Multiorder Graph Convolutional Network With Channel Attention for Hyperspectral Change Detection

Yuxiang Zhang, *Member, IEEE*, Rui Miao, Yanni Dong, *Senior Member, IEEE*, and Bo Du, *Senior Member, IEEE*

***Abstract*—Hyperspectral change detection (CD) aims to obtain the change information of objects in the multitemporal hyperspectral images (HSIs). Recently, with the advantages in fully extracting the image features of irregular areas, the graph convolutional network (GCN) has attracted increasing attention for hyperspectral CD. The existing GCN-based CD methods usually use a graph structure constructed by superpixels to reduce the computational cost, which ignores the multiorder difference information among graph nodes and the local difference information within superpixels. To address these problems, this article proposes an efficient multiorder GCN with a channel attention module (CAM) for hyperspectral CD. Specifically, the multiorder GCN module is designed by repeatedly mixing the feature representations of neighborhoods. The CAM is then proposed to enhance the difference features of bitemporal HSIs. After that, the pixel-wise CD is accomplished by a lightweight feature fusion module and a fully connected layer. Experiments on three hyperspectral datasets illustrated the effectiveness of the proposed algorithm.**

***Index Terms*—Change detection (CD), graph convolutional network (GCN), hyperspectral images (HSI).**

I. Introduction

With the rapid development of remote sensing techniques, remote sensing images have been widely used in Earth observation applications. Change detection (CD) is one of the fundamental tasks for understanding remote sensing images, which aims to obtain the change information of objects in the multitemporal remote sensing images [1], [2]. With high spectral resolution, hyperspectral image (HSI) can provide rich spectral information for CD [3]. Abundant spectral information can comprehensively reflect the composition of different objects, which can promote the detection of subtle changes of objects, thereby providing the probability of capturing fine changes of objects in the multitemporal HSIs [4], [5], [6]. HSI CD achieves significant application in reality scenarios, such as land cover mapping, agriculture and forestry detection, and resource exploration [7], [8], [9].

During the past decades, many traditional CD methods have been proposed, which can be roughly divided into the following four categories: image algebra; image transformation; image classification; and other traditional methods. The image algebra-based methods use the original information of the pixels to detect changes, such as the most common change vector analysis (CVA) [10]. The image transformation-based methods transform original HSI to explore the change information. The widely used methods include multivariate alteration detection (MAD) [11], iteratively reweighted MAD [12], and slow feature analysis [13]. The image classification-based methods distinguish the differences between the classified images to detect changes or directly utilize a classifier to detect changes [14]. Support vector machine (SVM) classifier [15] is a baseline method for CD. Many improved classifiers are also used in CD, including the multiclassifier framework [16]. Except these classic CD methods, researchers have introduced some basic techniques to CD [17]. For example, the statistical hypothesis is utilized to mine the change attributes [18], [19]. The distance metric is introduced to detect the forest cover changes [20]. What is more, the clustering and unmixing methods are also used in CD task [21], [22], [23], [24], [25]. The aforementioned traditional CD methods are generally simple, focusing on shallow features to identify change information.

With the powerful ability of exploiting features, deep learning architectures have attracted increasing attention in CD domain [26], [27], [28]. Many HSI CD methods are proposed based on convolution neural networks (CNNs). The primary deep learning-based CD method is based on U-Net [29] and Siamese network [30], which has the ability to capture the small change object. Besides, based on 2-D CNN, Wang et al. [31] applied the mixed-affinity matrix with subpixel representation to fuse multisource information of pixel level and subpixel level. Lin et al. [32] detected changes by bilinear CNNs (BCNNs), which can sufficiently capture the relationship between multitemporal images. Zhan et al. [33] accomplished CD task by a Siamese network, which can extract the joint spatial–spectral features.

Manuscript received 29 August 2023; revised 14 October 2023 and 20 November 2023; accepted 30 November 2023. Date of publication 5 December 2023; date of current version 15 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62071438, Grant 62222116, Grant 62225113, and Grant 62171417. *(Corresponding author: Yanni Dong.)*

Yuxiang Zhang and Rui Miao are with the School of Geophysics and Geomatics, China University of Geosciences, Wuhan 430074, China (e-mail: zhangyx@cug.edu.cn; miaorui797979@126.com).

Yanni Dong is with the School of Resource and Environmental Sciences, Wuhan University, Wuhan 430079, China (e-mail: dongyanni@whu.edu.cn).

Bo Du is with the School of Computer, Wuhan University, Wuhan 430079, China (e-mail: gunspace@163.com).

Digital Object Identifier 10.1109/JSTARS.2023.3339238

Song et al. [34] designed a new Siamese network structure with cross-temporal interaction symmetric attention module to extract the joint spatial–spectral–temporal features. Based on 3-D CNN, an adjacent-level feature fusion network [35] has been proposed to effectively extract and fuse deep features from bi-temporal images. Mou et al. [36] applied CNN and RNN to respectively learn the spectral–spatial features and the temporal features in multitemporal images. To better utilize temporal information, Wang et al. [37] used a CNN backbone to extract high-level semantic information and combined it with a visual transformer to extract change features. These CNN-based CD methods perform each patch convolution with fixed size and identical weights to extract spectral–spatial features [38], [39], [40].

Currently, graph convolution networks (GCNs) have been paid growing attention in the image processing field due to their ability of extending patch convolution to irregular data structure [41], [42], [43], [44], [45]. Compared to CNNs, GCNs can flexibly aggregate the adjacent nodes in the graph and effectively extract the spectral–spatial feature of irregular areas. In this way, GCNs can preserve the complex structural properties of the images and the dynamic boundaries of objects [46], [47]. Therefore, some researchers have made efforts to employ GCN for HSI CD. The first successful work is the dual-branch differential amplification graph convolutional network (D^2AGCN) [48]. Based on the benchmark dual-branch structure, the superpixel graph structure is constructed for each branch, the GCN is employed to extract image features from each branch and fuse the output features of two branches, and the differential amplification module is embedded to highlight the difference features between the graph nodes of two branches. However, this CD method has some limitations, which are as follows:

1) it applies the adjacency matrix to store the edge relationship between any two graph nodes and applies the GCN to fuse the outputs of two branches, which leads to a relatively high computational complexity due to the large number of stored graph edges and GCN layers;
2) it utilizes the vanilla GCN to realize the immediate feature aggregation in the neighborhood of graph nodes while ignoring sufficient feature aggregation of multiorder neighborhood information of graph nodes, which may weaken the performance of feature extraction;
3) it employs superpixel-wise CD to predict the probability of changes superpixel by superpixel, which ignores the local change difference within superpixels and weakens the probability of detecting subtle changes of objects.

To addressing the aforementioned limitations, we design a multiorder GCN with channel attention (MGCN) for HSI CD. The overall architecture includes three aspects: graph construction and store, feature extraction, and CD. First, it takes the advantage of superpixel segmentation to construct sparse graph structures. In addition, it introduces the superpixel sparse graph store technique to reduce the computational complexity by omitting the edge relationship between nonadjacent superpixel nodes out of the adjacency matrix. Then, inspired by [49], it constructs a dual-branch feature extraction framework composed of multiorder GCN and channel attention modules (CAMs) to extract sufficient features of multitemporal HSIs for CD. The multiorder GCN is designed by repeatedly aggregating the feature representations of neighborhood through graph nodes at different orders, which can learn a wider feature aggregation of graph nodes. The CAM is embedded in the dual-branch feature extraction framework to emphasize the difference features between two branches from channel dimension rather than graph nodes perspective. Finally, we design a pixel-wise CD method via feature fusion with association matrix and convolution layers rather than superpixel-wise CD. The computational complexity is further reduced by the lightweight feature fusion module via convolution layers rather than GCN. An association matrix is used to transform the superpixel level features of bitemporal HSIs to pixel level features. The convolutional layers are used to extract the correlation features of bitemporal HSIs, which allows the pixel level feature differences within one superpixel. The final CD result is obtained by predicting the probability of changes pixel by pixel. By considering the local difference information within one superpixel, the pixel-wise CD can extract the local change difference within superpixels and detect subtle changes of objects.

Our main contributions are summarized as follows.

1) A dual-branch feature extraction framework composed of multiorder GCN and CAMs is proposed, which can learn a wider feature aggregation for each branch and extract difference features between two branches from channel dimension.
2) The pixel-wise CD method is proposed via feature fusion with association matrix and convolution layers, which can extract the local change difference within superpixels and detect subtle changes of objects.
3) An efficient HSI CD network architecture MGCN is designed by utilizing the superpixel sparse graph store technique and the lightweight feature fusion module, which can perform well with relatively low computational complexity. The training time on RTX 3090 can be reduced from about an hour in D^2AGCN to one minute in MGCN.

The rest of this article is organized as follows. The related work is analyzed in Section II. Details of the proposed method are described in Section III. Experimental results and discussions are presented in Section IV. Finally, Section V concludes this article.

II. Related Work

The GCN was initially proposed by Bruna et al. [50] to handle data with irregular structures. This network extends the convolution process of fixed patch kernel to the topological graph structure. It utilizes the Laplace matrix to perform the convolution process of the graph. Due to the high computational complexity of Laplace matrix eigenvalue decomposition, this network shows good performance in calculating small-scale graphs and is usually not suitable for the large-scale graphs. To reduce the computational complexity, many works have been done, including small-scale graph construction, Polynomial expansion to approximate graph convolution, and Chebyshev polynomial with iterative convolution kernel [51], [52]. Recently,

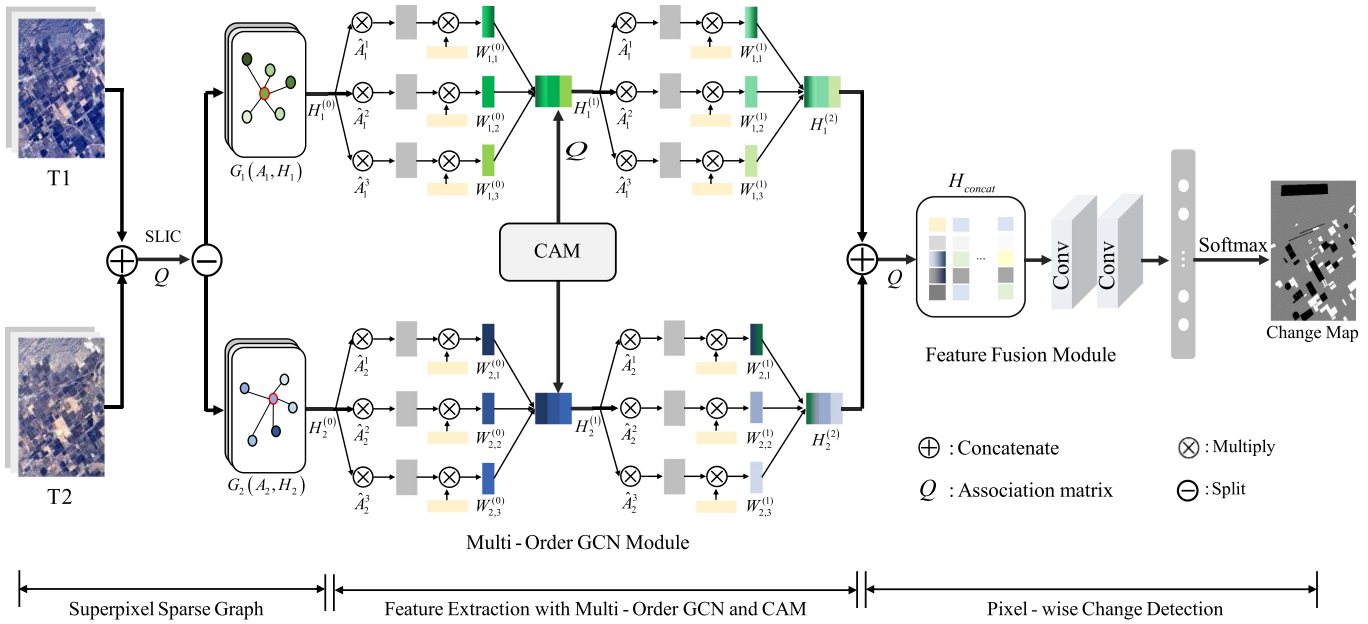

Fig. 1. Overall architecture of the proposed method.

many works have paid attention on representing the images as graph structures to extract the consistent image features with the topological structures in real scene [53], [54], [55]. Based on this, the GCN has been creatively applied to the field of image processing to obtain the graph representation of images.

In the HSI CD field, HSIs contain a large number of pixels. Therefore, it is not appropriate to perform convolution on a graph structure at pixel level. D^{2}AGCN constructs the graph structures at superpixel level by performing superpixel segmentation. As a typical segmentation algorithm, simple linear iterative clustering (SLIC) [56] is used to achieve the superpixel graph construction. Through SLIC, some changed pixels may be segmented into different superpixels, which will lead to the decrease of CD accuracy. To alleviate the problem, the bitemporal HSIs are cascaded along the channel dimension before the superpixel segmentation process. Moreover, in order to extract sufficient features, the superpixel graph structures of bitemporal HSIs are, respectively, sent to dual-branch network, where each branch is composed of two GCN layers. Behind each GCN layer, a difference amplification module is embedded into the dual-branch network to highlight the difference features between the graph nodes of two branches. In detail, for each node, the cosine similarity (CS) between output feature matrices of two branches is measured. Based on the CS measurement results, the difference coefficient for graph nodes were designed via the exponential function $\exp(\alpha(1-CS))-1$. It introduces a parameter α to control the difference coefficients to suppress the features of unchanged nodes and highlight the features of changed nodes in the dual-branch GCN network. Finally, the outputs of two branches are cascaded and fused by a GCN layer at the superpixel level. The superpixel-wise CD result is obtained based on the learned feature of superpixel level by a fully connected layer and a softmax nonlinear activation function. The final CD is obtained via the upsampling of the superpixel-wise CD result, where the predictions of pixels within superpixel remain the same.

The GCN in the HSI CD field is still in its infancy and many problems remain to be further studied. First, although the superpixel graph structure is used in D^{2}AGCN, the computational complexity of the whole architecture is still high. Second, the feature extraction deserves more attention. D^{2}AGCN introduces vanilla GCN to realize the immediate aggregation of image features in the neighborhood of graph nodes, which does not fully extract the features of the graph. Finally, due to the loss of local difference information within superpixels, the superpixel-wise CD ignores the local change difference within superpixels and weakens the probability of detecting subtle change of objects. The aim of the MGCN is to fully explore the feature aggregation of graph nodes and the local difference information within superpixels for CD with reduced computational complexity.

III. Methodology

The overall architecture of the proposed method is shown in Fig. 1. We implement the CD task through the following three main parts:

1) superpixel sparse graph construction and store;
2) feature extraction with multiorder GCN and CAMs;
3) pixel-wise CD.

The superpixel sparse graph is constructed for each temporal HSI and the superpixel sparse graph store technique is introduced to store the feature matrix and adjacency matrix. Then, in order to extract sufficient superpixel level features, the superpixel graph structures of bitemporal HSIs are, respectively, sent to the dual-branch network, where each branch is composed of two multiorder GCNs. Besides, we embed a CAM based

on the Sigmoid function between two multiorder GCN layers to adaptively emphasize the difference features between two branches from channel dimension. Finally, by considering the local difference information within superpixels, the superpixel level features are fused and transformed to pixel level features to obtain pixel-wise CD.

A. Superpixel Sparse Graph Construction and Store

The bitemporal HSIs are first cascaded and segmented into a series of superpixels by SLIC. After that, the cascaded segmentation results are splitted for bitemporal HSIs. And the superpixel sparse graph structure for each single temporal HSI is constructed based on the feature matrix and the sparse adjacency matrix, which is stored in the coordinate format to reduce the computational cost. Based on the segmentation results, an association matrix is calculated in order to restore the features from superpixel level back to original pixel level during the following pixel-wise change detection part.

We obtain a cascaded HSI cube $\boldsymbol{X}_{2d_0}$ with $N = w \times h$ pixels and $2 \times d_0$ spectral channels, where w, h, and d_0 respectively represent the image width, the image height, and the channel dimension. The segmentation results can be defined as

$$\begin{aligned} \text{Seg} &= \{p_1, p_i, \ldots, p_N\} \\ p_i &\in \{1, 2, \cdots, M\} \end{aligned} \tag{1}$$

where M is the total number of superpixels, $\{1, 2, \ldots, M\}$ represents the index of superpixels, the value of p_i denotes the index of superpixel the ith pixel belonging to.

With the segmentation results and the HSI cube, an association matrix $\boldsymbol{Q} \in \mathbb{R}^{N \times M}$ [57] representing the relationship between pixels and superpixels can be defined as follows:

$$\boldsymbol{Q}_{i,j} = \begin{cases} 1, & \text{if } \ p_i \ = j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\boldsymbol{Q}_{i,j}$ denotes whether the ith pixel belongs to the jth superpixel. In each column of the association matrix, value 1 will be assigned to pixels in the same superpixel.

For each single temporal HSI $\boldsymbol{X}_{d_0}$, the graph $\boldsymbol{G}$ with M nodes has a feature matrix $\boldsymbol{H} \in \mathbb{R}^{M \times d_0}$ with d features per node and an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{M \times M}$. A nonzero entry $\boldsymbol{A}_{i,j}$ indicates an edge between the ith and jth superpixel nodes. To make use of the spectral information of bitemporal HSIs, the feature value of a graph node is obtained by the mean spectral value of pixels contained in a superpixel. The feature matrix $\boldsymbol{H}$ can be defined as follows:

$$\bar{\boldsymbol{H}}_i = \frac{1}{K_i} \sum_{j=1}^{K_i} \boldsymbol{x}_{i,j} \tag{3}$$

where $\bar{\boldsymbol{H}}_i (i = 1, \ldots, M)$ denotes the ith row of $\boldsymbol{H}$ demonstrating the features of the ith graph node, K_i is the number of pixels contained in the ith superpixel, and $\boldsymbol{x}_{i,j}$ is the jth pixel in the ith superpixel.

And then, the adjacency matrix $\boldsymbol{A}$ can be calculated by the feature matrix as follows:

$$\boldsymbol{A}_{i,j} = \begin{cases} \exp^{-\frac{\|\bar{\boldsymbol{H}}_i - \bar{\boldsymbol{H}}_j\|^2}{\sigma^2}}, & \text{if two nodes are adjacent} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where σ is empirically set to 0.1 [44]. A nonzero entry $\boldsymbol{A}_{i,j}$ denotes the weight of the edge between two adjacent superpixel nodes, and a zero entry $\boldsymbol{A}_{i,j}$ indicates no edge between the ith and jth superpixel nodes.

It can be noticed that, there are some zero entries in the adjacency matrixes since some superpixels are not adjacent. Therefore, we construct a sparse graph with the aforementioned adjacency matrix and feature matrix. To further reduce the computational complexity of the graph convolution process in the following section, the sparse adjacency matrix is stored in the coordinate format [58], recording the nonzero values and their indexes in the matrix. For the case of sparse graph convolution, applying the same store transformation for $\boldsymbol{H}$. Thus, a superpixel sparse graph structure can be defined as

$$\boldsymbol{G}_{\text{sparse}}(\boldsymbol{A}, \boldsymbol{H}) = f_{\text{coo\_matrix}}(\boldsymbol{G}(\boldsymbol{A}, \boldsymbol{H})) \tag{5}$$

where $f_{\text{coo\_matrix}}$ denotes the coo_matrix function in python, which is employed to achieve the transformation of matrix store format. Therefore, the superpixel sparse graph structure is stored with nonzero values and their indexes in adjacency matrix and feature matrix. With this superpixel sparse graph store technique, zero entries in the adjacency matrix indicating no edge between two superpixel nodes are omitted.

B. Feature Extraction With Multiorder GCN and CAMs

Based on the constructed superpixel sparse graph structures of two branches $\boldsymbol{G}_1(\boldsymbol{A}_1, \boldsymbol{H}_1)$ and $\boldsymbol{G}_2(\boldsymbol{A}_2, \boldsymbol{H}_2)$, we perform feature extraction with multiorder GCN and CAMs for bitemporal HSIs. On the one hand, the multiorder GCN is introduced to aggregate the feature representations in the neighborhood of graph nodes at multiple orders. On the other hand, a CAM is designed for the dual-branch structure to automatically select and emphasize the difference features.

1) Multiorder GCN Module: The feature aggregation in the neighborhood of graph node is of vital importance for CD. To improve the feature aggregation, we mix the feature representations from the immediate (first-order) neighborhood of graph node and from its further P-order neighborhood in the GCN. Thus, a multiorder GCN module is designed to mix the feature representations of neighborhoods at different orders. The multiorder GCN module is composed of two multiorder graph convolutional (GC) layers on each branch. The first multiorder GC layer on each branch outputs the concatenation features of mixing orders of the adjacency matrix, and the second multiorder GC layer on each branch can further extract the features of mixed adjacency orders. Thus, the multiorder GCN module can fully capture the features in the neighborhoods of graph node by aggregating the feature representations of multiple adjacency orders. However, with more multiorder GC layers, the information of the entire graph may be aggregated to each node, the differentiation between nodes becomes less pronounced, and

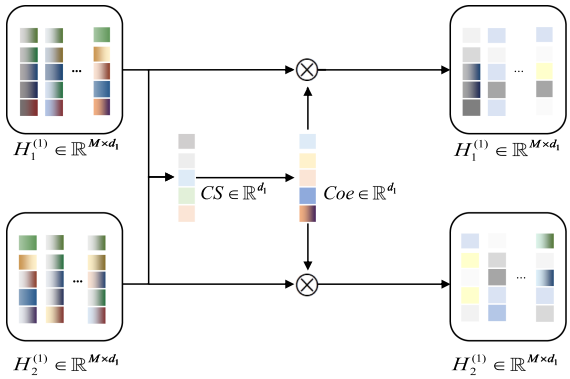

Fig. 2. Illustration of the CAM.

even the local information of the original nodes will be lost. This oversmoothing effect in turn decreases the performance of CD and even leads to worse performance compared to that with one multiorder GC layer. Each multiorder GC layer is composed of three vanilla GC layers, which is defined as

$$\boldsymbol{H}_q^{(l)} = \Big\|_{j\in P}\sigma\left(\hat{\boldsymbol{A}}_q^j \boldsymbol{H}_q^{(l-1)} \boldsymbol{W}_{q,j}^{(l-1)}\right) \tag{6}$$

where $P = \{1, 2, 3\}$ is a set of adjacency orders. Note that setting $P = \{1\}$ exactly recovers the original GC layer. $\|(\cdot)$ and $\sigma(\cdot)$ represent column-wise concatenation and activation function ReLU, respectively. $\boldsymbol{H}_q^{(l-1)} \in \mathbb{R}^{M\times d_{l-1}}$ and $\boldsymbol{H}_q^{(l)} \in \mathbb{R}^{M\times d_l} (l = 1, 2;\ q = 1, 2)$ denote the input and output for the lth multiorder GC layer on the qth branch, with $\boldsymbol{H}_1^{(0)} = \boldsymbol{H}_1$ and $\boldsymbol{H}_2^{(0)} = \boldsymbol{H}_2$. $\boldsymbol{W}_{q,j}^{(l-1)}$ is the weight matrix for the jth vanilla GC layer in the lth multiorder GC layer on the qth branch. $\hat{\boldsymbol{A}}_q^j$ denotes the adjacency matrix $\hat{\boldsymbol{A}}_q$ multiplied by itself j times. $\hat{\boldsymbol{A}}_q$ is a symmetrically normalized adjacency matrix with self-connections for the qth branch, which can be defined as

$$\hat{\boldsymbol{A}}_q = \boldsymbol{D}_q^{-\frac{1}{2}} (\boldsymbol{A}_q + \boldsymbol{I}) \boldsymbol{D}_q^{-\frac{1}{2}} \tag{7}$$

where $(\boldsymbol{D}_q)_{i,i} = \sum_j (\boldsymbol{A}_q)_{i,j}$.

2) Channel Attention Module (CAM):: The multiorder features are simply integrated by a concatenation operation for each branch. To better extract the difference information from multitemporal HSIs on different branches for CD, we embed a CAM between two multiorder GC layers to automatically select and emphasize the difference features between two branches. The illustration of the proposed CAM is shown in Fig. 2. The cosine similarity (CS) is used to measure the difference between the output feature matrices of the first multiorder GC layers on two branches. The output feature matrices are updated by multiplying with different weight coefficient (*Coe*) via the Sigmoid function and the CS results. The weights vary across different feature channels. The updated feature matrices are fed into the second multiorder GC layers for two branches.

The cosine similarity and the weight coefficient can be obtained by

$$\mathrm{CS}_i\left(\boldsymbol{H}_1^{(1)}, \boldsymbol{H}_2^{(1)}\right) = \frac{\left\langle \left(\boldsymbol{H}_1^{(1)}\right)_i, \left(\boldsymbol{H}_2^{(1)}\right)_i \right\rangle}{\left\| \left(\boldsymbol{H}_1^{(1)}\right)_i \right\| \left\| \left(\boldsymbol{H}_2^{(1)}\right)_i \right\|} \tag{8}$$

$$\mathrm{Coe}_i = \mathrm{Sigmoid}\left(1 - \mathrm{CS}_i\left(\boldsymbol{H}_1^{(1)}, \boldsymbol{H}_2^{(1)}\right)\right) \tag{9}$$

where $\mathrm{CS} \in \mathbb{R}^{d_1}$, $\boldsymbol{H}_q^{(1)} \in \mathbb{R}^{M\times d_1}$ denotes the output of the first multiorder GC layer on the qth branch, and $(\cdot)_i$ denotes the ith feature channel of $\boldsymbol{H}_q^{(1)}$. Coe_i denotes the weight coefficient assigned for the ith feature channel of $\boldsymbol{H}_q^{(1)}$.

Thus, the weight coefficient of the proposed CAM can be automatically calculated. The difference features with smaller similarities will be assigned larger weight coefficients. After that, the output feature matrix of the first multiorder GC layer on each branch is multiplied by the weight coefficient and input into the second multiorder

GC layer. The updated output feature matrix combined with channel attention can be expressed as

$$\left(\boldsymbol{H}_q^{(l)}\right)_i = \left(\boldsymbol{H}_q^{(l)}\right)_i \mathrm{Coe}_i. \tag{10}$$

C. Pixel-Wise CD

In this section, we describe how to perform pixel-wise change detection via assigning different probability of changes for pixels within each superpixel. The feature fusion module is designed to do the feature interaction between multitemporal images. Specifically, the obtained output feature matrices on two branches are cascaded and transformed from superpixel level to pixel level by the association matrix. Then, two convolution layers are utilized to capture the correlation information of two branches, which can allow local difference for the pixels within each superpixel. Finally, pixel-wise CD is accomplished by a fully connected layer and a softmax activation function, which can learn the underlying information of the obtained features to realize the judgment of the changed and unchanged areas. Therefore, the pixel-wise CD consists of an association matrix, two convolution layers, a fully connected layer, and a softmax nonlinear activation function. The feature fusion module is composed of an association matrix and two convolution layers.

The association matrix defined in Section III-A is used to transform the feature matrices on two branches from superpixel level to pixel level, the mathematical expression can be defined as

$$\boldsymbol{H}_{\mathrm{concat}} = \boldsymbol{Q} \otimes f_{\mathrm{concat}}\left(\boldsymbol{H}_1^{(2)}, \boldsymbol{H}_2^{(2)}\right) \tag{11}$$

where $\otimes$ denotes the matrix multiplication, and $f_{\mathrm{concat}}(\cdot)$ denotes the concatenation operation for the outputs of the feature extraction framework on two branches.

It can be noticed that all the pixels within a superpixel are assigned a same feature value with the association matrix. Then, these features of pixel level after two convolution layers will be learned to produce local difference for pixels within superpixels.

The output values of the softmax function are in the range of [0, [1]] with the sum of 1, which can be, therefore, regarded as the probability of the network to determine whether a pixel changes or not. Since the CD task is essentially a binary classification problem, the commonly used cross-entropy loss function is used to train the network, which is defined as

$$L\left(\hat{Y}, Y\right) = -\frac{1}{T}\sum_{t=1}^{T}\left\{Y_t \log\left(\hat{Y}_t\right) + (1 - Y_t)\log\left(1 - \hat{Y}_t\right)\right\} \tag{12}$$

where T denote the number of training samples, and Y_t and $\hat{Y}_t$, respectively, represents the truth label and the predict label of the sample t. L is used to measure the difference between Y_t and $\hat{Y}_t$, which should be reduced during the network training process. Due to the use of training samples to train the network, the proposed method is a supervised learning approach.

D. Computational Complexity

This section analyzes the computational complexity of the proposed method MGCN. The key of the MGCN is that we use two multiorder GC layers on each branch and two convolutional layers to fuse the two branches based on superpixel sparse graph. The adjacency matrix $\hat{\boldsymbol{A}}_q$ is stored as a sparse matrix with V_q nonzero entries. For (6), we multiply $\hat{\boldsymbol{A}}_q^j \boldsymbol{H}_q^{(l-1)}$ from right-to-left without computing $\hat{\boldsymbol{A}}_q^j$. Therefore, the computational complexity of two multiorder GC layers on the qth branch is calculated as $\mathrm{O}(\sum_{l=1}^{2} 3 \times V_q \times d_{l-1})$, where d_{l-1} denotes the channel dimension of the input features in the lth multiorder GC layers. For case of comparison, we pay attention to the trends of the computational complexity of feature fusion via two convolution layers, which can be represented as $\mathrm{O}(N \times d_2)$, where N denotes the pixel number and d_2 denotes the channel dimension of the input features. We express the overall computational complexity of MGCN as $\mathrm{O}(\sum_{q=1}^{2}\sum_{l=1}^{2} 3 \times V_q \times d_{l-1} + N \times d_2)$. In contrast, the key of D^{2}AGCN is that there are two GC layers on each branch and one GC layers to fuse the two branches. The overall computational complexity of D^{2}AGCN is $\mathrm{O}(\sum_{q=1}^{2}\sum_{l=1}^{2} M^2 \times d_{l-1} + M^2 \times d_2)$, where M^2 denotes the total entry of the adjacency matrix or the total number of stored graph edge. Under the realistic assumptions of $V_q \ll N < M^2$, MGCN greatly reduces the computational complexity by reducing the number of stored graph edges and replacing the GCN with the convolutional layers in feature fusion stage. Take the China dataset as an example, the pixel number N and the graph node number M for each branch is 58 800 and 12 801, respectively. With the superpixel sparse graph store technique, the number of stored graph edges can be reduced from 12 801 $\times$ 12 801 to 12 827 for the first branch and to 13 949 for the second branch.

IV. Experiments

A. Datasets and Experiment Setup

In this work, three hyperspectral datasets [59], [60] are applied to evaluate the performance of the proposed method.

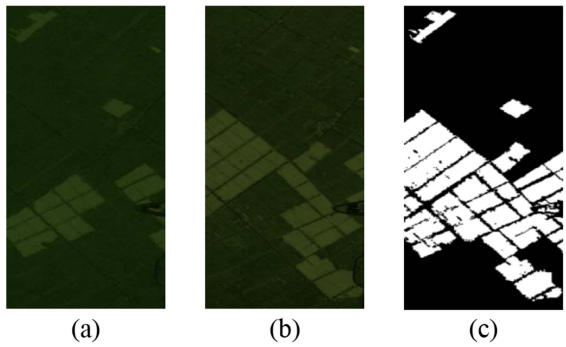

Fig. 3. China dataset. (a) Image acquired on May 3, 2006. (b) Image acquired on April 23, 2007. (c) Ground-truth image.

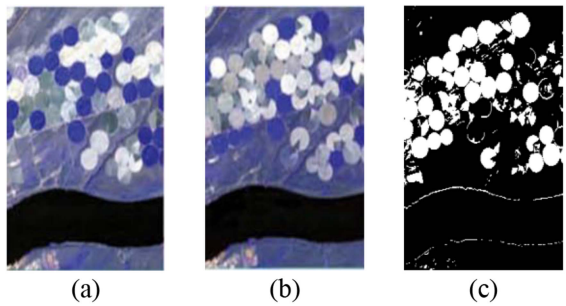

Fig. 4. USA dataset. (a) Image acquired on May 1, 2004. (b) Image acquired on May 8, 2007. (c) Ground-truth image.

1) *China dataset*: The two HSIs were obtained on May 3, 2006 and April 23, 2007, near the city of Yancheng, Jiangsu, China. They were obtained by the Earth Observing-1 (EO-1) Hyperion hyperspectral sensor, covering an area of farmlands with the size of 420 $\times$ 140 pixels and 155 bands. The images are shown in Fig. 3.
2) *USA dataset*: The two HSIs were obtained on May 1, 2004 and May 8, 2007 in Hermiston city, OR, USA. They were also obtained by the EO-1 Hyperion sensor, covering an area of irrigated agriculture with the size of 307 $\times$ 241 pixels and 156 bands. The images are shown in Fig. 4.
3) *Santa Barbara dataset*: The two HSIs were obtained in 2013 and 2014, near Santa Barbara, CA, USA. They were obtained by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor, making up of mountainous and farmlands. With size of 984 $\times$ 740 pixels and 224 bands, the ground-truth image contains 52 134 changed pixels, 80 148 unchanged pixels, and the rest are undetermined pixels. The images are shown in Fig. 5.

For the China and USA datasets, the ground-truth images consist of the changed regions in white and the unchanged regions in black, as shown in Figs. 3(c) and 4(c). For the Santa Barbara dataset, the ground-truth image consists of the changed regions in white, the unchanged regions in black, and the undetermined regions in gray, as shown in Fig. 5(c).

In this work, the number of epochs is set to 1000 with the early stopping strategy. The number of early stopping rounds is set to 100. We utilize the Adam optimizer with the learning rate of

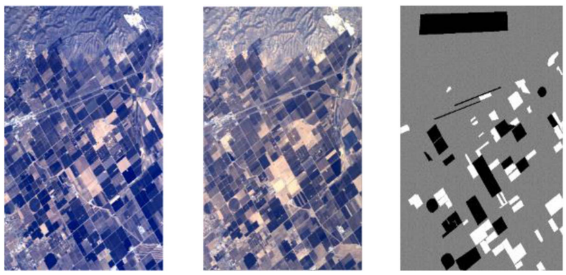

Fig. 5. Santa Barbara dataset. (a) Image acquired in 2013. (b) Image acquired in 2014. (c) Ground-truth image.

0.01 and weigh decay of 0.01 to optimize the proposed network. According to experiments, the numbers of hidden layers for three vanilla GC layers in the first multiorder GC layers on both branches are set to 128, 128, and 32, respectively. The numbers of hidden layers for the second multiorder GCN are set to 64, 64, and 8, respectively. We set the best segmentation scale for the China, USA, and Santa Barbara datasets are 5, 4, and 250, respectively. All the experiments are carried out by using 1% samples for validation. What is more, all the experiments are conducted on Python, Pytorch, and torch-sparse. The comparison experiments are conducted on NVIDIA GeForce RTX 3090.

B. Ablation Study

The multiorder GCN module, the CAM, and the feature fusion module play significant roles in our network. In this section, the effectiveness of these modules is demonstrated by ablation study.

1) Ablation Study on Different Modules: The basic architecture without these modules consists of two GC layers on each branch, a fully connected layer and a softmax activation function. The final CD is obtained via the upsampling of the superpixel-wise CD result. Based on the basic architecture, we apply different modules and fix the others to compare the CD accuracy. The following four cases are compared:

1) the basic architecture without three proposed modules;
2) the GC layers in the basic architecture are replaced with multiorder GC layers;
3) Based on the second architecture, the superpixel-wise CD is replaced with the pixel-wise CD via feature fusion module;
4) the CAM is embedded in the third architecture, forming the whole architecture with three proposed modules.

The results are provided via OA and Kappa, as shown in Table I.

As shown in Table I, the designed multiorder GCN module can improve the CD accuracy of the basic architecture on three datasets. In detail, the OA and Kappa are improved by 0.29% and 0.63% for Santa Barbara dataset; and applying the feature fusion can further improve the CD accuracy on three datasets. The OA and Kappa are further improved by 6.92% and 21.24% for USA dataset. By contrast of none-emphasizing the multiorder difference features, the CAM increases Kappa by 1.01% and 0.77% for China and USA datasets, respectively.

TABLE I
RESULTS OF ABLATION STUDY ON DIFFERENT MODULES

Dataset	Multi-order GCN	CAM	feature fusion	OA	Kappa
China	-	-	-	0.9496	0.8835
	√	-	-	0.9512	0.8859
	√	-	√	0.9620	0.9118
	√	√	√	**0.9662**	**0.9219**
USA	-	-	-	0.8788	0.6375
	√	-	-	0.8806	0.6400
	√	-	√	0.9498	0.8524
	√	√	√	**0.9511**	**0.8601**
Santa Barbara	-	-	-	0.9771	0.9517
	√	-	-	0.9800	0.9580
	√	-	√	0.9863	0.9712
	√	√	√	**0.9870**	**0.9728**

TABLE II
RESULTS OF MULTIORDER GCN MODULE WITH DIFFERENT ADJACENCY ORDERS (TIME UNIT: SECONDS)

Number of adjacency orders		1	2	3	4
China	OA	0.9527	0.9553	**0.9662**	0.9583
	Kappa	0.8899	0.8973	**0.9219**	0.9036
	Training time	**12.33**	16.56	55.24	39.24
	Testing time	**0.02**	0.03	0.09	0.04
USA	OA	0.9495	0.9489	**0.9511**	0.9477
	Kappa	0.8501	0.8475	**0.8601**	0.8458
	Training time	**23.39**	39.37	48.40	50.01
	Testing time	**0.02**	0.05	0.06	0.06
Santa Barbara	OA	0.9831	0.9822	**0.9870**	0.9846
	Kappa	0.9644	0.9626	**0.9728**	0.9677
	Training time	**37.65**	43.44	67.74	77.93
	Testing time	**0.01**	0.02	0.02	0.02

The experimental results demonstrate that the multiorder GCN, the channel attention, and the feature fusion modules are indeed helpful to learn information from multitemporal HSIs to improve the performance of CD.

2) Scale the Multiorder GCN Module: We further conduct experiments to analyze the performance of the multiorder GCN module with different number of adjacency orders. The number of adjacency orders ranges from one to four. When the number of adjacency order is one, the network architecture utilizes the vanilla GCN. The CAM module and pixel-wise CD are utilized in the whole network architecture. The experimental results are provided via OA and Kappa, as shown in Table II.

It can be seen that, when the number of adjacency orders is set as three, OA and Kappa obtain the optimal values, while the training time is about one minute. Take China dataset as an example, compared with the vanilla GCN, the OA and Kappa of three-order GCN are increased by 1.35% and 3.2%, respectively; while the training time is increased by about 42 s. Therefore, the detection performance and model complexity of the proposed MGCN method is moderate and acceptable.

C. CD Results

To demonstrate the performance of the proposed MGCN method, some recently proposed and commonly used methods are compared with the proposed method, including BCG-Net [6], HyperNet [61], BCNNs [32], SST-Former [62], CSANet [34], and D^{2}AGCN [48]. In detail, BCG-Net and HyperNet

TABLE III
PERFORMANCE OF CD ON THREE DATASETS (TIME UNIT: SECONDS)

Dataset	Metric	BCG-Net	HyperNet	BCNNs	SST-Former	CSANet	D^2AGCN	MGCN
China	OA	0.9398	0.9126	0.9446	0.9623	0.9537	0.9418	**0.9662**
	Kappa	0.8489	0.8026	0.8708	0.9134	0.8923	0.8597	**0.9219**
	Precision	0.9440	0.9180	0.9447	0.9501	0.9539	0.9471	**0.9570**
	Recall	0.8424	0.9126	0.9446	0.9642	0.9538	0.9164	**0.9652**
	F1	0.8903	0.9138	0.9446	0.9567	0.9538	0.9298	**0.9609**
	Training time	159.92	**16.71**	60.6	114.27	57.2	3554.93	55.24
	Testing time	290.37	0.71	2.52	5.69	7.6	9.91	**0.09**
USA	OA	0.9410	0.9267	0.9295	0.9461	0.9346	0.9258	**0.9511**
	Kappa	0.8339	0.7757	0.7940	0.8470	0.8096	0.7658	**0.8601**
	Precision	0.8529	0.9260	0.9284	0.9470	0.9338	0.9277	**0.9512**
	Recall	0.8924	0.9267	0.9295	0.9461	0.9346	0.9258	**0.9511**
	F1	0.8722	0.9240	0.9287	0.9465	0.9340	0.9216	**0.9511**
	Training time	153.83	**23.22**	34	59.42	59.8	4591.41	48.40
	Testing time	228.91	0.72	1.75	7.06	4.97	21.91	**0.06**
Santa Barbara	OA	0.9147	0.9228	0.9765	0.9775	0.9748	0.9796	**0.9870**
	Kappa	0.8245	0.8390	0.9507	0.9527	0.9468	0.9570	**0.9728**
	Precision	0.9192	0.9234	0.9765	0.9676	0.9750	0.9798	**0.9880**
	Recall	0.9147	0.9228	0.9765	0.9676	0.9748	0.9796	**0.9849**
	F1	0.9153	0.9230	0.9765	0.9676	0.9747	0.9795	**0.9864**
	Training time	201.85	251.53	77.9	114.04	94.8	1584	**67.74**
	Testing time	529.33	17.54	5.52	78.67	9.17	69.59	**0.02**
Trainable Parameters(M)		**0.05**	0.37	0.36	2.49	40.42	0.14	0.1

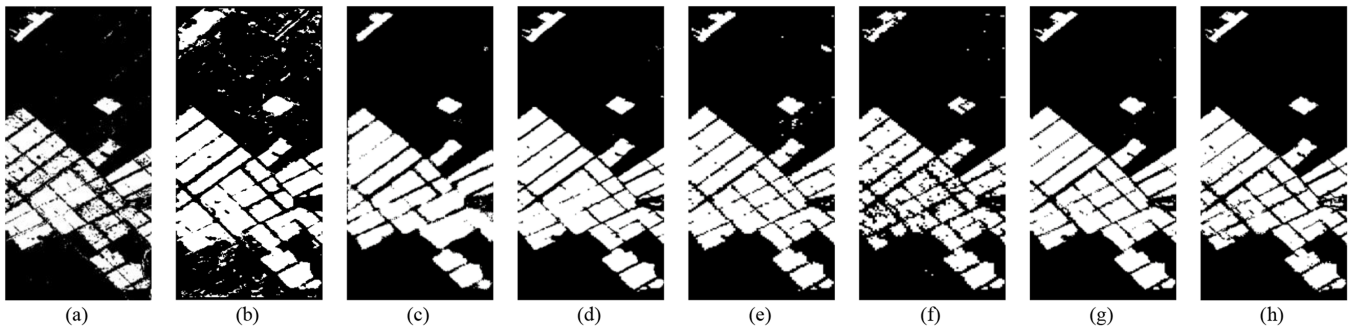

Fig. 6. Change maps of different methods on China dataset. (a) BCG-Net. (b) HyperNet. (c) BCNNs. (d) SST-Former. (e) CSANet. (f) D^2AGCN. (g) MGCN. (h) Ground-truth image.

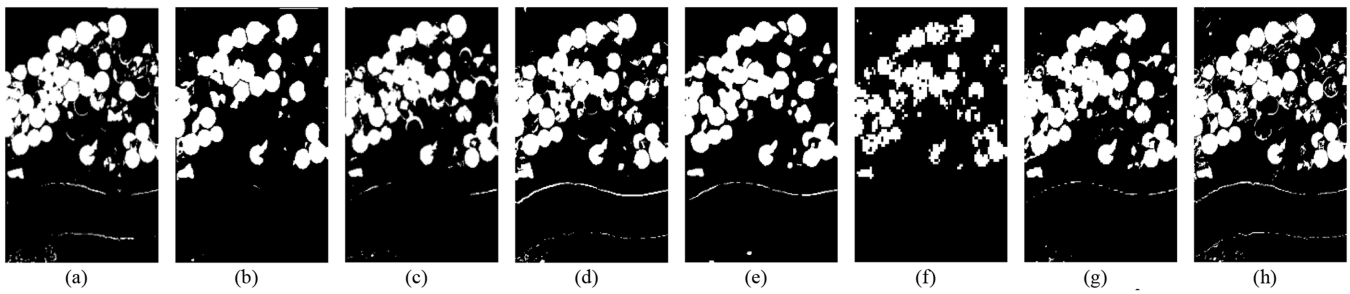

Fig. 7. Change maps of different methods on USA dataset. (a) BCG-Net. (b) HyperNet. (c) BCNNs. (d) SST-Former. (e) CSANet. (f) D^2AGCN. (g) MGCN. (h) Ground-truth image.

utilize pseudolabel to design an unsupervised network and a self-supervised network for CD tasks, respectively. For BCG-Net and HyperNet, the number of pseudolabel is set according to [6] and [61]. For the rest supervised methods, we use 0.5% samples for training. The performance of those methods is evaluated using a series of metrics including OA, Kappa, Precision, Recall, F1, training time, testing time, and Params. In order to alleviate the influence of random factors, the experiments of the proposed method are repeated five times. The value of each metric ranked in third is taken as the final evaluation value. Table III shows the results on three datasets, where the best value is in bold. And the change maps are shown from Figs. 6–8.

1) Experiment Results of China Dataset: As shown in Fig. 6, BCG-Net, HyperNet, and BCNNs miss some change regions in the lower half of the image. And HyperNet mistakenly detects the unchanged regions as changed regions in the upper and lower corner of the image. BCNNs and SST-Former cannot effectively capture the boundaries of the changed regions. CSANet mistakenly detects the unchanged regions as changed regions in the middle part of the image. Compared to D^2AGCN, MGCN

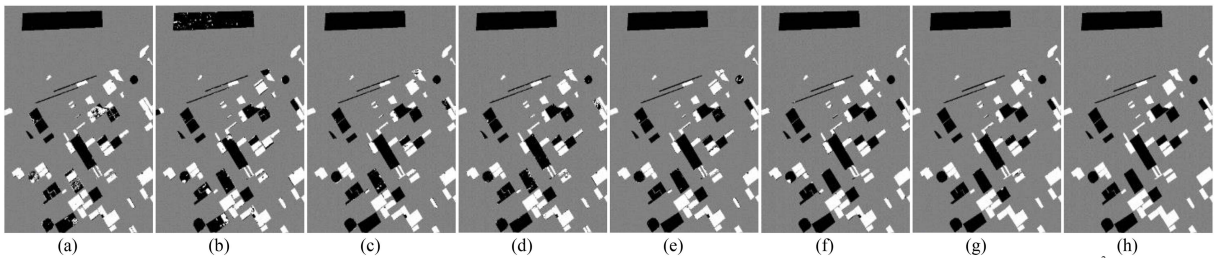

Fig. 8. Change maps of different methods on Santa Barbara dataset. (a) BCG-Net. (b) HyperNet. (c) BCNNs. (d) SST-Former. (e) CSANet. (f) D^{2}AGCN. (g) MGCN. (h) Ground-truth image.

exhibits fewer missed and false detection areas. In detail, the MGCN can preserve more precise details inside the changed objects and obtain more accurate boundaries of the changed objects. The reasons may be as follows. The multiorder GCN and CAMs are introduced to extract sufficient difference features in the neighborhood of superpixel graph nodes. And the pixel-wise CD can alleviate the error accumulation by superpixel segmentation and allow local difference within superpixels.

In addition, as shown in the Table III, the proposed MGCN method obtains higher values of the first five metrics than all the comparison methods. As for the two GCN-based methods, compared with D^{2}AGCN, the MGCN increases OA by 2.44%, Kappa by 6.22%, Precision by 0.99%, Recall by 4.88%, and F1 by 3.11%. Then, the proposed MGCN ranks second in the training time. Compared with D^{2}AGCN, the training and testing time of the MGCN is reduced by 3499.69 and 9.82 s, respectively.

2) Experiment Results of USA Dataset: The changes of land covers are complex. As shown in Fig. 7, HyperNet, BCNNs, and D2AGCN miss some changed regions, such as the regions within the circle and line shapes on the image. In addition, BCG-Net mistakenly detects the unchanged regions as changed regions in the upper half of the image. SST-Former and CSANet cannot effectively capture the boundaries of changed regions and miss the small changed regions. Compared to D2AGCN, MGCN can preserve more precise details for the circle changed regions and has less missed detection for the small changed regions.

In addition, as shown in Table III, the MGCN has better performance of detecting complex changes than all the comparison methods. Compared with SST-Former, the MGCN increases OA by 0.5%, Kappa by 1.31%, Precision by 0.42%, Recall by 0.5%, and F1 by 0.46%. Besides, the training time and testing time of the MGCN are also superior to SST-Former. By contrast of D^{2}AGCN, the MGCN increases OA by 2.53%, Kappa by 9.43%, Precision by 2.35%, Recall by 2.53%, and F1 by 2.95%, and the MGCN reduces the training time and testing time of D^{2}AGCN from 4591.41 to 48.4 s and 21.91 to 0.06 s, respectively.

3) Experiment Results of Santa Barbara Dataset: The change areas are relatively regular and scattered. As shown in Fig. 8, BCG-Net and HyperNet mistakenly detect the unchanged regions as changed regions, such as the regions in the lower corner and the upper half of the image. BCNNs, SST-Former, and CSANet mistakenly detect the unchanged regions as changed regions, such as the circle region in the upper and lower corner of the image. D^{2}AGCN also mistakenly detects the unchanged regions as changed regions, especially the boundary of the unchanged regions. The CD map of the MGCN is most similar to the ground-truth map.

As shown in the Table III, MGCN and D^{2}AGCN are superior to the other methods. Compared to D^{2}AGCN, the MGCN can improve OA by 0.74%, Kappa by 1.58%, Precision by 0.82%, Recall by 0.53%, and F1 by 0.69%. What is more, the MGCN reduce the training time and testing time by 1516.26 and 69.57 s, respectively.

Therefore, the results generally show that the metrics of the proposed MGCN, such as OA, Kappa, Precision, Recall, and F1, are more advantageous. The training time and testing time of the MGCN are also competitive. In addition, MGCN has fewer issues related to missed and false detections. Besides, the number of trainable parameters in the MGCN is about 0.1M, which is fewer than most comparison methods.

D. Parameter Analysis

We conduct two tasks for experiments denoted by evaluation of superpixel segmentation and evaluation of training samples. The first task is to illustrate the influence of superpixel segmentation on the MGCN. The second task is designed to evaluate the influence of training samples on CD.

1) Evaluation of Superpixel Segmentation: This section illustrates the influence of superpixel segmentation on the MGCN with three datasets. The experiment analyzes the performance of the MGCN with different superpixel segmentation scales. The OA and Kappa values on three datasets are shown in Fig. 9.

For China and Santa Barbara datasets, the OA and Kappa increases to the optimal value, and then, decreases when the segmentation scale gradually increases. For USA dataset, the OA and Kappa values generally decreases as the segmentation scale increases, and the optimal values of segmentation scale are set as 5, 4, and 250 for China, USA, and Santa Barbara datasets, respectively.

2) Evaluation of Training Samples: This section evaluates the influence of training samples on CD with three datasets. The experiments compare the performance of BCNNs, SST-Former, CSANet, D^{2}AGCN, and MGCN with different number of training samples. The training samples are set according to the range [0.2%, 0.5%, 1.6%, 3.2%, 6.4%, and 12.8%]. The OA and Kappa values of different methods with different training samples on threes datasets are shown in Figs. 10 and 11.

The results show that the performance of most methods increases as the number of training samples increases from 0.2% to 12.8%. The MGCN is superior to the other four methods,

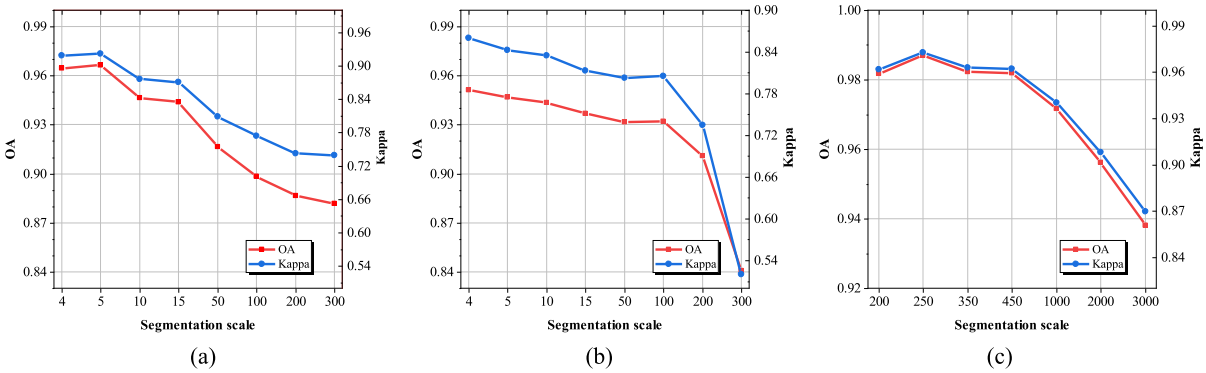

Fig. 9. Performance of MGCN with different segmentation scales on three datasets. (a) China. (b) USA. (c) Santa Barbara.

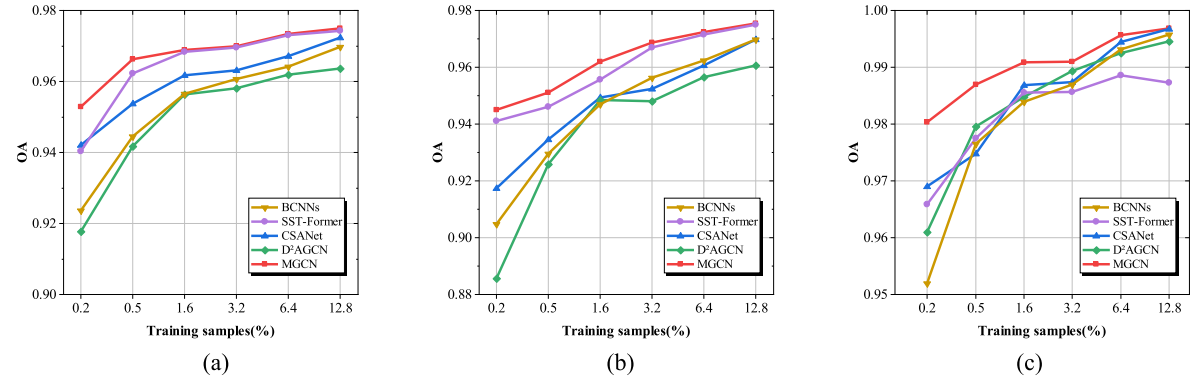

Fig. 10. OA of different methods with different training samples on three datasets. (a) China. (b) USA. (c) Santa Barbara.

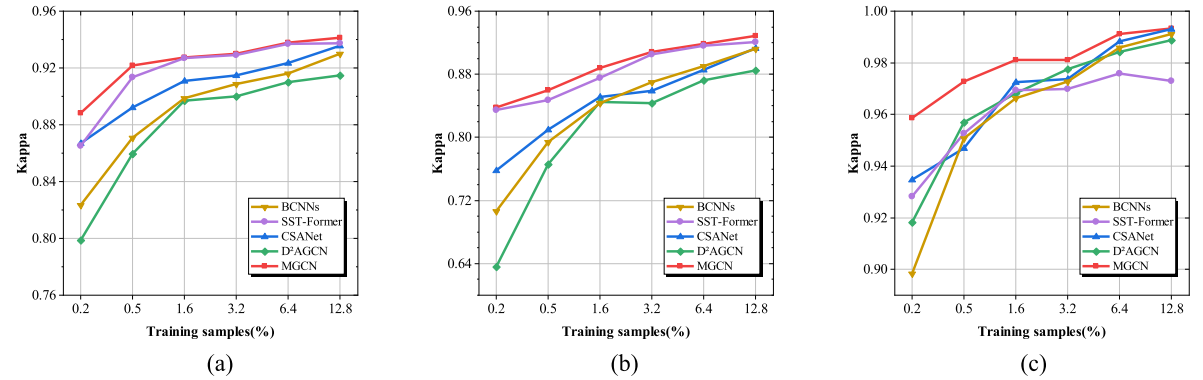

Fig. 11. Kappa of different methods with different training samples on three datasets. (a) China. (b) USA. (c) Santa Barbara.

especially when the number of training samples is lower than 1.6%, 3.2%, and 6.4% for China, USA, and Santa Barbara datasets, respectively. Besides, the MGCN still obtains good performance with limited number of training samples. In detail, with 0.2% training samples, OA of the MGCN approximately reaches 0.953, 0.945, and 0.965, for three datasets, respectively; while Kappa of the MGCN approximately reaches 0.89, 0.84, and 0.96 for three datasets, respectively.

V. Conclusion

In this article, we propose an MGCN method for hyperspectral CD. By designing a dual-branch feature extraction framework with multiorder GCN and CAMs, the MGCN method can extract sufficient features of multitemporal HSIs and emphasize the difference features between multitemporal HSIs. By introducing a feature fusion module with association matrix and convolution layers, the MGCN performs a pixel-wise CD to alleviate the error accumulation by superpixel segmentation and allow local difference within superpixels, and it adopts a superpixel sparse graph store technique and a lightweight feature fusion module via convolution layers rather than the GCN to improve the computational efficiency.

However, this method can be affected by the superpixel segmentation. To detect the complex changes of land covers, we require a fine superpixel segmentation to construct graph

structures for two temporal images. In order to obtain an appropriate superpixel graph structure, we will study how to constantly update the graph structure during training process. In addition, to fully utilize the rich spectral information of HSIs, we will extend the proposed method from binary CD to multiclass CD.

REFERENCES

[1] M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley, "Change detection from remotely sensed images: From pixel-based to object-based approaches," *ISPRS J. Photogramm. Remote Sens.*, vol. 80, pp. 91–106, 2013.

[2] A. Singh, "Review article digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, 1989.

[3] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[4] D. Datta, P. K. Mallick, A. K. Bhoi, M. F. Ijaz, J. Shafi, and J. Choi, "Hyperspectral image classification: Potentials, challenges, and future directions," *Comput. Intel. Neurosci.*, vol. 2022, vol. 2022, Art. no. 3854635.

[5] A. P. Tewkesbury, A. J. Comber, N. J. Tate, A. Lamb, and P. F. Fisher, "A critical synthesis of remotely sensed optical image change detection techniques," *Remote Sens. Environ.*, vol. 160, pp. 1–14, 2015.

[6] M. Hu, C. Wu, B. Du, and L. Zhang, "Binary change guided hyperspectral multiclass change detection," *IEEE Trans. Image Process*, vol. 32, pp. 791–806, 2023.

[7] L. Bruzzone and S. B. Serpico, "An iterative technique for the detection of land-cover transitions in multitemporal remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 35, no. 4, pp. 858–867, Jul. 1997.

[8] F. Huang, Y. Yu, and T. Feng, "Hyperspectral remote sensing image change detection based on tensor and deep learning," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 233–244, Jan. 2019.

[9] T. Hame, I. Heiler, and J. San Miguel-Ayanz, "An unsupervised change detection and recognition system for forestry," *Int. J. Remote Sens.*, vol. 19, no. 6, pp. 1079–1099, 1998.

[10] O. A. Carvalho Júnior, R. F. Guimarães, A. R. Gillespie, N. C. Silva, and R. A. T. Gomes, "A new approach to change vector analysis using distance and similarity measures," *Remote Sens*, vol. 3, pp. 2473–2493, 2011.

[11] 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, 1998.

[12] 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.

[13] 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.

[14] M. Volpi, D. Tuia, F. Bovolo, M. Kanevski, and L. Bruzzone, "Supervised change detection in VHR images using contextual information and support vector machines," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 20, pp. 77–85, 2013.

[15] H. Nemmour and Y. Chibani, "Multiple support vector machines for land cover change detection: An application for mapping urban extensions," *ISPRS J. Photogram. Remote Sens.*, vol. 61, no. 2, pp. 125–133, 2006.

[16] L. Bruzzone and R. Cossu, "A multiple-cascade-classifier system for a robust and partially unsupervised updating of land-cover maps," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 9, pp. 1984–1996, Sep. 2002.

[17] Z. Hou, W. Li, R. Tao, and Q. Du, "Three-order tucker decomposition and reconstruction detector for unsupervised hyperspectral change detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 6194–6205, Jun. 2021.

[18] T. Kasetkasem and P. K. Varshney, "An image change detection algorithm based on Markov random field models," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 8, pp. 1815–1823, Aug. 2002.

[19] J. Meola, M. T. Eismann, R. L. Moses, and J. N. Ash, "Application of model-based change detection to airborne VNIR/SWIR hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3693–3706, Oct. 2012.

[20] X. Huang and M. A. Friedl, "Distance metric-based forest cover change detection using MODIS time series," *" Int. J. Appl. Earth Observ. Geoinf.*, vol. 29, pp. 78–92, 2014.

[21] K. Ding, C. Huo, Y. Xu, Z. Zhong, and C. Pan, "Sparse hierarchical clustering for VHR image change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 3, pp. 577–581, Mar. 2015.

[22] A. Ertürk, M. D. Iordache, and A. Plaza, "Sparse unmixing-based change detection for multitemporal hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 708–719, Feb. 2016.

[23] A. Ertürk, M. D. Iordache, and A. Plaza, "Sparse unmixing with dictionary pruning for hyperspectral change detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 1, pp. 321–330, Jan. 2017.

[24] S. Liu, L. Bruzzone, F. Bovolo, and P. Du, "Unsupervised multitemporal spectral unmixing for detecting multiple changes in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 5, pp. 2733–2748, May 2016.

[25] S. Zhang et al., "Superpixel-guided sparse unmixing for remotely sensed hyperspectral imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 2155–2158.

[26] 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.

[27] 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.

[28] W. Dong et al., "Abundance matrix correlation analysis network based on hierarchical multihead self-cross-hybrid attention for hyperspectral change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Jan. 2023, Art. no. 5501513.

[29] M. Papadomanolaki, S. Verma, M. Vakalopoulou, S. Gupta, and K. Karantzalos, "Detecting urban changes with recurrent neural networks from multitemporal Sentinel-2 data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 214–217.

[30] 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.

[31] 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.

[32] 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.

[33] T. Zhan et al., "SSCNN-S: A spectral-spatial convolution neural network with siamese architecture for change detection," *Remote Sens*, vol. 13, pp. 2072–4292, 2021.

[34] R. Song, W. Ni, W. Cheng, and X. Wang, "CSANet: Cross-temporal interaction symmetric attention network for hyperspectral image change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, May 2022, Art. no. 6010105.

[35] Y. Ye, M. Wang, L. Zhou, G. Lei, J. Fan, and Y. Qin, "Adjacent-level feature cross-fusion with 3-D CNN for remote sensing image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Aug. 2023, Art. no. 5618214.

[36] L. Mou, L. Bruzzone, and 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.

[37] G. Wang, B. Li, T. Zhang, and S. Zhang, "A network combining a transformer and a convolutional neural network for remote sensing image change detection," *Remote Sens*, vol. 14, 2022, Art. no. 2228.

[38] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[39] J. Bai, B. Ding, Z. Xiao, L. Jiao, H. Chen, and A. C. Regan, "Hyperspectral image classification based on deep attention graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Mar. 2021, Art. no. 5504316.

[40] S. Baroud, S. Chokri, S. Belhaous, and M. Mestari, "A brief review of graph convolutional neural network based learning for classifying remote sensing images," *Proc. Comput. Sci.*, vol. 191, pp. 349–354, 2021.

[41] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[42] Q. Liu, L. Xiao, J. Yang, and Z. Wei, "CNN-enhanced graph convolutional network with pixel- and superpixel-level feature fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8657–8671, Oct. 2021.

[43] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.

[44] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 31, pp. 1559–1572, Jan. 2022.

[45] Q. Liu, Y. Dong, Y. Zhang, and H. Luo, "A fast dynamic graph convolutional network and CNN parallel network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, May. 2022, Art. no. 5530215.
[46] R. Singh, S. Bathla, and P. Meel, "State-of-the-art applications of graph convolutional neural networks," in *Proc. 6th Int. Conf. Recent Trends Comput.*, Cham, Switzerland, 2021, pp. 107–115.
[47] W. Dong, Y. Yang, J. Qu, S. Xiao, and Y. Li, "Local information-enhanced graph-transformer for hyperspectral image change detection with limited training samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Apr. 2023, Art. no. 5509814.
[48] J. Qu, Y. Xu, W. Dong, Y. Li, and Q. Du, "Dual-branch difference amplification graph convolutional network for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Dec. 2022, Art. no. 5519912.
[49] S. Abu-El-Haija et al., "MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 21–29.
[50] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2014, *arXiv:1312.6203v3*.
[51] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 3844–3852.
[52] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017, *arXiv:1609.02907v4*.
[53] X. Yang, B. Tu, Q. Li, J. Li, and A. Plaza, "Graph evolution-based vertex extraction for hyperspectral anomaly detection.," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2023.3303273.
[54] X. Liao, B. Tu, J. Li, and A. Plaza, "Class-wise graph embedding-based active learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Aug. 2023, Art. no. 5522813.
[55] B. Tu, Q. Ren, Q. Li, W. He, and W. He, "Hyperspectral image classification using a superpixel–pixel–subpixel multilevel network," *IEEE Trans. Instrum. Meas.*, vol. 72, May 2023, Art. no. 5013616.
[56] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[57] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Superpixel sampling networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 352–368.
[58] Z. Wang and M. Assoc Comp, "SparseRT: Accelerating unstructured sparsity on GPUs for deep learning inference," in *Proc. ACM Int. Conf. Parallel Architecture. Compilation Tech.*, 2020, pp. 31–42.
[59] M. Hasanlou and S. Seydi, "Hyperspectral change detection: An experimental comparative study," *Int. J. Remote Sens.*, vol. 39, no. 20, pp. 7029–7083, 2018.
[60] Y. Yuan, H. Lv, and X. Lu, "Semi-supervised change detection method for multi-temporal hyperspectral images," *Neurocomputing*, vol. 148, pp. 363–375, 2015.
[61] M. Hu, C. Wu, and L. Zhang, "HyperNet: Self-supervised hyperspectral spatial–spectral feature understanding network for hyperspectral change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Nov. 2022, Art. no. 5543017.
[62] Y. Wang et al., "Spectral–spatial–temporal transformers for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Aug. 2022, Art. no. 5536814.

Yuxiang Zhang (Member, IEEE) received the B.S. degree in sciences and techniques of remote sensing and the Ph.D. degree in the photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2011 and 2016, respectively.

She was a Visiting Scholar with the University of Sydney, Sydney, Australia, from 2019 to 2020. She is currently an Associate Professor with the School of Geophysics and Geomatics, China University of Geosciences, Wuhan. Her current research interests include hyperspectral image processing, pattern recognition, and machine learning.

Rui Miao received the B.E. degree in sciences and techniques of remote sensing from the College of Geodesy and Geomatics, Shandong University of Science and Technology, Qingdao, China, in 2022. She is currently working toward the master's degree in resources and environment with the School of Geophysics and Geomatics, China University of Geosciences, Wuhan, China.

Her current research current research interests include change detection and graph neural networks.

Yanni Dong (Senior Member, IEEE) received the B.S. degree in sciences and techniques of remote sensing from Wuhan University, Wuhan, China, in 2012, and the Ph.D. degree in photogrammetry and remote sensing from the State Key Lab of Information Engineering in Surveying, Mapping and Remote sensing, Wuhan University, in 2017.

She was a Hong Kong Scholar with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong. She is currently a Professor with the School of Resource and Environmental Sciences, Wuhan University, Wuhan, China. Her current research interests include hyperspectral image processing, pattern recognition, and machine learning.

Dr. Dong serves as a Reviewer of more than 20 international journals, including the IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, and PATTERN RECOGNITION (PR). She regularly serves as a Program Committee Member of the International Joint Conference on Artificial Intelligence and Association for the Advancement of Artificial Intelligence (AAAI).

Bo Du (Senior Member, IEEE) received the Ph.D. degree in photogrammetry and remote sensing from the State Key Lab of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China, in 2010.

He is currently a Professor with the School of Computer Science and Institute of Artificial Intelligence, Wuhan University. He is also the Director of National Engineering Research Center for Multimedia Software, Wuhan University. He has more than 80 research papers published in the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON CYBERNETICS (TCYB), IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), etc.; 14 of them are ESI hot papers or highly cited papers. His major research interests include machine learning, computer vision, and image processing.

Dr. Du serves as an Associate Editor for *Neural Networks*, *Pattern Recognition*, and *Neurocomputing*. He also serves as a Reviewer of 20 Science Citation Index (SCI) magazines including IEEE TPAMI, TCYB, TGRS, TIP, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, and IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. He won the Highly Cited Researcher (2019–2020) by the Web of Science Group. He was the recipient of IEEE Geoscience and Remote Sensing Society 2020 Transactions Prize Paper Award. He won the International Joint Conferences on Artificial Intelligence (IJCAI) Distinguished Paper Prize, IEEE Data Fusion Contest Champion, and IEEE Workshop on Hyperspectral Image and Signal Processing Best paper Award, in 2018. He regularly serves as senior Program Committee Member of the IJCAI and AAAI. He served as an Area Chair for International Conference on Pattern Recognition.