance and gradient explosion during training. The accuracy of these networks has improved compared with the traditional methods, but they still lack sufficient consideration of rich spectral data and attention to important information.

The attention mechanism is proposed to focus on key information, and it is widely used in the field of deep learning [28]. It is essentially a mechanism which learns a set of weighting coefficients by the network autonomously and dynamically emphasizes the regions of interest [28]. The features in the image include changing and unchanging, and the changing components are of interest and concern to us, which is just in line with the idea of attention mechanism. In CD, several methods have explored the attention mechanism to improve the detection performance. Moustafa et al. [29] proposed a attention residual recurrent U-Net (Att R2U-Net), which combines the classical U-Net with the attention mechanism and shows excellent performance in binary and multiclass CD of HSI. Chen et al. [30] added spatial channel double attention mechanism in their network to capture long-range correlations. Jiang et al. [31] proposed a pyramid-feature-based attention-guided Siamese network (PGA-SiamNet). It improves the long-range dependencies of the features using various attention mechanisms. Qu et al. [32] proposed a multilevel encoder–decoder attention network (MLEDAN), which introduces multiscale connection and attention mechanism to extract more effective spatial–spectral features. Then, LSTM is also used to analyze temporal dependence between multitemporal images. Gong et al. [33] proposed a spectral and spatial attention network (S2AN), which uses multiple repetitive spatial attention modules with adaptive Gaussian distributions to gradually enhance CD-related features. In summary, the attention mechanisms can help note the changing regions of the spatial–spectral information.

From the above research work, we found that there is a great potential about how to better extract information from the input HSIs and how to more fully integrate the features of different phases for HSI CD. First, shallow network structures often have the difficulty to extract the effective features, while complex network structures will lead to computational redundancy and may learn the irrelevant features for CD. Second, some current deep learning methods fuse the bitemporal features of two HSIs by RNN or LSTM, which simultaneously combines the changed components and unchanged components. For the CD task, it should significantly focus on the change components. Therefore, these methods do not separate changed and unchanged components to implement the CD tasks, which is very difficult to learn the subtle changed features of the bitemporal HSIs. And the feature fusion does not consider the importance of different features. We will pay more attention to the change components of features, so that we can more carefully explore the details of changes. In addition, multiscale features are conducive to learning fine changes, so our research is based on multiscale change features.

Based on this, we propose a multiscale diff-changed feature fusion network (MSDFFN) as shown in Fig. 1; it can learn fine and representative change features from bitemporal HSIs. MSDFFN is composed of a temporal feature encoder–decoder (TFED) subnetwork, a bidirectional diff-changed feature representation (BDFR) module, and a multiscale attention fusion (MSAF) module. The TFED subnetwork with reduced inception (RI) and skip layer attention (SLA) is designed to extract multiscale features from the input HSI patches. The BDFR module is proposed to specifically learn and enhance the discriminating features of change components obtained by the TFED subnetwork and the differential operation. Then, the MSAF module is used to fuse the multiscale features and obtain the final features with discriminatory power. The main contributions of this article are highlighted as follows.

1) We design a TFED subnetwork to extract the features of multitemporal HSIs, where an RI module is embedded to enrich the perceptual field, and skip connections containing channel-space coattention are added to fuse the contextual information.
2) The proposed BDFR module learns and fuses the refined change features, which pays special attention to the changed components in the entire features. With bidirectional representation, the subtle changes can be enhanced to improve the detection performance.
3) The MSAF module, focusing on the fusion of key information, is proposed to mine the intrinsic information of different feature maps and generate a discriminative spatial–spectral–temporal change features.

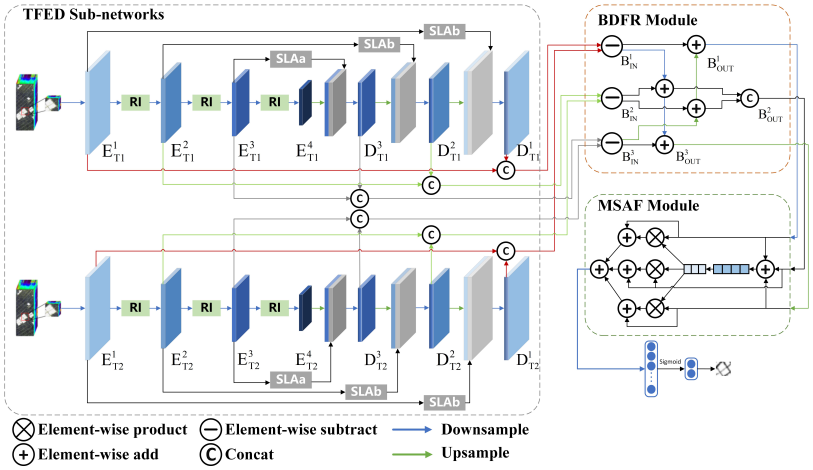


Fig. 1. Overview of the proposed MSDFFN.

4) By combining the above three thoughts, we propose the end-to-end MSDFFN, called MSDFFN, for HSI CD, which fuses the multiscale information to extract features with strong representational power. The experimental results on three HSI datasets show that MSDFFN outperforms the several state-of-the-art methods for HSI CD.

The rest of this article is organized as follows. Section II introduces the details of the proposed MSDFFN. Section III presents three experimental datasets, experiment setting, experimental results, and ablation study. In the end, Section IV draws some conclusions of this article and suggestions for future work.

II. Proposed Method

In this section, we introduce the proposed MSDFFN for the HSI CD task in Fig. 1, which is composed of the TFED subnetwork, BDFR module, and MSAF module. The bitemporal HSIs are passed through the temporal feature extraction subnetwork, which combines the RI module and the SLA module to obtain multiscale features. And then, the diff-changed features with fine representation power are acquired and learned by the BDFR module from these multiscale features. After that, the MSAF module fuses the multiscale diff-changed features adaptively with residual attention. In the following, we will explain each module of the network in detail.

A. Baseline

Before presenting the specific network details, we constructed a basic CD network framework. The baseline model is shown in Fig. 2. The encoder–decoder network [34] is widely used in the CD tasks because it can fuse image features well. In CNN, low-level features often have higher resolution and contain more details while lacking semantic information, and high-level features often have stronger semantic information while low resolution and poor perception of details [35]. The encoding and decoding network constructs skip connections between downsampling and upsampling layers to achieve the fusion of low-level and high-level features.

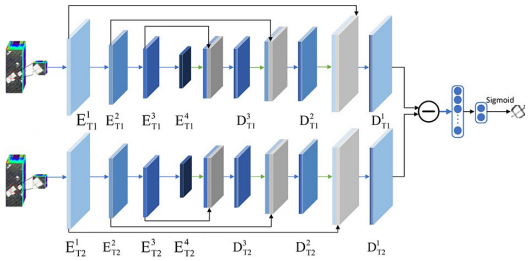


Fig. 2. Structure of the designed baseline.

As shown in Fig. 2, the patches of the bitemporal HSIs are passed through an encoder–decoder network to get the feature maps, and then the diff-changed feature map is obtained by the differential operation, which reflects the part of the changes between the bitemporal images. The diff-changed feature map is used to get the final CD result by sigmoid function. The network mainly includes three parts, i.e., bitemporal feature extraction, change feature extraction, and change feature classification. Among them, for change feature extraction, some studies have used RNN or LSTM to find the part of changes from temporal features, which simultaneously includes changed components and unchanged components. In this article, we mainly consider the change components to learn the subtle change features by the differential operation. Therefore, we construct a BDFR module to learn the fine features of change components.

B. Temporal Feature Encoder–Decoder Subnetwork

1) Architecture: Although the baseline model can perform basic CD, extraction and learning of changing features are not sufficient. It is difficult to extract discriminating features from complex HSIs. In this article, we propose a new TFED subnetwork with RI and SLA base on the baseline network. The architecture of the TFED network is divided into the encoder and decoder, including encoding path, decoding path, and three skip connection operations, as shown in Fig. 3. The encoder extracts features with a series of downsampling based on convolution operations. The RI module is added before downsampling to enrich receptive field and extract features at different scales. The decoder recovers resolution by continuously upsampling based on the deconvolution operations.

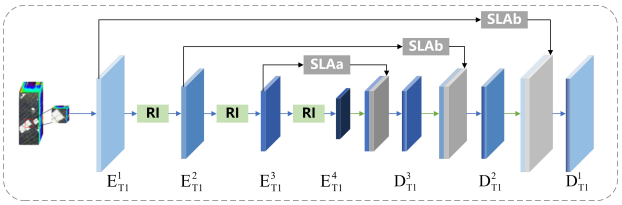


Fig. 3. Structure of the TFED subnetwork.

To better deliver the low-level features at the decoding stage, the SLA module is integrated into the skip connections to highlight the shallow features.

Given the initial feature map $E^0 \in \mathbb{R}^{C\times H\times W}$ as the input patch, the encoder stage of the TFED subnetwork can be summarized as follows:

$$E_T^{i+1} = \mathrm{ReLU}\left(f_{\mathrm{Conv}}{}^{3\times 3}\left[\mathrm{RI}\left(E_T^i\right)\right]\right), \quad i = 1, 2, 3 \tag{1}$$

where E_T^i represents the feature map generated by the ith convolutional layer, $E_T^1 = f_{\mathrm{Conv}}{}^{3\times 3}(E^0)$, $\mathrm{RI}(\cdot)$ represents the RI operation on the feature map, and $f_{\mathrm{Conv}}{}^{3\times 3}$ represents a convolution operation with a filter size of 3×3. All the convolution layers are followed by a batch normalization layer and a rectified linear unit (ReLU) layer. $\mathrm{ReLU}(\cdot)$ denotes the ReLU active function which can be described as

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2}$$

where x represents the value of feature map.

In the encoding path, the previous feature map is passed through an RI module, a convolution operation without padding, and an ReLU function to obtain the output feature map. The RI structure is able to extract the refined features of different shapes with several different convolutional kernels, which helps extract more effective features. The ReLU function makes the output of some neurons to be zero, which causes the sparsity of the network, reduces the interdependence of parameters, and alleviates overfitting. With successive downsampling operations, more abstract features can be learned from the input HSI patch step by step.

The decoder stage of the TFED subnetwork can also be summarized as follows:

$$D_T^i = \begin{cases} \mathrm{ReLU}\left(f_{\mathrm{Conv}}{}^{3\times 3}\left[f_{\mathrm{DConv}}{}^{3\times 3}\left(D_T^{i+1}\right); S_b\left(E_T^i\right)\right]\right), & i = 1, 2 \\ \mathrm{ReLU}\left(f_{\mathrm{Conv}}{}^{3\times 3}\left[f_{\mathrm{DConv}}{}^{3\times 3}\left(E_T^{i+1}\right); S_a\left(E_T^i\right)\right]\right), & i = 3 \end{cases} \tag{3}$$

where D_T^i represents the output feature map after the ith deconvolution operation, $f_{\mathrm{DConv}}{}^{3\times 3}$ represents a deconvolution operation with the filter size of 3×3, and $S_a(\cdot)$ and $S_b(\cdot)$ represent the SLA embedded into skip connections.

In the decoding path, the attention-enhanced feature maps from the encoder and the same-scale upsampled feature maps from the deconvolution are stacked in the channel dimension, which can achieve the fusion of high-level features and low-level features. With such a fusion of across-layer features, a better balance between fine-grained and semantic information can be achieved. The skip connections with attention enable early encoder layers to preserve the positional relation of pixel, which can better recover the detailed structure of the input.

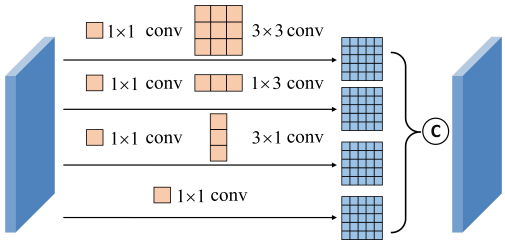


Fig. 4. Structure of the RI module.

2) Reduced Inception Module: The inception module was first proposed in GoogleNet [36]. It uses convolutional kernels of different sizes to obtain features from different receptive fields. The specific convolution kernel size is selected by the network itself by adjusting parameters in the process of training, so the architecture is highly tunable [37]. We can appropriately choose the number of filters and kernel size to maximize the retention of features that are beneficial for CD.

In the TFED subnetwork, we propose an RI module to improve the ability of feature extraction, as shown in Fig. 4. To reduce the calculation amount caused by multiple convolution layers, a 1×1 convolution is added before each convolutional layer to reduce the channel dimension. Considering the input size of the patch, we use the convolutional layers with the kernel size of 1×3, 3×1, and 3×3 to construct the RI module. After that, we concatenate these convolutional features to obtain the feature maps with the same size as the input of RI.

The specific process can be summarized as follows. Given an intermediate feature map $X \in \mathbb{R}^{C\times H\times W}$ as input, the RI can be represented as

$$\begin{aligned} \mathrm{RI}(X) = \big[& f_{\mathrm{Conv}}{}^{1\times 3}\left(f_{\mathrm{Conv}}{}^{1\times 1}(X)\right); \\ & f_{\mathrm{Conv}}{}^{3\times 1}\left(f_{\mathrm{Conv}}{}^{1\times 1}(X)\right); \\ & f_{\mathrm{Conv}}{}^{3\times 3}\left(f_{\mathrm{Conv}}{}^{1\times 1}(X)\right); \\ & f_{\mathrm{Conv}}{}^{1\times 1}(X)\big] \end{aligned} \tag{4}$$

where $\mathrm{RI}(X) \in \mathbb{R}^{C\times H\times W}$ is the final output. and $f_{\mathrm{Conv}}{}^{N\times M}$ represents the convolution operation with the filter size of $N \times M$. The asymmetric convolutional blocks formed by three different convolutional kernels can capture different shapes including square, horizontal, and vertical features from the input feature maps, which can improve the discriminative power of the features. Embedding the above RI module into the TFED network can extend the receptive field and improve the scale adaptability.

3) Skip Layer Attention Module: Attention mechanism has become one of the most widely used in deep learning [38]. When dealing with information, human beings often pay attention to the more important characteristics of input. The attention mechanism simulates the mechanism of human processing information. The essence function of attention mechanism is to apply the learned weights to the original features. Different parts of the input data or feature maps have different degrees of focus, and attention mechanism ignores irrelevant noise information and focuses on key information.

We design a spatial–spectral attention based on convolutional block attention module (CBAM) [39], [40] to learn the refined features of CD tasks adaptively. The attention

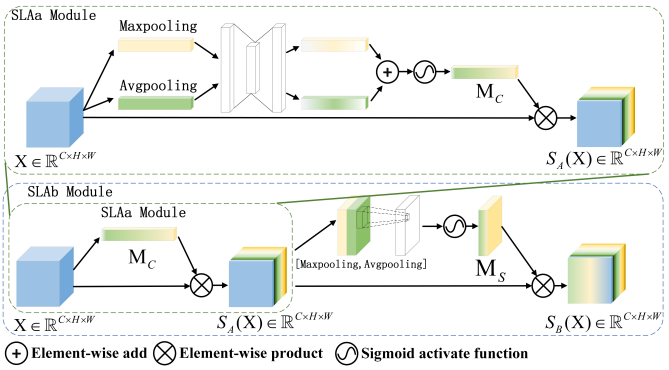


Fig. 5. Structure of the SLA module.

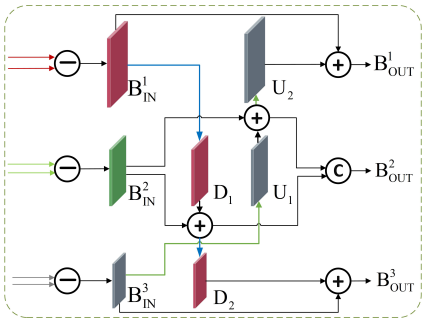


Fig. 6. Structure of the BDFR module.

module is embedded into the skip connections to highlight the feature maps from the encoder, which can enhance the regions of interest for the CD task. As shown in Fig. 5, to accommodate the input feature maps of different scales, we design two attention models that are SLAa and SLAb based on channel attention and spatial attention. We use the one containing only channel attention as SLAa, and the one cascading channel attention and spatial attention as SLAb. By multiplying the input features with the attention weights, we can obtain the enhanced feature maps on channels and spatially, respectively. The channel attention module refines the weight of each feature map to emphasize the meaningful channels and suppress the useless ones. The spatial attention module refines the weight of each spatial position to highlight informative regions and compress useless ones.

First, we generate two different spatial context descriptors, i.e., average-pooled descriptor F^c_{avg} and max-pooled descriptor F^c_{max}, by max-pooling and average-pooling operations. The spatial context descriptors can aggregate the spatial information of feature maps. After that, we input the two descriptors into the multilayer perception (MLP) with one hidden layer to compress and extract the features. To reduce the number of parameters, the size of the hidden layer is set as C/r to reduce the number of channels, where r is the reduction ratio. Then, we merge the output features of two descriptors using elementwise summation and activate the fused features with sigmoid function. Given an intermediate feature map $X \in \mathbb{R}^{C\times H\times W}$ as input, the channel attention maps are computed by

$$\begin{aligned} M_c(X) &= \sigma(\text{MLP}(\text{AvgPool}(X)) + \text{MLP}(\text{MaxPool}(X))) \\ &= \sigma\left(W_1\left(W_0\left(F^c_{\text{avg}}\right)\right) + W_1\left(W_0\left(F^c_{\text{max}}\right)\right)\right) \end{aligned} \tag{5}$$

where $M_c(X) \in \mathbb{R}^{C\times 1\times 1}$ infers the channel attention maps, and the MLP weights $W_0 \in \mathbb{R}^{C/r\times C}$ and $W_1 \in \mathbb{R}^{C\times C/r}$ are shared for inputs. $\sigma(\cdot)$ denotes the sigmoid function which can be described as

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{6}$$

where x represents a value of feature map. The attention added to the channels can adaptively adjust the feature response values of each channel to highlight the useful information.

Second, we generate two different channel context descriptors, including average-pooled descriptor $F^s_{\text{avg}} \in \mathbb{R}^{1\times H\times W}$ and max-pooled descriptor $F^s_{\text{max}} \in \mathbb{R}^{1\times H\times W}$, by max-pooling and average-pooling operations along the channels. The operations can aggregate the average pool features and the maximum pool features for the entire channels. Then, the two descriptors are stacked and passed through the standard convolution layer to generate the spatial attention maps. The spatial attention can be produced by

$$\begin{aligned} M_s(X) &= \sigma\left(f_{\text{Conv}}{}^{3\times 3}\left(\left[\text{AvgPool}(X); \text{MaxPool}(X)\right]\right)\right) \\ &= \sigma\left(f_{\text{Conv}}{}^{3\times 3}\left(\left[F^s_{\text{avg}}; F^s_{\text{max}}\right]\right)\right) \end{aligned} \tag{7}$$

where $M_S(X) \in \mathbb{R}^{1\times H\times W}$ infers the spatial attention maps.

The operation of the SLA module is summarized as follows:

$$S_A(X) = M_C(X) \otimes X \tag{8}$$

$$S_B(X) = M_S(S_A(X)) \otimes S_A(X) \tag{9}$$

where $\otimes$ denotes the elementwise product. During multiplication, the attention values are broadcasted accordingly.

In the specific TFED subnetwork, SLAa is used for deep-level features whose size of the feature maps is relatively small, and SLAb is used for shallow-level features whose size of the feature maps is relatively large. The reason is that smaller feature maps, such as 3×3, contain less spatial information. To simplify the network and reduce the computational cost, we do not focus on their spatial features.

C. Bidirectional Diff-Changed Feature Representation Module

After the input patches of bitemporal HSIs pass through the front TFED subnetwork, we can obtain feature maps with the three scales of 9, 7, and 5 for each temporal feature. The feature maps often contain the changed components and the unchanged components, where the changed can better find out the changes in detail. Our purpose is to discover the changed pixels in the bitemporal images; however, most of the previous studies do not focus on the learning of the changed components that can be represented by the diff-changed features. Therefore, we propose a BDFR module to learn the subtle difference features. This module makes use of the information of multilevel feature maps from the TFED subnetwork. The structure is shown in Fig. 6.

Differential operation can highlight difference information. To make the model more adaptive to bitemporal HSIs, the multiscale feature maps obtained from the TFED subnetwork are subtracted at the corresponding scales. The diff-changed

feature maps are used as the input of the subsequent processing, and the operation can be summarized as

$$B_{\mathrm{IN}}^{i} = f_{\mathrm{Conv}}{}^{1\times1}\left(\left[E_{T=2}^{i}; D_{T=2}^{i}\right] - \left[E_{T=1}^{i}; D_{T=1}^{i}\right]\right), \quad i = 1, 2, 3 \tag{10}$$

where B_{IN}^{i} is a diff-changed feature map as the input feature map of the BDFR module.

To better enhance the discriminative performance of diff-changed features, the BDFR module is composed of the upsampling and downsampling paths. In the process of downsampling, the information from high-level feature maps is brought to smaller scale feature maps, and the operation can be described as

$$D_1 = f_{\mathrm{Conv}}{}^{3\times3}\left(B_{\mathrm{IN}}^{1}\right) \tag{11}$$

$$D_2 = f_{\mathrm{Conv}}{}^{3\times3}\left(D_1 + B_{\mathrm{IN}}^{2}\right) \tag{12}$$

where D_1 and D_2 represent the output feature maps after the convolution operation.

Meanwhile, with the upsampling operation, the deep features are also transferred to a large scale. The operation of upsampling can be described as

$$U_1 = f_{\mathrm{DConv}}{}^{3\times3}\left(B_{\mathrm{IN}}^{3}\right) \tag{13}$$

$$U_2 = f_{\mathrm{DConv}}{}^{3\times3}\left(B_{\mathrm{IN}}^{2} + U_1\right) \tag{14}$$

where U_1 and U_2 represent the output feature maps after the deconvolution operation.

Now, we fuse the feature map of the middle scale by stack operation and can get three output feature maps with different scales

$$B_{\mathrm{OUT}}^{1} = U_2, \quad B_{\mathrm{OUT}}^{2} = [U_1; D_1], \quad B_{\mathrm{OUT}}^{3} = D_2 \tag{15}$$

where B_{OUT}^{i} represents the output feature maps of the BDFR module. The BDFR module recombines and learns the feature maps of different scales, which maximizes the utilization of feature information and improves the discrimination of diff-changed features.

D. Multiscale Attention Fusion Module

After BDFR, we obtain three feature maps with different scales. Like the proposed baseline, we can unify the scale of feature maps and directly learn through the fully connection layers, and then achieve the classification results by the sigmoid function. However, the features obtained in this way may lack attention to the key change information which is not conducive to the final detection accuracy.

As mentioned earlier, attention mechanisms focus on more important information, and there are more applications in the field of CD gradually [41]. For example, the classic channel attention squeeze–excitation attention [42] generates the channel weights by global average pooling from the channelwise level, which allows adaptive adjustment of the feature response values for each channel. Another classic attention, i.e., CBAM, introduces two different descriptors to aggregate the features with average-pooling and max-pooling. In this section, we propose an MSAF module to adaptively focus on the areas of CD, as well as fusing the feature maps of three scales. The specific structure is shown in Fig. 7.

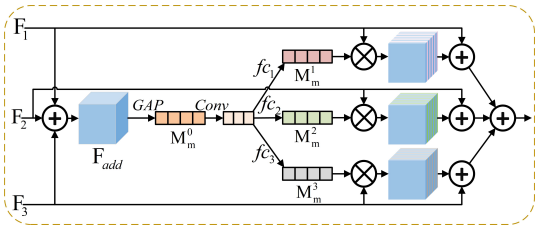


Fig. 7. Structure of the MSAF module.

First, to facilitate follow-up processing, the feature maps of the three scales are normalized to a unified scale by the convolution and deconvolution operations

$$\begin{aligned} F_{\mathrm{add}} &= f_{\mathrm{Conv}}{}^{3\times3}\left(B_{\mathrm{OUT}}^{1}\right) + B_{\mathrm{OUT}}^{2} + f_{\mathrm{DConv}}{}^{3\times3}\left(B_{\mathrm{OUT}}^{3}\right) \\ &= F_1 + F_2 + F_3 \end{aligned} \tag{16}$$

where F_i represents the output feature maps with the same size, and F_{add} represents the fused feature map by element-wise summation. To compute the channel attention efficiently, we squeeze the spatial dimension of the input feature map by global average pooling. A channelwise convolutional layer with the size of 1×1 is used to achieve the compact features and an ReLU activation is used to control the attention coefficient. Then, we generate an initial fusion channel attention feature as

$$M_m^0(F_{\mathrm{add}}) = \mathrm{ReLU}\left(f_{\mathrm{Conv}}{}^{1\times1}(\mathrm{GlobalAvgPool}(F_{\mathrm{add}}))\right) \tag{17}$$

where $M_m^0(F_{\mathrm{add}})$ represents the initial fusion feature

$$M_m^i\left(F_{\mathrm{add}}\right) = \sigma\left(\mathrm{fc}_i\left(M_m^0\left(F_{\mathrm{add}}\right)\right)\right), \quad i = 1, 2, 3 \tag{18}$$

where fc_i is a fully connection operation and has the same structure without parameter share.

To retain the original information, we add residual operation to build a connection between the original features and the attention features. The final output can be described as

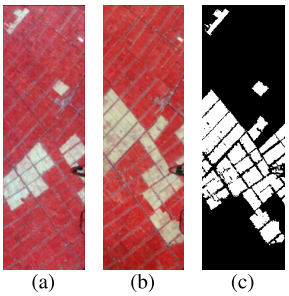
$$\begin{aligned} M_{\mathrm{out}}(X) = \big[&M_m^1\left(F_{\mathrm{add}}\right) \otimes F_1 + F_1; \\ &M_m^2\left(F_{\mathrm{add}}\right) \otimes F_2 + F_2; \\ &M_m^3\left(F_{\mathrm{add}}\right) \otimes F_3 + F_3\big]. \end{aligned} \tag{19}$$

By multiplying the input feature maps with the shared channel attention weights, the obtained feature maps highlight the useful regions and suppress the useless regions. The MSAF module adaptively selects the effective information in the multilayer features for fusion, so that the fused features achieve the complementarity of the multilayer information.

After the learning of previous several modules, a feature map, which contains rich changed details, is obtained. The probability estimate obtained from the fully connected layers can be used to predict the final labels of the input patches. Consistent with the processing in baseline, the final CD results can be described as

$$y_p = \sigma\left(\mathrm{fc}\left(f_{\mathrm{Conv}}{}^{3\times3}(M_{\mathrm{out}}(X))\right)\right) \tag{20}$$

where y_p is the predicted probability result, and $M_{\mathrm{out}}(X)$ is the output of the MSAF module. The fc(.) denotes the fully connected layers to extract the features and reduce the dimension.



(a) (b) (c)

Fig. 8. Farmland dataset. (a) Image acquired on May 3, 2006. (b) Image acquired on April 23, 2007. (c) Ground truth.

E. Loss Function

An appropriate loss function can optimize the designed network in model training to extract more effective features. The CD task considered as a classification task, and each pixel of bitemporal HSIs is divided into two categories, i.e., changed and unchanged. Therefore, the cross-entropy loss function is a popular and effective solution to optimize the network. The loss function is calculated as follows:

$$\text{Loss} = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log y_p + (1-y_i)\log\left(1-y_p\right)\right) \quad (21)$$

where n denotes the number of samples, and y_i is the ground-truth label of the given sample.

III. Experimental Results and Analysis

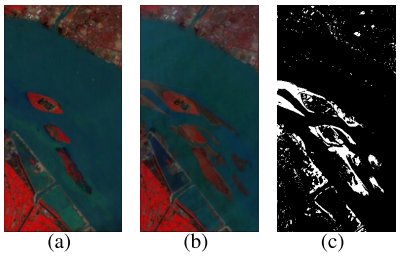
In this section, we first introduce the HSI CD datasets, and the evaluation measures are used to evaluate the effect of the proposed MSDFFN method. Then, we give a brief description of the comparison algorithms and introduce the relevant experimental details. At the same time, a series of ablation experiments are provided to verify the effectiveness of the proposed modules. Finally, we compare the impact of the training samples size on the network.

A. Datasets and Evaluation Measures

1) Datasets: The first dataset, named "Farmland," belongs to a farmland near the city of Yancheng, Jiangsu province in China, which was acquired by Earth Observing-1 (EO-1) Hyperion sensor on May 3, 2006, and April 23, 2007, respectively. The dataset has 242 bands in the range of 0.4–2.5 m with a spatial resolution of 30 m as shown in Fig. 8. After removing noise and water absorption bands, it contains 155 spectral bands for experiments and the spatial size of each image is 450 × 140 pixels. The main change areas are farmland.

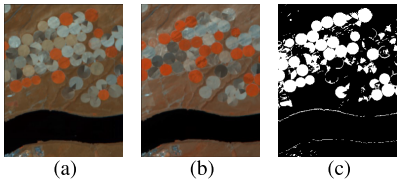
The second dataset, named "River," covers a river area from Jiangsu Province in China, as shown in Fig. 9, which was acquired on May 3, 2013 and December 31, 2013, respectively. They are also observed by the sensor EO-1. This dataset has a spatial size of 463 × 241 pixels with 198 bands available after noisy band removal. The main type of change on this dataset is the reduction of river course.

The third dataset, named "Hermiston," as shown in Fig. 10, belongs to an irrigated farmland from Hermiston City area (Oregon) in USA, which was acquired in 2013 and 2014.



(a) (b) (c)

Fig. 9. River dataset. (a) Image acquired on May 3, 2013. (b) Image acquired on December 31, 2013. (c) Ground truth.



(a) (b) (c)

Fig. 10. Hermiston dataset. (a) Image acquired on 2013. (b) Image acquired on 2014. (c) Ground truth.

This dataset was obtained by the Hyperion sensor mounted on the EO-1 satellite. The spatial size of each image is 307 × 241 pixels including 154 spectral bands after eliminating noise. The main change is farmland cover.

2) Evaluation Measures: To better quantify the performance of the proposed method, we mainly used the overall accuracy (OA) and kappa coefficient (KC) [43] as metrics, precision (Pr), recall (Re), and F1-score (F1) were introduced as an auxiliary evaluation.

The metrics are defined as follows:

$$\text{OA} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FN}+\text{FP}} \quad (22)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad (23)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad (24)$$

$$F1 = \frac{2\text{PR}}{\text{P}+\text{R}} \quad (25)$$

$$\text{Kappa} = \frac{\text{OA}-p_e}{1-p_e} \quad (26)$$

$$p_e = \frac{(\text{TP}\times\text{FN})+(\text{TP}\times\text{FP})+(\text{TN}\times\text{FN})+(\text{TN}\times\text{FP})}{N^2} \quad (27)$$

where true positive (TP) indicates the number of pixels correctly classified as changed region, true negative (TN) denotes the number of pixels correctly classified as unchanged regions, false positive (FP) represents the number of pixels misclassified as changed regions, and false negative (FN) is the number of pixels misclassified as unchanged regions. TP, TN, FP, and FN pixels are shown with white, black, green, and red in visualization results. The larger value of these evaluation metrics indicates better detection performance.

B. Compared Methods and Experimental Details

To evaluate the performance of the proposed architecture, we further compared our method with other CD methods. Some classic CD algorithms were implemented for comparison, including CVA [8], PCAKM [12], IRMAD [15], and

KNN [17]. CVA is a most commonly used method, which can provide change intensity and change direction. PCAKM uses the PCA method to project the original data into a new lower dimensional feature space, and CD is achieved by partitioning the feature vector space into two clusters using k-means. IRMAD is a CD algorithm based on CCA that aims to maximize the variance of projection feature difference. KNN uses proximity to classify or predict the classification of data points. Some deep architecture algorithms were also implemented for comparison, including ReCNN [25], BCNN [24], SiamCRNN [26], and ML-EDAN [32]. ReCNN and SiamCRNN use LSTM units to find the change information extracted by CNN. BCNN finds the relationship between bitemporal feature maps by combining bilinear feature. ML-EDAN learns the discriminating features by introducing multiscale features. All the codes of the above comparison algorithms were reproduced in this article. The mean and variance obtained from ten repeated experiments were used as the experimental results, which intuitively reflects the performance and robustness of the methods.

In our network, comprehensively considering the complexity of calculation and spatial–spectral information, we chose the input patch size as 9×9. In experiments, we selected 20% from the datasets as the training samples and the rest as the testing samples. Our network was trained and tested on a NVIDIA GTX A6000 GPU with 48-GB memory using the PyTorch [44] framework. In the stage of training, we used the stochastic gradient descent (SGD) optimizer [45] with weight decay 5e−3. The initial learning rate was designed to be 5e−3 and decayed by a factor of 0.1 at every 35 epoch. The number of total epochs was 100, and the batch size was set to 32. To avoid biased estimation, we conducted ten repeated experiments and took their average values with standard deviation as the final results.

According to the original paper, for BCNNs, we chose the input patch size as 11×11 and used the SGD optimizer with weight decay 5e−3. We trained the network for 100 epochs, and the initial learning rate was designed to be 1e−4 with 0.1 times decay at every 35 epochs. For ReCNN and SiamCRNN, we used the patch size of 5 and used the SGD optimizer in training. The initial learning rate was designed to be 2e−4 with a decay of 0.1 times for every 35 epochs. The total number of epochs is 150, and the batch size was set to 64 and 32, respectively. ML-EDAN method also used the patch size of 5, and we used the Adam optimizer to train. The initial learning rate was set to 1e−4, decaying by a factor of 10 at 100 and 150 epochs. The total number of epochs is 200, and the batch size is set to 16.

C. Experimental Results

1) Experimental Results on the Farmland Dataset: Table I and Fig. 11 show the results of each model on the Farmland dataset. Compared with the traditional CVA, PCAKM method, the supervised learning methods present a better Pr and have a great improvement in terms of KCs. It indicates that the CVA and PCAKM algorithms misjudge a large number of invariant regions into changing regions, thus having a high Re but a low KC. Deep-learning-based methods are generally more satisfactory compared with CVA, PCAKM, and IRMAD, because these methods can learn more deep features through convolutional layers. After introducing the multiscale features, ML-EDAN and the proposed MSDFFN have better performance than other deep learning methods which only consider the single-scale features. This indicates that the multiscale features are conducive to the network to learn more precise features for matching different shapes of land covers. Compared with ML-EDAN, the accuracy of our MSDFFN network is further improved due to only considering the changed components to learn the subtle changing features. Compared with all the methods, the proposed MSDFFN model has the best performance in OA, KC, F1-Score, Re, and Pr metrics.

From the visual observations, our proposed MSDFFN presents the fewest FP pixels, thus achieving the best visual performance. From Fig. 11(a)–(c), the traditional CVA, PCAKM methods exhibit more misclassified pixels, with significant "salt and pepper" noise in the unchanged areas (black regions) and a large number of misclassification pixels around

TABLE I
COMPARISONS BETWEEN MSDFFN AND VARIOUS METHODS ON THE FARMLAND DATASET

Method	OA%	KC(×100)	F1%	Pr%	Re%
CVA	95.25	88.6	91.97	90.33	93.66
PCAKM	95.14	88.37	91.82	89.78	93.96
IRMAD	95.57	90.13	93.14	91.46	94.89
KNN	97.89	94.85	96.33	96.97	95.70
ReCNN	97.30±0.05	93.46±0.12	95.36±0.08	95.02±0.17	95.72±0.19
SiamCRNN	97.15±0.12	93.08±0.30	95.09±0.22	94.78±0.46	95.41±0.53
BCNN	97.95±0.04	95.02±0.12	96.47±0.08	96.42±0.25	96.51±0.28
ML-EDAN	98.62±0.07	96.66±0.16	97.63±0.12	97.52±0.31	97.74±0.30
ours	**98.71±0.03**	**96.88±0.08**	**97.78±0.06**	**97.79±0.17**	**97.77±0.10**

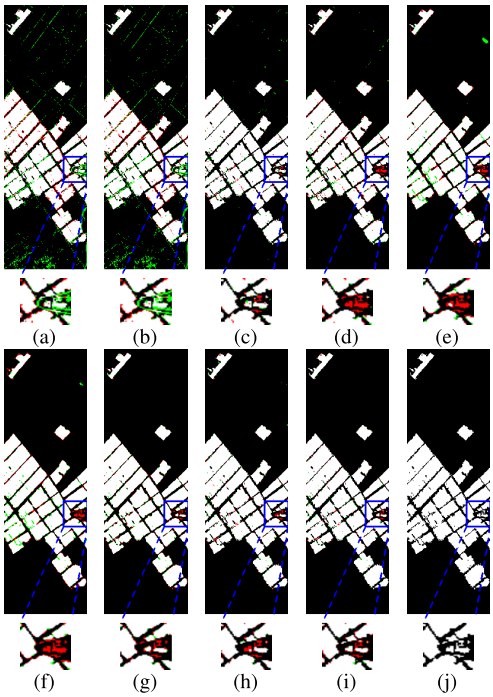


Fig. 11. Visualized results of different methods on the Farmland dataset. (a) CVA, (b) PCAKM, (c) IRMAD, (d) KNN, (e) ReCNN, (f) SiamCRNN, (g) BCNN, (h) ML-EDAN, (i) our MSDFFN, and (j) ground truth.

TABLE II
COMPARISONS BETWEEN MSDFFN AND VARIOUS METHODS ON THE RIVER DATASET

Method	OA%	KC(×100)	F1%	Pr%	Re%
CVA	92.81	66.18	69.92	54.93	96.17
PCAKM	92.72	65.91	69.69	54.60	96.29
IRMAD	94.07	62.96	66.21	65.60	66.83
KNN	94.36	61.74	64.78	70.85	59.67
ReCNN	95.96±0.23	70.56±2.19	72.66±2.09	88.17±0.57	61.86±3.12
SiamCRNN	96.50±0.14	75.59±0.89	77.45±0.83	88.14±1.96	69.12±1.51
BCNN	95.74±0.08	68.82±0.80	71.04±0.79	86.78±1.76	60.18±1.71
ML-EDAN	97.74±0.04	85.33±0.29	86.57±0.27	89.57±0.56	83.75±0.66
ours	**98.12±0.04**	**87.98±0.12**	**89.01±0.10**	**90.52±0.32**	**87.58±0.31**

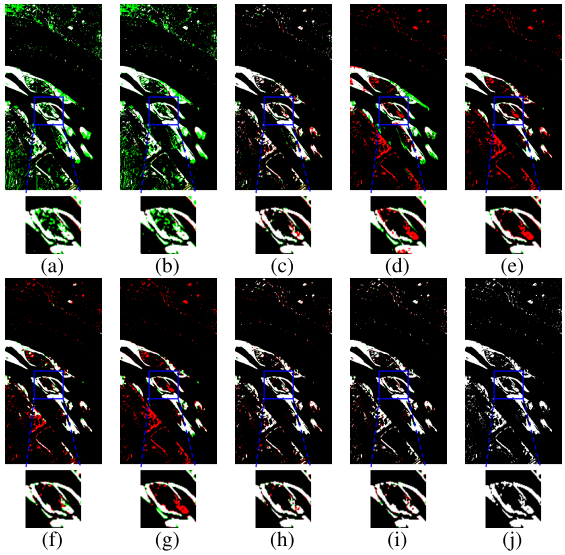


Fig. 12. Visualized results of different methods on the River dataset. (a) CVA, (b) PCAKM, (c) IRMAD, (d) KNN, (e) ReCNN, (f) SiamCRNN, (g) BCNN, (h) ML-EDAN, (i) our MSDFFN, and (j) ground truth.

the edges of the changed areas and small targets. IRMAD and KNN have less pixels of misclassification than CVA, but still have some pixels of FP in the upper right part of the image obviously. The deep learning algorithms, such as ReCNN, BCNN, and SiamCRNN, have better performance in distinguishing the unchanged pixels; however, the areas between the rice fields, as shown in the edges of the block areas of the image, still appear with some pixels of FN. MSDFFN has fewer misclassification points and better CD details than ML-EDAN, mainly shown in the middle right small target areas of the image.

2) Experimental Results on the River Dataset: Detection results of various algorithms on the River dataset are displayed in Table II. First, the unsupervised methods such as CVA and PCAKM have relatively low accuracy compared with the other methods, because it is more difficult to distinguish the changes when some features are very close to the invariant pixels without any labeled samples. For KNN, it is one of the classical supervised machine learning methods, and it achieves better results compared with the unsupervised algorithms. The accuracies of ML-EDAN and MSDFFN are higher than the single-scale methods, which also proves the effectiveness of multiscale features. Compared with ML-EDAN, the accuracies of the MSDFFN network are improved which shows that different multiscale fusion strategy has played a role.

TABLE III
COMPARISONS BETWEEN MSDFFN AND VARIOUS METHODS ON THE HERMISTON DATASET

Method	OA%	KC(×100)	F1%	Pr%	Re%
CVA	92.02	74.16	78.85	97.90	66.01
PCAKM	92.01	74.13	78.83	97.90	65.98
IRMAD	86.75	57.86	65.80	78.69	56.54
KNN	88.36	59.63	65.77	97.47	49.63
ReCNN	89.74±0.96	66.64±5.06	72.56±4.89	90.97±5.81	61.65±10.43
SiamCRNN	87.35±1.69	56.15±8.51	62.67±7.68	92.66±7.68	49.28±13.47
BCNN	96.75±0.11	90.56±0.34	92.65±0.27	94.46±0.80	90.92±0.99
ML-EDAN	97.19±0.11	91.87±0.35	93.68±0.28	94.88±0.86	92.53±1.15
ours	**97.59±0.04**	**93.06±0.12**	**94.61±0.10**	**95.55±0.32**	**93.69±0.31**

For the River dataset, as presented in Fig. 12, CVA and PCAKM have amount of pixels of FP (green regions), which means they can not distinguish positive and negative samples well. KNN has a lot of false pixels, because it just uses the shallow features with limited discriminating performance to detect the change pixels. Meanwhile, the deep learning algorithms such as ReCNN, BCNN, and SiamCRNN have many FN pixels (red regions), which means they identify many changing places as unchanging areas. Compared with ML-EDAN that also uses the multiscale strategies, the proposed MSDFFN has an advantage in distinguishing the small details.

3) Experimental Results on the Hermiston Dataset: The detection results on the Hermiston dataset are displayed in Table III. CVA and PCAKM have a bit better performance than IRMAD and KNN. This performance is inconsistent with these methods on the other two datasets. For the Hermiston dataset, the change areas, like some disks which overlap at a small part of the edge, are relatively scattered, and there are no large connected areas. The characteristics may lead to some misclassification, so the detection capability of simpler CNN structures may be marginally weaker than CVA. BCNN has higher accurate compared with ReCNN and SiamCRNN, because the combined linear features constructed by BCNN can better capture and fuse the features of the two phases than RNN and LSTM. In this scene, MSDFFN also achieves the best results than the other methods.

For the Hermiston dataset, the visual observations are presented in Fig. 13. For the detection results of CVA and PCAKM, a lap of unchanged pixels around the circular change areas are misclassified into the changes, and the classification results do not show enough details. ReCNN and SiamCRNN show a lot of FN pixels. Some large circular regions are lost in their CD results. Consistent with the performance on the above two datasets, compared with ML-EDAN, MSDFFN has fewer pixels of FN and FP and shows better CD details. MSDFFN detects some scattered target changed regions which ML-EDAN cannot detect.

D. Ablation Study

To more clearly show the effectiveness of each proposed module, we conducted the ablation experiments for each module, including RI, SLA, BDFR, and MSAF on the three datasets. Specifically, we added modules step by step and designed seven experiments. Noteworthy, the first experiment

TABLE IV
COMPARISONS ABLATION OF EACH MODULE ON THREE DATASETS

Dataset	Model	(1)	(2)	(3)	(4)	(5)	(6)	(7)
	RI	×	✓	×	✓	✓	✓	✓
	SLA	×	×	✓	✓	✓	✓	✓
	CMS	×	×	×	×	✓	✓	✓
	BDFR	×	×	×	×	×	✓	✓
	MSAF	×	×	×	×	×	×	✓
Farmland	OA	98.28±0.08	98.44±0.05	98.42±0.09	98.51±0.08	98.59±0.04	98.64±0.04	**98.71±0.03**
	KC(×100)	95.82±0.22	96.22±0.13	96.16±0.22	96.38±0.19	96.59±0.09	96.70±0.11	**96.88±0.08**
	F1	97.04±0.15	97.32±0.10	97.27±0.17	97.43±0.14	97.58±0.06	97.65±0.08	**97.78±0.06**
	Pr	96.59±0.35	96.99±0.27	97.07±0.21	97.34±0.30	97.69±0.19	97.67±0.23	**97.79±0.17**
	Re	97.50±0.17	97.66±0.43	97.48±0.30	97.50±0.26	97.47±0.15	97.63±0.25	**97.77±0.10**
River	OA	97.89±0.05	97.87±0.03	97.92±0.06	97.96±0.05	98.02±0.10	98.07±0.04	**98.12±0.04**
	KC(×100)	86.19±0.31	86.24±0.38	86.25±0.54	86.81±0.36	87.33±0.73	87.59±0.37	**87.98±0.12**
	F1	87.69±0.28	87.39±0.37	87.48±0.52	87.92±0.34	88.41±0.68	88.64±0.36	**89.01±0.10**
	Pr	88.82±0.81	90.13±1.53	**91.93±1.78**	90.70±1.12	90.11±1.19	90.99±1.19	90.52±0.32
	Re	86.61±0.74	84.87±1.96	83.52±2.20	85.34±1.32	86.81±1.70	86.46±1.71	**87.58±0.31**
Hermiston	OA	97.00±0.10	97.16±0.07	97.24±0.01	97.16±0.07	97.27±0.04	97.39±0.06	**97.59±0.04**
	KC(×100)	91.00±0.33	91.74±0.22	92.01±0.21	91.80±0.21	92.06±0.14	92.44±0.20	**93.06±0.12**
	F1	93.20±0.27	93.56±0.17	93.78±0.17	93.63±0.17	93.80±0.11	94.12±0.16	**94.61±0.10**
	Pr	95.23±0.86	95.70±0.68	95.16±0.38	94.55±0.49	**95.82±0.35**	95.62±0.30	95.55±0.32
	Re	91.27±1.11	91.52±0.60	92.45±0.50	92.74±0.60	91.89±0.44	92.67±0.42	**93.69±0.31**

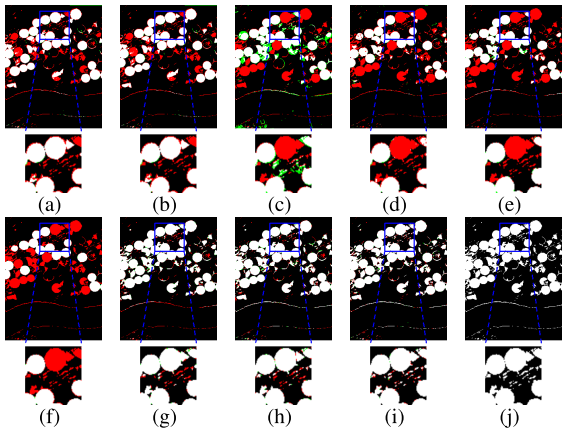


Fig. 13. Visualized results of different methods on the Hermiston dataset. (a) CVA, (b) PCAKM, (c) IRMAD, (d) KNN, (e) ReCNN, (f) SiamCRNN, (g) BCNN, (h) ML-EDAN, (i) our MSDFFN, and (j) ground truth.

is the baseline, whose structure is shown in Fig. 2. The fourth experiment is considered to only use the TFED module for CD. In the fifth experiment, the concatenated multiscale (CMS) operation was directly used to fuse the multiscale features from TFED.

It can be seen from the experimental results shown in Table IV, with the gradual addition of the proposed modules, the accuracies have been improved compared with the previous model in general, and the complete model has the highest accuracies. This proves the effectiveness of each module. For the Farmland dataset, the complete model presents optimal values in OA, Kappa, F1, Re, and Pr, respectively. For the River dataset and Hermiston dataset, the optimal values of Pr appear in "baseline + SLA" and "baseline + RI + SLA + CMS," respectively. Pr is related to the proportion of the correctly predicted samples to positive predictions. Since the proportion of positive and negative samples is not completely uniform, this may cause a high Pr. The model cannot be evaluated as good or bad by Pr alone. At this time, generally, F1 is more appropriate to evaluate the model, and F1 of our proposed complete MSDFFN is still optimal. For the River dataset, noteworthy, when only adding RI, the accuracy is slightly lower than that of the baseline. The inception module enriches the receptive field, while it may also introduce some data redundancy which causes some detection results with a slight decrease. For the Hermiston dataset, the complete model has the highest accuracies. Adding the RI module and SLA module separately can improve the accuracy, but when adding RI and SLA at the same time, there is a decline of accuracies. This may generate over learning when adding RI and SLA at the same time on the complex Hermiston dataset. In summary, although the proposed modules show side effect in a few cases for some datasets, they generally have advantage to improve the performance of CD under most conditions for all the experimental datasets.

E. Discussion

1) Discussion of the Feature Fusion Scale: In the proposed MSDFFN framework, the feature fusion scale is an inevitable parameter in our MSAF module, which is related to how to set a proper scale for feature fusion. In this article, we normalize the feature maps to the middle scale of 7, which indicates that the multiscale features are fused at the scale of 7×7. To verify the effectiveness of fusion at the middle scale, we tried to fuse the multiscale features at the scales of 9 and 5. To ensure fairness of the comparison, we used the same settings in all the experiments. The results are shown in Table V.

According to Table V, we can see that the best accuracy can be obtained at the scale of 7. When the scale is 9, it contains more features. This may introduce many additional information causing feature redundancy, which is not conducive to the subsequent detection. When the scale is 5, the window is smaller and less information is obtained. In this case, the features may be insufficient to discriminate the change

TABLE V
RESULTS OF DIFFERENT FEATURE FUSION SCALES ON THREE DATASETS

Scale		5	7	9
Farmland	OA	98.65±0.04	**98.71±0.03**	98.62±0.06
	KC(×100)	96.73±0.09	**96.88±0.08**	96.66±0.14
	F1	97.68±0.07	**97.78±0.06**	97.63±0.10
	Pr	97.52±0.25	**97.79±0.17**	97.61±0.15
	Re	97.86±0.28	**97.77±0.10**	97.65±0.24
River	OA	98.03±0.04	**98.12±0.04**	98.08±0.03
	KC(×100)	87.45±0.19	**87.98±0.12**	87.73±0.16
	F1	88.52±0.18	**89.01±0.10**	88.77±0.15
	Pr	89.77±1.44	90.52±0.32	**90.63±0.72**
	Re	87.37±1.53	**87.58±?.31**	87.00±0.59
Hermiston	OA	97.49±0.07	**97.59±0.04**	97.44±0.08
	KC(×100)	92.76±0.21	**93.06±0.12**	92.60±0.22
	F1	94.37±0.17	**94.61±0.10**	94.26±0.17
	Pr	95.50±0.51	95.55±0.32	**95.56±0.44**
	Re	93.27±0.49	**93.69±0.31**	92.99±0.36

TABLE VI
RESULTS OF THE MSAF MODULE WITH DIFFERENT ATTENTIONS ON THREE DATASETS

Attention		ECA	CBAM	ours
Farmland	OA	98.54±0.07	98.66±0.04	**98.71±0.03**
	KC(×100)	96.46±0.16	96.75±0.09	**96.88±0.08**
	F1	97.48±0.11	97.69±0.07	**97.78±0.06**
	Pr	97.49±0.24	97.82±0.16	**97.79±0.17**
	Re	97.48±0.23	97.57±0.15	**97.77±0.10**
River	OA	98.01±0.06	98.06±0.04	**98.12±0.04**
	KC(×100)	87.10±0.32	87.57±0.34	**87.98±0.12**
	F1	88.18±0.30	88.63±0.32	**89.01±0.10**
	Pr	**90.90±1.46**	90.30±0.87	90.52±0.32
	Re	85.67±1.42	87.05±1.19	**87.58±0.31**
Hermiston	OA	97.48±0.08	97.53±0.07	**97.59±0.04**
	KC(×100)	92.71±0.22	92.85±0.21	**93.06±0.12**
	F1	94.33±0.17	94.44±0.17	**94.61±0.10**
	Pr	95.65±0.44	**95.88±0.51**	95.55±0.32
	Re	93.04±0.36	93.05±0.49	**93.69±0.31**

areas, and some important details may be lost to reduce the detection accuracies. Based on the experimental results and comprehensive analysis, a suitable scale, with a larger or smaller scale both generating disadvantage for CD, is very important for the MSAF module, where we set the fusion scale as 7.

2) Application of the MSAF Module: In recent years, a lot of attention mechanisms have been used in computer vision, each with its own advantages and focus. For example, efficient channel attention (ECA) [46] introduced the adaptive 1-D convolution to replace the full connection layer, which simplifies the calculation. CBAM [39], cascading channel attention, and spatial attention use max pooling and average pooling operations to aggregate spatial and channel information. To validate the effectiveness of the attention fusion, we compared the proposed MSAF module with the other classic attention, i.e., ECA and CBAM. The results are shown in Table VI.

From the results, we can see that the proposed MSAF module can yield the best accuracies and is more suitable for the CD task based on the multiscale features. The attention mechanisms, such as ECA and CBAM which focuses on the information of channels and spaces, are directly applied to the fused feature maps. While the proposed MSAF module aims at multiscale feature fusion, considering the information sharing between different feature maps. The MSAF module shares the attention score into the three input feature maps and uses the residual connection to retain the original information. After stacking, the three feature maps will be better integrated and get more discriminating features.

TABLE VII
COMPUTATIONAL COST OF DIFFERENT METHODS ON THE FARMLAND DATASET

Methods	ReCNN	SiamCRNN	BCNNs	ML-EDAN	MSDFFN
#Params (K)	545.4	310.7	4542.8	93528.0	39452.4
Testing Times	2.15s	2.95s	3.21s	5.27s	8.97s

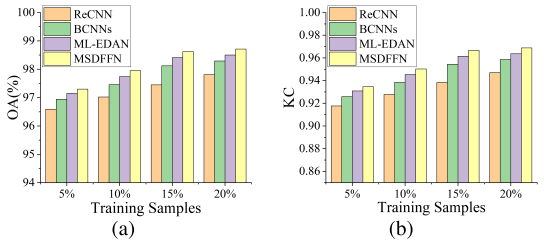


Fig. 14. CD results with different training sample sizes on the Farmland dataset. (a) OA. (b) KC.

3) Discuss of the Computational Cost: We tested the computational cost of the deep-learning-based methods. The computational cost of different methods on the Farmland dataset is shown in Table VII. The proposed method has fewer parameters than ML-EDAN and the most test time than the other methods. While MSDFFN achieves the best detection performance compared with all the compared algorithms. In further research, we will consider developing some more innovative model compression methods to reduce the computational cost and shorten the required time with guaranteed accuracy.

4) Discuss of the Training Sample Size: To comprehensively investigate the influence of the training sample size to the detection results, we tested the detection results under different training samples on the Farmland dataset with some deep learning methods. The methods included ReCNN, BCNN, and MSDFFN, and the selected training sample size was 5%, 10%, 15%, and 20%. We presented the experimental results in a histogram. The OAs and KCs are shown in Fig. 14.

With the reduced sample size, both the OAs and KCs show a downward trend in the several algorithms. This results indicate that the increased number of training samples will improve the detection results. Because more information can be used for training with the increase in training samples. Furthermore, as the training sample size decreases, the accuracies of the models with multiscale features are still higher than the models with single scale, and this shows that multiscale learning can enhance the robustness of the features. The reason is that multiscale learning has advantage to adaptively match the land covers with different shapes and extract purer features of different land covers. The proposed method still achieves the highest accuracies than the other three algorithms under different numbers of training samples, because MDFFN can

learn the subtle change features and adaptively fuse the discriminative information of different scales.

IV. Conclusion

In this article, an end-to-end framework named MSDFFN, including TFED, BDFR, and MSAF modules, was proposed to detect the changed regions of bitemporal HSIs. The proposed TFED, which combines RI and SLA, can extract rich multiscale features from the input patch pairs. The BDFR module with bidirectional representation can improve the discrimination performance of subtle changes, while MSAF adaptively fuses the features from different scales with attention mechanism. Our proposed method can better obtain and analyze the changed components and has advantages over the others in detecting small changes. The experimental results on three HSI datasets show that the proposed method can produce more accurate CD results than the other compared methods. There are also some limitations of our proposed method, which is a supervised algorithm and cannot use unlabeled samples. In the future work, a semisupervised algorithm [47] can be developed to use both labeled and unlabeled samples for further improved HSI CD.

References

[1] X. Zheng, X. Chen, X. Lu, and B. Sun, "Unsupervised change detection by cross-resolution difference learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, May 2022, Art. no. 5606616.

[2] W. Shi, M. Zhang, R. Zhang, S. Chen, and Z. Zhan, "Change detection based on artificial intelligence: State-of-the-art and challenges," *Remote Sens.*, vol. 12, no. 10, p. 1688, Jan. 2020.

[3] S. Liu, D. Marinelli, L. Bruzzone, and F. Bovolo, "A review of change detection in multitemporal hyperspectral images: Current techniques, applications, and challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 140–158, Jun. 2019.

[4] F. Luo, Z. Zou, J. Liu, and Z. Lin, "Dimensionality reduction and classification of hyperspectral image via multistructure unified discriminative embedding," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Nov. 2022, Art. no. 5517916.

[5] F. Luo, L. Zhang, X. Zhou, T. Guo, Y. Cheng, and T. Yin, "Sparse-adaptive hypergraph discriminant analysis for hyperspectral image classification," *IEEE Geosci. Remote. Sens. Lett.*, vol. 17, no. 6, pp. 1082–1086, Sep. 2020.

[6] A. Shafique, G. Cao, Z. Khan, M. Asad, and M. Aslam, "Deep learning-based change detection in remote sensing images: A review," *Remote Sens.*, vol. 14, no. 4, p. 871, Feb. 2022.

[7] M. Hansanlou and S. T. Seydi, "Hyperspectral change detection: An experimental comparative study," *Remote Sens.*, vol. 39, no. 20, pp. 7029–7083, Oct. 2018.

[8] F. Bovolo and L. Bruzzone, "A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 1, pp. 218–236, Jan. 2007.

[9] F. Thonfeld, H. Feilhauer, M. Braun, and G. Menz, "Robust change vector analysis (RCVA) for multi-sensor very high resolution optical satellite data," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 50, pp. 131–140, Aug. 2016.

[10] A. Tharwat, "Principal component analysis: An overview," *Pattern Recognit.*, vol. 3, no. 3, pp. 197–240, 2016.

[11] F. Luo, L. Zhang, B. Du, and L. Zhang, "Dimensionality reduction with enhanced hybrid-graph discriminant learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5336–5353, Aug. 2020.

[12] T. Celik, "Unsupervised change detection in satellite images using principal component analysis and k-means clustering," *IEEE Geosci. Remote. Sens. Lett.*, vol. 6, no. 4, pp. 772–776, 2009.

[13] A. A. Nielsen, K. Conradsen, and J. J. Simpson, "Multivariate alteration detection (MAD) and MAF postprocessing in multispectral, bitemporal image data: New approaches to change detection studies," *Remote Sens. Environ.*, vol. 64, no. 1, pp. 1–19, Apr. 1998.

[14] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.

[15] A. A. Nielsen, "The regularized iteratively reweighted mad method for change detection in multi-and hyperspectral data," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 463–478, Feb. 2007.

[16] C. Wu, B. Du, and L. Zhang, "Slow feature analysis for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2858–2874, May 2014.

[17] T. Abeywickrama, M. A. Cheema, and D. Taniar, "K-nearest neighbors on road networks: A journey in experimentation and in-memory implementation," *Proc. VLDB Endowment*, vol. 9, no. 6, pp. 492–503, Jan. 2016.

[18] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 565–1573, Sep. 1995.

[19] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1687–5265, Feb. 2018.

[20] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised deep change vector analysis for multiple-change detection in VHR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3677–3693, Jun. 2019.

[21] B. Du, L. Ru, C. Wu, and L. Zhang, "Unsupervised deep slow feature analysis for change detection in multi-temporal remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9976–9992, Dec. 2019.

[22] L. Khelifi and M. Mignotte, "Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis," *IEEE Access*, vol. 8, pp. 126385–126400, 2020.

[23] Q. Wang, Z. Yuan, Q. Du, and X. Li, "GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 3–13, Jan. 2019.

[24] Y. Lin, S. Li, L. Fang, and P. Ghamisi, "Multispectral change detection with bilinear convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 10, pp. 1757–1761, Oct. 2020.

[25] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning spectral-spatial–temporal features via a recurrent convolutional neural network for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 924–935, Feb. 2019.

[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, Jul. 2020.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] X. Zheng, B. Wang, X. Du, and X. Lu, "Mutual attention inception network for remote sensing visual question answering," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[29] M. S. Moustafa, S. A. Mohamed, S. Ahmed, and A. H. Nasr, "Hyperspectral change detection based on modification of UNet neural networks," *J. Appl. Remote Sens.*, vol. 15, no. 2, Jun. 2021, Art. no. 028505.

[30] 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, 2021.

[31] H. Jiang, X. Hu, K. Li, J. Zhang, J. Gong, and M. Zhang, "PGA-SiamNet: Pyramid feature-based attention-guided Siamese network for remote sensing orthoimagery building change detection," *Remote Sens.*, vol. 12, no. 3, p. 484, Feb. 2020.

[32] J. Qu, S. Hou, W. Dong, Y. Li, and W. Xie, "A multilevel encoder–decoder attention network for change detection in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Nov. 2022, Art. no. 5518113.

[33] M. Gong et al., "A spectral and spatial attention network for change detection in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Dec. 2022, Art. no. 5521614.

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

[35] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 202–211.

[36] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[38] G. Brauwers and F. Frasincar, "A general survey on attention mechanisms in deep learning," *IEEE Trans. Knowl. Data Eng.*, early access, Nov. 9, 2021, doi: 10.1109/TKDE.2021.3126456.
[39] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2018, pp. 3–19.
[40] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications*. Cham, Switzerland: Springer, 2016, pp. 179–187.
[41] 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.
[42] J. Hu, L. Shen, and G. Sun, "Squeeze- and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7132–7141.
[43] G. H. Rosenfield and K. Fitzpatrick-Lins, "A coefficient of agreement as a measure of thematic classification accuracy," *Photogram. Eng. Remote Sens.*, vol. 52, no. 2, pp. 223–227, 1986.
[44] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 32, 2019, pp. 1–12.
[45] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 421–436.
[46] 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. (CVPR)*, Jun. 2020, pp. 11531–11539.
[47] F. Luo, H. Huang, Z. Ma, and J. Liu, "Semisupervised sparse manifold discriminative analysis for feature extraction of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6197–6211, Oct. 2016.

Fulin Luo (Senior Member, IEEE) received the B.S. degree in mechanical engineering and automation from Southwest Petroleum University, Chengdu, China, in 2011, and the M.S. and Ph.D. degrees in instrument science and technology from Chongqing University, Chongqing, China, in 2013 and 2016, respectively.

He was an Associate Researcher and a Post-Doctoral Researcher with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, Wuhan, China, from 2017 to 2021. He was a Research Fellow with Nanyang Technological University, Singapore, from 2020 to 2021. He has been a Professor with the College of Computer Science, Chongqing University, since 2021. His research interests include remote sensing processing, computer vision, and biomedical analysis.

Dr. Luo received a fellowship from the National Post-Doctoral Program for Innovative Talents in 2017.

Tianyuan Zhou received the B.S. degree in electronic information engineering from the University of Shanghai for Science and Technology, Shanghai, China, in 2021. She is currently pursuing the M.S. degree in instrument science and technology with Chongqing University, Chongqing, China.

Her research interests include hyperspectral image change detection, remote sensing image processing, and machine learning.

Jiamin Liu received the M.S. and Ph.D. degrees in instrument science and technology from Chongqing University, Chongqing, China, in 1998 and 2001, respectively.

He is currently an Associate Professor with Chongqing University. His research interests include biometrics, image processing, and pattern recognition in general.

Tan Guo (Member, IEEE) received the M.S. degree in signal and information processing and the Ph.D. degree in communication and information systems from Chongqing University, Chongqing, China, in 2014 and 2017, respectively.

Since 2018, he has been with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing. His research interests include computer vision, pattern recognition, and machine learning.

Xiuwen Gong received the B.E. and M.E. degrees from Anhui Normal University, Wuhu, China, in 2011 and 2014, respectively. She is currently pursuing the Ph.D. degree in artificial intelligence with the Faculty of Engineering, The University of Sydney, Sydney, NSW, Australia.

Her research interests include machine learning, deep learning, and AI.

Jinchang Ren (Senior Member, IEEE) received the B.E., M.E., and D.E. degrees from Northwestern Polytechnical University, Xi'an, China, in 1992, 1997, and 2000, respectively, and the Ph.D. degree from the University of Bradford, Bradford, U.K., in 2019.

He is currently a Professor with the National Subsea Centre, Robert Gordon University, Aberdeen, U.K. His research interests include image processing, computer vision, machine learning, and big data analytics.