Multispectral Change Detection With Bilinear Convolutional Neural Networks

Yun Lin, Shutao Li, *Fellow, IEEE*, Leyuan Fang, *Senior Member, IEEE*, and Pedram Ghamisi, *Senior Member, IEEE*

***Abstract*— Recently, deep learning has been demonstrated to be an effective tool to detect changes in bitemporal remote sensing images. However, most existing methods based on deep learning obtain the ultimate change map by analyzing the difference image (DI) or the stacked feature vectors of input images, which cannot sufficiently capture the relationship between the two input images to obtain the change information. In this letter, a new method named bilinear convolutional neural networks (BCNNs) is proposed to detect changes in bitemporal multispectral images. The model can be trained end to end with two symmetric convolutional neural networks (CNNs), which are capable of learning the feature representation from bitemporal images and utilizing the relations between the two input images by a linear outer product operation in an effective way. Specifically, two sets of patches obtained from two multispectral images of different times are first input into two CNNs to extract deep features, respectively. Then, the matrix outer product is applied on the output feature maps to obtain the combined bilinear features. Finally, the ultimate change detected result can be produced by applying the softmax classifier on the combined features. Experimental results on real multispectral data sets demonstrate the superiority of the proposed method over several well-known change-detection approaches.**

***Index Terms*— Bilinear convolutional neural networks (BCNNs) model, change detection, multispectral images.**

I. Introduction

CHANGE detection aims to determine if the surface features of an area have changed by analyzing the images of the same land-cover region taken from different times [1]. Recently, since change detection techniques can provide a reliable basis for quick, accurate, and careful decision-making, it is widely used in many fields of remote sensing, such as urban extension studies [2], natural disaster assessment and warning [3], environmental monitoring [4], and land resource utilization [5].

During the past decades, numerous change detection methods have been introduced. In [6], the change vector analysis (CVA) was proposed to identify changed and unchanged areas by computing the magnitude and direction of the change vectors for the difference image (DI). In [7], Celik first adopted the principal components analysis (PCA) to extract the features of DI, and then applied k-means clustering on the PCA features to obtain the change map. The aforementioned methods do not require the labeling data sets for the model training [6], [7], which are considered as the unsupervised methods.

Recently, a deep neural network (DNN) that needs a large-scale training data set has been demonstrated to be a powerful tool to automatically extract features and learn the representation of the input data in the fields of computer vision and image processing [8]. In addition, the DNN-based model has been extended to change detection of remote sensing images [2], [9]–[11]. In [10], to further learn the representation of the relationships between two input images and improve change detection performance, the superpixel (SP) strategy is incorporated into the deep belief network (DBN) model and the change map is produced with the trained DBN. Liu *et al.* [11] introduced a symmetric convolutional coupling neural network (SCCN) model, which utilized one convolutional neural layer to extract feature representation and applied several coupling layers to transform the extracted features simultaneously. The difference map is generated in the feature space, and the change map is obtained by analyzing the difference map with a threshold. Among these deep learning-based change-detection techniques, some supervised models just need few pixels, rather than large-scale image data sets, to train the network, which are labeled based on prior knowledge [2], [10]. Compared with the unsupervised methods that need no labeled samples, the methods that use a certain amount of labeled data to train have great improvement in detection performance with little time consumption for labeling. In addition, since the training and test samples are from the same image, the relevance between the labeled and unlabeled samples can be captured more effectively in the feature space, which can assist in training the model and obtaining the change information of other unlabeled pixels of the same image pairs.

The methods mentioned above generally obtain the change results by analyzing the DI [6], [7], the difference map of the extracted features [11], or the stacked feature vectors of the images [9]. The DI is produced by applying pixelwise difference operation (e.g., image differencing and image ratio), which is often affected by the noise and other interferences,

Manuscript received February 15, 2019; revised June 28, 2019 and September 8, 2019; accepted November 11, 2019. This work was supported in part by the National Natural Science Fund of China for International Cooperation and Exchanges under Grant 61520106001, in part by the National Natural Science Foundation under Grant 61922029, in part by the Science and Technology Plan Project Fund of Hunan Province under Grant CX2018B171, Grant 2017RS3024, and Grant 2018TP1013, and in part by the Science and Technology Talents Program of Hunan Association for Science and Technology under Grant 2017TJ-Q09. *(Corresponding author: Leyuan Fang.)*

Y. Lin, S. Li, and L. Fang are with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China, and also with the Key Laboratory of Visual Perception and Artificial Intelligence of Hunan Province, Changsha 410082, China (e-mail: linyun@hnu.edu.cn; shutao_li@hnu.edu.cn; fangleyuan@gmail.com).

P. Ghamisi is with the Helmholtz-Zentrum Dresden-Rossendorf, Helmholtz Institute Freiberg for Resource Technology, 09599 Freiberg, Germany (e-mail: p.ghamisi@gmail.com).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2019.2953754

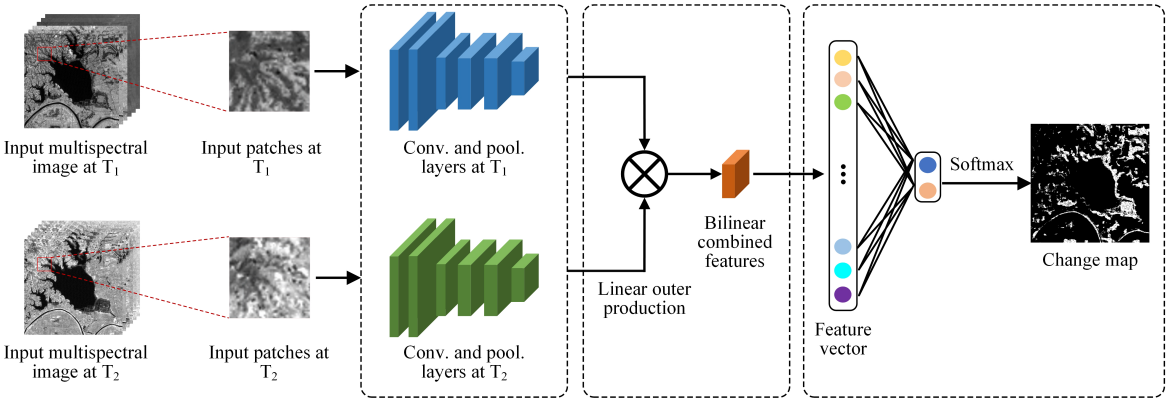

Fig. 1. Flowchart of the proposed BCNNs method for bitemporal multispectral image change detection. Each side of the BCNN network contains four convolutional layers and two pooling layers. A max-pooling layer with the kernel size of 2×2 and stride 2 is applied after each two convolutional layers. All the convolutional kernels are with the size of 3×3 with a stride of 1 pixel. The filter numbers of each convolutional layer are set to 64, 64, 128, and 128, respectively.

and thus cannot sufficiently provide reliable change information of the input images. Though the classification-based method [12], which classifies the land-cover objects before comparison, can obtain better performance, many additional categories of labels need to be provided. In addition, the classification-based method ignores or does not make good use of the relationships between the two input images. Therefore, it is necessary to effectively exploit the relationships between the two input images and avoid generating and analyzing the DI directly.

In this letter, to address the aforementioned issues, we propose a novel supervised end-to-end method called bilinear convolutional neural networks (BCNNs). In the BCNN model, a pair of multispectral image patches is first input into two symmetric convolutional neural networks (CNNs) to extract spectral–spatial features, respectively. Then, to integrate the representation of relationships between two multispectral images, the extracted features of different times are effectively combined by a linear outer product operation. Finally, the combined bilinear features are transformed into feature vectors and then fed into the softmax classifier to create the ultimate change-detection result.

II. Proposed BCNN Method

Fig. 1 shows the flowchart of the proposed method. Specifically, the BCNN method first learns the feature representation automatically through several convolutional layers of two CNNs. Then, the BCNNs combine the extracted information of two images by a linear matrix outer product operation to integrate their relations. Finally, the change map is obtained by the softmax classifier.

A. Feature Extraction

Given two multispectral images acquired from the same location at different times, we process them by means of radiometric correction, atmospheric correction, and geometric correction [13]. The patches of fixed size $s \times s$ from two processed images are first generated. Then, they are separately fed into the convolutional layers of two symmetric CNNs in the BCNNs model to extract features and learn the deep representation of original inputs automatically [14]. In the convolutional layers, the image patches successively convolve with convolutional kernels

$$\boldsymbol{x}_j^r = f\left(\boldsymbol{x}_i^{r-1} * \boldsymbol{w}_j^r + \boldsymbol{b}_j^r\right) \tag{1}$$

where $\boldsymbol{x}_i^{r-1}$ and $\boldsymbol{x}_j^r$ are the ith input feature map and the jth output feature map of the rth convolution layer, respectively. $\boldsymbol{w}_j^r$ and $\boldsymbol{b}_j^r$ are the jth convolution kernel and bias of the rth convolution layer, and $f(\cdot)$ represents the rectified linear unit (ReLU), which is defined as

$$f(\boldsymbol{x}) = \max(0, \boldsymbol{x}) \tag{2}$$

where $\boldsymbol{x}$ is the output feature map of the convolution operation. Here, we utilize the stochastic gradient descent (SGD) algorithm as the optimizer to learn the parameters during the training process. The supervised learning procedure is based on a backpropagation algorithm. After applying the convolutional layers, we utilize the max-pooling layers to remain the most representative features, thus avoiding the overfitting.

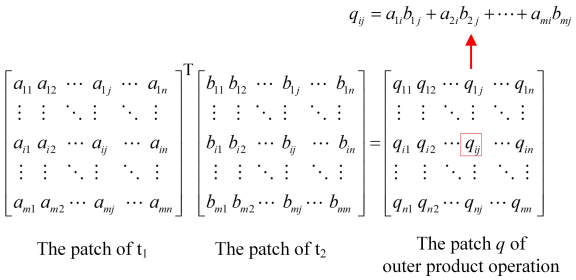

Fig. 2. Brief illustration of outer product operation.

B. Feature Bilinear Combination

After performing the convolutional layers and pooling layers, the matrix outer product is applied on two pooled feature maps $\mathbf{P}_1(f(\boldsymbol{x}_1))$ and $\mathbf{P}_2(f(\boldsymbol{x}_2))$, which can be defined as follows:

$$\Phi(q) = \sum_{l \in L} \mathbf{P}_1(f(\boldsymbol{x}_1))^{\mathrm{T}} \mathbf{P}_2(f(\boldsymbol{x}_2)) \tag{3}$$

where $\Phi(q)$ is the combined feature of two inputs at the position q, and $\mathbf{P}_1$ and $\mathbf{P}_2$ are the max-pooling operations

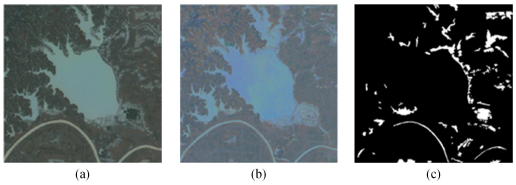

Fig. 3. Huarong data set. (a) True color composite image acquired on June 23, 2005. (b) True color composite image acquired on November 12, 2010. (c) Reference image.

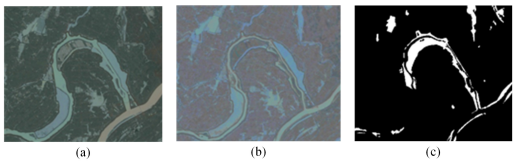

Fig. 4. Xiangyin data set. (a) True color composite image acquired on June 23, 2005. (b) True color composite image acquired on November 12, 2010. (c) Reference image.

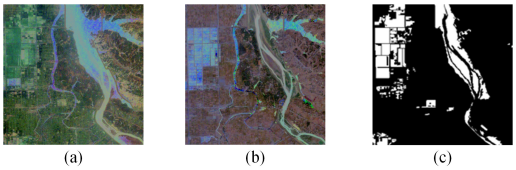

Fig. 5. Miluo data set. (a) True color composite image acquired on June 23, 2005. (b) True color composite image acquired on November 12, 2010. (c) Reference image.

at time t_1 and t_2 at a location $l \in L$. Compared with the difference operation [15] on the individual pixel to acquire change information, outer product can better combine the neighboring information of patches from different times. The reason can be described as follows. Mathematically, the outer product multiplies the feature map of t_2 with the transpose feature map of t_1. As shown in Fig. 2, after the outer product operation, the value of pixel q_{ij} is the product of the ith column for the patch in t_1 with the corresponding column jth for the patch in t_2. The outer product operation is a combination of information for two columns around the pixel q_{ij} in the patch q rather than subtraction, ratio, or log ratio of the corresponding individual pixel.

C. Change Map Prediction

According to [16], the combined bilinear features $\Phi(q)$ are reshaped to obtain the feature vectors v and processed by a signed square root operation and ℓ_2 normalization to improve performance

$$\mathbf{u} = \text{sign}(\mathbf{v})\sqrt{|\mathbf{v}|} \tag{4}$$

$$\mathbf{z} = \frac{\mathbf{u}}{\|\mathbf{u}\|_2}. \tag{5}$$

Then, we feed the normalized vector z into a fully connected layer and utilize the softmax function to compute the probability of the output belonging to each class, which can be described as

$$p_{z,k} = \frac{e^{\alpha_k}}{\sum_{k=1}^{K} e^{\alpha_k}} \tag{6}$$

where α_k is the score for the class k, K is the total number of classes, and $p_{s,k}$ is the probability of the sample s predicted to be the class k. Finally, the predicted label of each pixel can be determined by the maximal probability. Thus, the label of an unlabeled test pixel can be predicted by the proposed BCNNs model to generate the ultimate change map.

TABLE I
DESCRIPTIONS OF THREE MULTISPECTRAL DATA SETS

Data set	Huarong	Xiangyin	Miluo
Size	399 × 398	389 × 331	300 × 300
Location	Huarong, China	Xiangyin, China	Miluo, China
Changes	river, lake	river	river, buildings
Train-positive	1000	1000	1000
Train-negative	1000	4000	1000
Test-positive	9324	8051	14036
Test-negative	147478	115708	73964

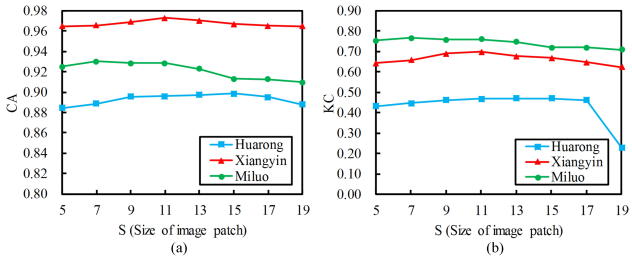

Fig. 6. Effects of different patch sizes on (a) CA and (b) KC for the proposed method on three test data sets.

III. EXPERIMENTS AND DISCUSSION

A. Data Set Descriptions and Evaluation Criteria

In the experiments, we test the BCNN method on three multispectral data sets shown in Figs. 3–5, which are captured by the Landsat-5 satellite on June 23, 2005 and November 12, 2010, respectively. They have six bands with a spatial resolution of 30 m (the thermal infrared band is removed, since its resolution is different from others). The descriptions of each data set are listed in Table I. Unlike other supervised classification tasks that need large-scale image data sets or those unsupervised methods that need no labeled data [7], [17], just a small amount of pixels in the image pairs are labeled to train the network. A certain amount of (e.g., thousands of) image patches in the bitemporal images are randomly selected as a pair of training samples, and the rest is for testing. Since the training and test samples are from the same image, relevance may be existed in the feature space between the training and test samples to some degree, which can be used to train the model and identify the categories of unlabeled samples afterward. The reference image is a binary map generated by labeling changed and unchanged regions manually based on the prior geographic information, where white pixels indicate the changed areas labeled as 1 and black pixels indicate the unchanged areas labeled as 0. The final results are reported by taking an average value over 20 runs. All the experiments are implemented on the desktop equipped with a single NVIDIA GeForce GTX 1080ti GPU, Intel Core i7-8700k CPU, and 16 GB DDR4 RAM.

False positives (FP), false negatives (FN), classification accuracy (CA), and Kappa coefficient (KC) are selected as the evaluation indexes [11], [18].

B. Effect of the Patch Size

The image patch size $s \times s$ is the only parameter that affects detection performance once the network structure of

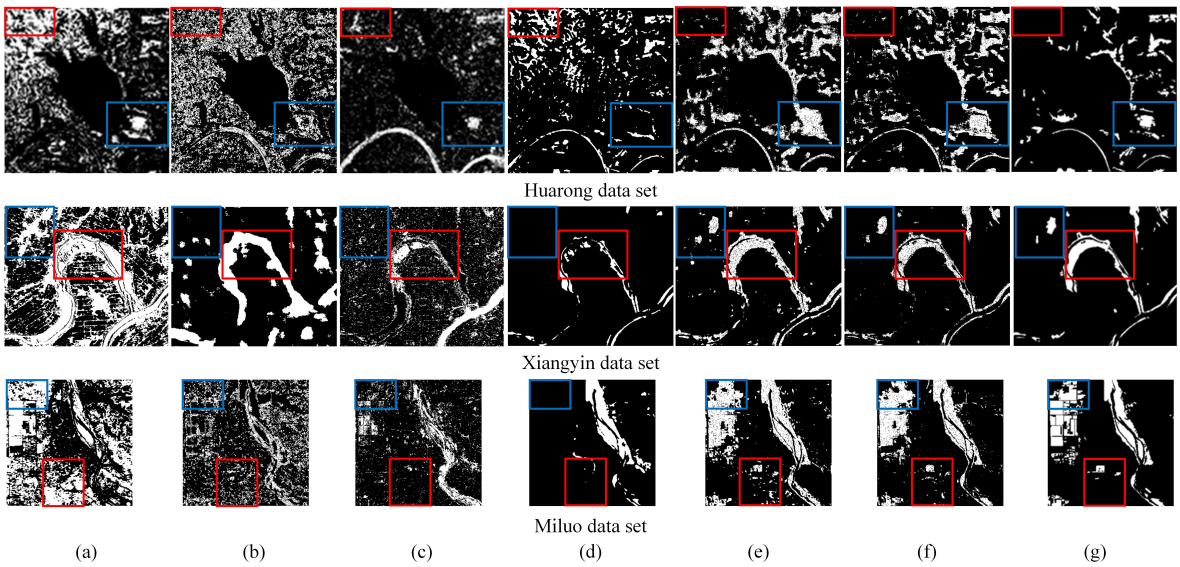

Fig. 7. Change maps obtained by different methods and the corresponding reference image. The first row is the Huarong data set. The second row is the Xiangyin data set. The third row is the Miluo data set. (a) CVA. (b) PCA-*k*-means. (c) IR-MAD. (d) SP-DBN. (e) SCNN-DI. (f) BCNNs. (g) Reference image.

TABLE II

COMPARISON OF QUANTITATIVE INDEXES FOR HUARONG DATA SET. THE BEST RESULTS IN THIS TABLE ARE LABELED

Methods	FN	FP	CA(%)	KC
CVA [6]	5309	29870	77.85	0.1351
PCA-*k*-means [7]	5007	35837	74.28	0.1145
IR-MAD [19]	6930	**13918**	86.87	0.1787
SP-DBN [10]	6211	14045	87.24	0.2245
SCNN-DI	**696**	19494	87.12	0.4079
BCNNs	697	15597	**89.61**	**0.4688**

TABLE III

COMPARISON OF QUANTITATIVE INDEXES FOR THE XIANGYIN DATA SET. THE BEST RESULTS IN THIS TABLE ARE LABELED

Methods	FN	FP	CA(%)	KC
CVA [6]	3519	52050	56.84	0.0507
PCA-*k*-means [7]	2906	17461	84.18	0.3058
IR-MAD [19]	4343	16063	84.15	0.2415
SP-DBN [10]	5382	**1014**	95.03	0.5108
SCNN-DI	**186**	6109	94.91	0.6885
BCNNs	872	2497	**97.28**	**0.7954**

TABLE IV

COMPARISON OF QUANTITATIVE INDEXES FOR MILUO DATA SET. THE BEST RESULTS IN THIS TABLE ARE LABELED

Methods	FN	FP	CA(%)	KC
CVA [6]	4828	28344	63.14	0.1851
PCA-*k*-means [7]	10258	15879	70.96	0.0922
IR-MAD [19]	9641	7916	80.49	0.2654
SP-DBN [10]	9086	**2199**	87.46	0.4485
SCNN-DI	**1037**	6923	90.95	0.7116
BCNNs	1145	5156	**92.84**	**0.7607**

the BCNNs is fixed. Fig. 6 illustrates the effect of patch size $s \times s$ on the CA and KC of the change result. We set the input patch size to be 11×11 for these three data sets in the experiments.

C. Experimental Results on Three Data Sets

In this section, the proposed BCNNs is compared with several existing well-known change-detection methods, such as CVA [6], PCA-*k*-means [7], iterative reweighted multivariate alteration detection (IR-MAD) [19], SP-DBN [10], and a symmetric CNN method based on the DI (SCNN-DI). All the parameters of the comparison methods are set to reach the best performance. The network structure of the SCNN-DI is the same as that of one side in the BCNNs.

1) Experiments on Huarong Data Set: Change results created by different methods are shown in the first row of Fig. 7, and the evaluation indexes are tabulated in Table II. In Fig. 7(a) and (b), many noisy spots exist in the results of CVA and PCA-*k*-means methods. Though the value of FP index is the smallest, the IR-MAD method misses many detailed areas and creates blurry appearance shown in Fig. 7(c). In the result of SP-DBN, main changed areas are wrongly detected. Though SCNN-DI can obtain the best performance on the FN index, its FP index is much worse than that of IR-MAD, SP-DBN, and BCNNs. The result of BCNNs shown in Fig. 7(f) has achieved significant improvement, which outperforms other methods, since it not only detects more precise changed regions (see the blue rectangle region) but also eliminates some small noisy spots (see the red rectangle regions). In addition, Table II demonstrates that BCNNs can deliver better quantitative results in terms of CA and KC indexes.

2) Experiments on Xiangyin Data Set: The second row of Fig. 7 shows the change maps obtained by different methods for the Xiangyin data set, and Table III shows the corresponding quantitative results. In Fig. 7(a), the change map obtained by CVA has many misclassifications. In Fig. 7(b), PCA-*k*-means can identify the changed and unchanged regions in general, while the changed boundaries are wrongly expanded. The result of IR-MAD reflects the changed areas with some noisy spots and miss-detected areas. Fig. 7(d) shows the change map obtained by the SP-DBN, which can identify the main changed areas, but cannot detect some less obvious regions (see the blue rectangle region). In Fig. 7(e), most of the changed and unchanged areas can be distinguished by the

TABLE V
COMPUTING TIME (SECONDS) OF DIFFERENT METHODS ON THREE DATA SETS

Datasets	CVA [6]	PCA-*k*-means [7]	IR-MAD [19]	SP-DBN [10]	SCNN-DI	BCNNs
Huarong	0.23	3.37	1.41	263.37	180.53	238.37
Xiangyin	0.81	2.78	2.70	175.97	140.51	177.61
Miluo	0.20	1.78	1.91	150.65	91.12	131.96

SCNN-DI with some false-detected spots (e.g., see the red rectangle region). Fig. 7(f) shows that BCNNs can generate the most accurate detection result and the least noisy spots. Table III further demonstrates that change result obtained by the proposed method is better than that of the other methods.

3) Experiments on Miluo Data Set: The change maps of Miluo data set generated by CVA, PCA-*k*-means, IR-MAD, SP-DBN, SCNN-DI, and BCNNs are shown in the third row of Fig. 7(a)–(f), and the related quantitative results are tabulated in Table IV. In the results of CVA, there are a large amount of false alarms. The PCA-*k*-means and IR-MAD methods still cannot provide very promising results, since the boundaries of the changed and unchanged areas are unclear. In Fig. 7(d), SP-DBN can well identify change information along the river but cannot detect the buildings (see the blue rectangle regions). In Fig. 7(e), SCNN-DI can identify most of the changed areas while still retaining much noise (see the red rectangle region). As can be seen in Fig. 7(f), BCNNs can preserve more detailed information. In addition, Table IV demonstrates that the BCNN method further reduces noise estimation and provides the best detection performance.

D. Computational Time on Three Data Sets

The execution time of different methods is tabulated in Table V for three data sets. As can be seen, the deep learning-based methods SP-DBN, SCNN-DI, and BCNNs generally require more computational time than the traditional methods CVA, PCA-*k*-means, and IR-MAD. This is because the deep learning-based methods need more computational cost to extract features and tune parameters in the deep network. In addition, the BCNNs and SCNN-DI are more efficient than the state-of-the-art deep learning-based method SP-DBN.

IV. CONCLUSION

In this letter, we presented a BCNN method to apply change detection on bitemporal multispectral images. The BCNNs method is able to learn the deep feature representation automatically. In addition, the BCNNs combine image information of different times effectively to generate the change map estimation. The experiments on three multispectral data sets demonstrate the effectiveness of our method.

In future, we will study how to modify the proposed model to be implemented in a semisupervised or unsupervised way, while still maintaining high detection accuracy. For example, a typical unsupervised classification method (e.g., PCA-*k*-means [7]) can be first adopted to create the initial result. Then, a small amount of data with high classification probability obtained by the unsupervised method can be used to train the BCNN model. After that, the trained model can be used for testing and the obtained data with high classification probability are then used to train the BCNN model iteratively. Finally, after several iterations, the ultimate change result can be generated.

REFERENCES

[1] A. Singh, "Digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, Jul. 1989.
[2] 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.
[3] D. Brunner, G. Lemoine, and L. Bruzzone, "Earthquake damage assessment of buildings using VHR optical and SAR imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2403–2420, May 2010.
[4] S. Liu, M. Chi, Y. Zou, A. Samat, J. A. Benediktsson, and A. Plaza, "Oil spill detection via multitemporal optical remote sensing images: A change detection perspective," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 324–328, Mar. 2017.
[5] 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.
[6] 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.
[7] 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, Oct. 2009.
[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[9] M. Gong, J. Zhao, J. Liu, Q. Miao, and L. Jiao, "Change detection in synthetic aperture radar images based on deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 1, pp. 125–138, Jan. 2016.
[10] M. Gong, T. Zhan, P. Zhang, and Q. Miao, "Superpixel-based difference representation learning for change detection in multispectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2658–2673, May 2017.
[11] J. Liu, M. Gong, K. Qin, and P. Zhang, "A deep convolutional coupling network for change detection based on heterogeneous optical and radar images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 545–559, Mar. 2018.
[12] G. Matasci, M. Volpi, M. Kanevski, L. Bruzzone, and D. Tuia, "Semi-supervised transfer component analysis for domain adaptation in remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3550–3564, Jul. 2015.
[13] Z. Zhu, "Change detection using Landsat time series: A review of frequencies, preprocessing, algorithms, and applications," *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 370–384, Aug. 2017.
[14] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4353–4361.
[15] L. Bruzzone and D. F. Prieto, "Automatic analysis of the difference image for unsupervised change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1171–1182, May 2000.
[16] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear convolutional neural networks for fine-grained visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1309–1322, Jun. 2018.
[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1106–1114.
[18] P. L. Rosin and E. Ioannidis, "Evaluation of global image thresholding for change detection," *Pattern Recognit. Lett.*, vol. 24, no. 14, pp. 2345–2356, 2003.
[19] 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.