IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 31, 2018, accepted December 7, 2018, date of publication December 24, 2018, date of current version February 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889326

Local Descriptor Learning for Change Detection in Synthetic Aperture Radar Images via Convolutional Neural Networks

HUIHUI DONG[1], WENPING MA[1], (Member, IEEE), YUE WU[2], MAOGUO GONG[3], (Senior Member, IEEE), AND LICHENG JIAO[1], (Fellow, IEEE)

[1]Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation, School of Artificial Intelligence, Xidian University, Xi'an 710071, China
[2]Xi'an Key Laboratory of Big Data and Intelligent Vision, School of Computer Science and Technology, Xidian University, Xi'an 710071, China
[3]Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Electronic Engineering, Xidian University, Xi'an 710071, China

Corresponding authors: Wenping Ma (wpma@mail.xidian.edu.cn) and Yue Wu (ywu@xidian.edu.cn)

This work was supported in part by the National Natural Science Foundation of China through the Foundation for Innovative Research Groups under Grant 61621005, in part by the National Natural Science Foundation of China under Grant U1701267 and Grant 61702392, and in part by the Fundamental Research Funds for the Central Universities under Grant JB181704 and Grant JBX170311.

ABSTRACT In this paper, we present a novel convolutional neural network (CNN)-based model for change detection in synthetic aperture radar (SAR) images. Considering that change detection task takes image pairs as an input, we first explore multiple neural network architectures, which are specifically adapted to the change detection task. There are several ways in which patch pairs can be processed by the network and how information sharing can efficiently learn the semantic difference between the changed and unchanged pixels. For this reason, we then design a ''Siamese samples'' CNN, which treats patch pairs as indiscriminate samples to extract descriptors and then joins for their outputs. During training, the two patch features are extracted by the same network instead of separate sub-networks, while the joining neuron measures the distance between the two feature vectors. Due to ''pseudo-labels'' with high accuracy that is difficult to obtain, we modify a joint classifier based on the fuzzy c-means method into joint-similarity classifier as preclassification to obtain coarse ''pseudo labels,'' and discard sample selection. Thus, the preclassification labels with a low accuracy are used to fine-tune the network. Finally, a significantly improved change detection result can be obtained from the network. The proposed architecture provides a better trade-off in terms of speed and accuracy among its counterparts (Siamese, Pseudo-Siamese, and 2-Channel networks). The experiments on several real SAR data sets demonstrate the state-of-the-art performance of the proposed method compared with the advanced change detection methods.

INDEX TERMS Convolutional neural networks, Siamese networks, change detection, synthetic aperture radar, local descriptor learning.

I. INTRODUCTION

Image change detection is to identify the changes between the two images captured in the same area but at different times. Change detection plays an important role in various fields such as medical diagnosis, environmental monitoring, video surveillance [2]–[4]. Especially, when a natural catastrophe strikes, an efficient change detection technology appears crucial for disaster relief and disaster evaluation. In addition, SAR sensor is independent of atmospheric and sunlight condition, so it can complete earth observation with features of all-day, round-the-clock work, and high resolution, etc. Given its unique advantages, it has been widely researched in the past few decades [5]–[9].

Early works on this problem as mentioned [5], [7], [10], change detection technologies for SAR images consist of three steps: 1) Image pre-processing; 2) Producing the difference images (DI) based on multitemporal images; and 3) Analysis of difference images. The image pre-processing

techniques including co-registration, radiation correction and denoising are first used. The research of change detection in many existing literatures focuses on the last two steps [7]–[9]. Furthermore, SAR images always suffer from the effect of speckle noises, which results in a more difficult process than optical images change detection. In order to overcome this difficulty, a variety of methods are proposed for reducing the corruption in SAR images [11]–[13]. In the step of generating DI, ratio operator is commonly used as its robustness and nonsensitive to speckle noises [14], [15]. In the step of analyzing the DI, images clustering and threshold strategies are often used, which can be seen as segmentation problem [3]. The fuzzy c-means method (FCM) is one of the most popular clustering algorithms, which outperforms hard clustering on retaining more information in some cases [10]. However, owing to the original FCM method without considering spatial context, it is sensitive to speckle noise. In [7], an FCM-based method (i.e., the MRFFCM method) for SAR image change detection is shown, which incorporates the MRF model with a novel form of energy function in the procedure of FCM, and focuses on the modification of the membership to reduce the effect of speckle noise. In the threshold methods, it detects changes by obtaining a decisive threshold to the histogram of the DI. For example, the Generalized Kittler&Illingworth (GKI) method is proposed in [9], which considers non-Gaussian distribution of the amplitude values of SAR images to find an optimal threshold. However, it is difficult to estimate a proper probability distribution from the DI in thresholding methods. Researchers have introduced many advanced approaches such as graph cut and level set based methods [16], [17].

Recently, there is the idea of deep learning that has accounted for the mainstream, especially in CNN-based research and applications [18]–[20]. That has made a great breakthrough in computer vision tasks such as image recognition, image classification and image segmentation [21]–[24]. To put it simply, deep learning is able to acquire the high-level expression of information contributed by multilayer neural network and the nature of nonlinearity. This makes it be of the capacity of automatically learning complex relationship from raw data. In [25], a fast greedy learning algorithm is proposed, i.e., deep belief networks (DBN), which greedily train for every layer. Convolutional neural network (CNN) is one of the earliest deep learning models, and it has become one of the most popular models. Due to remarkable performance of deep learning in image processing, it has been introduced to change detection. In existing works [26]–[29], denoising auto-encoders (DAE) seems to be popular for this problem. Owing to its favorable property of unsupervised reconstruction, it can be carried out with a post-classification comparison manner for feature learning and mapping [26], [30]. In most cases, this model is used by an integrated manner with other network models [27], which is similar to the restricted Boltzmann Machine (RBM) [25]. In general, RBM is used to initialize the weights and bias of the network so that derives the learning process towards the positive direction in an unsupervised way [28]. It can improve the network performance to some extent. In terms of the adoption of DBN including a stack of RBM, in [28], it concatenates directly two local patch vectors as a sample, and then feed them into deep neural network. It turned out that deep learning is potential and available for change detection. And this provides a feasible scheme for two parallel inputs. This mode of concatenation also is used in matching patch pairs using CNN [1], [31], [32], which join two descriptor vectors at the top fully connected layer of the network. Due to a natural adaption of CNN with two parallel inputs, recently, local descriptor learning has been widely studied using CNN [33], [34]. For our problem, multitemporal SAR images correspond rightly to two parallel inputs to the network. The learned descriptors by CNN are robust to geometric distortions of the input images. These advantages are perfect for SAR image change detection. However, to our knowledge, it has rarely been studied directly.

In addition, in existing change detection methods based on deep learning, it generally uses a pre-classification result (''pseudo-labels'') as sample labels to fine-tune the network [28]. Pre-classification labels are often obtained by traditional methods or manual [27], [28]. To better learn mapping relationship and avoid the effect of misclassification, the available algorithms tend to pick up the labeled data with high accuracy to train the network. The common ways are selecting samples as correctly classification as possible or most-unlikely-changed feature pairs as training data [27]. In this way, it comes at the costs in terms of computation and time complexities produced by these additional works.

In order to address the above problems and take advantage of the recent success in comparing patches via CNN [35], [36], we try to design and explore different network architectures based on CNN that can process patch pairs as input for SAR image change detection. Following this, we expect that can strongly reduce corruptions of speckle noise with low complexities of computation and time, and meanwhile lower the requirement of preclassification accuracy. We compare the proposed architecture to several advanced CNN-based architectures (i.e., Siamese, Pseudo-Siamese and 2-Channel networks shown in [1]) and DNN method [28] on four real SAR data sets. Experimental results demonstrate the effectiveness and advantages of the proposed approach. Above all, it can be summarized for main contributions of this paper into two aspects:

1) We explored and tested multiple CNN-based architectures to learn a classifier, which can classify changed and unchanged areas for SAR images change detection. These architectures are specifically adapted to our task. We then design information sharing of two patches just via a single branch convolutional networks that can accept patch pairs as parallel input. The key idea is refer to double samples, called a ''Siamese samples'' convolutional neural network. In contrast to available architectures, the experimental results demonstrate the designed architecture provides an optical

trade-off between accuracy and runtime. Owing to end-to-end local feature detector extracts high-level descriptors from two patches, the proposed method has strong robustness to speckle noises.

2) This paper presents a more simplified step for algorithm framework and reduces the requirement of the accuracy of pre-classification labels compared with [27] and [28]. That means there is no samples selection or designed additional processing. We modified JFCM [28] into simple and efficient to obtain ''pseudo labels''. Thus, this method is efficient and precludes the need for the complications in other works.

The remainder of this paper is organized as follows. Section II describes related works of CNN settled for matching patches by local descriptor learning. Section III describes the proposed method in detail. In Section IV, experimental results on real multitemporal SAR images are presented for verifying the performance of the approach. Finally, the conclusion is drawn in Section V.

II. RELATED WORK

The problem for matching image patches based on CNN plays an important role in a wide variety of computer vision community [1], [31]. Many related works have reported the significant improvements for matching performance by local descriptor learning [33]–[35]. Matching patches has lots of applicable situations such as signature verification [29], the stereo matching [32] and face verification [37]. The key to the issue is that the network needs to accept two parallel inputs, i.e., a pair of images, which is fit to the problem of SAR image change detection so well.

Hence, for our aim of change detection, given two SAR images, it can be concretely described as identifying whether change for every pixel based on local descriptor learning. This in itself is a binary classification problem, i.e., classify each pixel into changed classes or unchanged classes. The final change detection map is shown by the changed classes labeled for 1, others labeled for 0 or reverse. Our purpose is thus to predict such a final change map through the trained network based on discriminative descriptor learning.

In the example of signature verification, [31] proposed the time delay neural networks called Siamese network, which is one of the most popular networks. Based on this network, [37] have improved the loss function (i.e., proposed the contrastive loss function) to learn a similarity metric for face verification application, which primarily targets the case that samples have the large number of categories but each category has small number and only known a subset of the categories during training. When the loss function is minimized by training, the similarity metric is driven for being small if pairs of faces from the same person, and being large for pairs from different person. The model can later be used to compare or match new samples from previously-unseen categories. For matching over a pair of small baseline stereo, the method in [32] have presented a specific architecture of CNN, which refreshed the best record on the KITTI stereo dataset. However, it cannot work well on a different dataset. We need to design a specific CNN-based architecture for SAR images change detection. In [36], focusing on training samples, it proposed the network with positive and negative pairs (PN-Net) formed by triplets of patches for better learning discriminative local descriptors and a matching metric.

About the existing and various architectures based on CNN, the paper [1] provides a conclusive contribution. It investigated and exhibited different trade-offs and advantages among three basic models, also including Central-surround two-stream networks. However, in view of the low complexity for our project, we do not try to employ the latter network. The three basic structures are Siamese, Pseudo-siamese and 2-channel, respectively, described by [1]. As aforementioned, we use them as compared architectures, and analyzing and surveying their advantages and disadvantages on change detection of SAR images. By contract, the experimental results state the excellent performance of the proposed architecture on accuracy and runtime.

III. METHODOLOGY

The task of change detection is to distinguish the changed classes and non-changed classes from a pair of co-registered SAR images $R_{t1} = \{R_{t1}(i,j), 1 \leq i \leq I, 1 \leq j \leq J\}$ and $R_{t2} = \{R_{t2}(i,j), 1 \leq i \leq I, 1 \leq j \leq J\}$, which are obtained from the identical geographical area but at different times, i.e., $t1$ and $t2$, respectively. In general, the final result is presented by a binary image map $L = \{L(i,j) \in (0,1), 1 \leq i \leq I, 1 \leq j \leq J\}$, where $L(i,j)$ represents the pixel value. $I \times J$ is the size of the image.

In this section, we describe our method in details. The proposed method can obtain directly the result of change detection from raw image pairs, only requiring through the CNN without integrating other models. We still need the label data to fine-tune the network and learn mapping relationship, which is the key to our method. The ''pseudo labels'' are obtained by a pre-classification method, which no need for a high accuracy. In our network, discriminative representation is extracted from the raw data, and then pairs of descriptors are sent to the top network for learning a distance metric that identifies changes and unchanges.

A. ALGORITHM FRAMEWORK

The previous works related to algorithms framework [27], [28] can be divided into the following four steps as shown in Fig. 1: 1) pre-classification for obtaining label data with high accuracy. 2) selecting reliable examples for training the network. 3) training and fine-tuning the network. 4) classifying every pixel for changed and unchanged classes by the trained network. Specially, in [27], it first obtains each feature representations of two images through the network, and then generate a DI from two feature maps. Finally, analyzing the DI leads to produce the final change detection result. The framework of our proposed algorithm for change detection consists of three steps: 1) acquiring the label data by pre-classification method. 2) training and fine-tuning the network using back propagation (BP) algorithm.

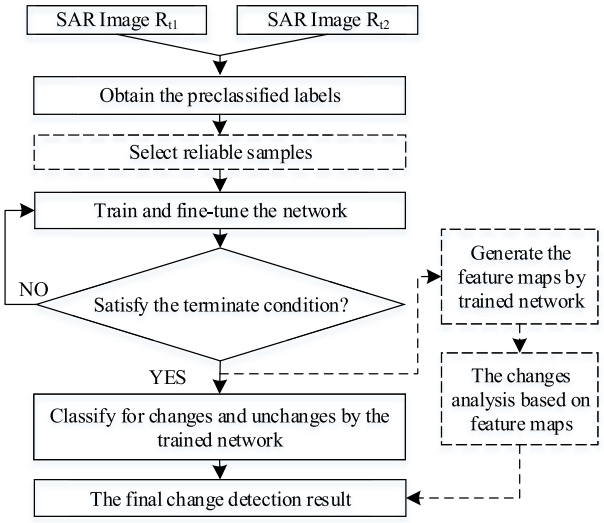


FIGURE 1. The whole algorithm framework is related to the previous methods based on deep learning. Only the full line sections are related to our proposed algorithm.

3) classifying each pixel for changes and unchanges by the well-trained network, i.e., generate the final change detection map. The work flow of our algorithm is explicitly shown in Fig. 2. During preprocessing phase, the training samples and their labels are obtained, and then both are fed into the network for training. After training, neighborhood patches of each pixel of two images are again fed into the network, which directly outputs the final change map. The class 0 represents being unchanged, the class 1 represents being changed.

In our system, it requires no sample selection. We use neighborhood patches of each pixel as training samples to train the network rather than selecting some special samples. Compared to [27] and [28], we reduce some additional works such as omitting the manual parameter used in sample selection [28]. No sample selection is beneficial for not only increasing samples diversity, but also can not further deteriorate the imbalance of samples. They contribute to the proposed methods for obtaining a competitive performance and offsetting the effect of mis-reclassification.

B. PRE-CLASSIFICATION METHOD

Our pre-classification method is derived from the joint classifier based on fuzzy c-means method (JFCM) [28]. It can be known as joint-similarity classifier (JSC). There is a clustering operation using FCM for each SAR images in JFCM. However, FCM is the algorithm with an iterative process, and the quality of result strongly depends on the initial cluster center. Therefore, it usually is difficult and time-consuming to obtain an available pre-classification result using JFCM. In order to improve algorithm efficiency and reduce the complexity, we design to remove the use of FCM from JFCM. JSC is thus high-efficiency and low-complexity. Although the accuracy of the result obtained by JSC is far lower than JFCM, one of the advantages is that the proposed detection algorithm reduces the requirement for the accuracy of label variations. That means the proposed method still obtains the excellent performance even though the pre-classification labels with strong noises and contamination. JSC is designed for 2-D images, and the operation is based on gray-level. The general process of JSC is described in Algorithm 1. The calculation of similarity is implemented pixel by pixel over two co-registered intensity SAR image, which is defined as:

$$S_{i,j} = \frac{|R^{t1}_{i,j} - R^{t2}_{i,j}|}{R^{t1}_{i,j} + R^{t2}_{i,j}} \tag{1}$$

where $R^{x}_{i,j}$ represents the gray value of images at time x and the location (i, j), $x \in (t1, t2)$.

Algorithm 1 The Process of JSC Method

input: Two original images R_{t1} and R_{t2}.
output: The final binary result L.

```
compute the values of S_{i,j}, v^{t1}_{i,j} and v^{t2}_{i,j} for each pixel.
compute the global threshold value T according to similarity map S.
for for each i ∈ [1, I] do
    for for each j ∈ [1, J] do
        if v^{t1}_{i,j} ≤ v^{t2}_{i,j} then
            if S_{i,j} ≤ T then
                R^{t2}_{i,j} = R^{t1}_{i,j}
            end if
        else
            if S_{i,j} ≤ T then
                R^{t1}_{i,j} = R^{t2}_{i,j}
            end if
        end if
    end for
end for
compare R*_{t1} and R*_{t2}.
```

After obtaining the value of similarity, we can further obtain the global threshold value T by iterative thresholding algorithm [38]. Furthermore, the whole process of our method only needs to iterate once after acquiring the global threshold value T. The result generated by JSC is certainly and extremely coarse. However, the proposed CNN-based method with it as preclassification can obtain state-of-the-art performance for change detection. To some extent, it indicates the fact that the network based on discriminative patch descriptor is of great robustness to noises and contamination, and it seems to be able to discover the hidden structure (changed regions) in extremely coarse label map.

Next, according to the principle of minimizing variance, the classifier determines the reference point from pixel pairs for classification. We design the pixel with small variance is chosen as the reference point. It guarantees the result that unchanged areas have the same pixel value; changed areas have the different pixel value. The formula of variance is written as follows:

$$v^{x}_{i,j} = w^{x}_{i,j}(R^{x}_{i,j} - G_{i,j})^2 \tag{2}$$

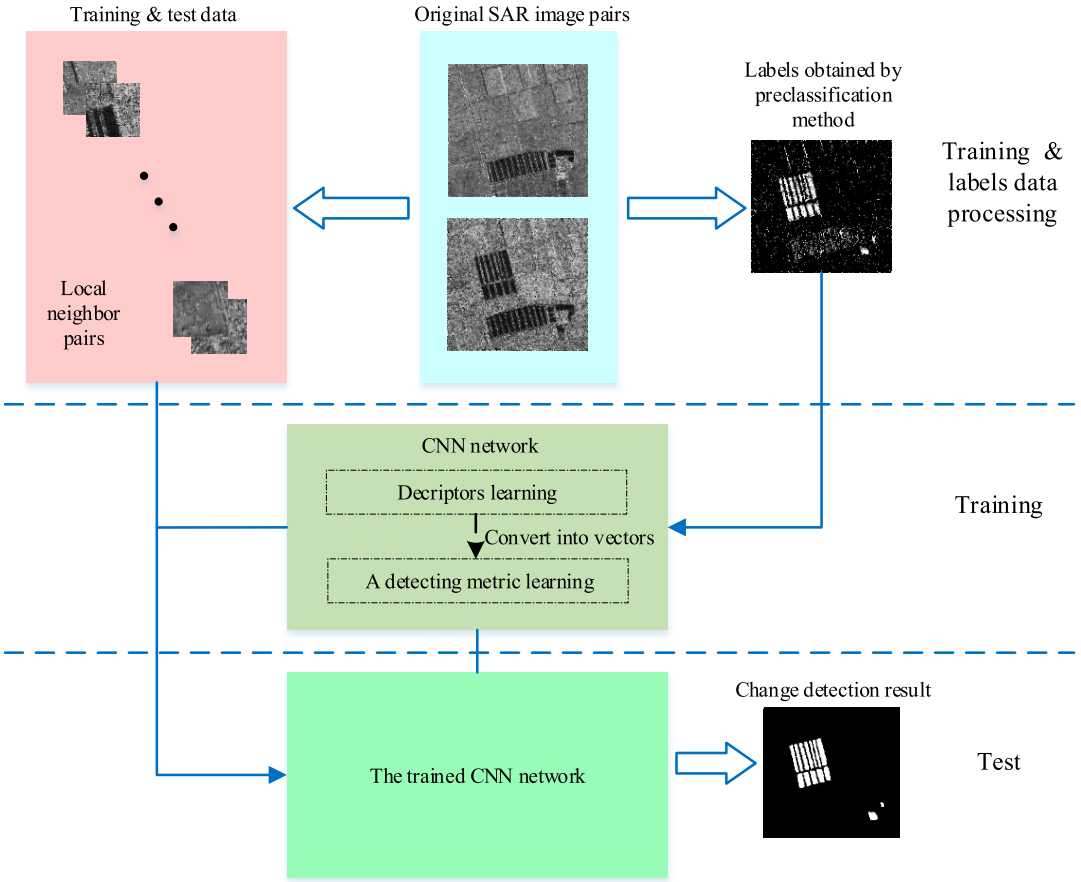


FIGURE 2. The work flow of the proposed algorithm. (1) Given two co-registered SAR images, producing the label data by pre-classification method and processing each pixel into corresponding neighborhood patch as training samples. (2) Learning a classifier by training and fine-tuning the network. (3) Generating directly the final change map by the trained network.

where $w_{i,j}^x$ is the weight of gray level, and $G_{i,j}$ is the average gray value of the weight. The definitions of both are written as:

$$w_{i,j}^x = \frac{R_{i,j}^x}{(R_{i,j}^{t1} + R_{i,j}^{t2})} \tag{3}$$

$$G_{i,j} = \sum_{x=t1}^{t2} w_{i,j}^x R_{i,j}^x \tag{4}$$

Then, bring (3) and (4) to (2), we can get the formula as follows:

$$v_{i,j}^x = \frac{R_{i,j}^x}{(R_{i,j}^{t1} + R_{i,j}^{t2})}[R_{i,j}^x - \frac{(R_{i,j}^{t1})^2 + (R_{i,j}^{t2})^2}{R_{i,j}^{t1} + R_{i,j}^{t2}}]^2 \tag{5}$$

Above all, we eventually obtain the following expressions of variance pixel by pixel:

$$v_{i,j}^{t1} = R_{i,j}^{t2} \frac{R_{i,j}^{t1} R_{i,j}^{t2}}{(R_{i,j}^{t1} + R_{i,j}^{t2})}[S_{i,j}]^2 \tag{6}$$

$$v_{i,j}^{t2} = R_{i,j}^{t1} \frac{R_{i,j}^{t1} R_{i,j}^{t2}}{(R_{i,j}^{t1} + R_{i,j}^{t2})}[S_{i,j}]^2 \tag{7}$$

Finally, the classified maps $R_{i,j}^{t1}$ and $R_{i,j}^{t2}$ of two images are gotten, in which unchanged areas is labeled for the same gray value, changed areas is labeled for the different gray value. By direct pixel-wise comparison, we can obtain the pre-classification result L.

In addition, we design the variant of JSC for surveying the influence of the mis-preclassification degree on the proposed architecture (see Section IV. F). The algorithm is described in Algorithm 2. Generally, JSC-variant can obtain the result with less white noise spots. But in some cases, it often loses more detail information of changed areas. The difference between JSC and JSC-variant is that whether classification is iterated according to iterative thresholding algorithm.

C. ARCHITECTURES

We first introduce three existing and basic models based on CNN [1] for change detection. These several models are specially adapted to our task. They all can complete information sharing for patch pairs. Grayscale patches are adopted for training the network. On the whole, the goal of the network is to construct a detector, which can learn semantic difference between changed and unchanged pixels. Inspired by these

Algorithm 2 The Process of JSC-Variant Method

input: Two original images R_{t1} and R_{t2}.
output: The final binary result L.

1: compute the values of $S_{i,j}$, $v_{i,j}^{t1}$ and $v_{i,j}^{t2}$ for each pixel.
2: compute an iteration value T according to similarity map S.
3: execute steps 3-15 in Algorithm 1.
4: repeat steps 1-3, until the termination of the iterative threshold algorithm.
5: compare R_{t1}^{*} and R_{t2}^{*}.

three basic models, we treat patches pairs as indiscriminate sample, proposing ''Siamese samples'' architecture, which extracts descriptors of two patches by only single network without sub-branches. And then the outputted descriptor pairs are joined to learn a detecting metric by minimizing objective function loss. In this section, we described the proposed architecture in detail.

1) SIAMESE NETWORK

As shown in Fig. 3, the Siamese network computes descriptors of patch pairs by two branches, each branch takes as input one of the two patches. For two sets of patches, $D^{t1} = \{d_n^{t1}(i,j), 1 \le i \le I, 1 \le j \le J, 1 \le n \le N\}$ and $D^{t2} = \{d_n^{t2}(i,j), 1 \le i \le I, 1 \le j \le J, 1 \le n \le N\}$, where $d_n^{t1}(i,j)$ and $d_n^{t2}(i,j)$ both represent the neighborhood patches at the position (i,j) in two images, respectively, N represents the number of samples and $N = I \times J$, the two sets are independently fed to the two branches. The same architecture and the same weight sets are shared in the two branches [31]. The output of two branches, i.e., computed descriptors, are concatenated and successively fed into the top fully connected layer. During training, the function loss is minimized, which drives a similarity metric to be discriminative. The updating of parameters is implemented by the sum of the gradients of two sub-branches [37].

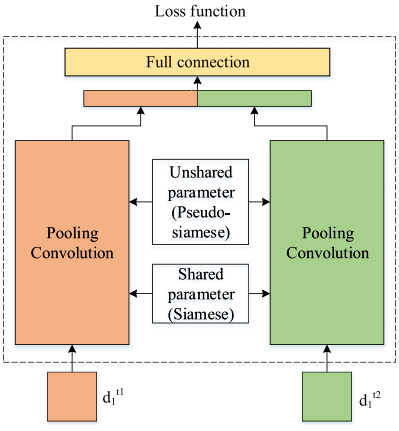


FIGURE 3. The shared parameter represents Siamese network. The unshared parameter represents Pseudo-siamese network.

2) PSEUDO-SIAMESE NETWORK

The Pseudo-siamese network differs from Siamese network only in that whether sharing parameters between two branches [1]. It is no shared for the weights of the two branches in Pseudo-siamese network. Similar to Siamese network, the two sub-network in the first half of the network aims at computing descriptors of patch pairs (see Fig. 3). Due to uncoupled weights and architectures, the Pseudo-siamese network increases the number of parameters, and generally with a large complexity. On the contrary, this also makes it more flexible.

3) 2-CHANNEL NETWORK

There are no two independent descriptors that are extracted by the networks for two patches. The architecture takes two patches as two feature maps, which are directly fed into the network [1]. Therefore, after the first convolutional layer of the network, pairs of patches $d_n^{t1}(i,j)$ and $d_n^{t2}(i,j)$ are integrated into one $d_n^{t}(i,j)$ [see Fig. 4(a)]. Then the fusion features are further computed for higher-level representation by convolutional, sigmoid and pooling layers. The output of this bottom part is then fed into a top fully connected layer. In general, the network runs more fast during training and test than other architectures due to it makes data by half at the first layer of the network.

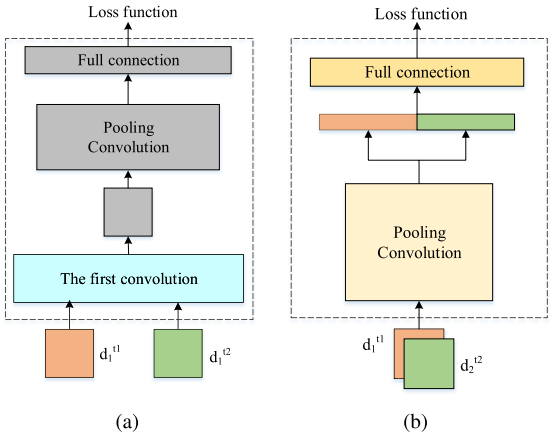


FIGURE 4. (a) 2-channel network. (b) The proposed network.

4) THE PROPOSED NETWORK

Above all, we simply consider the two patches of an input pair as two indiscriminate samples, which are directly fed to the first convolutional layer. Specifically, the network takes ordered sample sequence $D = \{[d_n^{t1}(i,j), d_n^{t2}(i,j)], 1 \le i \le I, 1 \le j \le J, 1 \le n \le N\}$ as input, (where $[\cdot,\cdot]$ represents the two samples with respect to patch pairs at the position (i,j) and indicates the location relationship of patch pair as two independent samples, i.e., two samples corresponding to patch pairs is next to each other when feeding into the network. *Notice that the total number of samples is* $2N$). Different from Siamese network with two sub-network, per-branch accepts one of pairs of patches as input. Comparatively, our network with a single branch directly

accepts double samples as input. After the descriptors are obtained by bottom network, i.e., convolutional, sigmoid and pooling layers, and then the concatenation vectors of descriptor pairs are fed to the top fully connected layer to learn a detecting metric for identifying whether changes. The concatenated descriptors can be expressed by equation as $D^{v*} = \{d_n^{v*} = (d_n^{t1}(i,j) \circ d_n^{t2}(i,j), 1 \le i \le I, 1 \le j \le J, 1 \le n \le N\}$, (where d_n^{v*} is one sample and $(\cdot \circ \cdot)$ represents the concatenating of vectors of two descriptors, and it is worth noticing where the number of samples is N instead of $2N$).

In a sense, this network [see Fig. 4(b)] can be viewed as sharing the parameters via a single branch for two parallel inputs. It treats a pair of patches as indiscriminate samples at descriptor learning phase, but needs to keep their positional relationship for convenience to concatenating two descriptors at a top decision layer. In this case, the gradient of the loss function with respect to the parameter vector controlling double samples is computed using back-propagation. The gradient is updated as ordinary network but with double samples. This differs from Siamese network that the parameters are updated using the sum of the gradients contributed by two sub-branches. The proposed architecture contributes beneficially to the robustness to speckle noises for learning a better representation. On the contrary, 2-channel network makes two patches information joint, which may drives speckle noises joint and not easy to remove as shown in Section IV. The proposed architecture gives a better trade-off between accuracy and runtime than comparative architectures under the same environment configurations.

D. IMPLEMENTATION DETAIL AND TRAINING PROTOCOL

Our network is easy to implement. Compared with the ordinary single branch CNN, we only need to adjust corresponding parameters to adapt double and siamese samples for running. The work detail of the proposed architecture is illustrated in Fig. 5. The specific architecture consists of four layers. In the experiments, we try various architectures such as setting more layers, different size of convolutional kernel and mean pooling, etc. These results have no significant difference.

Fig. 5 shows one of best-performing structures. In order to follow-up easy to use, this architecture can be represented as (C4-2)-S2-(C2-6), where (Cx-y) denotes a convolutional layer, x denotes the size of convolutional kernel, y denotes the number of feature map; Sz denotes a mean pooling layer, z denotes the pooling scale, and there is one fully connected layer at the top layer by default for ease of notation. During training, the network takes neighborhood patches of each pixel as input, and the pre-classification provides their corresponding labels (change or unchange). The parameters of the network are randomly initialized. We use mini-batch stochastic gradient descent (mini-SGD) algorithm and the squared-error cost function to fine-tune the parameter set for optimal classification. The squared-error cost function is represented by:

$$\theta^* = \arg\min_{\theta} \frac{1}{2N} \sum_{i}^{N} \|y_i - \widehat{y_i}\|_2^2 \tag{8}$$

where y_i is the classification result, $\widehat{y_i}$ is the label of sample i, θ represents the parameters of the network, N represents the number of samples in one of image pairs. In addition, we adopt sigmoid function as activation function and output label is set for two classes (0, 1) corresponding to unchanged pixels and changed pixels, respectively. We set the batch size and epoch to 100 and 30, respectively, and the learning rate with $\alpha = 1.0$. After training, the neighborhood patch of each

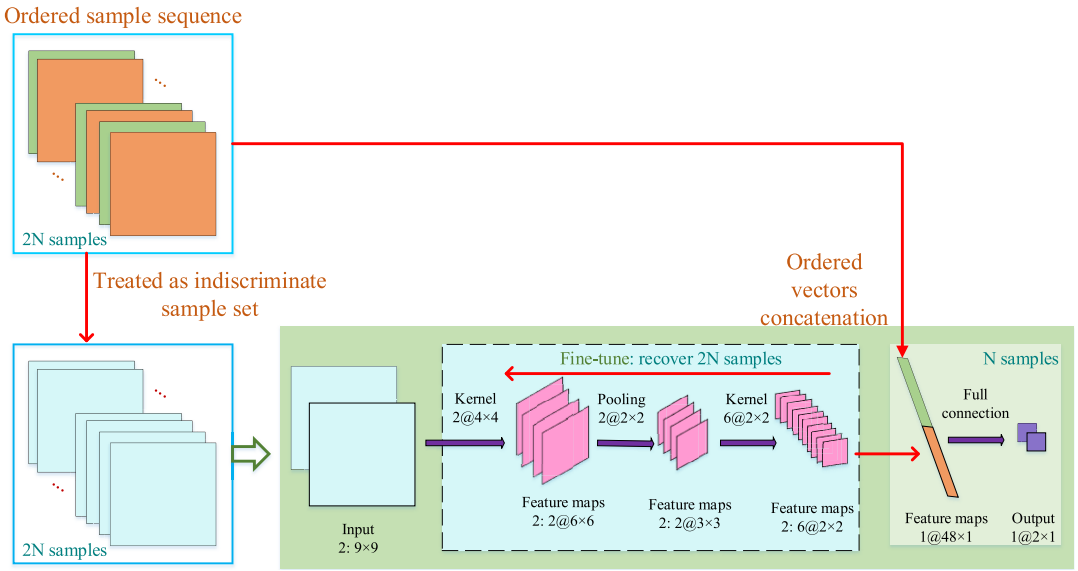


FIGURE 5. The specification of proposed "Siamese samples" architecture. The input is ordered sample sequence. A pair of patches at the same position in two images is treated as indiscriminate samples, extracting descriptors at the bottom layer and then concatenating their vector descriptors to learn a detection metric at the top fully connected layer. During back-propagation at the bottom layer, the gradient is processed with 2N samples by correspondingly reshaping.

pixel in image pair (i.e., training sample) is fed into the trained network for predicting their joined class label.

IV. EXPERIMENTS

The proposed network is very small but unusually effective to reduce speckle noises. We find that the networks are able to learn correct semantic difference between changed and unchanged pixels even though the labels with much mis-preclassification. In this section, the experiments are presented for evaluating the proposed methods on four real SAR image data sets. We compare our method with popular Siamese, Pseudo-siamese and 2-channel networks as well as DNN method [28]. For a fair comparison, the preclassification method (JSC) is used in all neural network classifiers and the same hyper-parameter set is used in these CNN-based architectures. Besides, all networks can receive image pair of any size as input and thus scale well. The training set of each network is collected according to its own structure and design partly described in Section III. C. We also listed the more description about training sets of different networks shown in Table. 1. Experimental results verified the state-of-the-art performance of the proposed method. We made the best results bold.

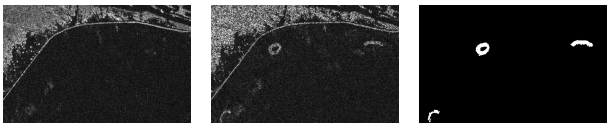
TABLE 1. Details of training sets with 5 × 5 neighborhood for different networks.

Architectures	branch.	number of sets.	size of sets.	sample selection.	channel.
DNN	no	1	$N \times 50$	yes	1
2-channel	no	1	$N \times 5 \times 5 \times 2$	no	2
Pseudo-siamese	yes	2	$2 \times N \times 5 \times 5$	no	1
Siamese	yes	2	$2 \times N \times 5 \times 5$	no	1
The proposed	no	1	$2N \times 5 \times 5$	no	1

Our codes are written in Matlab language. The environment of running codes is shown as follows: Intel(R) Core(TM) i5-4200M CPU @ 2.50GHz, RAM 12.0GB, Windows7 Pro (64-bit) and Matlab R2015b (8.6.0.267246).

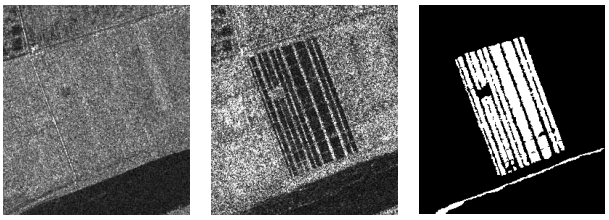
A. DATA SETS

The Coastline (450 × 280 pixels) and Farmland (257 × 289 pixels) data are from the same areas, i.e., the yellow River Estuary areas of China. The original size of yellow River Estuary image pairs is 7666 × 7692, which is captured in June of 2008 and 2009 by Radarsat-2 senior, respectively. It is necessary for two SAR images pairs to highlight that they are single-look and four-look images, respectively. That means they are affected by noises at the different level, the single-look one is much greater than the four-look one. As shown in Fig. 6 and Fig. 7, the two data sets are selected at the different and special segment from original yellow River Estuary image pairs. For the two data sets, it is more difficult to detect the changed regions. Besides, reference image (ground truth) shows the actual changed areas based on the original multitemporal SAR images, which is acquired by integrating prior information with the photo-interpretation.



(a) (b) (c)

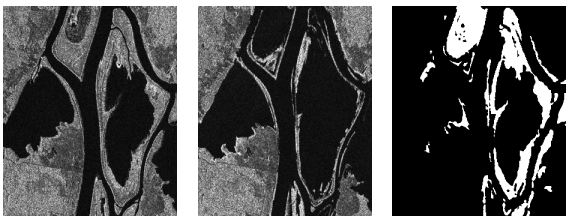
FIGURE 6. The Coastline data set: (a) The SAR image obtained in June 2008. (b) The SAR image obtained in June 2009. (c) The ground truth.



(a) (b) (c)

FIGURE 7. The Farmland data set: (a) The SAR image obtained in June 2008. (b) The SAR image obtained in June 2009. (c) The ground truth.

The Ottawa data set is a section (290 × 350 pixels) of SAR images pairs, which is related to the city of Ottawa acquired by the Radarsat SAR sensor and provided by the Defence Research and Development Canada, Ottawa. The available ground truth is shown in Fig. 8(c), which is acquired by integrating prior information with the photo-interpretation based on two input images Fig. 8(a) and (b).



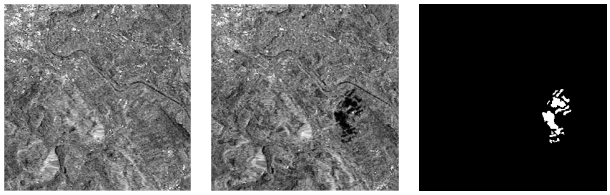
(a) (b) (c)

FIGURE 8. The Ottawa data set: (a) The SAR image obtained in July 1997 during summer flooding. (b) The SAR image obtained in August 1997 after the summer flooding. (c) The ground truth.

The Bern data set is a section (301 × 301 pixels) of SAR images pairs, which is related to an area near the city of Bern, Switzerland, in April and May 1999, and acquired by European Remote Sensing 2 satellite SAR sensor. Between the two dates, River Aare flooded parts of the cities of Thun and Bern and the airport of Bern entirely. Therefore, the Aare valley between Bern and Thun was selected as a test site to detect flooded areas. The ground truth presented the actual changed areas, i.e., affected areas, which is acquired by integrating prior information with the photo-interpretation based on Fig. 9(a) and (b).

B. EVALUATION CRITERIA

We evaluate the performance of our algorithm using the popular strategy in SAR images change detection [7], [10], [14], [28]. The change detection result is a binary image,



(a) (b) (c)

FIGURE 9. The Bern data set: (a) The SAR image obtained in April 1999 before the flooding. (b) The SAR image obtained in May1999 after the flooding. (c) The ground truth.

where white areas represent the changes, black areas represent the unchanges. Given the result and available ground truth, a quantitative analysis can be carried out. False negative (FN) and false positive (FP) both represent the number of pixels that are wrongly detected. The former is changed pixels but undetected, the latter is unchanged pixels but detected wrongly as changed pixels. The overall error (OE) is the sum of FN and FP, i.e., $OE = FP + FN$. The percentage of correct classification (PCC) is calculated by the following expression:

$$PCC = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

Where TN and TP also are the number of pixels but correctly detected. TN is true negative, i.e., changed pixels is detected correctly. TP is true positive, i.e., unchanged pixels is detected correctly.

Kappa coefficient is another measure for accuracy assessment [39], whose value ranges from 0 to 1. And, the higher value kappa is, the better the change detection result is. Generally speaking, it is of more authority than PCC to assess the result of change detection. Because the kappa is involved in more information based on difference between the error matrix and change agreement. It is obtained by:

$$KC = \frac{PCC - PRE}{1 - PRE} \tag{10}$$

where

$$PRE = \frac{(TP + FP) \cdot (TP + FN) + (FN + TN) \cdot (FP + TN)}{(TP + TN + FP + FN)^2} \tag{11}$$

A better result will be lower values of both FP and FN, the higher values of both PCC and Kappa.

C. DISCRIMINATIVE PATCH DESCRIPTOR ANALYSIS FOR DENOISING

As described above, speckle noises are eliminated well by the proposed network. However, it is not clear yet how the speckle noises are eliminated and also what the network have learned. Therefore, we set the size of input examples for 9×9 to train the developed network in Ottawa data set, and then we visualized convolutional kernels, activation maps and processed feature maps of two original images over $C1$ and $C3$ (see Fig. 10). In the first column of Fig. 10, we can see that the filters of initialization present disorderly as their values are random. However, after training, they become well-organized. Furthermore, two kernels of $C1$ seem to learn a white filter (represents changes) and a black filter (represents unchanges), respectively. Due to the kernel of $C3$ is too small 2×2, it is difficult to observe visually the pattern. In the second column of Fig. 10, the more discriminative representations of input patches are extracted through higher layer. In addition, with the patch descriptor more clearly represented for white or black, the speckle noises are reduced well. Similarly, in the higher layers, the feature maps of original SAR image pair are more abstract and discriminative for identifying the changed and unchanged areas (see the third and fourth columns of Fig. 10). To sum up, the network can grasp the crucial information for learning a binary classifier, in which discriminative descriptor is extracted and simultaneously speckle noise is reduced.

D. RESULTS ON DIFFERENT DATA SETS

1) RESULTS ON THE COASTLINE DATA SET

In this data set, on the one hand, it has an imbalance data between changes and unchanges, which is adverse to the methods based on deep learning. That means unchanged samples account for a much great proportion while training the network, which usually lead to a harder process to learn an excellent estimator for detecting changes. On the other hand, the preclassification result [see Fig. 11(a)] is extremely coarse, changed areas are drowned in dense white noise spots and only changed outline can be observed. These two aspects both are challenges for classifiers to obtain a better performance. The final results obtained by six methods are shown in Fig. 11 and their quantitative analysis is presented in Table. 2. First of all, in terms of accuracy, although the deep neural network (DNN) [28] method selects samples as correctly preclassification as possible to train the network,

TABLE 2. Change detection results of Coastline data set.

Method/architectures	FP(%)	FN(%)	PCC(%)	Kappa(%)	Train-time	Test-time
JSC	29.5429	0.1770	70.2802	3.6769	-	-
DNN	0.3984	0.4373	99.1643	59.7970	-	-
2-channel	0.5412	0.0952	99.3635	75.0710	**5.3257s**	**1.0308s**
Pseudo-siamese	0.1151	0.1095	99.7754	89.4166	8.3820s	1.4595s
Siamese	0.1349	**0.0896**	99.7825	89.6068	8.4704s	1.4961s
The proposed	**0.0904**	0.1103	**99.7992**	**90.4264**	6.6879s	1.3353s

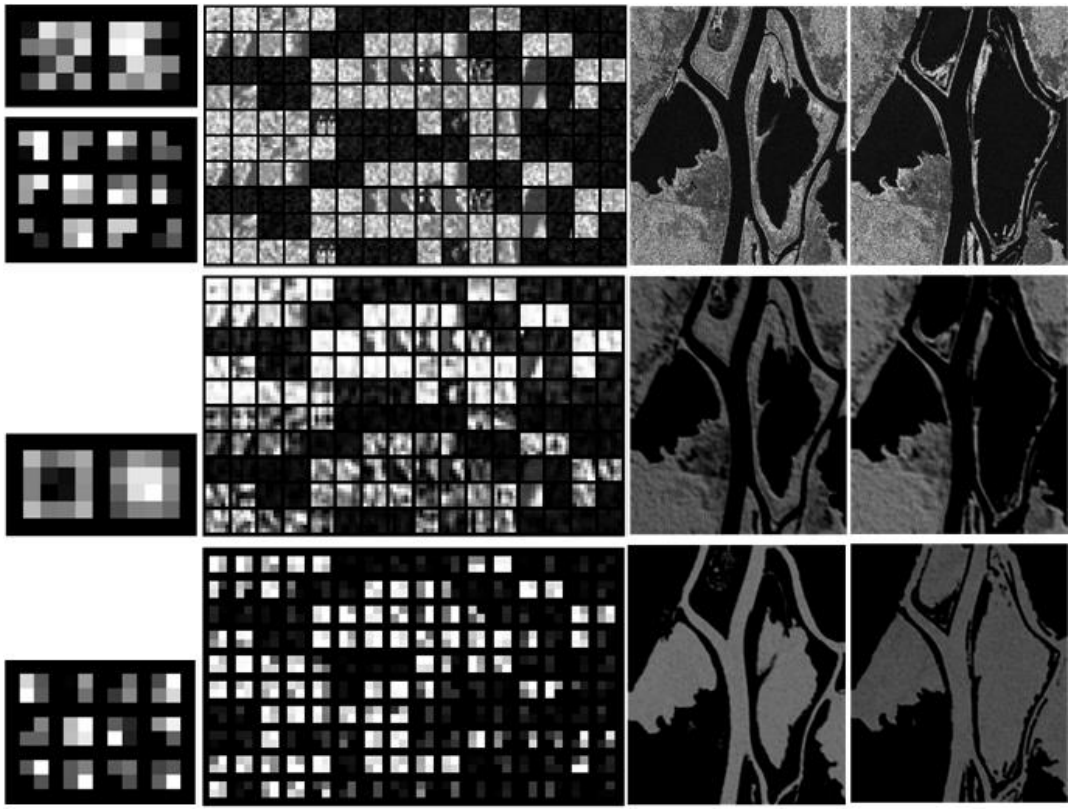


FIGURE 10. Feature visualization of convolutional layers. (a) The first column: randomly initialized convolutional kernels of $C1$ and $C3$ layers, trained convolutional kernels, (b) The second column: input patches, extracted descriptors/activation maps at $C1$ and $C3$ layers, (c) The third and fourth columns: original SAR image pair $I1$ and $I2$, processed feature maps at $C1$ and $C3$ layers. The first, second and third rows represent initialization, $C1$ and $C3$ layers, respectively.

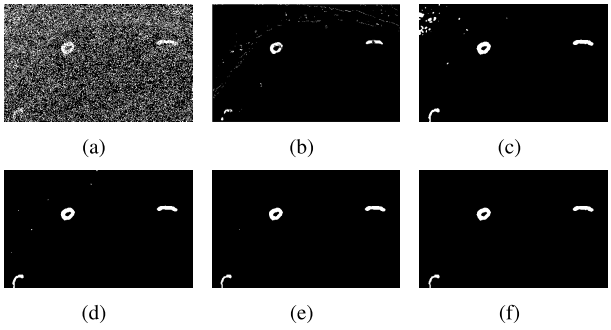


FIGURE 11. Pre-classification and change detection results on Coastline data set obtained by (a) JSC, (b) DNN, (c) 2-channel, (d) Pseudo-siamese, (e) Siamese, (f) The proposed.

it can be observed that it missed some changed details and existed many noise spots in Fig. 11(b). Under our algorithm framework, the CNN-based methods all acquired excellent performance even though omitting sample selection. In addition, there are the extraction of discriminative descriptors in Siamese, Pseudo-siamese and the proposed network from patch pairs, then the joined descriptors are fed to a fully connected layer for deciding the classes. Therefore, they are more robust to speckle noises and contamination than JSC and DNN methods. It can be observed that the proposed method yields the highest kappa value of 90.4264%. The result obtained by 2-Channel network is the worst among Siamese, Pseudo-siamese and the proposed networks. Focusing on Fig. 11(c), there are many noise spots in the upper left corner. It is because 2-Channel network processes the two patches jointly, which preserves relative much detail information, especially speckle noises. It has opposite effect on eliminating noises, while outstanding the changed regions. Secondly, in terms of runtime, the proposed architecture ranks only second to 2-channel network. Due to Siamese and Pseudo-Siamese networks have two branches in their networks, the proposed has only one with double numbers of samples. Both of them thus spent more train-time and test-time than the proposed (see Table. 2). Note that the train-time in the table is the average time of each epoch. In all, our method performs favorably against state-of-the-art algorithms in accuracy, while achieving much faster runtime performance. The proposed architecture can learn better representations for our task.

2) RESULTS ON THE FARMLAND DATA SET

On this data set, experimental results are shown in Fig. 12 and the quantitative evaluations are listed in Table. 3. As shown in Fig. 12(a), the preclassification result obtained by

TABLE 3. Change detection results of Farmland data set.

Method/architectures	FP(%)	FN(%)	PCC(%)	Kappa(%)	Train-time	Test-time
JSC	22.9155	5.5956	71.4890	29.9201	-	-
DNN	1.2494	3.8318	94.9187	81.8396	-	-
2-channel	4.1630	3.4158	92.4212	74.8256	**6.0841s**	**2.0978s**
Pseudo-siamese	**0.5197**	4.8012	94.6791	80.2162	8.1127s	3.2894s
Siamese	0.7917	4.0459	96.1624	82.4414	8.1840s	2.7933s
The proposed	1.4204	**2.4935**	**96.0861**	**86.4772**	7.5218s	2.7197s

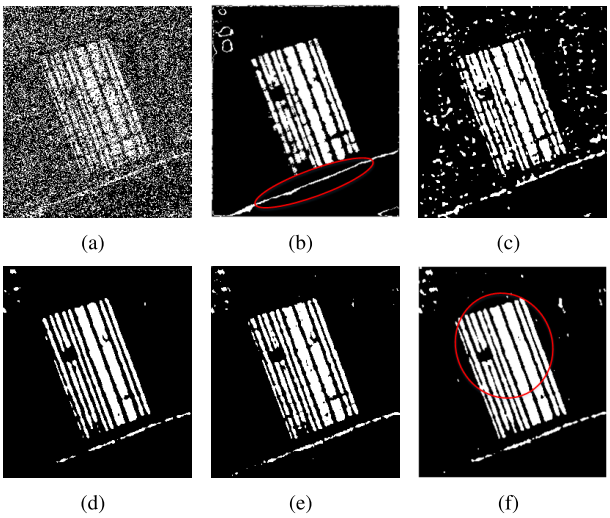


FIGURE 12. Pre-classification and change detection results on Farmland data set obtained by (a) JSC, (b) DNN, (c) 2-channel, (d) Pseudo-siamese, (e) Siamese, (f) The proposed.

JSC method is polluted by catastrophic noise spots. It is merely able to be observed a profile of changed regions. The DNN method gives the best performance about the oblique line section surrounded by the red ellipse in Fig. 12(b), but the rest white changed areas lost lots of detail information. The result obtained by the proposed network is rightly opposite as shown in Fig. 12(f), which gives the better performance of changed areas surrounded by the red ellipse. However, in terms of detecting accuracy, Siamese and the proposed architectures are higher than DNN method (see Table. 3). In Fig. 12(c), it can be seen that the result obtained by 2-Channel architecture is the worst, i.e., false alarms and missed alarms both are high. In all, different method or architectures give a different balance between FP and FN. The proposed architecture exhibits the best Kappa and FN. Due to the weights of the network are randomly initialized and the difference on gradient updating, the performance of the developed network is better than Siamese network. In addition, the runtime of the proposed network outperforms Siamese and Pseudo-siamese networks. Although 2-channel network spent the least time, the accuracy is low. Our method not only yielded the best accuracy, but also spend relatively less training and test times.

3) RESULTS ON THE OTTAWA DATA SET

As a result, the above two experiments show that the proposed method can significantly overcome the effect of mis-preclassification and filter out speckle noises, as well as preserve more detail information. On this data set, the detection results and quantitative analysis are shown in Fig. 13 and Table. 4, respectively. In the preclassification result obtained by JSC [see Fig. 13(a)], dotted and sparse speckle noises are distributed unlike the above two preclassification results. Our proposed method and comparative algorithms all yield good results, but the performance of the proposed method is best. On conserving detail, our architecture outperforms comparative methods. This experiment on this data set states the proposed architecture presents the best performance under the pseudo label with less mis-preclassification. On the other hand, our proposed architecture yields the best FN of 2.4935% and dominant time-consuming, compared with Siamese, Pseudo-Siamese and 2-Channel networks.

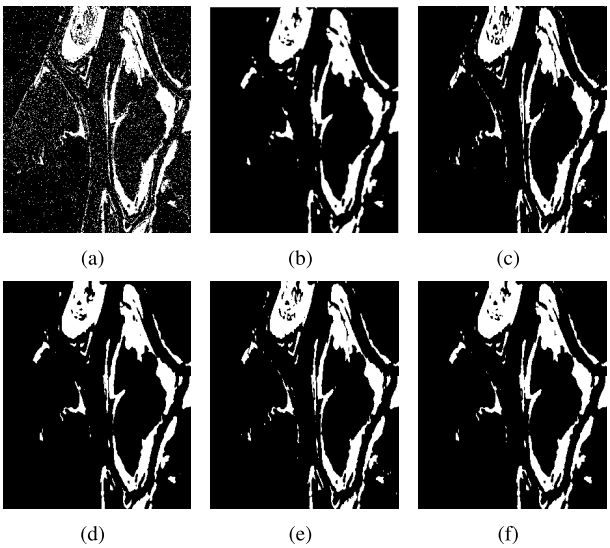


FIGURE 13. Pre-classification and change detection results on Ottawa data set obtained by (a) JSC, (b) DNN, (c) 2-channel, (d) Pseudo-siamese, (e) Siamese, (f) The proposed.

4) RESULTS ON THE BERN DATA SET

From Fig. 14 and Table. 5, we can see that our proposed algorithm shows better performance than comparative

TABLE 4. Change detection results of Ottawa data set.

Method/architectures	FP(%)	FN(%)	PCC(%)	Kappa(%)	Train-time	Test-time
JSC	5.4660	1.7882	92.7458	75.1042	-	-
DNN	0.7172	1.0020	98.2808	93.4949	-	-
2-channel	0.8847	1.4158	97.6995	91.2396	**5.5792s**	**0.7306s**
Pseudo-siamese	**0.4887**	1.1823	98.3291	93.6100	8.8430s	1.4206s
Siamese	0.5823	1.3842	98.0335	92.4583	9.0034s	1.5765s
The proposed	0.5626	**0.9123**	**98.5251**	**94.4100**	5.4485s	0.9019s

TABLE 5. Change detection results of Bird data set.

Method/architectures	FP(%)	FN(%)	PCC(%)	Kappa(%)	Train-time	Test-time
JSC	9.8476	**0.0486**	90.1039	17.9857	-	-
DNN	0.4227	0.1336	99.4437	80.1252	-	-
2-channel	0.1689	0.2373	99.5938	83.4244	3.3820s	**0.3224s**
Pseudo-siamese	0.7483	0.1247	99.1269	72.0608	5.5732s	0.5768s
Siamese	0.2605	0.2528	99.4868	79.6710	5.2633s	0.5194s
The proposed	**0.1324**	0.2329	**99.6347**	**84.8988**	**3.1535s**	0.3285s

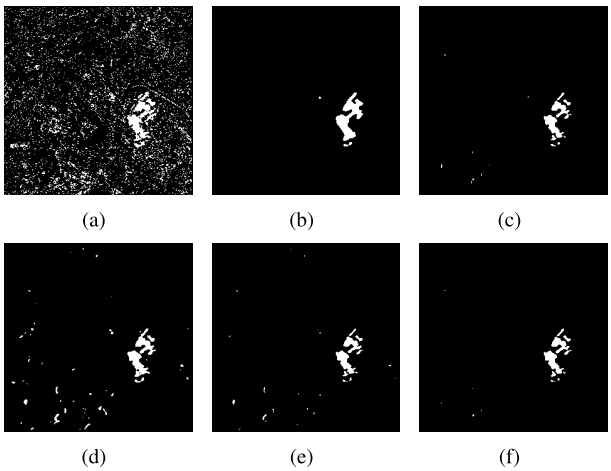


FIGURE 14. Pre-classification and change detection results on Bird data set obtained by (a) JSC, (b) DNN, (c) 2-channel, (d) Pseudo-siamese, (e) Siamese, (f) The proposed.

methods. In terms of the quality of preclassification result [see Fig. 14(a)], the difficulty of this data set is that the noise spots of mis-preclassification gathered into large or small blocks. That is a more difficult for classifiers to correctly detect changes and preserve details. In this data set, the result obtained by Pseudo-siamese network is worse and only outperforms JSC method (see Table. 5). However, the proposed still obtains the excellent performance. Although 2-Channel network spent less runtime, it was worse on reducing noises. DNN method obtained relative good result, but failed on reserving much changed details. In addition, it can be observed the proposed is better on reducing false alarms than Siamese network under the same parameter configuration. In general, the proposed network has a good performance on the accuracy and runtime.

E. ANALYSIS OF PARAMETERS

1) THE SIZE OF THE NEIGHBORHOOD

The size $n \times n$ of the neighborhood patch is an important parameter, which has an effect on speed and accuracy of detection. In [28], it sets the size of the neighborhood to 5×5, which obtained the best performance on Ottawa data set. In [13], the size 11×11 of the neighborhood is adopted by experiment verification. Based on these previous works and our specific network architecture (shown in Section III. D), as well as considering the dimension of neighborhood features reduces with increasing network layers, so we set n to 7, 9, 11, 13 and 15 to survey the different trade-off among FP, FN, PCC and Kappa on Coastline data set. As shown in Fig. 15, when $n = 9$, values of FP and FN are the nearest, values of PCC and Kappa are the highest, i.e., it yields the best balance between noise restriction and detail preservation. However, if the size of the neighborhood is too large or too small, detail information both are more lost with the high FN. In addition, the ablation studies over the variation of training time with increasing size of neighborhood indicate: the larger size of neighborhood, the longer training time is. The larger

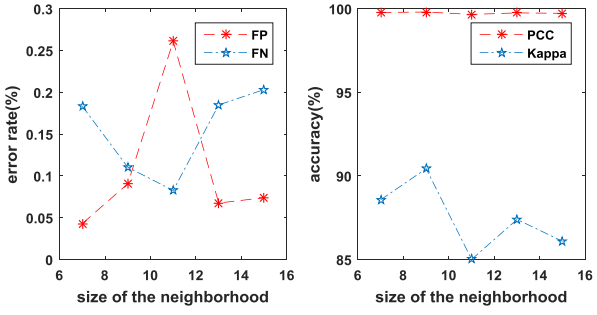


FIGURE 15. The influence of size of the neighborhood with (C4-2)-S2-(C2-6) specification architecture on Coastline data set.

image is the same case. Because it is easily foreseeable, we did not show the corresponding experimental results.

2) THE PARAMETERS OF THE NETWORK

Furthermore, we explore the influence of different sizes of convolutional kernel and pooling on detection performance. The following two networks are shown for comparing. One is described in the part of Section III. D, which is represented as (C4-2)-S2-(C2-6). Another is (C2-2)-S2-(C2-6). They differ in the size of the first convolutional kernel. The results of the two networks with different sizes of the neighborhood are shown in Fig. 16. We can see that the former results is better and more stable. The reason is that the size of the latter convolutional kernel is too small, which leads to the descriptor extracted by the network is not robust enough to noises.

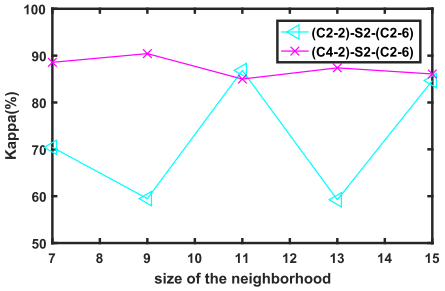


FIGURE 16. The influence of (C4-2)-S2-(C2-6) and (C2-2)-S2-(C2-6) in terms of Kappa accuracy on Coastline data set.

In addition, we study other network architectures such as 9-(C4-2)-S3-(C1-6), 11-(C2-2)-S3-(C2-6), 13-(C5-2)-S3-(C2-6), 15-(C4-2)-S3-(C2-6), where the first number represents the size of the neighborhood. And we try to set various combinations of the number of feature maps at the first and third convolutional layers. We also attempt deeper layers. However, they all have no significant improvement in the detection results.

F. INFLUENCE OF MIS-PRECLASSIFICATION

Due to deep learning algorithm is usually supervised, needing the label data to fine-tune the network, so we can use a preclassifcation strategy to obtain the pseudo labels. However, mis-preclassfication in the pseudo labels always exists. For simplicity and remaining more changed samples, we have no selecting samples pre-classified correctly to train our network. So, what is the effect of mis-preclassification on the proposed method? In order to study this, we use JSC-variant as contrast to carry out this experiment on two data sets. As shown in Fig. 17 and Fig. 18, we can observe that the results based JSC-variant exist less white noise spots on the whole. Quantitative evaluations are listed in Table. 6. On Coastline data set, the result obtained by JSC-variant is less noises, but lost much more changed detail information [see Fig. 17(a)]. Therefore, we could not acquire a better result than based JSC method after fine-tuning. That means missed changed details can not be recovered completely passages through the network. In contrast, the result obtained by JSC is covered by dense white noise spots, but we can see the changed information is more remained [see Fig. 11(a)]. In this case, the network overcomes the interference and yields the better performance. However, on Bern data set, JSC-variant eliminates the nubby noises. The proposed method based on the results of JSC-variant yields state-of-the-art detection result with Kappa of 87.6037%. That is because the distribution of noises is block-shaped in the result obtained by JSC method, which is learnt as changed classes by the networks. From that, we can see that the proposed method can capture useful features when the preclassification with dense noise spots. However, it is helpless and wondering for nubby noises.

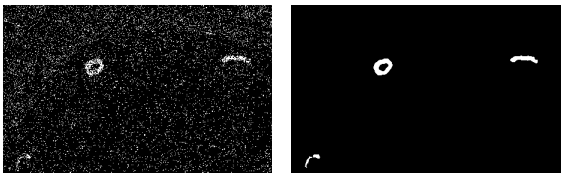


FIGURE 17. Pre-classification and change detection results on Coastline data set obtained by (a) JSC-variant, (b) The proposed network.

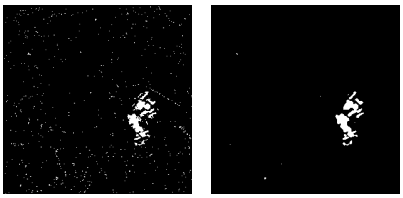


FIGURE 18. Pre-classification and change detection results on Bern data set obtained by (a) JSC-variant, (b) The proposed network.

All in all, for sample label is very coarse, the proposed algorithm can exploit potential changes under noises

TABLE 6. Change detection results of Coastline data set and Bern data set with sample labels obtained by JSC-variant.

Data-set	Methods	FP(%)	FN(%)	OE(%)	PCC(%)	Kappa(%)
Coastline	JSC-variant	6.6103	0.4754	7.0857	92.9143	12.7423
	The proposed	**0.0048**	**0.3024**	**0.3072**	**99.6929**	**83.1752**
Bern	JSC-variant	1.0739	0.1821	1.2560	98.7439	62.9068
	The proposed	**0.1678**	**0.1468**	**0.3146**	**99.6854**	**87.6037**

masking; for the noise is relatively less, it can be fruitful to reduce noise, and eventually obtain a significant improvement for the accuracy of detection result. Our network is constructed like a strong noise canceler. The deep learning algorithm has a strong capacity of learning features even if fine-tuning using corrupted labels. Be similar to DNN method [26], the preclassification labels can be obtained by the traditional suitable algorithms to further improve the performance for different data sets.

V. CONCLUSIONS

In this paper, we explore multiple available CNN-based architectures, which are especially appropriate for change detection task, i.e., take two patches as input. Following the different trade-offs among them, we design ''Siamese samples'' network to receive patch pair as input. The ''Siamese samples'' network learns discriminative patch descriptor by single branch with double samples and then establishes a binary classifier (i.e., identify changed class and unchanged class). The proposed architecture offers a better balance between accuracy and runtime than Pseudo-Siamese, Siamese and 2-channel networks, and it is very robust to speckle noise. Moreover, the proposed method is with the more simplified algorithm framework and reduces the requirement for the accuracy of preclassification label compared with the state-of-the-art.

Experimental results on several real SAR image data sets show the significant improvement in terms of preserving detail features and reducing speckle noises. Taken together, the proposed method is like a strong speckle noise canceler, which improves significantly the final detection performance under an extremely coarse label map. It is simple and efficient to collect labeled samples so that train a classifier with supervision. However, if an initial and available result cannot be obtained, the networks will be unable to work like existing most change detection methods based on deep learning. That will be our next work to decrease the dependency on labeled data and develop semi-supervised or unsupervised methods. Furthermore, we consider to apply deep learning to change detection of heterogeneous and hyperspectral images, which also is a promising area in the future.

ACKNOWLEDGMENT

The authors would like to thank Hao Zhu for his guidance to revise the paper. They would also like to thank the anonymous reviewers for their constructive criticism.

REFERENCES

[1] S. Zagoruyko and N. Komodakis, ''Learning to compare image patches via convolutional neural networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4353–4361.
[2] F. Chatelain, J.-Y. Tourneret, and J. Inglada, ''Change detection in multisensor SAR images using bivariate gamma distributions,'' *IEEE Trans. Image Process.*, vol. 17, no. 3, pp. 249–258, Mar. 2008.
[3] S. P. Chatzis and T. A. Varvarigou, ''A fuzzy clustering approach toward hidden Markov random field models for enhanced spatially constrained image segmentation,'' *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 5, pp. 1351–1361, Oct. 2008.
[4] D.-M. Tsai and S.-C. Lai, ''Independent component analysis-based background subtraction for indoor surveillance,'' *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 158–167, Jan. 2009.
[5] E. J. M. Rignot and J. J. van Zyl, ''Change detection techniques for ERS-1 SAR data,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 31, no. 4, pp. 896–906, Jul. 1993.
[6] F. Bovolo and L. Bruzzone, ''A detail-preserving scale-driven approach to change detection in multitemporal SAR images,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 12, pp. 2963–2972, Dec. 2005.
[7] 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.
[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] 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.
[10] 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.
[11] G. Moser and S. B. Serpico, ''Generalized minimum-error thresholding for unsupervised change detection from SAR amplitude imagery,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 10, pp. 2972–2982, Oct. 2006.
[12] R. Xiao, R. Cui, M. Lin, L. Chen, Y. Ni, and X. Lin, ''SOMDNCD: Image change detection based on self-organizing maps and deep neural networks,'' *IEEE Access*, vol. 6, pp. 35915–35925, 2018.
[13] Y. Li, M. Gong, L. Jiao, L. Li, and R. Stolkin, ''Change-detection map learning using matching pursuit,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4712–4723, Aug. 2015.
[14] M. Gong, Y. Cao, and Q. Wu, ''A neighborhood-based ratio approach for change detection in SAR images,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 2, pp. 307–311, Mar. 2012.
[15] J. Ma, M. Gong, and Z. Zhou, ''Wavelet fusion on ratio images for change detection in SAR images,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 6, pp. 1122–1126, Nov. 2012.
[16] G. Moser and S. B. Serpico, ''Unsupervised change detection with high-resolution SAR images by edge-preserving Markov random fields and graph-cuts,'' in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2012, pp. 1984–1987.
[17] C. Li, C. Xu, C. Gui, and M. D. Fox, ''Level set evolution without re-initialization: A new variational formulation,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 430–436.
[18] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, ''Discriminative learning of deep convolutional feature point descriptors,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2015, pp. 118–126.
[19] Y. Le Cun *et al.*, ''Handwritten digit recognition with a back-propagation network,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, no. 2, 1990, pp. 396–404.
[20] A. Dosovitskiy *et al.*, ''FlowNet: Learning optical flow with convolutional networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2015, pp. 2758–2766.
[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[22] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[23] K. He, X. Zhang, S. Ren, and J. Sun, ''Spatial pyramid pooling in deep convolutional networks for visual recognition,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[24] J. Long, E. Shelhamer, and T. Darrell, ''Fully convolutional networks for semantic segmentation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 79, no. 10, Jun. 2015, pp. 3431–3440.
[25] 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.
[26] 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. 2016.

[27] P. Zhang, M. Gong, L. Su, J. Liu, and Z. Li, "Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 116, pp. 24–41, Jun. 2016.
[28] 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. 2015.
[29] M. Gong, X. Niu, P. Zhang, and Z. Li, "Generative adversarial networks for change detection in multispectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2310–2314, Nov. 2017.
[30] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[31] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'siamese' time delay neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 737–744.
[32] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1592–1599.
[33] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3279–3286.
[34] T. Trzcinski, M. Christoudias, and V. Lepetit, "Learning image descriptors with boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 597–610, Mar. 2015.
[35] C. Osendorfer, J. Bayer, S. Urban, and P. van der Smagt, "Convolutional neural networks learn compact local image descriptors," in *Proc. Int. Conf. Neural Inf. Process.*, 2013, pp. 624–630.
[36] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk, "PN-Net: Conjoined triple deep network for learning local image descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.
[37] S. Chopra, R. Hadsell, and Y. Lecun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 539–546.
[38] 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.
[39] G. H. Rosenfield and K. Fitzpatrick-Lins, "A coefficient of agreement as a measure of thematic classification accuracy," *Photogramm. Eng. Remote Sens.*, vol. 52, no. 2, pp. 223–227, 1986.

HUIHUI DONG received the B.E. degree in computer science and technology from Henan University, Kaifeng, China, in 2015, where she is currently pursuing the Ph.D. degree with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an. Her current research interests include computer vision and remote sensing image understanding.

WENPING MA (M'07) received the B.S. degree in computer science and technology and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2003 and 2008, respectively. Since 2006, she has been with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University, where she is currently an Associate Professor. She has published more than 30 SCI papers in international academic journals, including the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON IMAGE PROCESSING, *Information Sciences*, *Pattern Recognition*, *Applied Soft Computing*, *Knowledge-Based Systems*, *Physica A-Statistical Mechanics and its Applications*, and the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. Her research interests include natural computing and intelligent image processing. She is a member of the Chinese Institute of Electronics and the China Computer Federation.

YUE WU received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2011 and 2016, respectively. His research interests include computer vision and computational intelligence.

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. Since 2006, he has been a Teacher with Xidian University. In 2008 and 2010, he was promoted as an Associate Professor and as a Full Professor, respectively, both with exceptive admission. His current research interests include computational intelligence with applications to optimization, learning, data mining, and image understanding. He is an Executive Committee Member of the Chinese Association for Artificial Intelligence and a Senior Member of the Chinese Computer Federation. He 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 currently the Vice-Chair of the IEEE Computational Intelligence Society Task Force on Memetic Computing.

LICHENG JIAO 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. Since 1992, he has been a Professor with the School of Artificial Intelligence, Xidian University, Xi'an, where he is currently the Director of the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China. He is in charge of about 40 important scientific research projects and has published more than 20 monographs and 100 papers in international journals and conferences. His research interests include image processing, natural computation, machine learning, and intelligent information processing. He is a Fellow of the IEEE Xi'an Section Execution Committee, the Chairman of the Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, a Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, and an expert of the Academic Degrees Committee of the State Council.