Transferred Deep Learning-Based Change Detection in Remote Sensing Images

Meijuan Yang, Licheng Jiao, *Fellow, IEEE*, Fang Liu, *Senior Member, IEEE*, Biao Hou, *Member, IEEE*, and Shuyuan Yang, *Senior Member, IEEE*

***Abstract*—Supervised deep neural networks (DNNs) have been extensively used in diverse tasks. Generally, training such DNNs with superior performance requires a large amount of labeled data. However, it is time-consuming and expensive to manually label the data, especially for tasks in remote sensing, e.g., change detection. The situation motivates us to resort to the existing related images with labels, from which the concept of change can be adapted to new images. However, the distributions of the related labeled images (source domain) and unlabeled new images (target domain) are similar but not identical. It impedes a change detection model learned from source domains being well applied to the target domain. In this paper, we propose a transferred deep learning-based change detection framework to solve this problem. It consists of pretraining and fine-tuning stages. In the pretraining process, we propose two tasks to be learned simultaneously, namely, change detection for the source domain with labels and reconstruction of the unlabeled target data. The auxiliary task aims to reconstruct the difference image (DI) for the target domain. DI is an effective feature, such that the auxiliary task is of much relevance to change detection. The lower layers are shared between these two tasks in the training process. It mitigates the distribution discrepancy between the source and target domains and makes the concept of change from the source domain adapt to the target domain. In addition, we evaluate three modes of the U-net architecture to merge the information for a pair of patches. To fine-tune the change detection network (CDN) for the target domain, two strategies are exploited to select the pixels that have a high possibility of being correctly classified by an unsupervised approach. The proposed method demonstrates an excellent capacity for adapting the concept of change from the source domain to the target domain. It outperforms the state-of-the-art change detection methods via experimental results on real remote sensing data sets.**

***Index Terms*—Adaptation, change detection, deep neural networks (DNNs), reconstruction, remote sensing.**

I. Introduction

Change detection in remote sensing aims to analyze two (or more) images acquired over the same geographical area at different times and identify the changes that may occur. It plays an important role in diverse applications, including assessing natural or anthropic disasters, monitoring land-use development over time, investigating deforestation, and studying urban growth [1], [2]. The available earth observation satellites lead to large numbers of remote sensing images to monitor the environmental changes. Synthetic aperture radar (SAR) images have the advantage of being insensitive to atmospheric and sunlight conditions [3], and optical images are cheap to be obtained. Therefore, these two kinds of images are most commonly used in change detection studies [4], [5].

Image preprocessing is necessary to suppress some distortions and facilitate change detection. To be specific, radiometric correction [6] and geometric registration [7], [8] make the two (or more) remote sensing images radiometrically and spatially comparable. Since SAR images are prone to be disturbed by speckle noise [9], [10], a number of low-pass filters are exploited, e.g., the Lee filter [11] and the sigma filter.

Several approaches have been developed for change detection in remote sensing images over the past decades. These methods are grouped into two categories: unsupervised and supervised methods. Unsupervised methods do not need prior knowledge from the labeled data, while supervised methods require training data.

Generation and analysis of a difference image (DI) are two major steps in the unsupervised methods. Generally, the DI can be generated by a ratio operator [10], [12]–[14] or subtraction [15]. For multispectral images, DI is represented by the spectral change vector (SCV). Some mid-level features [10], [15], [16] extracted from raw images or a DI can gain rich information to assist the comparison of two (or more) remote sensing images, such as PCA. Analysis of the DI can be accomplished by the change vector analysis (CVA) [17], thresholding [18], or clustering [15]. CVA-based methods, e.g., the compressed CVA [19], can distinguish different kinds of changes. Statistical modeling is usually combined with thresholding to deal with the change detection problem. The magnitude of the DI can be modeled by the generalized Gaussian distribution [13] or a Rayleigh–Rice mixture density [20]. Among clustering methods, a reformulated fuzzy local information C-means algorithm [21] drew much attention for making the spatial context incorporated in a novel fuzzy way. More techniques are investigated in [1] when applied to change detection in remote sensing images. Unsupervised methods are more flexible and widely used in practice. However, they do not take advantage of prior knowledge from the labeled data. Furthermore, they rely on some model assumptions or

Manuscript received September 29, 2018; revised January 30, 2019 and March 28, 2019; accepted March 29, 2019. Date of publication April 30, 2019; date of current version August 27, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant U1701267 and in part by the State Key Program of National Natural Science of China under Grant 61836009. *(Corresponding author: Licheng Jiao.)*

The authors are with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education and the Joint International Research Laboratory of Intelligent Perception and Computation, International Research Center for Intelligent Perception and Computation, School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: mjuanyang@gmail.com; lchjiao@mail.xidian.edu.cn).

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

Digital Object Identifier 10.1109/TGRS.2019.2909781

similarity rules to discriminate between changed and unchanged classes. With these disadvantages, it is hard for unsupervised methods to achieve a breakthrough in change detection.

Efforts were also put in the supervised (or semisupervised) methods for change detection in remote sensing images. Supervised methods require prior knowledge from labels: image classification maps (in postclassification comparison methods) or change detection maps. In postclassification comparison methods, each remote sensing image is classified independently, and then, two (or more) classification maps are compared to derive the final change detection map (CD map) [22]. Accordingly, the accuracy of the CD map depends on the quality of the image classification maps. Compound classification [23], [24] that incorporates temporal dependence between multitemporal images can increase the accuracy of the postclassification comparison techniques. In some works, temporal dependence is modeled by the joint probabilities of land-use transitions. With change detection maps as supervision, support vector machine (SVM) [14], [25]–[28] and one-class [29] classifiers are popular in achieving CD maps by supervised learning. Generally, training supervised CD methods usually requires a large amount of labeled data. However, it is time-consuming and expensive to manually label the data, especially for tasks in remote sensing, e.g., change detection. Moreover, the procedures of feature extraction and classifier training are separated in most existing supervised CD methods. It will affect the overall performance.

To sum up, the aforementioned change detection methods only use the information from the current images themselves. The performance of change detection methods will be boosted if they can adapt additional knowledge from the existing related images with labels to the new images. When this knowledge is reused, data shift between two scenes in remote sensing becomes a challenging problem. The data shift stems from sensor characteristics, noise and distortions, and the spectral and radiometric of the ground object. Due to the data shift, it should be noted that a change detection model trained from existing related images (source domain) cannot be directly applied to new images (target domain) with satisfied performance. In machine learning, multitask learning can always help a model adapt to a new data space. When a model learns two tasks simultaneously, a general representation can be derived from the model to facilitate the adaptation.

Drawing inspiration from this assumption, we pretrain a change detection network (CDN) that cooperates with a reconstruction network. A deep reconstruction detection network (RDN), shown in Fig. 2, is proposed to achieve this goal. RDN includes two tasks: the change detection is performed in the source domain with image labels, and the reconstruction is fulfilled in the target domain without labels. In Fig. 2, discriminate features related to changes can be learned from the source domain, while they may be only suitable for the source domain. In the target domain, we propose to reconstruct a DI map. The motivation behind this idea is that DIs are highly effective features to change detection. The reconstruction of a DI map has much relevance to detecting changes. With the reconstruction task, the concept of change from the source domain can be consequently adapted to the target domain.

In this paper, a transferred deep learning-based change detection framework is proposed in remote sensing images. We aim at learning a CDN that performs well on the unlabeled target data with the additional knowledge from related labeled source data. From this perspective, the proposed framework comprises two stages (see Fig. 1). At the pretraining stage, the CDN is pretrained by the source data. Meanwhile, reconstructing the DI for the target domain by the reconstruction network is proposed as an auxiliary task to change detection. Two networks are integrated to form RDN and share the lower layers. As a result, the lower layers are suitable not only for the source domain but also for the target domain. It mitigates the discrepancy of the feature distributions between these two domains. At the fine-tuning stage, the lower layers of the CDN are frozen, and the higher layers are fine-tuned for the target domain. To achieve this goal, region- and boundary-based strategies are proposed to select the pixels that have a high possibility to be correctly classified by an unsupervised approach. In this paper, we not only reuse the prior knowledge from the source domain but also exploit the information from the target domain for change detection. In consequence, the CD map for the target domain is improved significantly through the proposed framework.

The main contributions of this paper are concluded as follows.

1) We propose a transferred deep learning-based change detection framework, in which unlabeled images gain additional knowledge from related labeled images.
2) At the pretraining stage, RDN is proposed to learn the change detection task for the source domain and reconstruction task for the target domain simultaneously. It mitigates the distribution discrepancy between the source and target domains from the feature level.
3) Region- and boundary-based strategies are proposed to select the pixels that have a high possibility of being correctly classified by an unsupervised approach. It is an important step in fine-tuning the CDN for the target domain.

The remainder of this paper is organized as follows. In Section II, a brief review is given about change detection methods based on DNNs and transfer learning methods. In Section III, we present the framework of the proposed change detection method, followed by the CDN, the reconstruction network, joint pretraining of the two tasks, and the fine-tuning process in detail. In Section IV, we provide some experimental results and detailed analysis to demonstrate the effectiveness of the proposed method. Finally, this paper is concluded in Section V.

II. Related Work

A. Change Detection Based on DNNs

DNNs can extract high-level and task-specific image features in a data-driven fashion [30], [31]. In recent years, DNNs methods made significant progress in many disciplines [32], [33]. Besides the natural images and videos, the DNN-based methods are also applied in remote sensing. Several deep neural network (DNN) architectures are used for change detection, e.g., restricted Boltzmann machines (RBMs) [34], [35],

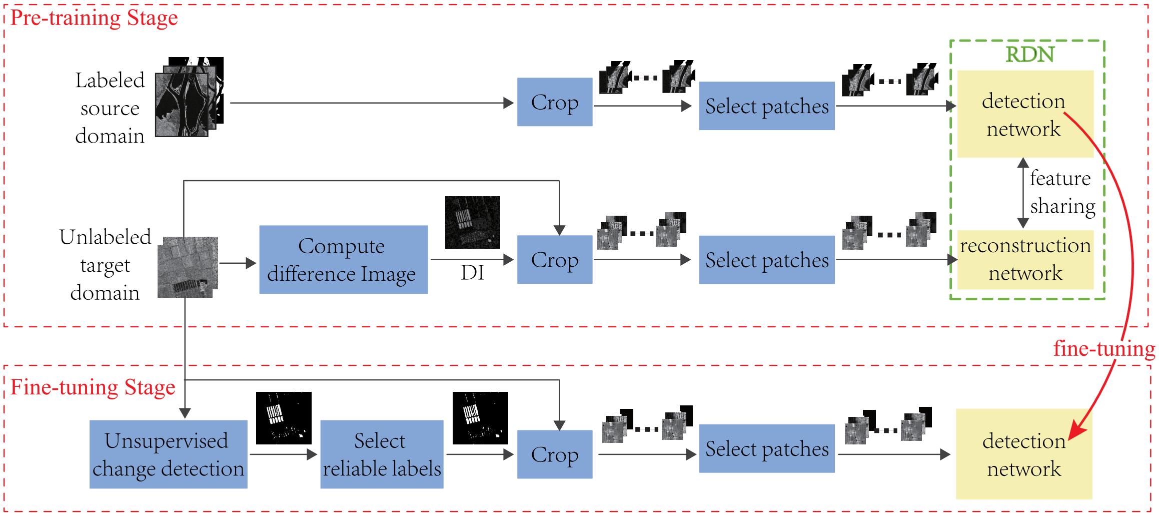


Fig. 1. Pipeline of the proposed transferred deep learning-based change detection method in remote sensing images.

autoencoders [36], Siamese convolutional networks [37], and VGG-16 network [38]. Among these methods, the general representations are pretrained layerwise [34], [35] or by ImageNet [38], [39]. The task-specific features extracted in higher layers are learned with the supervision of the training data from the same domain [37]. The supervision can also be the pixels with a high possibility of being correctly classified from a preclassification map [34], [35]. In this paper, we propose a transferred DNN-based change detection framework inspired by multitask learning. It uses the information not only from the current image (target domain) itself but also from the related labeled images (source domain).

B. Pixel-Level Domain Adaptation

Maximum mean discrepancy [40] and adversarial learning [41] are two kinds of approaches used in the DNN-based domain adaptation. Most of the existing domain adaption methods are applied in image-level tasks. However, domain adaptation for pixel-level tasks has not been explored extensively, e.g., semantic segmentation and change detection. It is still challenging to adapt the pixel-level knowledge from the source domain to the target domain. In this paper, we propose to reconstruct the DI of the target domain to mitigate the domain discrepancy between the source and target samples.

III. Method

A pair of remote sensing images $\mathbf{X}_1$ and $\mathbf{X}_2$ are considered. $\mathbf{X}_1 = \{x_1(i,j), 1 \leqslant i \leqslant H, 1 \leqslant j \leqslant W\}$ and $\mathbf{X}_2 = \{x_2(i,j), 1 \leqslant i \leqslant H, 1 \leqslant j \leqslant W\}$ with a size of $H \times W$ are acquired at t_1 and t_2 over the same geographical area, respectively. Given $\mathbf{X}_1$ and $\mathbf{X}_2$, we aim to generate a change detection map $\mathbf{CM} = \{\text{cm}(i,j), 1 \leqslant i \leqslant H, 1 \leqslant j \leqslant W\}$. Let $\boldsymbol{\Omega} = \{\omega_u, \omega_c\}$ be the set of labels and $\text{cm}(i,j) \in \boldsymbol{\Omega}$. ω_u and ω_c denote the unchanged class and changed class, respectively. $\mathbf{X}_1$ and $\mathbf{X}_2$, which have been registered geometrically and radiometrically, are preprocessed by a Lee filter [11]. This step can remove the speckle noise from SAR images.

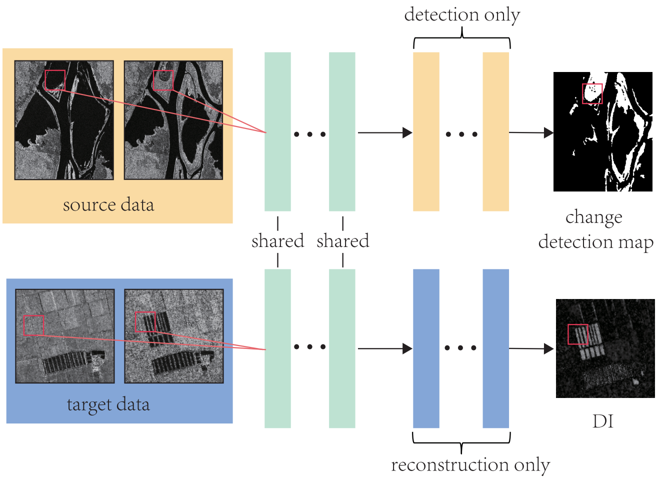


Fig. 2. Proposed RDN includes a CDN for the source domain and a reconstruction network for the target domain. The lower layers (green) are general and shared between the two tasks. The features from higher layers depend on the specific data set and task that are learned separately.

In this paper, the remote sensing images are divided into overlapping patches that are fed into the change detection and reconstruction networks. The outputs are also patch-based. The size of these patches is $m \times m$. The spatial shift between two adjacent patches is h. Then, some patches are selected to train the CDN. Our attention is focused on some patches that contain much information about changes. On the one hand, the patches containing the changed pixels should be selected. On the other hand, the patches where pixels are prone to be misdetected into the change classes also have priorities. Inspired by [42], using a deep learning model for face verification, a pair of patches are exchanged to generates two input modes. As a result, the information in the training process is doubled. For example, one area is changed from farmland (at t_1) to water (at t_2). After exchanging, we also have the change information from water to farmland.

Fig. 1 shows the pipeline of the proposed transferred deep learning-based change detection method that consists of two stages. At the pretraining stage, RDN is trained with both the source and target data. Two temporal images and the reference from the source domain are cropped into overlapping patches to pretrain the CDN. For the target domain, the patches from two temporal images and the DI are used to train the reconstruction network. With these selected patches from two domains, RDN is trained until it converges. After pretraining, lower layers of the CDN are frozen. At the fine-tuning stage, higher layers of the CDN are fine-tuned for the target domain with some reliable labels. Two strategies are exploited in selecting the pixels that have a high possibility of being correctly classified from an initial CD map. After fine-tuning, the CDN is adapted to fit the target domain.

In the testing phase, all the patches cropped from the target domain are inferred by the detection network that has excellent adaptation capacity to the target domain after training. To get a whole CD map for the target domain, the overlapping area between two adjacent patches is averaged to avoid the block artifacts.

A. RDN

The proposed deep RDN is composed of two branches, as shown in Fig. 2. One is the CDN for the source domain, and the other is the reconstruction network for the target domain. The former task can learn the concept of change from the source domain, while the latter aims to reconstruct the DI for the target domain. On the one hand, RDN makes the shared features of the two tasks generalized to both the source and target domains. On the other hand, reconstruction of DI facilitates the change detection in the target domain since DI is an effective feature for change detection.

The detailed architecture of RDN is presented in Section IV-B. The existing research on DNNs has revealed that the features extracted from lower layers are general and that from higher layers are task-specific [40], [43], [44]. In other words, features from lower layers show superior transferable ability. Based on the above-mentioned analysis, the features from lower layers are shared by both the change detection and reconstruction networks, and the features from higher layers are learned separately for these two tasks.

The labeled samples from the source domain are denoted by $\{\mathbf{x}_k^s, \mathbf{y}_k^s\}_{k\in[N^s]}$, where $\mathbf{x}_k^s$ are the kth pair of patches with a size of $m \times m \times 2$. $\mathbf{y}_k^s$ is the corresponding CD map with a size of $m \times m$. N^s is the number of selected training patches from the source domain. There are N^t training patches $\{\mathbf{x}_k^t\}_{k\in[N^t]}$ for the target domain without labels. Let P denote the probability distribution for the source domain and Q for the target domain, where $P \neq Q$. We aim to learn a shared representation that can minimize the discrepancy between P and Q and a change detection model that can be adapted to the target domain.

In the following, the architecture of the patch-based CDN is illustrated as well as three modes to merge information between patch pairs. Second, how to reconstruct the DI for the target domain is described in detail. Third, we present how to jointly train the change detection and reconstruction networks and learn a shared representation. Fourth, region- and boundary-based strategies are illustrated in the fine-tuning process.

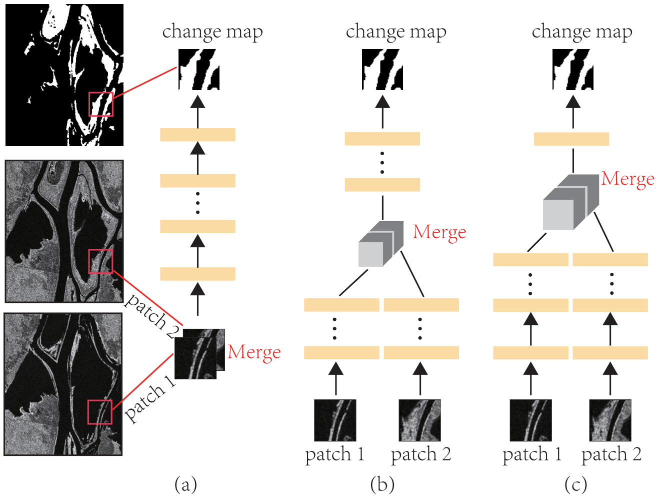


Fig. 3. Three modes for merging information for the patch pairs. (a) Two patches are concatenated in the input in the early mode. (b) Feature maps of the two patches are concatenated after the second max-pooling layer in the middle mode. (c) Feature maps are merged until the last two layers in the late mode. The detailed information about the detection networks with three modes is provided in Section IV-B.

B. Change Detection Networks

As a pixel-level task, change detection aims to predict the class label (changed or unchanged) for each pair of pixels. U-net [45] is promising in pixel-level tasks. It consists of a downsampling path and a symmetric upsampling path. The context information from the downsampling path is propagated into the upsampling path. The skip connections between the downsampling path and the upsampling path make U-net architecture yield segmentation maps with high resolution. Therefore, we use U-net architecture to design the CDN. Our patch-based CDN has fewer layers than that in [45]. The detailed architecture is provided in Section IV-B.

As the input of the detection network is a pair of patches, when and how to merge the information for a pair of patches play an important role. In this paper, the information can be merged in the lower layers (early mode), the middle layers (middle mode), or the higher layers (late mode). In the early mode, we concatenate a pair of patches in the input [see Fig. 3(a)]. As the two patches are jointly processed from the first layer, the network provides great flexibility and fast to converge. When merging the information in the middle mode, we extract the feature maps for each patch separately until the feature maps reach the lowest resolution [see Fig. 3(b)]. Then, we concatenate the feature maps of the pair of patches serving as the input of the upsampling path. When merging information in the late mode, we extract the feature maps for each patch separately except the last two layers [see Fig. 3(c)]. The CDN with the late mode is similar to the traditional feature matching methods, in which the features are extracted independently and then matched with a similarity function. As the feature maps are extracted for each patch separately in the middle and late modes, the parameters can be shared or not shared between the two patches. The number of the parameters increases if not shared. The detailed description about the detection networks with three modes is provided in Section IV-B. We will decide when and how to merge the information for patch pairs via experiments.

We then define the CDN formally. Given a patch pair $\mathbf{x}_k^s$ from the source domain, the inference $\hat{\mathbf{y}}_k^s$ by the detection network is a binary map indicating the changes that occur

$$\hat{\mathbf{y}}_k^s = f_{\theta_s}(g_{\theta_c}(\mathbf{x}_k^s)) \tag{1}$$

where g and θ_c correspond to the first few layers and their parameters, respectively. They are shared with the reconstruction network in the target domain. f denotes the final layers used by the CDN only, and θ_s is the corresponding parameter. The loss function to the detection network is defined as

$$L^s = -\sum_{k=1}^{N_s}\sum_{i=1}^{m}\sum_{j=1}^{m} l_{\text{bce}}(\mathbf{y}_k^s(i,j), \hat{\mathbf{y}}_k^s(i,j)) \tag{2}$$

where $\mathbf{y}_k^s(i,j), \hat{\mathbf{y}}_k^s(i,j) \in \Omega$ and $l_{\text{bce}}(\mathbf{y}_k^s(i,j), \hat{\mathbf{y}}_k^s(i,j)) = (\mathbf{y}_k^s(i,j)\log(\hat{\mathbf{y}}_k^s(i,j)) + (1-\mathbf{y}_k^s(i,j))\log(1-\hat{\mathbf{y}}_k^s(i,j)))$ is the binary cross-entropy loss for the pixel located at (i,j) in the kth patch pair.

C. Reconstruction Networks

In remote sensing, data shift always exists between two scenes. With data shift, a CDN trained from one domain will perform poorly when it is applied in another domain. In order to minimize the gap between the source and target domains, we put forward a reconstruction network to reconstruct the DI of the target domain. DI map is an effective feature for change detection in remote sensing images. The task of reconstructing a DI map has much relevance to detecting changes in the target domain. Consequently, when the lower layers are shared between the change detection task in the source domain and the reconstruction task in the target domain, they are also suitable for detecting changes in the target domain.

To generate a DI, the ratio operator (including log-ratio and mean-ratio operators) and the subtraction operator are widely used. Considering that the speckle noise of SAR imaging is multiplicative, we select to use the log-ratio operator in this paper. The DI $\mathbf{R} = \{r(i,j), 1 \leqslant i \leqslant H, 1 \leqslant j \leqslant W\}$ for the target domain is generated by

$$r(i,j) = \left|\log\frac{x_1(i,j)}{x_2(i,j)}\right|. \tag{3}$$

A reconstruction network is proposed to reconstruct the DI, the architecture of which is provided in Section IV-B in detail. The DI for the kth patch from the target domain is reconstructed by

$$\hat{\mathbf{r}}_k^t = \varphi_{\theta_t}(g_{\theta_c}(\mathbf{x}_k^t)) \tag{4}$$

where φ and θ_t denote the final layers and the corresponding parameters of the reconstruction network, respectively. g and θ_c are shared between the change detection and the reconstruction tasks.

For the reconstruction network, the loss function L^t is defined as follows:

$$L^t = \sum_{k=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{m} l_{sqr}(\mathbf{r}_k^t(i,j), \hat{\mathbf{r}}_k^t(i,j)) \tag{5}$$

Algorithm 1 Joint Pretraining of the Deep RDN

```
Input:
  Labeled data from the source domain: $\{\mathbf{x}_k^s, \mathbf{y}_k^s\}_{k\in[N^s]}$;
  Unlabeled data from the target domain: $\{\mathbf{x}_k^t\}_{k\in[N^t]}$;
  A DI for the target domain: $\{\mathbf{r}_k^t\}_{k\in[N^t]}$;
  Parameters: learning rate $lr$, threshold $T_l$, weight $\alpha$;
while ($\sigma_l < T_l$) do
  for each batch from the source domain do
    Compute $L^s$ for the detection network using (1), (2)
    Update $\theta_c$ and $\theta_s$: $\theta_c \leftarrow \theta_c - \alpha \cdot lr \cdot \frac{\partial L^s}{\partial \theta_c}$,
      $\theta_s \leftarrow \theta_s - \alpha \cdot lr \cdot \frac{\partial L^s}{\partial \theta_s}$
  end for
  for each batch from the target domain do
    Compute $L^t$ for the reconstruction network using (4), (5)
    Update $\theta_c$ and $\theta_t$: $\theta_c \leftarrow \theta_c - (1-\alpha) \cdot lr \cdot \frac{\partial L^t}{\partial \theta_c}$,
      $\theta_t \leftarrow \theta_t - (1-\alpha) \cdot lr \cdot \frac{\partial L^t}{\partial \theta_t}$
  end for
  Compute the standard deviation $\sigma_l$ for the average $L^t$ within the last 10 epoches
end while
Output:
  Parameters for the proposed RDN: $\theta_c$, $\theta_s$, and $\theta_t$;
```

where $\mathbf{r}_k^t$ is the DI for the kth patch cropped from $\mathbf{R}$ and $\hat{\mathbf{r}}_k^t$ is the reconstructed version of $\mathbf{r}_k^t$. $l_{sqr}(\mathbf{r}_k^t(i,j), \hat{\mathbf{r}}_k^t(i,j)) = \|\mathbf{r}_k^t(i,j) - \hat{\mathbf{r}}_k^t(i,j)\|_2^2$ is the squared loss of each pixel in reconstructing $\mathbf{r}_k^t$ for the target domain.

D. Source–Target Joint Pretraining

The last two layers of the CDN and the reconstruction network are considered as domain specific layers $(f_{\theta_s}, \varphi_{\theta_t})$. The common subnetwork g_{θ_c} is shared by the two networks. To learn the shared features from g_{θ_c}, the CDN for the source domain and the reconstruction network for the target domain are optimized alternately. $(f_{\theta_s}, \varphi_{\theta_t})$ are updated separately (see Algorithm 1). The loss function of the two tasks are minimized by

$$L^{\text{pre}} = \alpha L^s + (1-\alpha)L^t \tag{6}$$

where $0 \leqslant \alpha \leqslant 1$ is a constant parameter balancing the importance between the change detection and reconstruction tasks. We set α via experiments. To optimize the total loss function L^{pre}, L^s and L^t are minimized alternately via stochastic gradient descent (SGD).

When the reconstruction loss L^t stabilizes, RDN stops training (see Algorithm 1). After that, the shared representations extracted from the lower layers with θ_c are also suitable for change detection in the target domain. We visualize the shared representations with feature maps. Six feature maps (16 maps in total produced by the common subnetwork) and the predicted CD map for each pair of patches are shown in Fig. 4.

E. Fine-Tuning

After pretraining, the extracted features from g_{θ_c} are also suitable for the target domain. The next step is to fine-tune f_{θ_s}

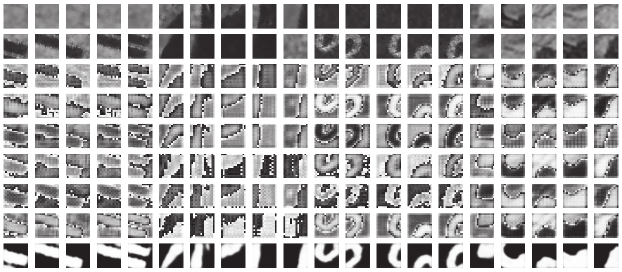


Fig. 4. Feature maps of some patch pairs are produced by the common subnetwork (parameterized by θ_s) shared between change detection and reconstruction tasks. The patch pairs are shown in the first and second lines, respectively. From the third line to the eighth line, six (16 in total) feature maps are provided. The predicted CD maps for patch pairs are shown in the last line.

```
Algorithm 2 Fine-Tuning of the CDN
Input:
    Pseudo-labeled target data: $\{\mathbf{x}_k^t, \mathbf{y}_k^t\}_{k\in[N^t]}$;
    A weight map: $\mathbf{w}$;
    Parameters: $\theta_c$, $\theta_s$, learning rate $lr$;
for each epoch do
    for each batch from the source domain do
        Compute $L^{ft}$ using (10), (11)
        Update $\theta_s \leftarrow \theta_s - lr \cdot \frac{\partial L^{ft}}{\partial \theta_s}$
    end for
end for
Output:
    Parameters for the proposed RDN: $\theta_s$;
```

with labeled data from the target domain to make the change detection model perform well on the target data. Therefore, the lower layers g_{θ_c} are frozen, and only the high layers f_{θ_s} are optimized at the fine-tuning stage (see Algorithm 2).

However, the target domain is fully unlabeled for change detection in remote sensing. The conventional way is to use the CD maps from unsupervised methods as the pseudolabels. Since the pseudolabels are not entirely correct, patches and pixels that have a high possibility of being correctly classified should be chosen. In this paper, we propose two strategies to choose reliable labels for the target domain. The region-based strategy places stress on the difference between each changed area and the surrounding unchanged area. Based on the difference, the label of all the pixels in one small changed area is updated. The boundary-based strategy puts emphasis on the pixels that are close to the center of the changed or unchanged areas. These pixels have a high possibility of being correctly classified and can be used at the fine-tuning stage.

1) Region-Based Strategy: For an initial CD map, we divide it into small regions based on the connectivity of each pixel. Pixels in each region are not connected to those in another region (e.g., a small region in Fig. 5). Then, we evaluate the land-cover variation for the corresponding small patches A_1 and A_2 (cropped from $\mathbf{X}_1$ and $\mathbf{X}_2$), respectively

$$\text{var}_1 = \left|\log\frac{\text{mean}(A_{c,1})}{\text{mean}(A_{u,1})}\right| \tag{7}$$

$$\text{var}_2 = \left|\log\frac{\text{mean}(A_{c,2})}{\text{mean}(A_{u,2})}\right|. \tag{8}$$

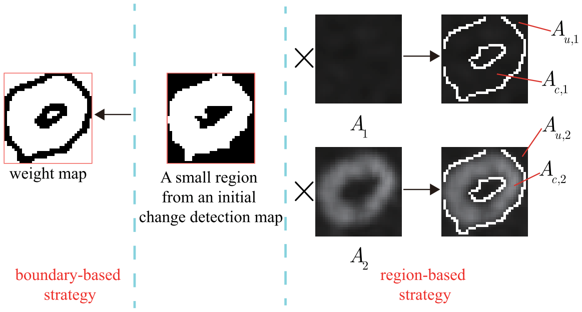


Fig. 5. Two strategies used in choosing reliable labels for the target domain.

According to the initial CD map, change occurs between $A_{c,1}$ and $A_{c,2}$, and there is no change between $A_{u,1}$ and $A_{u,2}$. Referring to (7), a small value of var$_1$ indicates a high probability that only one kind of ground object exists in A_1. When var$_1$ and var$_2$ are both small, it means that the ground object is not varied in both A_1 and A_2. In this case, it is essential to compare A_1 and A_2. If the difference between A_1 and A_2 is large, we conclude that A_1 and A_2 are totally changed. Otherwise, they are totally unchanged.

2) Boundary-Based Strategy: In the initial CD map, the pixels close to the border between the changed and unchanged areas are difficult to be correctly classified. On the contrary, the pixels close to the center of the changed or unchanged area have a high possibility of being correctly classified. Based on this observation, the former does not have the confidence to fine-tune the CDN ($w(i, j) = 0$), while the latter gets the confidence ($w(i, j) = 1$).

Let $\mathcal{N}_{ij}$ denote the neighborhood of the pixel at position (i, j) with a size of $n_b \times n_b$. Then, a sum of the changed pixels in $\mathcal{N}_{ij}$ is obtained, sum_{ij}. When $sum_{ij} \leqslant \delta$, the pixel (i, j) is close to the center of the unchanged area. $0 \leqslant \delta \leqslant n_b^2$ is a constant value. When $sum_{ij} \geqslant n_b^2 - \delta$, it means that the pixel (i, j) is close to the center of the changed area. The pixels are located close to the border between the changed and unchanged areas if $\delta < sum_{ij} < n_b^2 - \delta$. Hence, a weight map $\mathbf{w}_k = \{w(i, j), 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant m\}$ is precomputed for each pixel by

$$w(i, j) = \begin{cases} 1, & \text{sum}_{ij} \geqslant n_b^2 - \delta \text{ or sum}_{ij} \leqslant \delta \\ 0, & \text{else.} \end{cases} \tag{9}$$

In this paper, n_b and δ are both set to 5 empirically.

With the weight map $\mathbf{w}_k$, the loss function in the fine-tuning process is defined by

$$\hat{\mathbf{y}}_k^t = f_{\theta_s}\left(g_{\theta_c}\left(\mathbf{x}_k^t\right)\right) \tag{10}$$

$$L^{ft} = -\sum_{k=1}^{N_t}\sum_{i=1}^{m}\sum_{j=1}^{m} \mathbf{w}_k(i, j) \cdot l_{\text{bce}}\left(\mathbf{y}_k^t(i, j), \hat{\mathbf{y}}_k^t(i, j)\right). \tag{11}$$

After pretraining, the features from the lower layers g_{θ_c} are aligned for the source and target domains. After fine-tuning, the higher layers f_{θ_s} of the CDN are also suitable for the target domain. Consequently, the CDN is capable to infer an accurate CD map for the target domain.

IV. Experiments

After evaluating three modes of merging information for patch pairs (in Fig. 3), the architecture of the CDN is specified. When only source data are used, the CDN is considered as our baseline (s-CDN). When CDN is pretrained by both the source and target data (see Algorithm 1), we denote it as pt-CDN. Furthermore, when pt-CDN is fine-tuned by the reliable labels of the target domain, the method is called pt-CDN+ft (see Algorithm 2). The proposed method is validated on two data sets: Yellow River data set and Mexico data set. We also compare it with some of the state-of-the-art change detection algorithms. The experimental results and detailed analysis demonstrate the adaptation capability of our proposed method. In the following, the description of data sets, evaluation measures, and experimental settings for change detection are provided first.

A. Data Set Descriptions and Evaluation Measures

The first data set used in our experiments was acquired at the region of Yellow River Estuary in China. Two SAR images were captured by Radarsat-2 in June, 2008, and June, 2009, respectively. The speckle noise is large in the SAR image captured in June, 2009, which makes change detection a challenging problem. The size of the two SAR images is 7666×7692. In order to validate the proposed and the state-of-the-art change detection methods, we select three areas with small size from the Yellow River data set. They are named Farmland, River, and Coastline and shown in Figs. 11–13, respectively. The corresponding reference images are shown in Figs. 11(c)–13(c). The reference images were created by integrating prior information with photograph interpretation based on the input images.

The second data set was acquired at urban Mexico by Landsat-7 ETM+. Two temporal optical images were derived from band 4, providing a view of the vegetation in April, 2000, and May, 2002, respectively. In Fig. 14, two images with a size of 512×512 are shown, as well as the corresponding reference image.

Five measures are used to evaluate the predicted CD maps quantitatively. False positives (FPs) error indicates the number of the unchanged pixels classified into the changed class incorrectly. False negatives (FNs) error indicates the number of the changed pixels classified into the unchanged class incorrectly. Overall error (OE) is the sum of the FP and FN. The percentage of correct classified pixels (PC) [46] is calculated as follows:

$$\mathrm{PC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{N}} \tag{12}$$

where TP and TN represent true positives and true negatives, respectively. N is the number of all the pixels to be classified. The last measure that we used is the Kappa coefficient [23], [46] that measures the agreement between the predicted CD map and the reference image. It is calculated by

$$\mathrm{Kappa} = \frac{\mathrm{PC} - \mathrm{PE}}{1 - \mathrm{PE}} \tag{13}$$

TABLE I
Source Data Descriptions

Cite	Sensor	t_1	t_2	Size
Ottawa	Radarsat	May, 1997	Aug., 1997	290×350
Bern	ERS-2	Apr., 1999	May, 1999	301×301
de Gaulle airport	ERS-1	July, 1997	Oct., 1998	240×370

where PE is defined as

$$\mathrm{PE} = \frac{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN}) + (\mathrm{FN} + \mathrm{TN})(\mathrm{TN} + \mathrm{FP})}{\mathrm{N}^2}. \tag{14}$$

B. Experimental Settings

1) Source Data and Training Samples Selection: Three data sets are used as the source data to train the proposed RDN, including Ottawa, Bern, and de Gaulle airport. Theoretically, the source domains should include diverse remote sensing data sets, e.g., with different sensor characteristics, noise, and distortions. However, it is time-consuming and expensive to manually label these data. In this paper, three source domains are selected according to the sensors from which the data sets are captured (see Table I). In addition, the speckle noise is small in the Ottawa data set, while it is considerable in the de Gaulle airport data set. The detailed information about the three source data sets is listed in Table I, in which ERS is short for European Remote Sensing satellite. From Table I, we can see that all the source data sets are SAR images. Two temporal images of the three source data sets and the corresponding references are shown in Fig. 7.

From the source data sets, we select the patches that contain changed pixels of more than 5%. In addition, we also select the patches where unchanged pixels are prone to be classified into the changed class incorrectly. These selected patches contain rich information for distinguishing between the changed class and the unchanged one. There are 21 926 training samples for the Ottawa data set. For the Bern and de Gaulle airport data sets, we select 2080 and 35 508 training samples, respectively. Since labels are not available in the target domain, we concentrate on the patch pairs that have a large difference. In the target domain, we select 12 974, 17 692, 51 248, and 21 526 training samples for the Farmland, River, Coastline, and Mexico data sets, respectively. In the testing, all the patches cropped from the target domain are used as testing samples.

2) Patch Size m and Spatial Shift h: For patch size m and spatial shift h, we have done a systematic search. We set m in the range $\{16, 28, 48, 64\}$ and h in the range $\{2, 4, 8, 16\}$. Different values of h in the training phase have an influence on the number of training samples, while h in the testing phase affects the resolution of the CD maps. Therefore, the value of h is determined for these two phases independently. Note that the training data and testing data should be derived from the same domain. If they are derived from two different domains, transferability will also affect the performance of the CDN. In our experiments, we divide the Ottawa data set into two parts: the upper and lower parts. Then, the upper part is used for training and the lower part for testing, and vice versa. The change detection maps of these two parts are stitched together

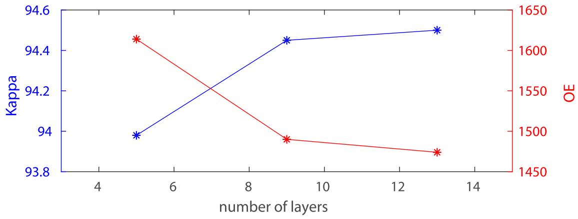


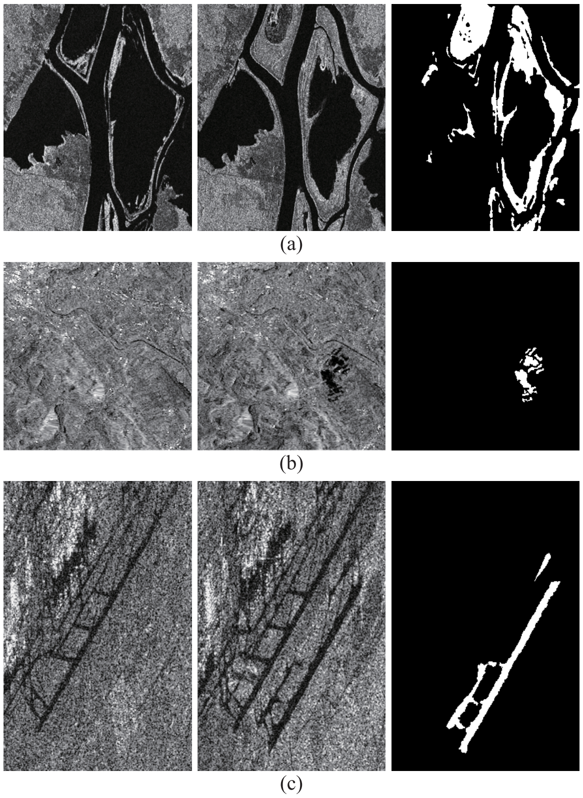
Fig. 6. Comparison of the performance for CDNs with five, nine, and 13 layers.

to get the overall accuracy. Fig. 8 shows the overall accuracy of the CD map when m and h are set to different values. From Fig. 8(a), it can be seen that the CDN performs best when m is set to 28. When h is set to 2, 4, and 8 both in the training phase and the testing phase, the accuracy of the CD maps varies little. Therefore, h is set to 2 in the training phase while being set to 4 in the testing phase to save testing time.

3) Network Architecture and Training: The architectures of the CDNs with three modes in Fig. 3 are then illustrated in detail. A CDN with the early mode has the following layers: C(8,3,1)-P(2,2)-C(16,3,1)-P(2,2)-C(32,3,1)-D(16,3,2)-C(16,3,1)-D(8,3,2)-C(1,3,1). As for the middle mode, the detection network consists of two branches C(8,3,1)-P(2,2)-C(16,3,1)-P(2,2), one for a patch from $\mathbf{X}_1$ and the other for the corresponding patch from $\mathbf{X}_2$, followed by C(32,3,1)-D(16,3,2)-C(16,3,1)-D(8,3,2)-C(1,3,1). When the late mode is used, two branches C(8,3,1)-P(2,2)-C(16,3,1)-P(2,2)-C(32,3,1)-D(16,3,2)-C(16,3,1), each for one of the patch pair, followed by D(8,3,2)-C(1,3,1) are included in the CDN.

The number of layers also has an influence on the overall performance of the CDN. Taking the early mode as an example, the number of layers is chosen according to the experimental results. Similar to choosing m and h, two parts of the Ottawa data serve as training and testing data. Fig. 6 records the overall accuracy of the CD maps that are inferred by the CDNs with five, nine, and 13 layers. The architecture for the CDN with five layers is C(8,3,1)-P(2,2)-C(32,3,1)-D(8,3,2)-C(1,3,1). The architecture for that with nine layers is provided earlier. We can further make the CDN deeper with 13 layers: C(8,3,1)-P(2,2)-C(16,3,1)-P(2,2)-C(32,3,1)-P(2,2)-C(32,3,1)-D(32,3,2)-C(16,3,1)-D(16,3,2)-C(16,3,1)-D(8,3,2)-C(1,3,1). From Fig. 6, we can see that the CDNs with nine and 13 layers get similar Kappa coefficient, and both perform better than that with five layers. In the following, the number of layers for the CDN is set to 9 by taking account of the network parameters and the accuracy.

The proposed RDN is shown in Fig. 2. The common subnetwork shared by the change detection (using the early mode) and the reconstruction tasks has the following layers: C(8,3,1)-P(2,2)-C(16,3,1)-P(2,2)-C(32,3,1)-D(16,3,2)-C(16,3,1). θ_c corresponds to these layers. Two layers D(8,3,2)-C(1,3,1) with parameters θ_s following the common subnetwork are only applicable to the change detection task. Similar to the detection task, there are also two layers D(8,3,2)-C(1,3,1) with parameters θ_t for the reconstruction task only. In the above-shorthand notations, C(n, l, p) denotes a convolutional layer with n filters of spatial size $l \times l$ with stride p, P(l, p) is a max-pooling layer of size $l \times l$ with stride p, and D(n, l, p) denotes a deconvolutional layer with n filters of spatial size $l \times l$ with stride p.


(a)

(b)

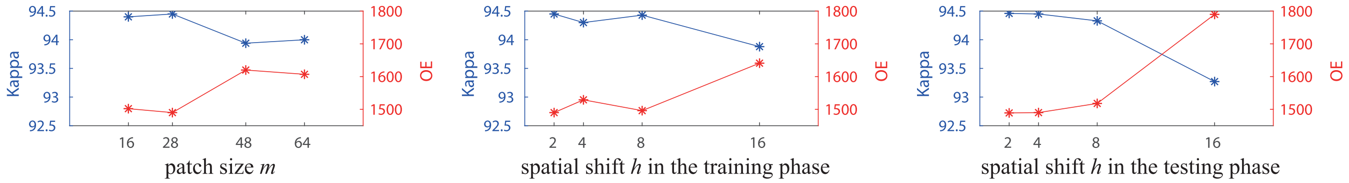
(c)

Fig. 7. Three training data sets used as source domains. (a) Ottawa. (b) Bern. (c) de Gaulle airport. Left and middle: two SAR images acquired at t_1 and t_2. Right: the reference images.

All the networks are trained from scratch with Tensorflow. In the implementation, we use a learning rate of 0.01 with a batch size of 64 for both the source and target domains. We initialize the convolutional and deconvolutional weights with the Xavier initializer [47] and bias with a constant value 0. The parameters, θ_c, θ_s, and θ_t, are updated after each batch. The threshold T_l is set to 0.003. It means that the proposed RDN stops training when the standard deviation σ_l of the average reconstruction loss within the last 10 epoches is less than 0.003. At the fine-tuning stage, the initial CD map is obtained by RFLICM [21]. The reason is that RFLICM is an effective and robust method for change detection in SAR images. Other unsupervised change detection methods can also be alternatives. The threshold for var_1 and var_2 is set to 1.6 for the Yellow River data set and 1.3 for the Mexico data set empirically.

C. Three Modes for Merging Information

When and how to merge the information for a pair of patches have quite an impact on the CDN. There are three modes for merging information, including the early mode, the middle mode, and the late mode. In the middle and late modes, parameters can be shared or unshared between a pair of patches. In all, we have five methods (listed in Table II) to merge the information. In order to evaluate the five methods,



(a) (b) (c)

Fig. 8. Kappa and OE vary with different (a) patch size m, (b) spatial shift h in the training phase, and (c) spatial shift h in the testing phase.

TABLE II

Change Detection Results Using Five Methods to Merge Information for Patch Pairs

Modes	Training	Testing	Measure				
			Kappa	PC	FP	FN	OE
Early	Upper Part	Lower Part	93.82	98.71	436	439	875
	Lower Part	Upper Part	94.92	98.19	190	425	615
	Overall		**94.45**	**98.53**	**626**	**864**	**1490**
Middle Share	Upper Part	Lower Part	93.61	98.69	227	658	885
	Lower Part	Upper Part	95.70	98.45	230	295	525
	Overall		**94.72**	**98.61**	**457**	**953**	**1410**
Middle Unshare	Upper Part	Lower Part	93.08	98.58	254	703	957
	Lower Part	Upper Part	95.95	98.45	216	278	494
	Overall		**94.56**	**98.57**	**470**	**981**	**1451**
Late Share	Upper Part	Lower Part	91.79	98.34	246	878	1124
	Lower Part	Upper Part	94.89	98.18	199	420	619
	Overall		**93.41**	**98.28**	**445**	**1298**	**1743**
Late Unshare	Upper Part	Lower Part	91.50	98.27	291	876	1167
	Lower Part	Upper Part	95.36	98.34	200	364	564
	Overall		**93.47**	**98.29**	**491**	**1240**	**1731**

five CDNs are trained, each with one of the five information merging methods.

The quantitative change detection results for these two parts are shown in Table II. When the upper part is used to train the CDNs, we can see that the network with the early mode achieves the most accurate CD map on the lower part. When the lower part acts as training data and the upper part as testing data, the CDN using the middle mode with unshared parameters performs best. When the two parts of the Ottawa data are stitched together, we find that the CDNs with the early mode, middle share, and middle unshare get similar Kappa coefficients, 94.45%, 94.72%, and 94.56%. However, the networks using two methods in the late mode to merge information perform a little worse. From an empirical point of view, it is not advisable to merge information in the upsampling path of the U-net architecture. We also compare CDNs using shared parameters with that using unshared parameters in both the middle and late modes. After comparison using Kappa coefficients, 94.72% versus 94.56% and 93.41% versus 93.47% from Table II, we can find whether to share parameters makes little difference. The reason may be that we exchange a pair of patches in the input of the CDNs. It makes the extracted feature maps of each branch not be specific to one of two remote sensing images. The parameters of the two branches are similar even though they are not explicitly shared.

Fig. 9 provides the training accuracy for the five methods of merging information. From Fig. 9, it can be seen that the CDNs using the early mode converge fastest. With the late mode, it needs most epoches to converge. Taking both the convergence speed and the overall accuracy into account, we select the early mode to be used in the CDN.

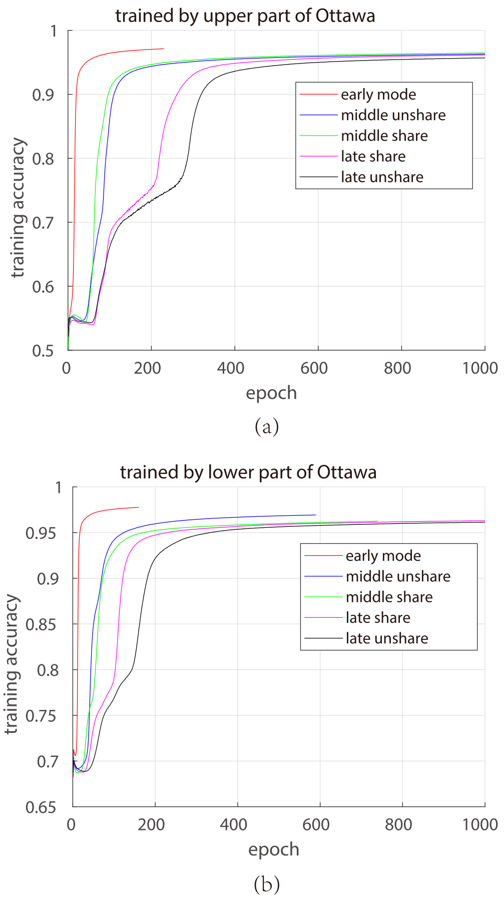
TABLE III

Change Detection Results on the Farmland Data Set by the pt-CDNs Using Different Values of α

Method		Kappa	PC	FP	FN	OE
s-CDN		84.25	98.21	914	680	1594
pt-CDN	α = 0.7	85.85	98.39	831	601	1432
	α = 0.6	**87.15**	**98.54**	**764**	**536**	**1300**
	α = 0.5	85.10	98.26	1041	507	1548
	α = 0.4	84.91	98.23	1080	494	1574

D. α Selection in Joint Pretraining

In the proposed RDN, α is a parameter balancing the importance of the reconstruction and detection tasks. Referring to [43], α ranges from 0.4 to 0.7. s-CDN is thought to be a special case of the RDN with $\alpha = 1.0$. In this experiment, the Ottawa data set is used as the source domain and the Farmland data set as the target domain to evaluate the pt-CDN with different values of α. The evaluation results are shown in Table III. From this Table III, we can see that the pt-CDN with $\alpha = 0.6$ performs best, with the Kappa coefficient of 87.15% and OE of 1300. Table III also



(a)

(b)

Fig. 9. Training accuracy of the CDNs with five methods to merge information for patch pairs. The CDNs are trained by (a) upper and (b) lower parts of the Ottawa data, respectively.

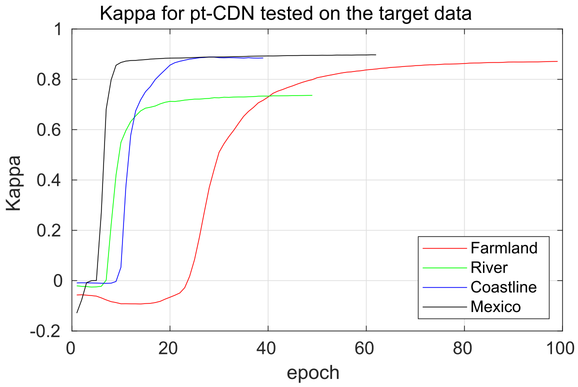
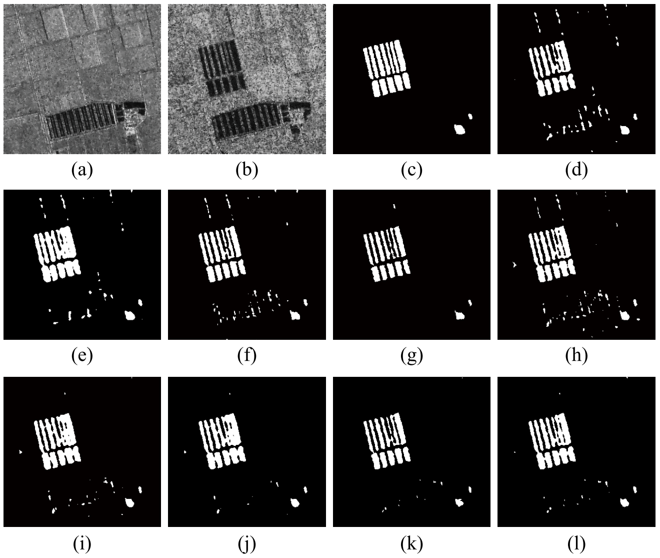


Fig. 10. Kappa coefficient for pt-CDN tested on the target domain.

indicates that FP increases significantly when α is equal to or less than 0.5. In this case, the reconstruction network contributes more to the total loss. Large speckle noise in the Farmland data set makes FP increase significantly. Based on the above-mentioned analysis, α is set to 0.6 in the following experiments. Fig. 10 shows the Kappa coefficient for pt-CDN that is tested on the target data.

E. Fine-Tuning CDN for the Target Domain

In this section, we show the experimental results on the Farmland, River, Coastline, and Mexico data sets. We evaluate the proposed method with both one-source and multisource domains. When there is a one-source domain, the Ottawa data set is used. For multisource domains, the Ottawa, Bern, and de Gaulle airport data sets are all used. The proposed method is also compared to a variety of state-of-the-art change detection methods: GKI [13], PCANet [14], PCA *k*-means [15], and RFLICM [21]. These methods are implemented using the default parameters provided in [13]–[15] and [21], respectively. GKI is a thresholding algorithm. In PCA *k*-means, h and S are set to 5 and 3, respectively. RFLICM is a clustering method and widely used in remote sensing images. For PCANet, we use the publicly available code of [14].



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 11. Farmland data and the comparison results. (a) Farmland data acquired in June, 2008. (b) Farmland data acquired in June, 2009. (c) Reference image. CD maps from (d) GKI, (e) PCA *k*-means, (f) RFLICM, (g) PCANet, (h) CDN with one source, (i) CDN after pretraining with one-source, (j) CDN after pretraining with one source + fine-tuning, (k) CDN with multisource, and (l) CDN after pretraining with multisource + fine-tuning.

TABLE IV
CHANGE DETECTION RESULTS ON THE FARMLAND DATA

Method		Kappa	PC	FP	FN	OE
GKI		80.50	97.81	1017	931	1948
PCA *k*-means		84.61	98.14	1298	356	1654
RFLICM		85.26	98.37	692	761	1453
PCANet		85.53	98.56	52	1232	1284
One source	s-CDN	84.25	98.21	914	680	1594
	pt-CDN	87.15	98.54	764	536	1300
	pt-CDN + ft	87.70	98.64	553	656	1209
Multi-source	s-CDN	87.18	98.69	139	1031	1170
	pt-CDN + ft	**88.83**	**98.78**	**435**	**651**	**1086**

The change detection maps for Farmland, River, and Coastline are shown in Figs. 11–13, respectively. The corresponding quantitative results are listed in Tables IV–VI, respectively. With target information incorporated in the training process, the pt-CDN improves the CD maps. Especially, for the Farmland and Coastline data, both FP and FN decrease. When reliable labels of the target domain are used to fine-tune

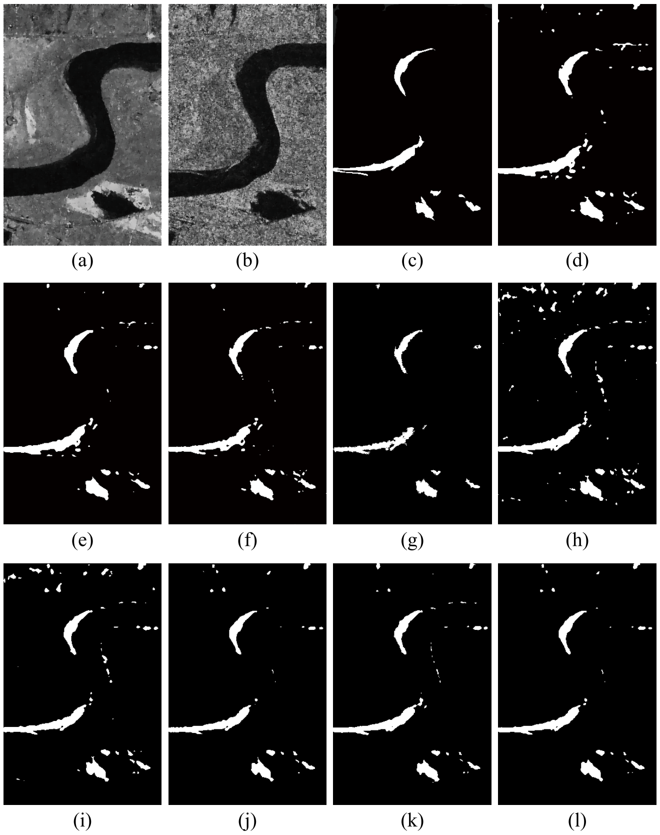


Fig. 12. River data and the comparison results. (a) River data acquired in June, 2008. (b) River data acquired in June, 2009. (c) Reference image. CD maps from (d) GKI, (e) PCA k-means, (f) RFLICM, (g) PCANet, (h) CDN with one source, (i) CDN after pretraining with one-source, (j) CDN after pretraining with one source + fine-tuning, (k) CDN with multisource, and (l) CDN after pretraining with multisource + fine-tuning.

TABLE V

CHANGE DETECTION RESULTS ON THE RIVER DATA

Method		Kappa	PC	FP	FN	OE
GKI		68.74	97.55	2595	569	3164
PCA k-means		73.23	98.04	1918	612	2530
RFLICM		76.01	98.32	1525	646	2171
PCANet		75.23	98.52	683	1229	1912
One source	s-CDN	69.14	97.50	2826	410	3236
	pt-CDN	73.61	98.05	1967	553	2520
	pt-CDN + ft	78.41	98.58	1065	771	1836
Multi-source	s-CDN	77.73	98.48	1313	656	1969
	pt-CDN + ft	**79.40**	**98.63**	**1083**	**688**	**1771**

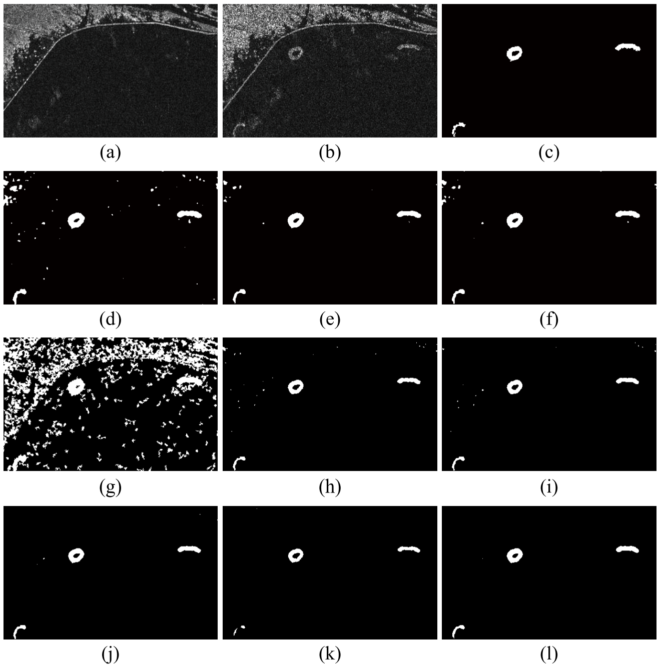


Fig. 13. Coastline data and the comparison results. (a) Coastline data acquired in June, 2008. (b) Coastline data acquired in June, 2009. (c) Reference image. CD maps from (d) GKI, (e) PCA k-means, (f) RFLICM, (g) PCANet, (h) CDN with one source, (i) CDN after pretraining with one source, (j) CDN after pretraining with one source + fine-tuning, (k) CDN with multisource, and (l) CDN after pretraining with multisource + fine-tuning.

the pt-CDN, Kappa coefficients of pt-CDN+ft increase for all the data (especially for River data: from 73.61% to 78.41%). It demonstrates the effectiveness of the region- and boundary-based strategies that are used in selecting reliable labels from an initial CD map.

When multisource is used, the proposed method is also effective in improving the adaptation ability of the CDN (see Tables IV–VI). For the Coastline data, the discrepancy between the target domain and the multisource domains is increased. In this case, the proposed method still improves the CD maps (from 83.45% to 89.99%).

In the existing algorithms, GKI and PCA k-means show poor results, for which FP is always large. RFLICM and PCA-net perform better than GKI and PCA k-means in the Farmland and River data sets. However, PCA-net performs poor in the Coastline data set. In PCA-net, the Gabor wavelet and a coarse-to-fine FCM algorithm are utilized to obtain a preclassification CD map. In this step, a large number of unchanged pixels are misclassified into the changed class for the Coastline data set. As a result, the high FPs (FP = 22 809) are retained in the final change detection map. In summary, our method using three data sets as source domain performs best among these methods. It proves that the knowledge from the source domain is well used by the RDN.

The source data sets of Ottawa, Bern, and de Gaulle airport are all SAR images, while the Mexico data set is captured by an optical sensor. When only one source data (Ottawa) is used to train the CDN, both s-CDN and pt-CDN achieve accurate CD maps [see Fig. 14(h) and (i)]. However, when all the three data sets are used as the source domain, the performance of the s-CDN degrades on the Mexico data set. The reason is that the Ottawa data set has a little speckle noise, and the s-CDN trained by the Ottawa data set is relatively easy to be adapted to the Mexico data set with no speckle noise. However, the data set of de Gaulle airport has large speckle noise. When the CDN is trained using all data set and tested on the Mexico data set, there is a large number of pixels to be misclassified into unchanged class. For s-CDN, FN is larger with multisource (FN = 6093) than that with one source (FN = 3299) (see Table VII). FN decreases (FN = 3261) after pretraining and fine-tuning. From the experimental results, we can conclude

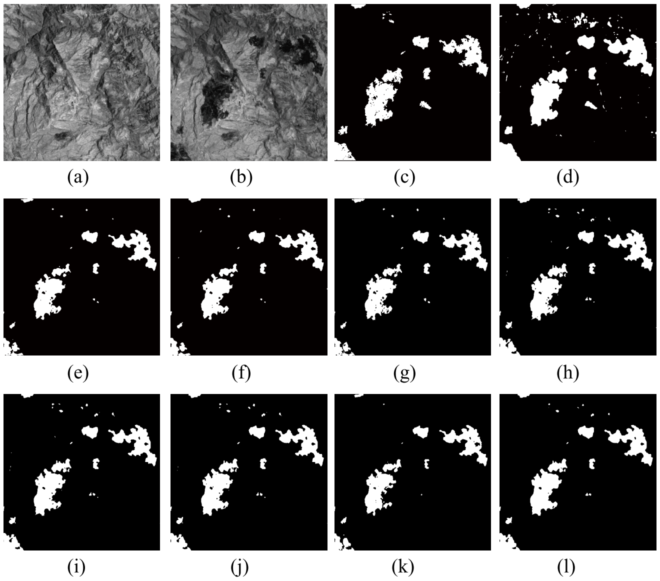


Fig. 14. Mexico data and the comparison results. (a) Mexico data acquired in April, 2000. (b) Mexico data acquired in May, 2002. (c) Reference image. CD maps from (d) GKI, (e) PCA k-means, (f) RFLICM, (g) PCANet, (h) CDN with one source, (i) CDN after pretraining with one-source, (j) CDN after pretraining with one source + fine-tuning, (k) CDN with multisource, and (l) CDN after pretraining with multisource + fine-tuning.

TABLE VI
CHANGE DETECTION RESULTS ON THE COASTLINE DATA

Method		Kappa	PC	FP	FN	OE
GKI		64.58	98.88	1382	31	1413
PCA k-means		84.20	99.63	381	86	467
RFLICM		75.98	99.51	261	357	618
PCANet		8.70	81.90	22809	3	22812
One source	s-CDN	86.20	99.71	180	187	367
	pt-CDN	88.41	99.76	124	179	303
	pt-CDN + ft	**90.55**	**99.80**	**126**	**126**	**252**
Multi-source	s-CDN	83.45	99.70	15	369	384
	pt-CDN + ft	89.99	99.79	133	134	267

TABLE VII
CHANGE DETECTION RESULTS ON THE MEXICO DATA

Method		Kappa	PC	FP	FN	OE
GKI		86.60	97.40	6380	447	6827
PCA k-means		86.47	97.80	684	5084	5768
RFLICM		85.33	97.63	757	5462	6219
PCANet		86.52	97.80	743	5020	5763
One source	s-CDN	89.55	98.22	1363	3299	4662
	pt-CDN	89.74	98.24	1477	3125	4602
	pt-CDN + ft	**89.84**	**98.27**	**1325**	**3214**	**4539**
Multi-source	s-CDN	83.42	97.34	867	6093	6960
	pt-CDN + ft	89.69	98.25	1339	3261	4600

that the proposed method shows superior performance (see Table VII and Fig. 14) by making use of the knowledge from both the source and target domains.

In Table VIII, the running time is listed for all the aforementioned change detection methods (taking the Farmland data set as an example). All the experiments are done with the computer of a four-core CPU Intel i5-3470 @3.2 GHz and 8-GB RAM. GKI, PCA k-means, and PCANet are implemented in MATLAB, RFLICM in C programming language, and the proposed method in Python with a GPU (1050 Ti with 4-GB memory). In multisource domains, the running time is boosted with a large number of training samples. Besides the running time, we also record the parameters used in the aforementioned change detection methods in Table IX.

TABLE VIII
RUNNING TIME (S) OF COMPARED METHODS ON THE FARMLAND DATA

Method		Training	Testing	Total	GPU
GKI		-	1.2	1.2	w/o
PCA k-means		-	153.1	153.1	w/o
RFLICM		-	6.2	6.2	w/o
PCANet		1336.4	503.7	1840.1	w/o
One source	s-CDN	320.1	12.4	332.5	w/
	pt-CDN	374.7	12.4	387.1	w/
	pt-CDN + ft	480.7	15.8	496.5	w/
Multi-source	s-CDN	1660.5	12.3	1672.8	w/
	pt-CDN + ft	2652.0	12.5	2664.5	w/

TABLE IX
PARAMETERS OF THE COMPARED METHODS

Method	Parameters (by learning)	Hyper-Parameters (by setting empirically)
GKI	-	N_{max}
PCA k-means	-	h, S
RFLICM	-	m, ϵ
PCANet	-	k, σ
s-CDN	14k	m, h
pt-CDN	15k	m, h, α, T_l
pt-CDN + ft	15k	m, h, α, T_l, var_1, var_2, n_b, δ

V. CONCLUSION

In this paper, a transferred deep learning-based change detection framework is proposed in remote sensing. It adapts the concept of change that is learned from the source domain to the target domain by reducing the distribution discrepancy between two domains. The proposed method consists of pretraining and fine-tuning stages. Two tasks are included in the pretraining stage: change detection in the source domain with labels and reconstruction in the target domain without labels. In the source domain, a CDN of the U-net architecture is designed to detect changes in a supervised way. The concept of change from the source domain is not suitable well for the target domain. An auxiliary task is proposed to reconstruct the DI for the target domain with a reconstruction network. The lower layers are shared between the two tasks, while the final layers corresponding to each task are trained separately. In addition, we exploit and evaluate three modes of the U-net architecture to merge the information for a pair of patches. After pretraining, reliable labels that are selected from a CD map generated in an unsupervised way are used in fine-tuning the CDN for the target domain. Experiments on two remote sensing data sets confirm that the proposed method is capable to adapt a change detection model from the source domain to the target domain with satisfied performance. In this paper, each source domain is equally important to the target domain in the adaptation. However, the samples from the source domain should contribute more when they are similar to those from the target domain. In the future, we will focus on how to select important samples from multiple source domains to improve the accuracy of the CD map for the target domain.

References

[1] Y. Ban and O. Yousif, “Change detection techniques: A review,” in *Multitemporal Remote Sensing*. Cham, Switzerland: Springer, 2016, pp. 19–43.

[2] L. Castellana, A. D’Addabbo, and G. Pasquariello, “A composed supervised/unsupervised approach to improve change detection from remote sensing,” *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 405–413, 2007.

[3] L. Jiao, J. Zhao, S. Yang, and F. Liu, *Deep Learning, Optimization and Recognition*. Beijing, China: Tsinghua Univ. Press, 2017.

[4] G. Liu, L. Jiao, F. Liu, H. Zhong, and S. Wang, “A new patch based change detector for polarimetric SAR data,” *Pattern Recognit.*, vol. 48, no. 3, pp. 685–695, 2015.

[5] M. Janalipour and M. Taleai, “Building change detection after earthquake using multi-criteria decision analysis based on extracted information from high spatial resolution satellite images,” *Int. J. Remote Sens.*, vol. 38, no. 1, pp. 82–99, 2017.

[6] L. Paolini, F. Grings, J. A. Sobrino, J. C. Jiménez Muñoz, and H. Karszenbaum, “Radiometric correction effects in Landsat multi-date/multi-sensor change detection studies,” *Int. J. Remote Sens.*, vol. 27, no. 4, pp. 685–704, 2006.

[7] G. Chen, K. Zhao, and R. Powers, “Assessment of the image misregistration effects on object-based change detection,” *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 19–27, Jan. 2014.

[8] S. Wang, D. Quan, X. Liang, M. Ning, Y. Guo, and L. Jiao, “A deep learning framework for remote sensing image registration,” *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 148–164, Nov. 2018.

[9] 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.

[10] 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.

[11] J.-S. Lee, “Speckle suppression and analysis for synthetic aperture radar images,” *Opt. Eng.*, vol. 25, no. 5, p. 255636, 1986.

[12] R. Shang, L. Qi, L. Jiao, R. Stolkin, and Y. Li, “Change detection in SAR images by artificial immune multi-objective clustering,” *Eng. Appl. Artif. Intell.*, vol. 31, pp. 53–67, May 2014.

[13] 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.

[14] F. Gao, J. Dong, B. Li, and Q. Xu, “Automatic change detection in synthetic aperture radar images based on PCANet,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1792–1796, Dec. 2016.

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

[16] H.-C. Li, T. Celik, N. Longbotham, and W. J. Emery, “Gabor feature based unsupervised change detection of multitemporal SAR images based on two-level clustering,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2458–2462, Dec. 2015.

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

[18] L. Bruzzone and D. F. Prieto, “Automatic analysis of the difference image for unsupervised change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1171–1182, May 2000.

[19] F. Bovolo, S. Marchesi, and L. Bruzzone, “A framework for automatic and unsupervised detection of multiple changes in multitemporal images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 6, pp. 2196–2212, May 2012.

[20] M. Zanetti, F. Bovolo, and L. Bruzzone, “Rayleigh-Rice mixture parameter estimation via EM algorithm for change detection in multispectral images,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5004–5016, Dec. 2015.

[21] 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.

[22] 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.*, Jul. 2007, pp. 4327–4330.

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

[24] B. Demir, F. Bovolo, and L. Bruzzone, “Detection of land-cover transitions in multitemporal remote sensing images with active-learning-based compound classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 5, pp. 1930–1941, May 2012.

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

[26] M. Hao, W. Shi, K. Deng, H. Zhang, and P. He, “An object-based change detection approach using uncertainty analysis for VHR images,” *J. Sensors*, vol. 2016, Nov. 2016, Art. no. 9078364.

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

[28] S. V. Gadhiraju, H. Sahbi, B. Banerjee, and K. M. Buddhiraju, “Supervised change detection in satellite imagery using super pixels and relevance feedback,” *Geomatica*, vol. 68, no. 1, pp. 5–14, 2014.

[29] S. Ye, D. Chen, and J. Yu, “A targeted change-detection procedure by combining change vector analysis and post-classification approach,” *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 115–124, Apr. 2016.

[30] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, “Evaluating the visualization of what a deep neural network has learned,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 11, pp. 2660–2673, Nov. 2017.

[31] M. Bianchini and F. Scarselli, “On the complexity of neural network classifiers: A comparison between shallow and deep architectures,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1553–1565, Aug. 2014.

[32] S. Lin and G. C. Runger, “GCRNN: Group-constrained convolutional recurrent neural network,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4709–4718, Oct. 2017.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[34] 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.

[35] G. Cao, B. Wang, H. C. Xavier, D. Yang, and J. Southworth, “A new difference image creation method based on deep neural networks for change detection in remote-sensing images,” *Int. J. Remote Sens.*, vol. 38, no. 23, pp. 7161–7175, Dec. 2017.

[36] Y. Li, L. Zhou, G. Lu, B. Hou, and L. Jiao, “Change detection in synthetic aperture radar images based on log-mean operator and stacked auto-encoder,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2017, pp. 3090–3096.

[37] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, “Change detection based on deep Siamese convolutional network for optical aerial images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1845–1849, Oct. 2017.

[38] S. H. Khan, X. He, F. Porikli, and M. Bennamoun, “Forest change detection in incomplete satellite images with deep neural networks,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5407–5423, Sep. 2017.

[39] A. M. E. Amin, Q. Liu, and Y. Wang, “Convolutional neural network features based change detection in satellite images,” in *Proc. Int. Workshop Pattern Recognit.*, 2016, Art. no. 100110W.

[40] M. Long, Y. Cao, J. Wang, and M. Jordan, “Learning transferable features with deep adaptation networks,” in *Proc. Int. Conf. Mach. Learn.*, Jun. 2015, pp. 97–105.

[41] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7167–7176.

[42] Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1489–1496.

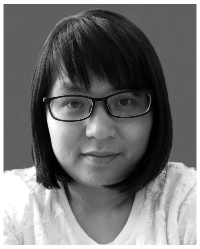
[43] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 597–613.

[44] O. Sener, H. O. Song, A. Saxena, and S. Savarese, "Learning transferrable representations for unsupervised domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2110–2118.

[45] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Assist. Intervent.* Cham, Switzerland: Springer, Oct. 2015, pp. 234–241.

[46] 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.

[47] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, Mar. 2010, pp. 249–256.



Meijuan Yang received the B.S. degree from Xidian University, Xi'an, China, in 2010, and the M.S. degree from the Xi'an Institute of Optics and Precision Mechanics, University of Chinese Academy of Sciences, Xi'an, in 2013. She is currently pursuing the Ph.D. degree with the School of Artificial Intelligence, Xidian University, Xi'an.

Her research interests include deep learning and image processing.

Licheng Jiao (SM'89–F'17) 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 Electronic Engineering, 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 authored over 20 monographs and 100 papers in international journals and conferences. His research interests include image processing, natural computation, machine learning, and intelligent information processing.

Dr. Jiao is also a member 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.

Fang Liu (SM'07) received the B.S. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 1984, and the M.S. degree in computer science and technology from Xidian University, Xi'an, in 1995.

She is currently a Professor with the School of Computer Science, Xidian University. She has authored or coauthored five books and over 80 papers in journals and conferences. Her research interests include signal and image processing, synthetic aperture radar image processing, multiscale geometry analysis, learning theory and algorithms, optimization problems, and data mining.

Biao Hou (M'07) was born in Longxian, China, in 1974. He received the B.S. and M.S. degrees in mathematics from Northwest University, Xi'an, China, in 1996 and 1999, respectively, and the Ph.D. degree in circuits and systems from Xidian University, Xi'an, in 2003.

Since 2003, he has been with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University, where he is currently a Professor. His research interests include dictionary learning, deep learning, and synthetic aperture radar image processing.

Shuyuan Yang (SM'14) received the B.A. degree in electrical engineering and the M.S. and Ph.D. degrees in circuits and systems from Xidian University, Xi'an, China, in 2000, 2003, and 2005, respectively.

She has been a Professor of electrical engineering with Xidian University. Her research interests include machine learning and multiscale geometric analysis.