Change Detection in Synthetic Aperture Radar Images Based on Deep Neural Networks

Maoguo Gong, *Senior Member, IEEE*, Jiaojiao Zhao, Jia Liu, Qiguang Miao,
and Licheng Jiao, *Senior Member, IEEE*

Abstract—This paper presents a novel change detection approach for synthetic aperture radar images based on deep learning. The approach accomplishes the detection of the changed and unchanged areas by designing a deep neural network. The main guideline is to produce a change detection map directly from two images with the trained deep neural network. The method can omit the process of generating a difference image (DI) that shows difference degrees between multitemporal synthetic aperture radar images. Thus, it can avoid the effect of the DI on the change detection results. The learning algorithm for deep architectures includes unsupervised feature learning and supervised fine-tuning to complete classification. The unsupervised feature learning aims at learning the representation of the relationships between the two images. In addition, the supervised fine-tuning aims at learning the concepts of the changed and unchanged pixels. Experiments on real data sets and theoretical analysis indicate the advantages, feasibility, and potential of the proposed method. Moreover, based on the results achieved by various traditional algorithms, respectively, deep learning can further improve the detection performance.

Index Terms—Deep learning, image change detection, neural network, synthetic aperture radar (SAR).

I. INTRODUCTION

IMAGE change detection is a process to identify the changes that have occurred between the two images of the same scene but taken at different times. It is an important issue in both civil and military fields. This depends on the fact that, for many public and private institutions, the knowledge of the dynamics of either natural resources or man-made structures is a valuable source of information in decision making [1]. It has found wide use in diverse disciplines, such as remote sensing, disaster evaluation, medical diagnosis, and video surveillance [2]. In particular, when a natural catastrophe strikes, an effective and efficient change detection task appears

critical when lives and properties are at stake. It proves to be an important application of remote sensing technology. In particular, due to their independence on atmospheric and sunlight conditions, synthetic aperture radar (SAR) images have become valuable sources of information in change detection. However, with the presence of the speckle noise, SAR images exhibit more difficulties than optical ones [3].

There are two major thoughts for change detection in SAR images according to the literature available. First, postclassification comparison, which means separately classifying the two SAR images and then comparing their classification results to achieve the changed and unchanged regions [2], [4]. This method has an advantage of avoiding radiation normalization of multitemporal remote sensing images, which are obtained from different sensors and different environmental conditions. However, there exists an issue of accumulated classification error and a requirement for high accuracy of classifying each of the two images. Second, postcomparison analysis, that is to say that first making a difference image (DI) between multitemporal SAR images, and then analyzing it to gain change detection results. Thus, it is also called DI analysis [5]. It is the current mainstream with excellent performance. However, besides the method for analyzing a DI, the quality of a DI also affects the final detection results.

Most algorithms presented are based on the latter thought. The common technique to generate a DI is the ratio method [6]. In addition, considering the influence of speckle noise, the log ratio is widely used [7], [8]. In the DI-analysis step, there are two conventional methods being used, the thresholding method and the clustering method. Classical thresholding methods, such as [9] and [10], have been applied to determine the threshold in an unsupervised manner. Actually, in the thresholding method, it is necessary to establish models to search for an optimal threshold. A Kittler-Illingworth (KI)-based method [Generalized Kittler-Illingworth method (GKI)] was generalized to take the non-Gaussian distribution of the amplitude values of SAR images into account [1]. Among the most popular clustering methods, the fuzzy c-means algorithm (FCM) can retain more information than hard clustering in some cases. In [5], we proposed an FCM-based SAR image change detection method [i.e., reformulated fuzzy local-information c-means algorithm (RFLICM)]. In recent years, graph cut [11], principal component analysis [12], and Markov random field [13] are also increasingly applied to solve the change detection problem, but neural networks are rarely considered in the

field. Just a few works were presented, such as a Hopfield-type neural network proposed to model spatial correlations in [14] and a neural-network-based change-detection system for multispectral images in [15].

As mentioned above, change detection technology in SAR images has developed well. However, with data acquisition channels and the scope of applications increasing, the involved algorithms cannot satisfy the requirements for higher accuracy and more flexible applications. As for the widely used DI-analysis methods, there currently exist three problems in change detection as follows.

1) *How to Suppress Speckle Noise?* As an important characteristic of SAR images, speckle makes image details blurred and reduces intensity and spatial resolution of images, thus it is difficult to interpret SAR images. When changed information is extracted, the presence of speckle results in some false alarms. Therefore, suppressing noise is a regular step before detection and many algorithms have been produced to solve the problem.

2) *How to Generate a DI With Good Performance?* The DI-analysis method is regarded as an effective one and used widely. The separability of a DI has direct impacts on classification results of the DI. It can be seen that a good DI is a prerequisite for the correct detection.

3) *How to Design an Efficient Classification Method?* This is the most crucial step in the whole algorithm, which determines the final change detection results.

Inspired by the architectural depth of the brain, deep learning [16]–[19] has become a new kind of machine learning method and has been paid increasing attention in recent years. Deep learning algorithms seek to exploit the unknown structure in the input distribution in order to discover good representations, with higher level learned features defined in terms of lower level features [20]. Convolutional neural networks (CNNs) are the early proposed deep architectures. They are inspired by the receptive fields in neural cortex [21] mainly designed for 2-D data, such as images and videos. With the development of deep learning, both the frameworks and training algorithms of CNNs have been improved [22], [23]. The breakthrough of the deep learning is a fast learning algorithm for deep belief networks proposed in [24], a learning algorithm that greedily trains one layer at a time. They exploited the restricted Boltzmann machine (RBM) [25] and an unsupervised learning algorithm, for each layer. Shortly after, related algorithms based on autoencoders, which apparently exploit the same principle, were proposed [26]. Recently, regularization methods for deep networks have been proposed to improve the performance of the networks [27]–[29]. Dropout training, whose key idea is to randomly drop units from a neural network during training, significantly reduces overfitting [27]. In addition, Goodfellow *et al.* [28] proposed maxout network. It is designed to both facilitate optimization by dropout and improve the accuracy of dropout's fast approximate model averaging technique. Moreover, the benefit of a network with local winner-take-all blocks was demonstrated in [29]. Deep learning has
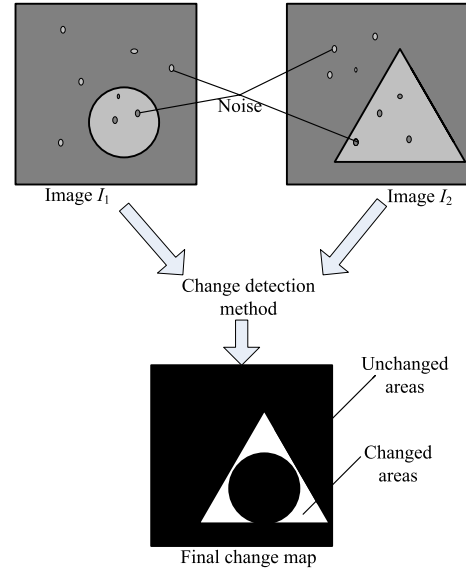

Fig. 1. Change detection problem.

important empirical successes in a number of traditional artificial intelligence applications, such as natural language processing [30] and image processing [31]–[33], especially the classification [34]–[37] and recognition [38]–[41] tasks.

In this paper, we try to apply deep learning to SAR images rather than standard images. The proposed algorithm includes three aspects as follows: 1) preclassification for obtaining some data with labels of high accuracy; 2) constructing a deep neural network for learning images features, and then fine-tuning the parameters of the neural network; and 3) using the trained deep neural network for the classification of changed and unchanged pixels.

The rest of this paper is organized as follows. Section II will give problem statements and the proposed algorithm framework. In Section III, the proposed method will be described in detail. Section IV will present the experimental results on real multitemporal SAR images to verify the feasibility of the method. Finally, the conclusion is drawn in Section V.

II. PROBLEM STATEMENTS AND ALGORITHM FRAMEWORK

The two coregistered intensity SAR images $I_1 = \{I_1(i, j), 1 \leq i \leq A, 1 \leq j \leq B\}$ and $I_2 = \{I_2(i, j), 1 \leq i \leq A, 1 \leq j \leq B\}$ are considered, which have the same size $A \times B$ and are acquired over the same geographical area at two different times t_1 and t_2, respectively. The two original images are polluted by noise. The change detection problem can be shown in Fig. 1. We should design efficient change detection methods to find the changes between the two images.

The procedure of commonly used change detection method in SAR images can be divided into three steps: 1) image preprocessing; 2) generation of a DI; and 3) analysis of the DI [42]. The framework is shown in Fig. 2(a). In general, geometric correction and registration are usually implemented to align two images in the same coordinate frame before change detection. In the first step, it is only denoising that we need to consider. As described in Section I, in the three steps,
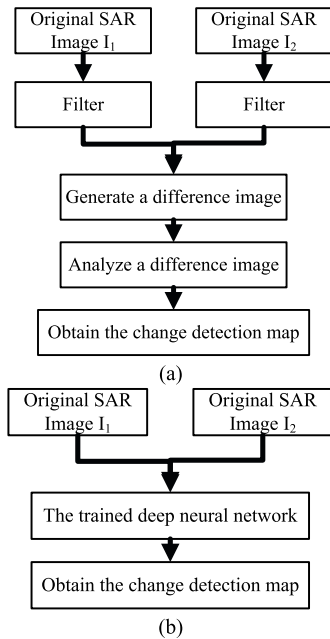
(a)

(b)

Fig. 2. Algorithm frameworks for the change detection problem. (a) Framework based on DI-analysis. (b) Framework based on the proposed deep neural networks.

Fig. 3. Flowchart of generating the deep neural network.

every step is actually seen as a separate issue. Most previous studies deal, respectively, with the three issues and propose different algorithms for each issue, respectively. Our objective aims at simplifying the change detection issue without the two steps of filtering or generating a DI. The final detection results can be directly obtained from the two original images. Deep neural networks that have the powerful ability to learn the complicated relationships from the two images prove to be the first choice to realize the target. Fig. 2(b) shows a simple framework of this method.

Here, we analyze the cause of choosing deep neural networks. Shallow networks cannot efficiently represent the task of interest [43]. The two original images without being filtered have complex relationships and we want to achieve the changes directly from the two images. Traditional deep neural networks would have sufficient representational power to encode the task, but they are difficult to train because error gradients decay exponentially with depth. We, therefore, adopt a layerwise unsupervised pretraining strategy to train the deep network. A deep architecture is composed of multiple levels of nonlinear operations, such as in neural nets with many hidden layers, which is different from shallow neural networks. Deep learning algorithms can discover multiple levels of distributed representations, with higher levels representing more abstract concepts. Automatically learning features without supervision at multiple levels of abstraction allows a system to learn complex functions, which maps the input to the output directly from the data. For change detection issue, deep neural networks are able to learn the nonlinear relations from the two original images. There is no need to filter or generate a DI, which reaches the goal that we should enable change detection process to be concise to some extent.

Change detection is often applied to disaster evaluation or medical diagnosis. It is difficult to obtain the prior knowledge
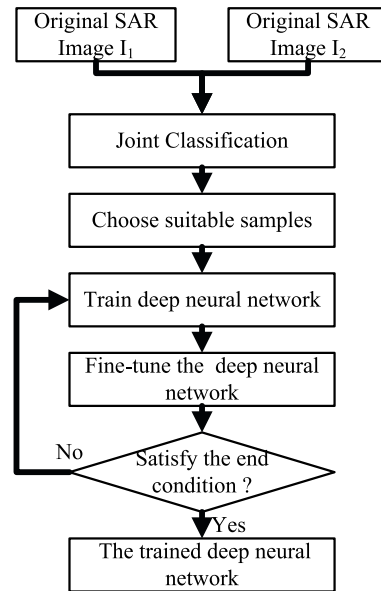
in these practical applications. Therefore, unsupervised change detection methods are urgently needed and very important. Deep neural network is a method for unsupervised feature learning and supervised classification. It can learn from the data sets, which have a few labeled data. In addition, some labeled data can be obtained by a preclassification. The flowchart of generating the deep neural network is shown in Fig. 3.

III. METHODOLOGY

A. Preclassification and Sample Selection

To avoid generating a DI, we make a joint classification of the two original images. Here, a joint classifier of the two original images based on FCM [joint classifier based on FCM (JFCM)] is used. The method takes gray levels as inputs. The similarity of gray levels relating to two pixels at the corresponding position in the two images is obtained through similarity operator. Then, the global threshold value of similarity is gotten, which is used to control the joint classifier to classify the two images. The classification results are represented by $\Omega = \{\Omega^1, \Omega^2\}$.

The specific process is shown in Algorithm 1. The similarity of gray levels relating to two pixels at the corresponding position (i, j) in the two original images is defined as

$$S_{ij} = \frac{|I_{ij}^1 - I_{ij}^2|}{I_{ij}^1 + I_{ij}^2} \qquad (1)$$

where $S_{ij} \in [0, 1]$, and I_{ij}^t is the gray level at the position (i, j) in the t-temporal image ($t = 1, 2$). The global threshold value (T) of similarity is gotten by iterative threshold method [44]. The classifier determines reference points for classification according to the principle of minimum variance. The variance at the position (i, j) in the t-temporal image is given by

$$\delta_{ij}^t = \omega_{ij}^t \left(I_{ij}^t - G_{ij} \right)^2 \qquad (2)$$

Algorithm 1 Joint Classification Based on FCM

Input: Two original images I_1 and I_2
　　　Set end conditions
　　　while (! end conditions)
　　　　for each $i \in [1, A]$ do
　　　　　for each $j \in [1, B]$ do
　　　　compute similarity of gray-levels S_{ij} and variance
　　　　　　　　　$\delta_{ij}^1, \delta_{ij}^2$
　　　　　　　if $\delta_{ij}^1 \leq \delta_{ij}^2$
　　　　classify I_{ij}^1 using FCM algorithm
　　　　　　　　if $S_{ij} \leq T$　// T is the global
　　　　threshold value of similarity
　　　　　　　$\Omega_{ij}^2 = \Omega_{ij}^1$
　　　　　　else classify I_{ij}^2 using FCM algorithm
　　　　　　end if
　　　　　else classify I_{ij}^2 using FCM algorithm
　　　　　　　　if $S_{ij} \leq T$
　　　　　　$\Omega_{ij}^1 = \Omega_{ij}^2$
　　　　　　else classify I_{ij}^1 using FCM algorithm
　　　　　end if
　　　　　　　end if
　　　　　　end for
　　　　end for
　　　end while
Output: $\Omega = \{\Omega^1, \Omega^2\}$

---

where

$$\omega_{ij}^t = I_{ij}^t / (I_{ij}^1 + I_{ij}^2) \tag{3}$$

$$G_{ij} = \sum_{t=1}^{2} \omega_{ij}^t I_{ij}^t \tag{4}$$

where ω_{ij}^t represents the weight of gray value and G_{ij} represents the weighted average gray value. Therefore, (3) and (4) are substituted into (2) and δ_{ij}^t are also written by

$$\delta_{ij}^t = \frac{I_{ij}^t}{I_{ij}^1 + I_{ij}^2} \left[I_{ij}^t - \frac{(I_{ij}^1)^2 + (I_{ij}^2)^2}{I_{ij}^1 + I_{ij}^2} \right]^2. \tag{5}$$

Derived from (1) and (5), we obtain the following two equations:

$$\delta_{ij}^1 = I_{ij}^2 \frac{I_{ij}^1 I_{ij}^2}{I_{ij}^1 + I_{ij}^2} [S_{ij}]^2 \tag{6}$$

$$\delta_{ij}^2 = I_{ij}^1 \frac{I_{ij}^1 I_{ij}^2}{I_{ij}^1 + I_{ij}^2} [S_{ij}]^2. \tag{7}$$

If $\delta_{ij}^1 \geq \delta_{ij}^2$, then $I_{ij}^1 \leq I_{ij}^2$. Suppose that reference points for classification are chosen according to the principle of maximum variance. I_{ij}^1 is viewed as the reference point. When $S_{ij} \leq T$, $\Omega_{ij}^2 = \Omega_{ij}^1$ and when $S_{ij} > T$, $\Omega_{ij}^2 \neq \Omega_{ij}^1$. After a few iterations of classification, the category to which I_{ij}^2 belongs has a greater clustering center than the one to which I_{ij}^1 belongs. In this case, the possibility that changed information in different images has the same label increases,
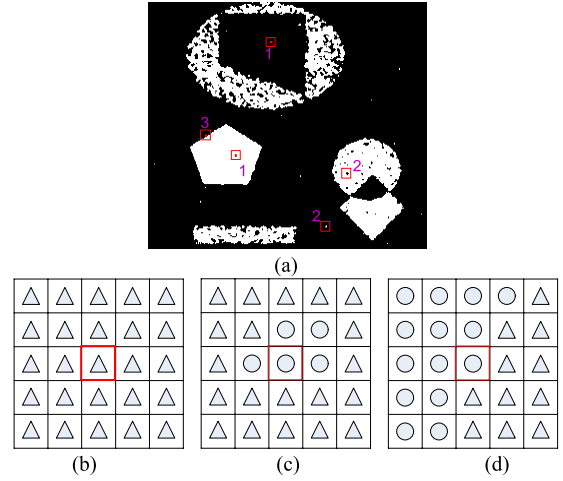

Fig. 4. Example to illustrate how to choose suitable samples. $\triangle$ and $\bigcirc$: two classes, respectively. (a) Simulated image. (b) Point 1. (c) Point 2. (d) Point 3.

which causes changed information lost. Suppose that reference points for classification are chosen according to the principle of minimum variance. I_{ij}^2 is seen as the reference point. When $S_{ij} \leq T$, $\Omega_{ij}^1 = \Omega_{ij}^2$ and when $S_{ij} > T$, $\Omega_{ij}^1 \neq \Omega_{ij}^2$. Through a few iterations of classification, the category to which I_{ij}^1 belongs has a similar clustering center with the category to which I_{ij}^2 belongs. It makes sure that in different images, the unchanged information has the same label, and the changed information has different labels.

The results of preclassification are not entirely correct. The pixels that have a high possibility of being correctly classified are chosen to train the networks. Suppose that the pixel p_{ij} at the position (i, j) in the preclassification result map has the label Ω_{ij}. Let N_{ij} be a neighborhood with a center at the position (i, j) and of size $n \times n$. A simulated image, as an example, is shown in Fig. 4, and there are three kinds of points should be considered. A point in the changed areas or unchanged areas, such as Point 1 in Fig. 4, has a neighborhood in which almost all pixels have the same labels with it. A point on the edge, such as Point 3 in Fig. 4, has a neighborhood in which about half all pixels have the same labels with it. These two kinds of point can be chosen as samples. A point that is wrongly classified (also called noise spot), such as Point 2 in Fig. 4, has a neighborhood in which a few or none of the pixels have the same labels with it. This kind of point should be eliminated. In conclusion, if the point p_{ij} has a neighborhood N_{ij}, which satisfies (8), the point p_{ij} can be chosen as a sample

$$\frac{Q(p_{\xi\eta} \in N_{ij} \wedge \Omega_{\xi\eta} = \Omega_{ij})}{n \times n} > \alpha \tag{8}$$

where the point $p_{\xi\eta}$ is in the neighborhood N_{ij}, and $Q(p_{\xi\eta} \in N_{ij} \wedge \Omega_{\xi\eta} = \Omega_{ij})$ means the number of pixels with the label equals to Ω_{ij} in the neighborhood N_{ij}. The parameter α, which decides whether the p_{ij} is chosen as a sample, is very important. α cannot be set too small or too large. If α is set too small, the result will not be robust to noise; and if α is set too large, the diversity of samples will decrease, which results in more missed alarms.

B. Deep Neural Network Establishment

Training a deep neural network is the core part of the algorithm. It is difficult to optimize the weights and biases in nonlinear networks, which have multiple hidden layers. Starting with random weights, multilayer BP network cannot always find a satisfactory result. If the initial weights are large, the result typically traps into local optimization. However, small initial weights lead the gradients in the early layers to be tiny, thus making it infeasible to train networks with many hidden layers. The initial weights close to a good solution can make gradient descent works well, but finding such initial weights requires a very different type of algorithm that learns one layer of features at a time. The RBM [37] can help to solve the problem.

The process of the proposed method is shown in Fig. 5. First, inputting neighborhood features of each position. Second, a stack of RBM is learned for pretraining. Next, the RBMs are unrolled to create a deep neural network for training. In addition, the deep neural network is fine-tuned using BP of error derivatives [45].

As shown in Fig. 5(a), let $N_{ij}^{I_1}$ represents a neighborhood with a center at position (i, j) and of size $n \times n$ in image I_1. The size of neighbor may influence the change detection results. The analysis of the parameter will be made next. $N_{ij}^{I_2}$ represents a corresponding neighborhood in image I_2. Convert $N_{ij}^{I_1}$ and $N_{ij}^{I_2}$ to vectors $M_{ij}^{I_1}$ and $M_{ij}^{I_2}$, respectively. The feature vector of the sample at the position (i, j) can be written by $M_{ij} = [M_{ij}^{I_1} \ M_{ij}^{I_2}]$ (M_{ij} is the connection of $M_{ij}^{I_1}$ with $M_{ij}^{I_2}$). The values of features lie between 0 and 1 and are non-Gaussian, so we use logistic output units.

An ordinary structure of RBM network is shown in Fig. 6. RBM has l visible units $(v_1, v_2, \ldots, v_l)$ corresponding to features of its inputs and m hidden units $(h_1, h_2, \ldots, h_m)$ that are trained. In addition, each connection in an RBM must connect a visible unit to a hidden unit. $W_{l \times m}$ represents a weight matrix between visible layer and hidden layer; $b = (b_1, b_2, \ldots, b_l)$ are biases of visible units and $c = (c_1, c_2, \ldots, c_m)$ are biases of hidden units. A joint configuration (v, h) of the visible and hidden units has an energy [45], [46] given by

$$E(v, h) = - \sum_{i \in \text{pixels}} b_i v_i - \sum_{j \in \text{features}} c_j h_j - \sum_{i,j} v_i h_j W_{ij}. \tag{9}$$

Suppose that $\forall i, j, v_i \in \{0, 1\}, h_j \in \{0, 1\}$. For a given v, the binary state, h_j, of each hidden unit, j, is set to 1 with the probability

$$P(h_j = 1 | v) = \sigma \left(\sum_{i=1}^{l} W_{ij} \times v_i + c_j \right) \tag{10}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is a sigmoid function. Once binary states have been chosen for the hidden units, a reconstructive data are produced by setting each v_i to 1 with the probability

$$P(v_i = 1 | h) = \sigma \left(\sum_{j=1}^{m} W_{ji} \times h_j + b_i \right). \tag{11}$$
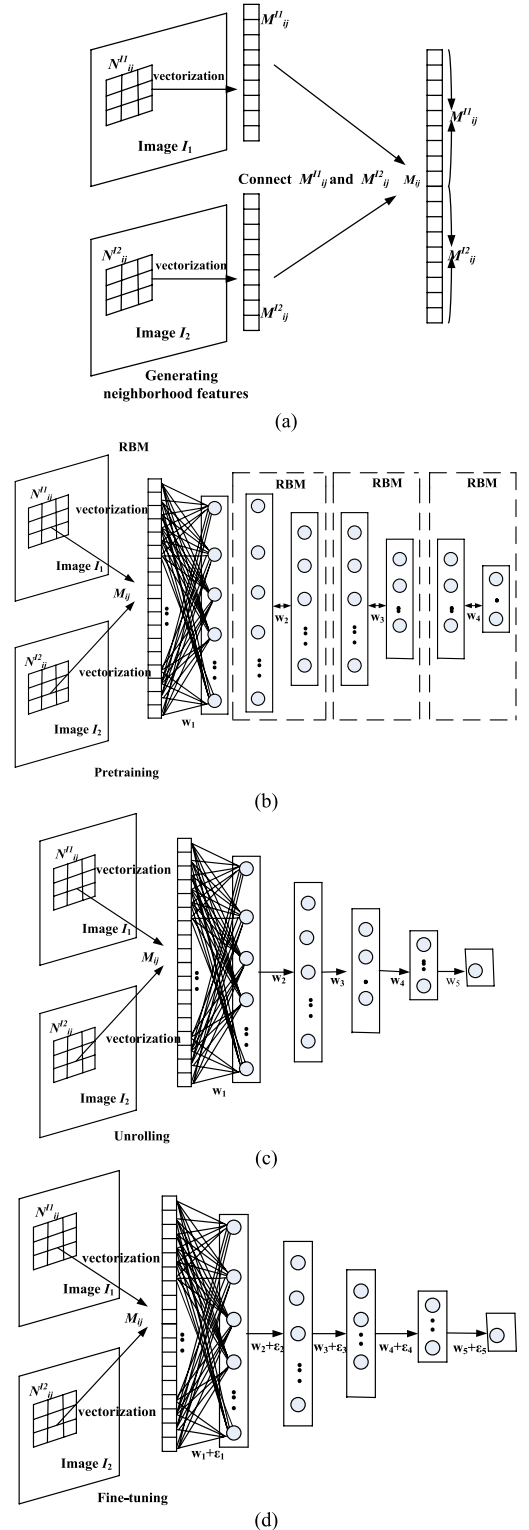

Fig. 5. Training the deep neural network for the change detection problem. (a) Neighborhood features for each position as inputs. (b) RBMs are used for pretraining. (c) After pretraining, RBMs are unrolled to create a deep neural network. (d) Fine-tuning using BP.

The states of the hidden units are then updated once more so that they represent features of the reconstructive data. The change in a weight is given by

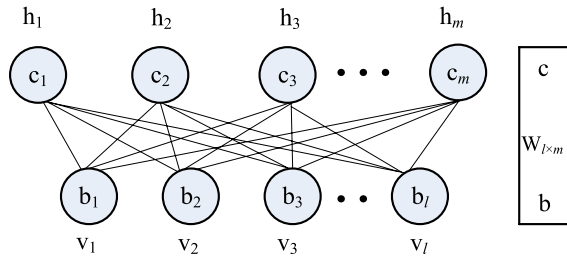$$\Delta W_{ij} = \varepsilon(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{re}}) \tag{12}$$
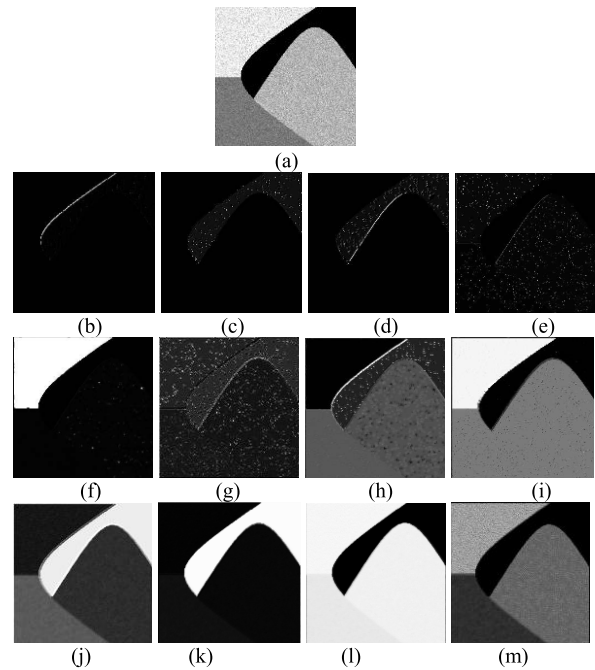
Fig. 6.   Structure of RBM.

Fig. 7.   Feature images extracted from hidden layers. (a) Input SAR image. (b)–(e) Feature images extracted from the first layer. (f)–(i) Feature images extracted from the second layer. (j)–(m) Feature images extracted from the third layer.

where ε is a learning rate and $\langle v_i h_j \rangle_{\text{data}}$ is the fraction for data when the feature detectors are being driven by data, and $\langle v_i h_j \rangle_{\text{re}}$ is the corresponding fraction for reconstructions. A simplified version of the same learning rule is used for the biases.

A two-layer RBM network in which stochastic, binary features are connected to stochastic, binary feature detectors using symmetrically weighted connections can be used to model neighborhood features. The features correspond to visible units of the RBM because their states are observed; the feature detectors correspond to hidden units. The network assigns a probability to every possible pixel via this energy function in (9) [47]. Given the set of training samples, which are chosen previously, a stack of RBM is used to pretraining. The pretraining does not use any information of the class labels. Here, a layer-by-layer learning algorithm is applied. As shown in Fig. 5(b), the output of the upper layer is taken as the input of the next layer. Here, every layer is a two-layer RBM network, which is trained according to the rules described previously.

After pretraining, the RBM model is unfolded to produce a deep neural network that initially uses the same weights and biases, as shown in Fig. 5(c). The cross-entropy error backpropagation strategy is used through the whole network to fine-tuned the weights for optimal classification, as shown in Fig. 5(d). The cross-entropy error is represented by

$$E = -\sum_i e_i \log \hat{e}_i - \sum_i (1 - e_i) \log(1 - \hat{e}_i) \qquad (13)$$

where e_i is the label of the sample i and $\hat{e}_i$ is the classification result.

By training and fine-tuning the network, the final deep neural network is established. The neighborhood features of each position are fed into the deep neural network. In addition, the network outputs the class label of the pixel. The class label 0 represents the pixel being changed, and the class label 1 represents the pixel being unchanged.

C. Feature Analysis and Denoising

Therefore, what features do the deep network learns and why can it perform well? During training of each layer, hidden layer is trained to represent the dominate factors of the data in visible layer. After being trained, the RBM gives a reliable representation of the input data statistically. The knowledge of the input data is learned. In addition, in higher layers, the knowledge is more abstract and helpful to predict

the output. To observe the layerwise process clearly, we input an SAR image, as shown in Fig. 7(a), and train the network without supervision. Then, we extract typical features of units in each layer and draw the feature images, as shown in Fig. 7(b)–(m).

It can be seen in Fig. 7 that different hidden layers learn different features. In low layers, such as the first hidden layer, the features learned are simple shapes, including lines and points. Typically, Fig. 7(b) labels an obvious line in input image, Fig. 7(c) shows noises in an area, and Fig. 7(e) shows all the noises and a line. In the second layer, some structure features are showed, but noises are obvious. In high layers, such as the third layer, the features are more complex than that in lower layers. Such as Fig. 7(k), it gets rid of the impact of noise, but two different areas are recognized as one. In Fig. 7(m), it recognizes four areas, but it is corrupted by some unobvious noises. It can be seen that the deep network finally learns the major features of the input image and weaken the influence of noises. Deep learning applies to not only standard images, which are the main data sources in the previous research, but also SAR images. Although SAR images have speckle noise that is very difficult to tackle.

The feature images, shown in Fig. 7, are typical features extracted from hidden layers. In addition, the actual number of features is very large. A feature image can be seen as a kind of interpretation of the input image. Maybe one kind of interpretation is not complete, but a large number of features can represent the image well. Therefore, the network has grasped the skill of how to interpret and represents the image after unsupervised learning. However, it has no idea what to
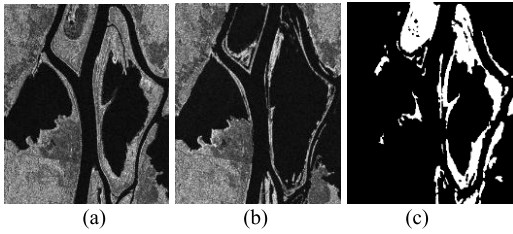
Fig. 8. Multitemporal images relating to Ottawa. (a) Image acquired in July 1997, during the summer flooding. (b) Image acquired in August 1997, after the summer flooding. (c) Ground truth.
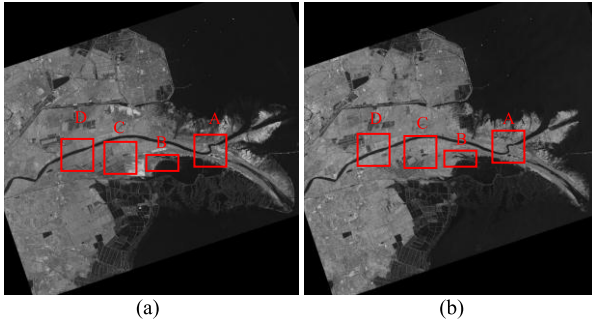

Fig. 9. Multitemporal images relating to Yellow River Estuary. (a) Image acquired in June 2008. (b) Image acquired in June 2009.

do with the features learned and how to deal with the image. Therefore, examples are needed.

IV. EXPERIMENTAL STUDY

A. Introduction to Data Sets

The Ottawa data set is a section (290×350 pixels) of two SAR images over the city of Ottawa acquired by RADARSAT SAR sensor and provided by the Defence Research and Development Canada, Ottawa. The available ground truth (reference image), which is shown in Fig. 8(c), was created by integrating prior information with photointerpretation based on the input images [Fig. 8(a) and (b)]. The experiment on Ottawa data set is an instance of disaster evaluation. The changed areas represent the affected areas.

The Yellow River data set used in the experiments consists of two SAR images acquired by Radarsat-2 at the region of Yellow River Estuary in China in June 2008 and June 2009, as shown in Fig. 9. It is worth noting that the two images are single-look image and four-look image, respectively. This means that the influence of speckle noise on the image acquired in 2008 is much greater than that of the one acquired in 2009. The huge difference of speckle noise level between the two images used may complicate the processing of change detection. The original size of these two SAR images acquired by Radarsat-2 is 7666×7692. They are too huge to show the detail information in such small pages.

We select four typical areas (the Inland water, the coastline, and the two farmlands), where different kinds of changes occur. Inland water where the changed areas are concentrated on the borderline of the river is shown in Fig. 10. It is comparatively hard to detect. Fig. 11 shows the coastline where the changed areas are relatively small, compared with
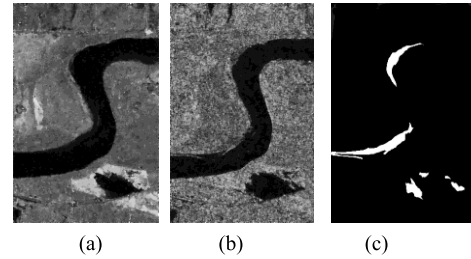

Fig. 10. Multitemporal images relating to Inland water of Yellow River Estuary. (a) Image acquired in June 2008. (b) Image acquired in June 2009. (c) Ground truth.
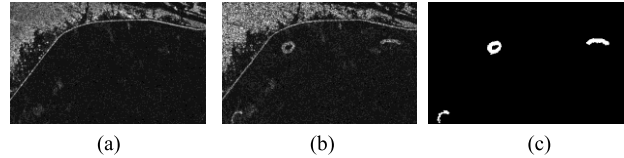

Fig. 11. Multitemporal images relating to Coastline of Yellow River Estuary. (a) Image acquired in June 2008. (b) Image acquired in June 2009. (c) Ground truth.
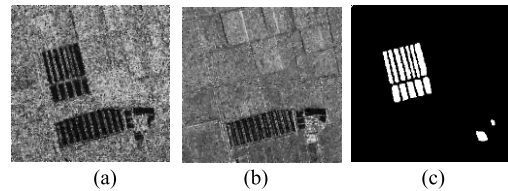

Fig. 12. Multitemporal images relating to Farmland C of Yellow River Estuary. (a) Image acquired in June 2008. (b) Image acquired in June 2009. (c) Ground truth.
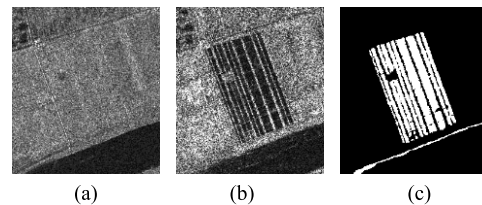

Fig. 13. Multitemporal images relating to Farmland D of Yellow River Estuary. (a) Iimage acquired in June 2008. (b) Image acquired in June 2009. (c) Ground truth.

the farmlands with large and regular changes. It can be seen that the changed areas appear as newly reclaimed farmlands in Figs. 12 and 13, respectively. The experiments on Yellow River data sets are instances of environmental monitoring. The changed areas show the changes on the surface.

B. Evaluation Criteria

The quantitative analysis of change detection results is set as follows: 1) the false negatives (FNs) (changed pixels that undetected) and 2) the false positives (FPs) (unchanged pixels wrongly detected as changed) should be calculated. Overall error (OE) is the sum of FN and FP [48]. In the experiments, they are reported as percentages. We calculate the percentage
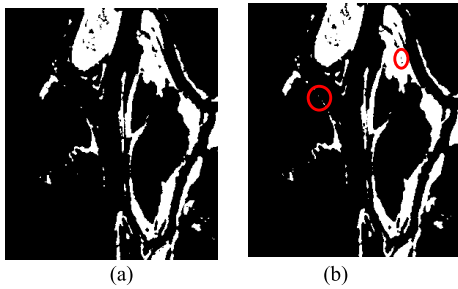
Fig. 14. Change detection results of Ottawa data set achieved by (a) supervised method and (b) proposed method.

correct classification (PCC). It is given by

$$PCC = (TP + TN)/(TP + FP + TN + FN) \qquad (14)$$

where TP is short for true positives, which is the number of pixels that are detected as the changed area in both the reference image and the result. TN is short for true negatives, which is the number of pixels that are detected as the unchanged area in both the reference image and the result. For accuracy assessment, Kappa statistic is a measure of accuracy or agreement based on the difference between the error matrix and chance agreement [49]. According to the map, we count the actual number of pixels belonging to the unchanged class and changed class (written as Mu and Mc, respectively). Kappa is calculated as

$$Kappa = \frac{PCC - PRE}{1 - PRE} \qquad (15)$$

where

$$PRE = \frac{(TP + FP) \cdot Mc + (FN + TN) \cdot Mu}{(TP + FP + FN + TN)^2}. \qquad (16)$$

C. Performance of the Preclassification and Sample Selection

The first set of experiments aims at proving it effective that samples obtained by preclassification are used for fine-tuning of the deep neural network. A supervised deep learning algorithm is used to make a comparison with the proposed method. In both the methods, training and testing are implemented on the same image. It is noted that in the proposed method, the samples used for training are the pixels that should satisfy the rule for selecting samples, not all the pixels in the image. To be fair, in the two methods, we use the same network topology and the same training set, which has a few samples accounting for 10% of the total pixels. However, in different methods, the training set has different labels. In the supervised method, the labels are given according to the ground truth, and in the proposed method, the labels are given according to the preclassification results. Although the testing set has all the pixels of an image, we remove the samples used for training from the testing set when calculating the evaluation criteria. The experiments are carried out on the Ottawa data set and the Farmland C data set, one from water area and the other from farmland.

Figs. 14 and 15 show the final maps on the Ottawa data set and the Farmland C data set. It can be seen that the proposed
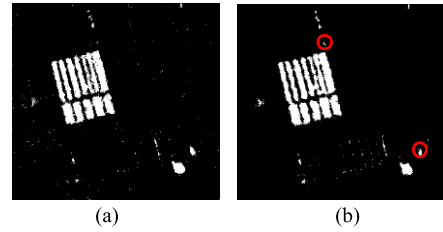

Fig. 15. Change detection results of Farmland C data set achieved by (a) supervised method and (b) proposed method.

TABLE I
CHANGE DETECTION RESULTS OF OTTAWA DATA SET AND FARMLAND C DATA SET OBTAINED BY SUPERVISED AND PROPOSED METHODS

Dataset	Criterion	Method	
		Supervised Method	Proposed Method
Ottawa	FP (%)	0.47	0.56
	FN (%)	0.67	0.61
	OE (%)	1.14	1.17
	PCC (%)	98. 86	98.83
	Kappa(%)	95.68	95.59
Farmland C	FP (%)	0.63	0.73
	FN (%)	0.75	0.71
	OE (%)	1.38	1.44
	PCC (%)	98.62	98.56
	Kappa(%)	87.59	86.92

method achieves at similar results to the supervised method visually. After all, due to the samples are not from the ground truth in the proposed method, there are some obvious false alarms or missed alarms, such as the areas surrounded by the red circles in Figs. 14(b) and 15(b). Furthermore, a quantitative comparison between the two methods on the two data sets is shown in Table I. For the Ottawa data set, the PCC yielded by the proposed method equaling to 98.83% approaches that of 98.86% by the supervised method. It is because that the two sample sets, respectively, in the two methods have the same number for a fair comparison. For the Farmland C data set, the PCC resulted by the proposed method is 98.56%, a little lower than but close to that of the supervised method. In conclusion, the proposed method can have similar effect to the supervised method on the change detection to some extent, which makes sure that it is feasible to use the samples obtained by preclassification for fine-tuning of deep neural network.

D. Performance of the Deep Learning Method

The second set of experiments is conducted on three data sets (Inland water data set, Coastline data set, and Farmland D data set) to analyze the impact of deep learning on the change detection results. The three data sets are all selected from the Yellow River data set. It is a greater challenge for dealing with the change detection. For all data sets, every hidden layer is pretrained 50 passes through the entire training set and a 50-250-200-100-1 network is used. In addition, a clustering algorithm RFLICM [5], a threshold algorithm GKI [1], and JFCM are presented as comparative methods, respectively. In the tables, we bold the best results.
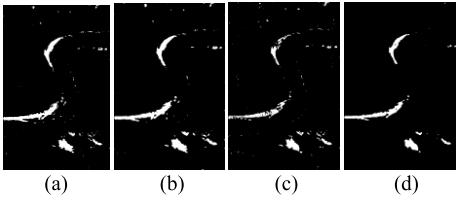
Fig. 16. Change detection results of Inland water data set achieved by (a) GKI, (b) RFLICM, (c) JFCM, and (d) proposed method.
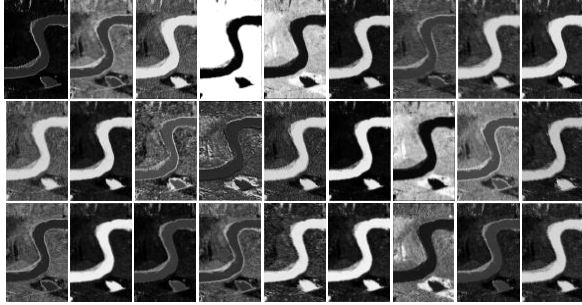

Fig. 17. Feature images of Inland water data set extracted from the third layer.

1) Results on the Inland Water Data Set: The change detection results generated by the proposed method and the three comparative methods on the Inland water data set are presented in Fig. 16. As shown in Fig. 16(a), the final map generated by GKI is polluted by many white noise spots. It is due to the necessity of modeling data for searching an optimal threshold in the GKI method. Any little error in threshold may result in the presence of noise on the final map. As for RFLICM, there are also many noise spots emerging on the black ground. Clustering methods are sensitive to noise. Although RFLICM, which incorporates both local spatial and gray information, is an improved method, it cannot give a best result in this case. Since the influence of the speckle noise on the Yellow River data set is much great. JFCM is used for preclassification just because a DI can be avoided. In the method, FCM is separately implemented on two original images. Although a joint-classification strategy is used, false alarms or missed alarms are high. By contrast, the proposed method applying deep learning has an obvious improvement. In particular, it appears much effective without a DI generated. In Fig. 17, we exhibit some feature images extracted from the highest hidden layer. It is clear that the deep network is able to learn meaningful features and overcome the noise that is characteristic of SAR. The features learnt are local features and the stacked RBMs can represent the difference features of the two images. In Fig. 17, different feature images have different representations and even some feature images highlight the changed areas. Even if the two images have different looks of speckle, deep learning learns the meaningful features of the images and restricts the impact of noise. Furthermore, with reliably training samples, which are selected to avoid the influence of noise, the method is robust. That is why FP yielded by the proposed method is much lower. In Table II, the performances of the four methods are given quantitatively. The proposed method exhibits the best results.

TABLE II
CHANGE DETECTION RESULTS OF INLAND WATER DATA SET

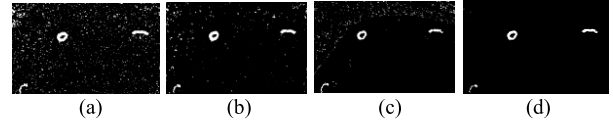| Method	FP(%)	FN(%)	OE(%)	PCC (%)	Kappa (%)
GKI	0.90	1.21	2.11	97.88	69.44
RFLICM	0.97	1.04	2.01	97.98	71.74
JFCM	0.60	1.20	1.80	98.19	68.94
Proposed Method	**0.27**	**1.04**	**1.31**	**98.67**	**76.45**

Fig. 18. Change detection results of Coastline data set achieved by (a) GKI, (b) RFLICM, (c) JFCM, and (d) proposed method.
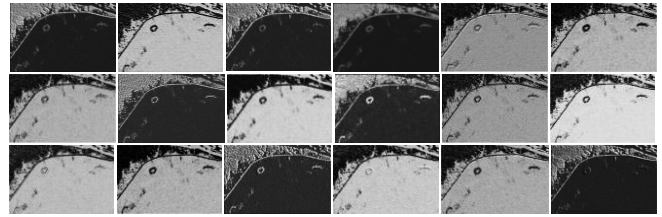

Fig. 19. Feature images of Coastline data set extracted from the third layer.

TABLE III
CHANGE DETECTION RESULTS OF COASTLINE DATA SET

Method	FP(%)	FN(%)	OE(%)	PCC (%)	Kappa (%)
GKI	4.16	**0.09**	4.25	95.74	30.25
RFLICM	1.08	0.13	1.21	98.78	59.98
JFCM	1.34	0.17	1.51	98.48	48.51
Proposed Method	**0.02**	0.16	**0.18**	**99.81**	**88.76**

2) Results on the Coastline Data Set: For the Coastline data set, Fig. 18 shows the final maps of the four methods. GKI presents the worst performance. The final map swarms with many white spots. In Fig. 18(b), the final map obtained by RFLICM has many false alarms because of the existence of noise. As shown in the figure, the proposed method based JFCM is the best to complete the detection task. Fig. 19 shows some features images learned by the network. It can be seen that noises are not obvious. Moreover, quantitative analysis in Table III also declares this point. The PCC yielded by the proposed method equals to 99.81% is higher than that of 98.48% by JFCM, and has a big promotion. Serving as an overall evaluation, PCC and Kappa of the proposed method exhibit best among, although FN of it is not best. Besides, our method has another advantage that it can result in lower FP.

3) Results on the Farmland D Data Set: The results of the experiments on the Farmland D data set are shown in Fig. 20 and listed in Table IV, respectively. The proposed method appears much better than the three other methods. In Table IV, our method exhibits the best PCC and Kappa. In all, the proposed method wins in the competition both from the figure and from the table. For both the land and the water areas, our method can present good performance.
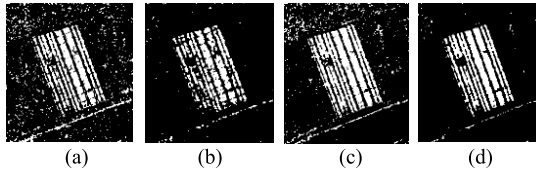
Fig. 20.   Change detection results of Farmland D data set achieved by (a) GKI, (b) RFLICM, (c) JFCM, and (d) proposed method.

TABLE IV

CHANGE DETECTION RESULTS OF FARMLAND D DATA SET

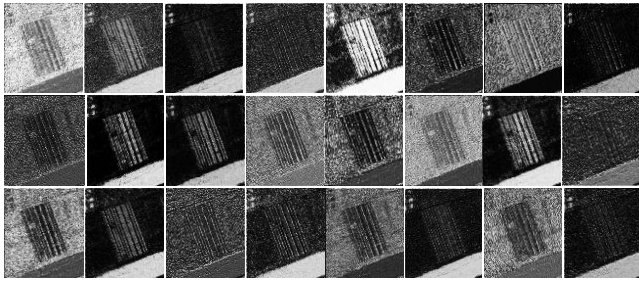| Method	FP(%)	FN(%)	OE(%)	PCC (%)	Kappa (%)
GKI	5.23	5.06	10.30	89.65	65.35
RFLICM	0.47	7.24	7.72	92.27	69.47
JFCM	3.67	3.29	6.96	93.03	76.67
Proposed method	**0.39**	**4.89**	**5.29**	**94.70**	**80.20**

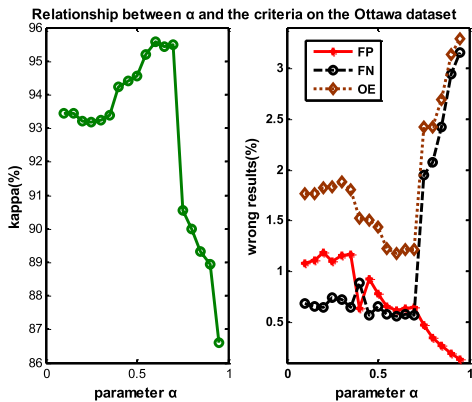Fig. 21.   Feature images of Farmland D data set extracted from the third layer.

Fig. 22. Relationship between the parameter α and the criteria on the Ottawa data set.

Feature images selected in Fig. 21 show evidence of denoising. Deep network has no special requirement for data distribution and it just learns many useful features for tasks.

4) Analysis of Parameters: In the step of selecting suitable samples (Section III-A), α is an important parameter. We set α to 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, and 0.95 for testing it. The relationship between the parameter α and the criteria on the Ottawa data set is shown in Fig. 22. It is seen that the results are stable when α is between 0.45 and 0.7. When α is too small, the result is sensitive to noise, which yields a high FP; when α is
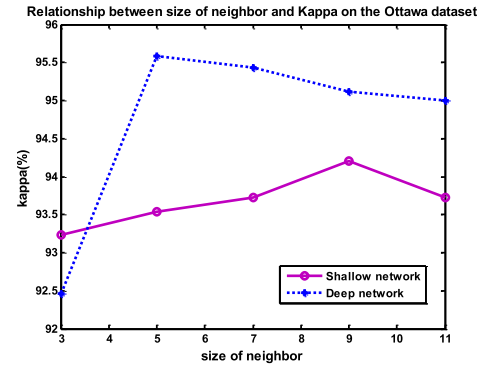

Fig. 23.   Relationship between the size of neighbor and Kappa on Ottawa data set.
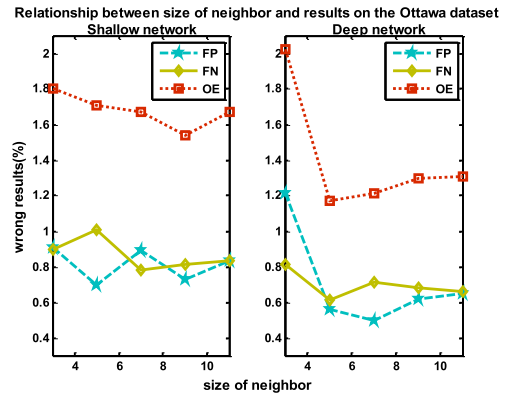

Fig. 24.   Relationship between the size of neighbor and FP, FN, OE on Ottawa data set.

too large, the diversity of sample is decreased, which yields a high FN. Furthermore, when α is set suitably, the proposed method can keep a balance between FP and FN.

To generate neighborhood features, we should select neighborhood of size $n \times n$ for each position (Section III-B). The size of the neighborhood has an effect on the final detection results. We set n to 3, 5, 7, and 9 to indicate the relationship between n and FP, FN, OE, Kappa on Ottawa data set. We analyze the parameter n using a deep network and a shallow network (a 50-250-1 architecture by experience). As shown in Figs. 23 and 24, if $n = 3$, the result by the deep network is even worse than that by the shallow network. This is because deep learning always requires lots of data and does not suitable for learning too few features. The difference of the results obtained by, respectively, setting n between 5 and 11 can be accepted for the two networks. However, when $n > 9$, the results by the two networks are both increasingly bad. Setting $n = 5$ is the best choice for the deep network but $n = 9$ for the shallow network. Furthermore, shown in Fig. 24, the results obtained by neural network actually can gain a balance between FP and FN.

Furthermore, in the process of training the deep neural network, we try to set various values of the weight decay, momentum, and learning rate to experiments. In addition, the RBMs are trained for more epochs. However, no any significant difference in the detection results after the fine-tuning is
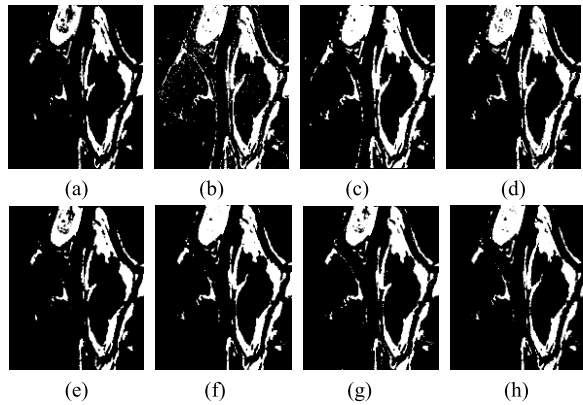
Fig. 25. Change detection results of Ottawa data set achieved by (a) RFLICM, (b) GKI, (c) LFS_KGC, (d) MRFFCM, (e) D_RFLICM, (f) D_GKI, (g) D_LFS_KGC, and (h) D_MRFFCM.

TABLE V
VALUES OF EVALUATION CRITERIA OF OTTAWA DATA SET

Criteria / Method	FP(%)	FN(%)	OE(%)	PCC (%)	Kappa (%)
RFLICM	0.16	2.49	2.66	97.33	89.35
D_RFLICM	**0.14**	**2.13**	**2.27**	**97.72**	**90.99**
GKI	2.76	**0.21**	2.98	97.02	89.49
D_GKI	**0.36**	1.34	**1.71**	**98.29**	**93.73**
GC	3.05	**0.22**	3.28	96.71	88.48
D_GC	**1.25**	1.04	**2.30**	**97.70**	**91.41**
MRFFCM	1.61	0.70	2.31	97.68	91.51
D_MRFFCM	**0.98**	**0.69**	**1.67**	**98.32**	**93.74**

Fig. 26. Change detection results of Inland water data set achieved by (a) RFLICM, (b) GKI, (c) LFS_KGC, (d) MRFFCM, (e) D_RFLICM, (f) D_GKI, (g) D_LFS_KGC, and (h) D_MRFFCM.

Fig. 27. Change detection results of Coastline data set achieved by (a) RFLICM, (b) GKI, (c) LFS_KGC, (d) MRFFCM, (e) D_RFLICM, (f) D_GKI, (g) D_LFS_KGC, and (h) D_MRFFCM.

observed when the depth of the network is given. As stated in [47], the precise weights found by the greedy pretraining do not matter as long as it finds a good region from which to start the fine-tuning. Such a problem that BP is easily to trap into local optimal can be avoided and the adjustment of parameters becomes less complicated. As for the depth configuration and every layer size, they are selected based on experience and many experiments. Our data sets are not enough large to call for more layers. We try different layer size, and actually the results have no much difference. In the experiments, we use a deep network with a 50-250-200-100-1 architecture and a shallow network with a 50-250-1 architecture.

E. Flexibility of the Proposed Deep Learning Algorithm

The third set of experiments is carried out on all the five data sets. Since deep learning algorithm contains a supervised fine-tuning stage, we can select some samples from the results obtained by the available algorithms, such as improved FCM algorithm based on Markov random field (MRFFCM) [13], GKI [1], RFLICM [5], and local fit-search model (LFS)_kernel-induced graph cuts (KGC) [11], respectively. We make a comparison between the four methods and the proposed algorithm based on the results obtained by the four methods.

1) Results on the Ottawa Data Set: As for the Ottawa data set, the results are shown in Fig. 25 and listed in Table V.
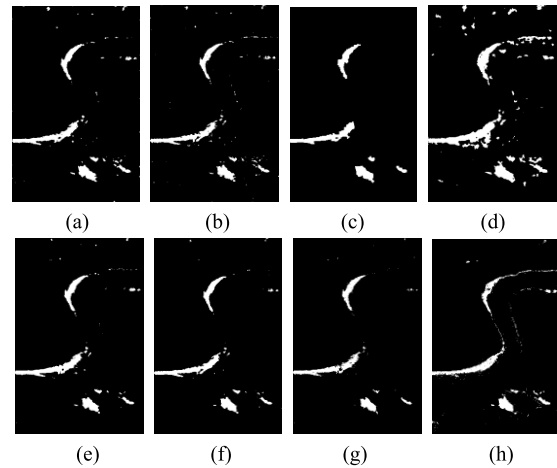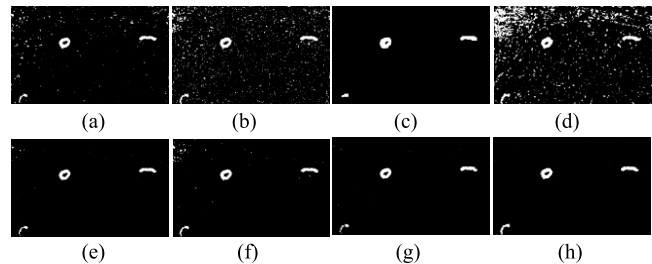
The influence of noise on the Ottawa data set is less great. Therefore, RFLICM, which applies local information, actually reduces the noise but causes the loss of details, as shown in Fig. 25(a). In addition, the FN yielded by RFLICM is as high as 2.49%. The proposed method based on the results of RFLICM performs better with the fact that the four criteria are improved. GKI, which needs to establish a model, is sensitive to the noise. It yields a very high FP. The proposed method based on the results of GKI can balance FP and FN, and significantly reduces OE. As for LFS_KGC, it can decrease the influence of speckle noise to some extent but would lead to poor results at the boundaries. The changed areas detected by LFS_KGC always have a good continuation and the final map generated by it has little discrete noise. Therefore, from the vision, the change detection map is good. It yields the highest FP and OE, as shown in Table V. The proposed method based on the result of LFS_KGC presents a better performance. MRFFCM is a clustering method with a modified MRF energy function. It can perform well on the Ottawa data set, which is influenced by noise less greatly. The proposed method based on the result of MRFFCM outperforms the other methods.

In all, the proposed method based on the results obtained by the available algorithms, respectively, arrives at significantly improved results. The deep learning algorithm has a strong capacity of learning features and can interpret images sufficiently. It can find a balance between denoising and
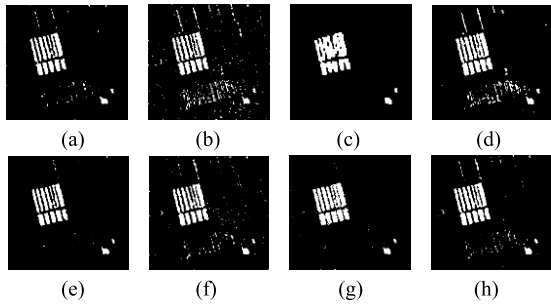
Fig. 28. Change detection results of Farmland C data set achieved by (a) RFLICM, (b) GKI, (c) LFS_KGC, (d) MRFFCM, (e) D_RFLICM, (f) D_GKI, (g) D_LFS_KGC, and (h) D_MRFFCM.
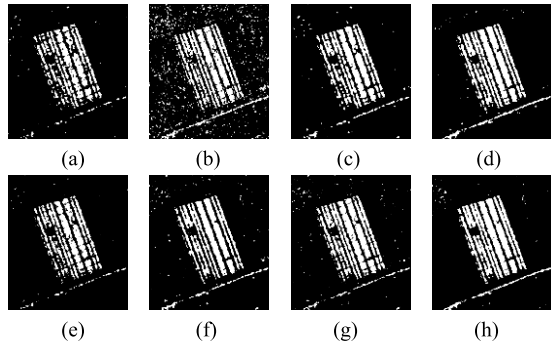

Fig. 29. Change detection results of Farmland D data set achieved by (a) RFLICM, (b) GKI, (c) LFS_KGC, (d) MRFFCM, (e) D_RFLICM, (f) D_GKI, (g) D_LFS_KGC, and (h) D_MRFFCM.

TABLE VI

VALUES OF EVALUATION CRITERIA OF INLAND WATER DATA SET

Criteria / Method	FP(%)	FN(%)	OE(%)	PCC (%)	Kappa (%)
RFLICM	0.97	1.04	2.01	97.98	71.74
D_RFLICM	**0.85**	**0.62**	**1.47**	**98.52**	**77.59**
GKI	0.90	1.21	2.11	97.88	69.44
D_GKI	**0.59**	**0.81**	**1.41**	**98.59**	**77.10**
GC	0.60	1.14	1.75	98.25	73.85
D_GC	**0.41**	**0.87**	**1.28**	**98.71**	**78.32**
MRFFCM	3.35	**0.67**	4.02	95.97	60.28
D_MRFFCM	**1.31**	1.15	**2.47**	**97.52**	**66.35**

TABLE VII

VALUES OF EVALUATION CRITERIA OF COASTLINE DATA SET

Criteria / Method	FP(%)	FN(%)	OE(%)	PCC (%)	Kappa (%)
RFLICM	1.08	**0.13**	1.21	98.78	59.98
D_RFLICM	**0.09**	0.17	**0.26**	**99.73**	**86.75**
GKI	4.16	**0.09**	4.25	95.74	30.25
D_GKI	**0.32**	0.13	**0.45**	**99.55**	**80.30**
GC	0.21	0.15	0.36	99.63	83.35
D_GC	**0.10**	0.15	**0.25**	**99.75**	**87.80**
MRFFCM	8.52	0.10	8.62	91.38	16.82
D_MRFFCM	**0.85**	**0.08**	**0.94**	**99.06**	**67.26**

TABLE VIII

VALUES OF EVALUATION CRITERIA OF FARMLAND C DATA SET

Criteria / Method	FP(%)	FN(%)	OE(%)	PCC (%)	Kappa (%)
RFLICM	0.53	1.00	1.53	98.46	85.68
D_RFLICM	**0.19**	**0.91**	**1.10**	**98.89**	**89.44**
GKI	2.87	0.97	3.84	96.15	69.96
D_GKI	**1.12**	**0.83**	**1.95**	**98.04**	**82.82**
GC	0.98	0.71	1.59	98.29	85.02
D_GC	**0.75**	**0.61**	**1.36**	**98.64**	**87.89**
MRFFCM	2.08	**0.53**	2.62	97.38	79.03
D_MRFFCM	**0.95**	0.63	**1.58**	**98.41**	**86.09**

TABLE IX

VALUES OF EVALUATION CRITERIA OF FARMLAND D DATA SET

Criteria / Method	FP(%)	FN(%)	OE(%)	PCC (%)	Kappa (%)
RFLICM	0.47	7.24	7.72	92.27	69.47
D_RFLICM	**0.23**	**6.71**	**6.94**	**93.05**	**72.75**
GKI	5.23	**5.06**	10.30	89.65	65.35
D_GKI	**0.93**	5.32	**6.26**	**93.74**	**76.65**
GC	**0.79**	5.50	6.29	93.70	76.35
D_GC	0.87	**4.82**	**5.69**	**94.30**	**79.00**
MRFFCM	**0.86**	4.00	4.86	95.13	82.37
D_MRFFCM	0.88	**3.60**	**4.49**	**95.51**	**83.91**

preserving details, which is reflected in a balance between FP and FN.

2) Results on the Yellow River Data Sets: As mentioned above, it is hard to detect the changes occurring on the Yellow River data set, which is influenced by noise much greatly. The results on the four typical areas are shown in Figs. 26–29 and listed in Tables VI–IX. Although the two clustering methods, RFLICM and MRFFCM, use local information for the purpose of suppressing noise, the final maps generated by them are polluted by some spots. The final maps generated by GKI swarm with noise but LFS_KGC keeps good performance of denoising. No matter the data set is from the water (Inland water and Coastline data sets) or the land (Farmland C and Farmland D data sets), the proposed method based on the results obtained by the available methods can arrive at

excellent results. In addition, no matter the change areas are small (Coastline data set) or large (Ottawa data set), regular (Inland water data set) or irregular (Farmland C and Farmland D data sets), the proposed method is applicative.

V. CONCLUSION

This paper has presented a novel change detection algorithm specifically toward analyzing multitemporal SAR images based on deep learning. Different from the traditional methods, the deep neural networks can achieve at the final maps directly from the two original images. Thus, the change detection issue is simplified as a classification problem even without generating a DI. We joint classify the two original images to realize preclassification. Next, a deep neural network is established to complete the change detection task. Compared with a clustering method RFLICM and a thresholding method GKI, the proposed method exhibits good performance. Moreover,

the proposed method based on the results obtained by the available methods can arrive at excellent results.

The experiments on the data sets, which have different features, verify the effectiveness of the proposed method. Furthermore, the deep neural network we construct can deal with two images having different noise characteristics, such as the Yellow River data set. In the future, we will do research on the application of deep learning to the change detection in two images, which have not been registered or come from different sensors.

REFERENCES

[1] Y. Bazi, L. Bruzzone, and F. Melgani, "An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 874–887, Apr. 2005.

[2] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 294–307, Mar. 2005.

[3] E. E. Kuruoglu and J. Zerubia, "Modeling SAR images with a generalization of the Rayleigh distribution," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 527–533, Apr. 2004.

[4] D. Haboudane and E.-M. Bahri, "Deforestation detection and monitoring in cedar forests of the moroccan Middle-Atlas mountains," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Barcelona, Spain, Jul. 2007, pp. 4327–4330.

[5] M. Gong, Z. Zhou, and J. Ma, "Change detection in synthetic aperture radar images based on image fusion and fuzzy clustering," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2141–2151, Apr. 2012.

[6] C. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images*. Norwood, MA, USA: Artech House, 1998.

[7] F. Bujor, E. Trouvé, L. Valet, J.-M. Nicolas, and J.-P. Rudant, "Application of log-cumulants to the detection of spatiotemporal discontinuities in multitemporal SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 10, pp. 2073–2084, Oct. 2004.

[8] J. Inglada and G. Mercier, "A new statistical similarity measure for change detection in multitemporal SAR images and its extension to multiscale change analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, pp. 1432–1445, May 2007.

[9] P.-S. Liao, T.-S. Chen, and P.-C. Chung, "A fast algorithm for multilevel thresholding," *J. Inf. Sci. Eng.*, vol. 17, no. 5, pp. 713–727, 2001.

[10] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognit.*, vol. 19, no. 1, pp. 41–47, 1986.

[11] M. Gong, M. Jia, L. Su, S. Wang, and L. Jiao, "Detecting changes of the Yellow River Estuary via SAR images based on a local fit-search model and kernel-induced graph cuts," *Int. J. Remote Sens.*, vol. 35, nos. 11–12, pp. 4009–4030, May 2014.

[12] O. Yousif and Y. Ban, "Improving urban change detection from multitemporal SAR images using PCA-NLM," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2032–2041, Apr. 2013.

[13] M. Gong, L. Su, M. Jia, and W. Chen, "Fuzzy clustering with a modified MRF energy function for change detection in synthetic aperture radar images," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 1, pp. 98–109, Feb. 2014.

[14] S. Ghosh, L. Bruzzone, S. Patra, F. Bovolo, and A. Ghosh, "A context-sensitive technique for unsupervised change detection based on Hopfield-type neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 778–789, Mar. 2007.

[15] X. L. Dai and S. Khorram, "Remotely sensed change detection based on artificial neural networks," *Photogram. Eng. Remote Sens.*, vol. 65, no. 10, pp. 1187–1194, 1999.

[16] G. E. Hinton, "Learning to represent visual input," *Philos. Trans. Roy. Soc. London B, Biol. Sci.*, vol. 365, no. 1537, pp. 177–184, 2009.

[17] B. Chen, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, and L. Carin, "Deep learning with hierarchical convolutional factor analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1887–1901, Aug. 2013.

[18] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning— A new frontier in artificial intelligence research [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 13–18, Nov. 2010.

[19] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[20] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proc. ICML Workshop Unsupervised Transf. Learn.*, 2012, pp. 17–36.

[21] T. N. Wiesel and D. H. Hubel, "Receptive fields of single neurones in the cat's striate cortex," *J. Physiol.*, vol. 148, no. 3, pp. 574–591, 1959.

[22] J. Masci, J. Angulo, and J. Schmidhuber, "A learning framework for morphological operators using counter–harmonic mean," in *Mathematical Morphology and Its Applications to Signal and Image Processing*. Berlin, Germany: Springer-Verlag, May 2013, pp. 329–340.

[23] J. Masci, A. Giusti, D. C. Cireşan, G. Fricout, and J. Schmidhuber, "A fast learning algorithm for image segmentation with max-pooling convolutional networks," in *Proc. Int. Conf. Image Process.*, Melbourne, VIC, Australia, Sep. 2013, pp. 2713–2717.

[24] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[25] H. Larochelle and Y. Bengio, "Classification using discriminative restricted Boltzmann machines," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, Jul. 2008, pp. 536–543.

[26] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layerwise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 153–160.

[27] N. Srivastava, "Improving neural networks with dropout," Ph.D. dissertation, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2013.

[28] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, Jun. 2013, pp. 1319–1327.

[29] R. K. Srivastava, J. Masci, S. Kazerounian, F. Gomez, and J. Schmidhuber, "Compete to compute," in *Proc. Adv. Neural Inf. Process. Syst.*, Stateline, NV, USA, Dec. 2013, pp. 2310–2318.

[30] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, Jul. 2008, pp. 160–167.

[31] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1717–1724.

[32] J. Masci, U. Meier, D. C. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. 21st Int. Conf. Artif. Neural Netw.*, Espoo, Finland, Jun. 2011, pp. 52–59.

[33] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[34] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proc. Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, Jul. 2011, pp. 1237–1242.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Stateline, NV, USA, Dec. 2012, pp. 1097–1105.

[36] A. Stuhlsatz, J. Lippel, and T. Zielke, "Feature extraction with deep neural networks by a generalized discriminant analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 596–608, Apr. 2012.

[37] D. Prokhorov, "A convolutional learning system for object classification in 3-D lidar data," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 858–863, May 2010.

[38] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[39] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[40] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, 2010.

[41] S. L. Phung and A. Bouzerdoum, "A pyramidal neural network for visual pattern recognition," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 329–343, Mar. 2007.

[42] L. Bruzzone and D. F. Prieto, "An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 452–466, Apr. 2002.

[43] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 27, NO. 1, JANUARY 2016

[44] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *J. Electron. Imag.*, vol. 13, no. 1, pp. 146–168, 2004.

[45] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[46] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, USA, Tech. Rep. UTML TR 2010-003, Aug. 2010.

[47] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[48] 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.

[49] G. H. Rosenfield and K. Fitzpatrick-Lins, "A coefficient of agreement as a measure of thematic classification accuracy," *Photogram. Eng. Remote Sens.*, vol. 52, no. 2, pp. 223–227, 1986.

Jia Liu received the B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2013, where he is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems with the School of Electronic Engineering.

His current research interests include computational intelligence and image understanding.

Maoguo Gong (M'07–SM'14) received the B.S. degree in electronic engineering and the Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2003 and 2009, respectively.

He has been a Teacher with Xidian University, since 2006, where he was promoted to Associate Professor and Full Professor, both with exceptive admission, in 2008 and 2010. He has authored over 50 papers in journals and conferences, and holds 14 granted patents. His current research interests include computational intelligence with applications to optimization, learning, data mining, and image understanding.

Dr. Gong received the prestigious National Program for the support of Top-Notch Young Professionals from the Central Organization Department of China, the Excellent Young Scientist Foundation from the National Natural Science Foundation of China, and the New Century Excellent Talent in University from the Ministry of Education of China. He is the Vice Chair of the IEEE Computational Intelligence Society Task Force on Memetic Computing, an Executive Committee Member of the Chinese Association for Artificial Intelligence, and a Senior Member of the Chinese Computer Federation.

Qiguang Miao received the M.Eng. and Ph.D. degrees in computer science from Xidian University, Xi'an, China.

He is currently a Professor with the School of Computer Science and Technology, Xidian University. His current research interests include intelligent image processing and multiscale geometric representations for images.
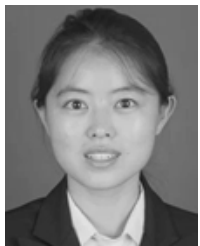
Jiaojiao Zhao received the B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2012, where she is currently pursuing the M.S. degree with the School of Electronic Engineering.

Her current research interests include change detection in remote sensing images.

Licheng Jiao (SM'89) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

He has been a Professor with the School of Electronic Engineering, Xidian University, Xi'an, since 1992. His current research interests include image processing, natural computation, machine learning, and intelligent information processing.