Pattern Recognition 66 (2017) 213–228

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Deep learning and mapping based ternary change detection for information unbalanced images

Linzhi Su, Maoguo Gong*, Puzhao Zhang, Mingyang Zhang, Jia Liu, Hailun Yang

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an 710071, China

ARTICLE INFO

Keywords:
Change detection
Information unbalanced images
Deep neural networks
Feature representation
Feature mapping

ABSTRACT

This paper mainly introduces a novel deep learning and mapping (DLM) framework oriented to the ternary change detection task for information unbalanced images. Different from the traditional intensity-based methods available, the DLM framework is based on the operation of the features extracted from the two images. Due to the excellent performance of deep learning in information representation and feature learning, two networks are used here. First, the stacked denoising autoencoder is used on two images, serving as a feature extractor. Then after a sample selection process, the stacked mapping network is employed to obtain the mapping functions, establishing the relationship between the features for each class. Finally, a comparison between the features is made and the final ternary map is generated through the clustering of the comparison result. This work is highlighted by two aspects. Firstly, previous works focus on two images with similar properties, whereas the DLM framework is based on two images with quite different properties, which is a usually encountered case. Secondly, the DLM framework is based on the analysis of feature instead of superficial intensity, which avoids the corruptions of unbalanced information to a large extent. Parameter tests on three datasets provide us with the appropriate parameter settings and the corresponding experimental results demonstrate its robustness and effectiveness in terms of accuracy and time complexity.

1. Introduction

The problem of change detection has been treated as a significant issue for decades due to its wide applications in remote sensing [1–6]. In the literatures, it is usually viewed as a process to detect the changes from two images reflecting the same area but taken at different times. For the remote sensing images, these changes usually exhibit as land-cover transitions [7,8], such as the emergence or disappearance of the natural landscape or artificial architectures. This task aims to let us know where it is that the changes have occurred and sometimes what kind of changes have happened. Obviously, for the researchers, the prior knowledge of the region to be studied will largely facilitate the task since it provides much information about the ground truth and thus they are able to get acquainted with some possible changed areas. Based on such prior information, some supervised techniques were developed. However, such information is sometimes quite difficult to obtain (or even impossible to retrieve), so recently, researches have focused more on the unsupervised techniques in which no ground truth is available.

Early literature (such as [1,9,10]) shows out some initial concept and basic methods of change detection. In [1], the authors provided a classical process in which two preprocessed images are compared through an operator to generate a change index (CI) image and then by analyzing the CI image we can get a final change-detection map. This procedure is followed by many developed techniques, and is referred to as the change vector analysis (CVA) in the related follow-up studies [7,8]. In the incipient researches on change detection, the changed area is detected from the unchanged area, a process to generate a CI image and to segment (or classify) it into the unchanged class and the changed class. There is much research on this issue especially on the synthetic aperture radar (SAR) images that are quite independent on weather and light conditions. For example, in [9], Bazi et al. proposed a threshold method for segmenting the log-ratio CI image by using the Kittler–Illingworth (KI) criterion [11] based on the generalized Gaussian model, and then some other flexible models (such as the log-normal, Nakagami-ratio and Weibull-ratio models) were proposed to estimate the class of distribution better [12]. Besides, the expectation maximization (EM) algorithm can also be adopted to establish a threshold iteratively [13]. Researchers also noticed the noise corruption and tried to develop some methods for solving such problem. One of the popular ways is to utilize the spatial information. In fact, a pixel in the image is not isolated but influenced and constrained by the

* Corresponding author.
E-mail address: gong@ieee.org (M. Gong).

http://dx.doi.org/10.1016/j.patcog.2017.01.002
Received 22 June 2016; Received in revised form 22 November 2016; Accepted 2 January 2017
Available online 04 January 2017

others, and there was an awareness that utilizing the spatial information can effectively alleviate noise corruption. Therefore, many improved clustering methods were then proposed. These techniques are mainly based on the improved fuzzy *c*-means (FCM) algorithm, such as the fast generalized FCM (FGFCM) [14], the reformulated FCM by using local information (RFLICM) [15] and the Markov random field based FCM (MRFFCM) [16], in which the spatial information is fully considered.

Recent researches focus more on the discrimination of the changes, a further and higher requirement of the change-detection task. In 2007, Bovolo and Bruzzone proposed a comprehensive framework for multi-change detection based on the CVA technique for multi-spectral image [7]. In the approach, the CI image from the two multi-spectral images is analyzed in the polar domain. The unchanged class and the changed class are discriminated according to the magnitude, whereas the multiple changes are distinguished according to the direction angle. This involves the calculation of the Bayesian estimation, a complicated process that includes establishing the model and computing the posterior probability. In 2012, Bovolo et al. improved the CVA technique into one called compressed CVA (C^2VA), compressing the direction angle from $[0, 2\pi)$ to $[0, \pi]$ by using the vectorial angle [8]. Besides, C^2VA does not involve much about the complicated Bayesian estimation when discriminating different changes but uses the convenient *k*-means (KM) clustering algorithm, largely reducing the time complexity. Based on this framework, several techniques designed for some other complex images are proposed, such as the hyper-spectral (HS) images and very high resolution (VHR) images. Based on CVA, the approaches for the high-spectral (HS) images and very high resolution (VHR) images have been developed. Liu et al. made an analysis of the HS images, and a CVA based hierarchical strategy was proposed and elaborated in [17,18], respectively. Besides, in [19], a context-sensitive technique is designed for the VHR images, robust to registration noise.

In summary, in the literature available, several techniques have been developed to tackle the problem when the two images possess similar properties. However, in reality, there are still several situations that may be encountered and therefore should not be overlooked. Firstly, there may be an update of the imaging system between the two times when the two images were taken respectively. Secondly, the two images are from two different sensors, and thus the subsequent analysis will be based on the multi-sensor images. For instance, one image is from a SAR sensor, whereas the other is from an optical one. Thirdly, some outer conditions, such as weather condition and light condition, may also directly or indirectly influence the imaging effect (especially for the optical remote sensing images). These aspects mainly lead to the discrepancy in brightness, contrast and above all, noise distribution, which purports that the two images possess unbalanced information. We refer to the two images as an information unbalanced (IU) images. Actually, in most cases, hardly can we obtain two images with high similarity at two different times, especially when the second image was taken long after the time at which the first was taken. In the change-detection task for IU images, changed areas are detected and then subdivided, and this process may be affected by the unbalanced information. Such superficial information belonging to the two images is usually influenced by the disparity in noise distribution, brightness or contrast, and thus fails to reflect the nature of ground object features. Besides, the typical intensity-based approaches above conduct the computations that are dependent on complicated and fixed equations, having limited ability to undertake the inner feature extraction and learning. Therefore, inaccurate or unsteady detection may be resulted in when a traditional approach is adopted even in the simple one-dimensional (1-D) imagery. Hence, it is necessary to find out the inner features of the two images if this problem is taken into account and the deep neural networks (DNNs) are considered as good and opportune tools to extract and represent such features. Inspired by the DNN and the related research techniques, we proposed a novel deep learning and mapping (DLM) framework for dealing with the 1-D IU images change-detection task. In the DLM framework, the features of the two images are first extracted through the stacked denoising autoencoder (SDAE) [20]. Then by using a stacked mapping network (SMN), we establish the functions to map the features and three feature channels are generated through comparison. Finally, the final ternary map is generated through the classification of a change feature index consisting of the channels.

This paper is organized into six sections. Section 2 introduces the background knowledge and the related DNNs are detailed in Section 3. Section 4 makes a full introduction to the framework. The experimental data and some settings are shown in Section 5 and the experimental results as well as the corresponding analysis are exhibited in Section 6. Finally, the conclusions are drawn in Section 7.

2. Background knowledge

2.1. CVA techniques and ternary change detection

Consider two coregistered images, $I_1 = \{I_1(i, j), 1 \le i \le A, 1 \le j \le B\}$ and $I_2 = \{I_2(i, j), 1 \le i \le A, 1 \le j \le B\}$, acquired over the same geographical area at two different times, respectively. In the CVA framework, a CI image I is generated through subtraction, ratio or some other appropriate operators (e.g. those introduced in [21–23]) and then two spectral channels I_{s_1} and I_{s_2} are selected from the entire S channels in the CI image. The magnitude ρ and the direction angle ϑ can be calculated as [7]

$$\begin{cases} \rho(i, j) = \sqrt{I_{s_1}^2(i, j) + I_{s_2}^2(i, j)} \\ \tan[\vartheta(i, j)] = \dfrac{I_{s_1}(i, j)}{I_{s_2}(i, j)}. \end{cases} \tag{1}$$

ρ and ϑ serve as two basic elements in the polar coordinate system. The basic idea of CVA contains two points. One is to divide the CI image into the unchanged class (U) and the changed class (C) through a threshold T to ρ. Therefore, in the polar coordinate system, the unchanged area is shown as a circular region and the changed area as an annulus as shown in Fig. 1(a). The other point is to subdivide the changed class into k categories according to ϑ. So multiple changes are exhibited as k annulus sectors, which is shown in Fig. 1(b).

C^2VA is the improved version of CVA proposed in 2012, and it considers all the S spectral channels rather than only two as in CVA. The magnitude ρ and the direction angle α (whose nature is the vectorial angle) are calculated as [8]:

$$\begin{cases} \rho(i, j) = \sqrt{\sum_{r=1}^{S} I_r^2(i, j)} \\ \alpha(i, j) = \arccos\left(\dfrac{1}{\sqrt{S}} \dfrac{\sum_{r=1}^{S} I_r(i, j)}{\sqrt{\sum_{r=1}^{S} I_r^2(i, j)}} \right). \end{cases} \tag{2}$$

According to the property of the function $\arccos(\cdot)$, $\alpha \in [0, \pi]$, and the polar domain is compressed into a semi-circle domain as shown in Fig. 1(c). Now let us consider the case where S=1, and thus there is only one channel in the image (i.e. 1-D image as usually referred to). According to Eq. (2), we have $\rho = I$ and we can also calculate the value of α:

$$\alpha(i, j) = \begin{cases} 0, & \text{if } I(i, j) > 0 \\ \pi, & \text{if } I(i, j) < 0. \end{cases} \tag{3}$$

This special case is shown in Fig. 1(d), and the polar coordinate system degrades to a straight line. Thus, the final change-detection map is ternary. Bazi et al. proposed a KI-based double-threshold approach for the ternary change detection [24] according to the histogram of the CI image, and from the analysis above, it is just the special case in C^2VA technique. Since the framework to be proposed is

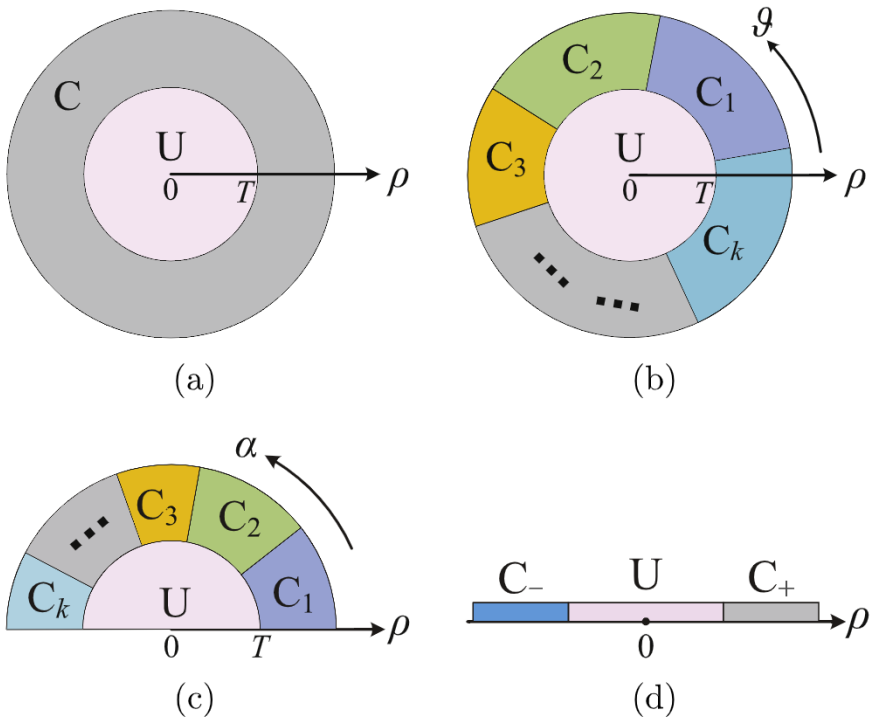

Fig. 1. Illustrations of CVA and C^{2}VA in the polar coordinate system. (a) The division of the unchanged class and changed class through a threshold T to ρ. (b) The subdivision of changed class into k categories according to ϑ. (c) The C^{2}VA method in the semi-circle domain. (d) The special case where there is only one spectral channel and the polar domain degrades to a straight line.

based on the 1-D image, we just consider the ternary task, in which the changed class is subdivided into the positive changed class (C+) and negative changed class (C−). In reality, C+ usually corresponds to the establishment of artificial buildings or the land after (let us say) the receding flood. C− often corresponds to the opposite trends, i.e. the destruction of artificial buildings or the submergence of land by water.

2.2. Change detection for IU images

Suppose the original terrain features of I_1 and I_2 are Y_1 and Y_2, respectively. I_1 and I_2 are obtained through two imaging systems and this process is a mapping process between them. This simplified model can be shown in the following equation:

$$\begin{cases} I_1 = G_1(Y_1) \\ I_2 = G_2(Y_2), \end{cases} \tag{4}$$

where G_1 and G_2 are two stochastic mapping functions from the original feature representation to the pixel-based representation. The establishment of the function depends on the imaging system itself and the environmental factors. So unbalanced information should come from different environments or sensors, and thus I_1 and I_2 differ much in their statistical property.

By "unbalanced information" is meant the information that characterizes two images in different qualitative and/or quantitative level. Such information can be shown as the disparity of several properties, such as the disparity in brightness, contrast and/or noise distribution. To illustrate the concept of IU images further, two main factors are considered. One is the disparity of outer environment. For example, atmosphere and sunlight conditions may affect the intensity of the reflection wave and therefore represent in the images as the disparity of brightness and contrast. The other factor is the disparity noise level or noise type, which is usually due to different sensors. This serves as the dominant factor and in many cases, different noise may also generate different brightness or contrast. Let us consider two kinds of indispensable 1-D images, optical images and SAR images, which are corrupted by the additive Gaussian noise and multiplicative speckle noise, respectively. For the two optical or SAR images, their simplified models can be represented as

$$\begin{cases} I_{\text{op1}} = Y_1 + g_1 \\ I_{\text{op2}} = Y_2 + g_2 \end{cases} \begin{cases} I_{\text{SAR1}} = Y_1 \cdot p_1 \\ I_{\text{SAR2}} = Y_2 \cdot p_2, \end{cases} \tag{5}$$

where g_1 and g_2 are two independently distributed Gaussian noises and p_1 and p_2 are two independently distributed speckle noises (which follow the Rayleigh distribution as proved in [25]), respectively. Due to the property of the noise shown in Eq. (5), traditional approaches adopt subtraction-based and ratio-based operators to the optical images and SAR images to generate the corresponding CI indexes I_{op} and I_{SAR}, respectively:

$$\begin{cases} I_{\text{op}} = I_{\text{op2}} - I_{\text{op1}} = (Y_2 - Y_1) + (g_2 - g_1) \\ I_{\text{SAR}} = \dfrac{I_{\text{SAR2}}}{I_{\text{SAR1}}} = \dfrac{Y_2}{Y_1} \cdot \dfrac{p_2}{p_1}. \end{cases} \tag{6}$$

For the optical images, if g_1 and g_2 are independent identically distributed (i.i.d.), $\mathbb{E}(g_2 - g_1) = 0$ ($\mathbb{E}(\cdot)$ denotes the expectation of a random variable). However, this is a statistical property and does not purport that the noise will be largely reduced for each pixel. If g_1 and g_2 involve quite different parameters, $\mathbb{E}(g_2 - g_1) \neq 0$, they will affect the quality of I_{op}, which also leads to an inaccurate detection in the following classification process. For the SAR images, the situation is more complicated. When they have the same number of looks, p_1 and p_2 are i.i.d. In this case, the distribution of I_{SAR} can be determined as discussed in [9,21], and the noise corruption can be largely reduced by using the ratio-based operators. However, if the two SAR images have different numbers of looks, p_1 and p_2 are no longer i.i.d. and ratio itself is not capable of reducing the noise corruption. Moreover, there is another case where an optical image and a SAR one are involved:

$$\begin{cases} I_{\text{op1}} = Y_1 + g \\ I_{\text{SAR2}} = Y_2 \cdot p. \end{cases} \tag{7}$$

If the subtraction or the ratio operator is used, we have $I_{\text{mix}} = Y_2 \cdot p - Y_1 - g$ or $I_{\text{mix}} = (Y_2 \cdot p)/(Y_1 - g)$, respectively. Apparently,

neither subtraction nor ratio-based operators is suitable.

In summary, a simple subtraction or ratio cannot distinguish each class well due to their limited abilities to tackle such disparity. Besides, as the unbalanced information affects not only the individual pixels but also the whole image, some spatial-based improved operators may also meet difficulty if a well-distinguished CI image is in need. This is mainly to the fact that these techniques are based on the direct use of pixel intensity which may be highly affected by the outer and inner factors. Therefore, to cope with the problem, it is considered to recover the features Y_1 and Y_2 and make a comparison between them. In this way, changes can be detected with much less influence from the unbalanced information.

2.3. Deep learning for change detection

Feature is considered as something that can reflect the important inner nature of an object through its appearance. Recent literatures have focused on the study of the artificial neural networks with some deep architecture structures stacked by shallow layers, or deep neural networks (DNNs) as often referred to. By deep learning is meant the machine learning techniques that use supervised and/or unsupervised strategies to automatically learn hierarchical feature representations in deep architectures. According to the recent works in [26,27], a deep architecture has multiple levels of feature representation, and higher levels usually correspond to more abstract information [28]. Chang analyzed the deep and shallow architecture of multi-layer neural networks in [29], with the mathematical foundation for the deep and shallow architectures set up through 11 theorems. Besides, Szymanski and McCane made a comparative theoretical analysis of the performance of the deep and shallow architectures, demonstrating that it is effective for the deep networks to encode the periodicity [30]. The research investigates the power of neural networks solely due to depth and manifests the distinguished performance of DNNs. Enlightened by the fact that DNN is able to extract inner robust features, many researchers turn to apply it to image processing, such as ship or vehicle detection [28,31], image feature coding [32], image quality assessment [33], image super-resolution [34] and image restoration [35].

Deep networks have also found their applications to change detection. For example, in [36], Gong et al. proposed a novel approach by using the binary restricted Boltzmann machine (RBM) network [36]. Several RBMs are utilized to extract the features from the two images, and such features are passed layer by layer by expanding them into DNNs. Zhang et al. also proposed a novel deep learning based framework incorporating unsupervised feature learning and mapping analysis. The framework involves the denoising autoencoder (DAE) [20] for extracting the deep features and a mapping network to establish the relationship between features [37]. In the two works, the spatial information is utilized flexibly and based on such information, the whole network is optimized through the minimization of the reconstruction error. Due to the combination of the spatial information and deep learning techniques, these approaches are able to cope with some complicated demandings in change detection. Specifically, the technique in [36] largely improves the performance in SAR image change detection, while the framework in [37] is particularly designed for the images with different resolutions. These two works demonstrate the feasibility of using deep learning in change detection. Therefore, for the complicated ternary change detection in IU images, a more detailed framework will be designed based on the SDAE and SMN.

3. Introduction to the SDAE and SMN

Before presenting the framework for IU images, we first make an introduction to the SDAE and SMN, which serve as two key architectures for extracting the features and establishing the mapping function, respectively.

3.1. Autoencoder, DAE and SDAE

Given a normalized N-dimensional input vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)^{\mathrm{T}} \in [0, 1]^N$, an autoencoder generates a hidden layer $\mathbf{y} = (y_1, y_2, \ldots, y_{N'})^{\mathrm{T}} \in [0, 1]^{N'}$ as its feature representation:

$$\mathbf{y} = s(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{8}$$

where $\mathbf{W}$ is the weight matrix and $\mathbf{b}$ is the corresponding bias vector. s is the sigmoid function usually defined as $s(t) = 1/(1 + \mathrm{e}^{-t})$. $\mathbf{y}$ is then mapped back as $\mathbf{x}^{\dagger}$:

$$\mathbf{x}^{\dagger} = s(\mathbf{W}^{\mathrm{T}}\mathbf{y} + \mathbf{c}), \tag{9}$$

where $\mathbf{c}$ is a different bias vector. The reverse weight matrix can optionally be constrained by $\mathbf{W}^{\mathrm{T}}$, and in this case the autoencoder is said to have tied weights [28], which is usually considered. For simplicity, we use θ to denote the parameter set as $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$. By minimizing the error between $\mathbf{x}$ and $\mathbf{x}^{\dagger}$, θ can be determined. This is done through the minimization of the loss function $\mathcal{L}$ (also called the reconstruction error in some literature), and in [20], two kinds of loss functions are given. One is the traditional squared error:

$$\mathcal{L}(\mathbf{x}, \mathbf{x}^{\dagger}) = \|\mathbf{x} - \mathbf{x}^{\dagger}\|^2. \tag{10}$$

The other alternative form is referred to as average reconstruction cross-entropy:

$$\mathcal{L}(\mathbf{x}, \mathbf{x}^{\dagger}) = -\sum_{k=1}^{N} [x_k \ln x_k^{\dagger} + (1 - x_k)\ln(1 - x_k^{\dagger})]. \tag{11}$$

In [20,28], the minimization problem is solved by the stochastic gradient descent algorithm. In fact, more compared optimization algorithms can be found in [38], where the authors have given out a full research and shown the comparison of the methods available.

As a summary, Fig. 2(a) shows a basic autoencoder intuitively. In fact, Vincent et al. have demonstrated that by using corrupted inputs, the structure is able to achieve better learning accuracy [20]. This is the DAE, whose basic idea is illustrated in Fig. 2(b). First, the original input vector $\mathbf{x}$ is corrupted into a noisy version $\tilde{\mathbf{x}}$ by means of a stochastic mapping $\tilde{\mathbf{x}} \sim \mathbf{q}_D(\tilde{\mathbf{x}}|\mathbf{x})$, and then the reconstructed input $\mathbf{x}^{\dagger}$ is generated. The loss function is then established based on $\mathbf{x}$ (rather than $\tilde{\mathbf{x}}$) and $\mathbf{x}^{\dagger}$, and finally the related parameters can be determined.

Suppose we have n input elements, and for each training sample $\mathbf{x}^{(r)}$ is mapped to a corresponding feature $\mathbf{y}^{(r)}$ and a reconstruction $\mathbf{x}^{\dagger(r)}$, $r = 1, 2, \ldots, n$. The objective function of autoencoder is given in [28] as:

$$J_{\mathrm{DAE}} = \frac{1}{n}\sum_{r=1}^{n} \mathcal{L}(\mathbf{x}^{(r)}, \mathbf{x}^{\dagger(r)}). \tag{12}$$

And the steps of DAE can be summarized as follows.

1. Determine $\theta^{(1)}$ according to the first input element $\mathbf{x}^{(1)}$ through the first DAE. r=1.
2. If $r < n$, for the input element $\mathbf{x}^{(r+1)}$, obtain the training result $\theta^{(r+1)}$ by using the DAE with $\theta^{(r)}$ as pre-training initialization and go to Step 3. If r=n, go to Step 4. (n denotes the number of the input vectors.)
3. Let r:=r + 1 and go back to Step 2.
4. Establish the objective function J_{DAE} as Eq. (12) and apply $\theta^{(n)}$ to all the input elements as fine-tuning initialization.
5. Obtain θ^* through the minimization process $\theta^* = \mathrm{argmin}_{J_{\mathrm{DAE}}}$ by using the back-propagation algorithm.

More intuitively, Fig. 3 shows the steps above.

The DAEs comprise the basic block of SDAE, which can learn deep features by stacking several DAEs and thus establishing a deep neural network. In [39] where SDAE is introduced, the process above is viewed as a pre-training stage and a fine-tuning stage is added after it.

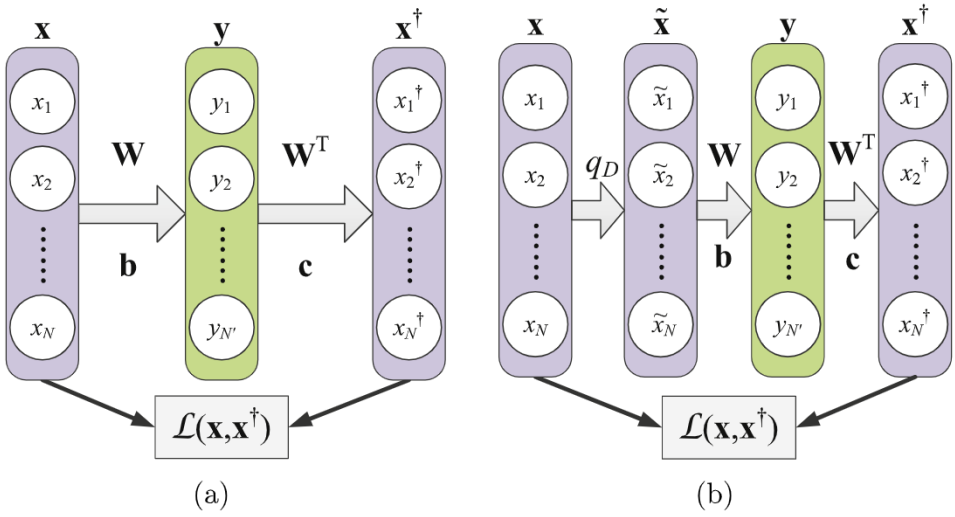

Fig. 2. Illustration of the basic autoencoder (a) and DAE (b).

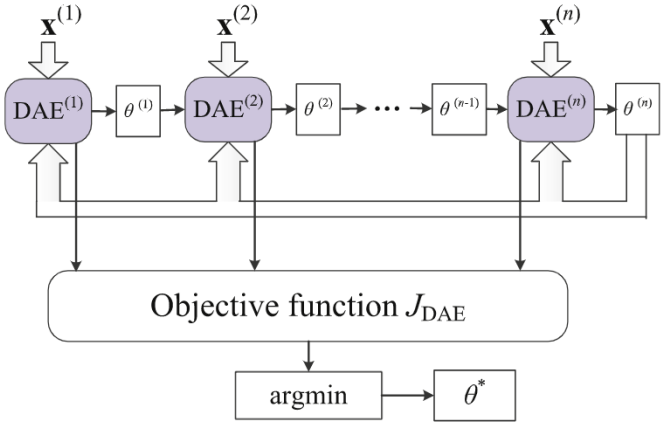

Fig. 3. Illustration of the DAE with respect to all the inputs.

3.2. SMN

In [40], the extracted features from two situations are mapped, and this idea will also be adopted in the DLM framework. Here we make an introduction to the SMN for establishing such a mapping function. Actually, an SMN can be designed with large similarity to SDAE. We slightly modify the autoencoder to propose MN as shown in Fig. 4, where $\mathbf{h}$ is an $M'-$ dimensional hidden layer and the loss function $\mathcal{L}$ can also be defined like Eq. (10) or (11).

The related formulae of MN are given in the following equation:

$$\begin{cases} \mathbf{h} = \mathbf{s}(\mathbf{Q}\mathbf{y}^1 + \mathbf{a})) \\ \mathbf{y}^\dagger = s(\mathbf{Q}^{\mathrm{T}}\mathbf{h} + \mathbf{d}), \end{cases} \tag{13}$$

where $\mathbf{Q}$, $\mathbf{a}$ and $\mathbf{d}$ are the corresponding parameters which can be denoted as $\xi = \{\mathbf{Q}, \mathbf{a}, \mathbf{d}\}$. Then by minimizing J_{SMN} (shown as Eq. (14)), $\xi^* = \mathrm{argmin}_{J_{\mathrm{MN}}}$ can be determined.

$$J_{\mathrm{MN}} = \frac{1}{m}\sum_{r=1}^{m} \mathcal{L}(\mathbf{y}^{2(r)}, \mathbf{y}^{\dagger(r)}), \tag{14}$$

where m is the number of the input vectors. Fig. 5 illustrates the MN with respect to all the inputs.

Similarly, SMN also contains the pre-training stage and the fine-tuning stage, which help to establish the deep network and built a robust mapping function F is established between $\mathbf{y}^1$ and $\mathbf{y}^2$:

$$\mathbf{y}^2 = F(\mathbf{y}^1). \tag{15}$$

In summary, SDAE aims at extracting and outputting the feature from one group of input vectors, whereas SMN aims at generating a mapping function between two groups of vectors. This is their main difference and these two deep structures play a pivotal and indispensable role in the DLM framework.

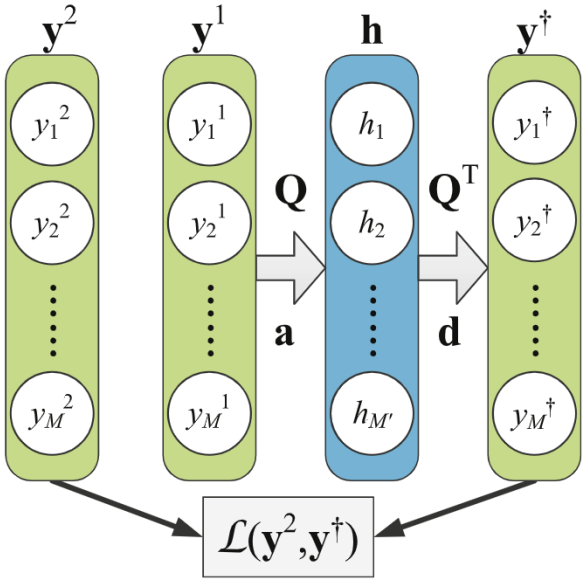

Fig. 4. Illustration of the MN, a similar network to the autoencoder.

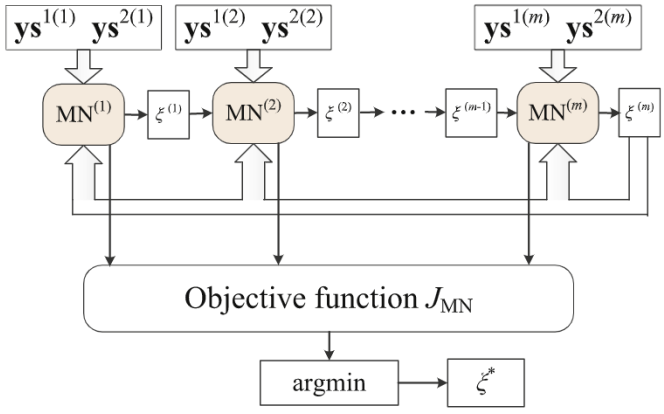

Fig. 5. Illustration of the MN with respect to all the inputs.

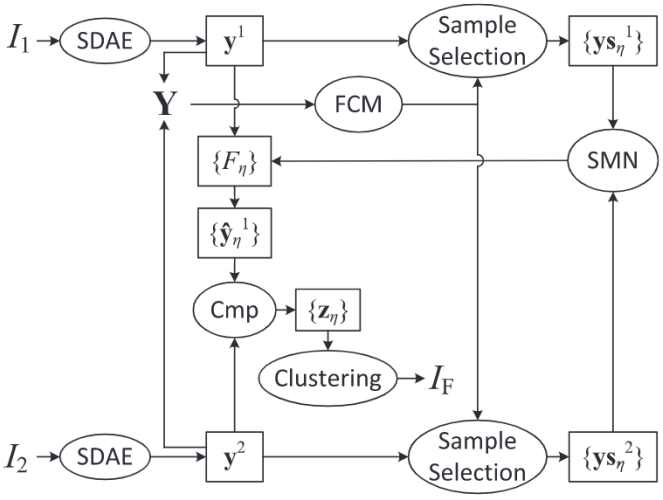

Fig. 6. Procedure of the DLM framework.

4. DLM framework

This section mainly introduces the proposed DLM framework, whose main procedure is summarized as follows.

1. Extract the features from I_1 and I_2 as $\mathbf{y}^1$ and $\mathbf{y}^2$, respectively, by utilizing the spatial information through SDAE.
2. Select the training feature samples $\{\mathbf{ys}_\eta^1\}$ and $\{\mathbf{ys}_\eta^2\}$ from $\mathbf{y}^1$ and $\mathbf{y}^2$, respectively, $\eta \in \{\mathrm{U}, \mathrm{C}+, \mathrm{C}-\}$.
3. Establish the three feature mapping functions $\{F_\eta\}$ between $\{\mathbf{ys}_\eta^1\}$ and $\{\mathbf{ys}_\eta^2\}$ through SMN so that $\mathbf{ys}_\eta^2 = F_\eta(\mathbf{ys}_\eta^1)$.
4. Apply the $\{F_\eta\}$ to $\mathbf{y}^1$ separately and obtain the three mapped features $\{\hat{\mathbf{y}}_\eta^1\}$.
5. Make a comparison between $\mathbf{y}^2$ and $\{\hat{\mathbf{y}}_\eta^1\}$ separately to generate three change feature channels $\{\mathbf{z}_\eta^1\}$ and construct a change feature index $\mathbf{Z}$ by putting them in series.
6. Classify $\mathbf{Z}$ into three classes by using a clustering algorithm to generate a final ternary map I_F.

Fig. 6 illustrates the procedure of DLM framework intuitively. And in what follows, it will be introduced in detail.

4.1. Feature extraction by using SDAE

The process of feature extraction by using SDAE has been introduced in Section 3.1. Here we consider how to design the input vector $\mathbf{x}(i, j)$ for the SDAE.

It is believed that every pixel is not individual but highly related to its neighbors, and i.e. local spatial information should be taken into account appropriately when an image is processed [15,16,41]. So first a neighborhood $\Omega_{i,j}$ is selected and the entire pixels in $\Omega_{i,j}$ are then drawn into a vector, which serves as the corresponding input. Since each output feature representation is an $N'-$D vector, the final entire outputs exhibit as an $N'-$layer feature map. Fig. 7 shows the input and output intuitively.

4.2. Training sample selection

Before establishing of the mapping function, we need to select the training feature samples $\mathbf{ys}$ automatically in advance. This can be implemented by generating a rough feature map $\mathbf{Y}$ by using subtraction as

$$\mathbf{Y}(i, j) = \mathbf{y}^2(i, j) - \mathbf{y}^1(i, j). \tag{16}$$

Then we can assign a label to each pixel through a clustering method, and this can be done by adopting the FCM algorithm. Thus, according to the membership matrices $\{\mathbf{R}_\eta\}$ by FCM ($\eta \in \{\mathrm{U}, \mathrm{C}+, \mathrm{C}-\}$), a rough label assignment result is obtained and each pixel is labeled initially. In the light of the initial label, the corresponding features $\{\mathbf{ys}_\eta^1\}$ and $\{\mathbf{ys}_\eta^2\}$ can be selected. It is worth noting that the aim of generating the rough $\mathbf{Y}$ to get the samples from two images and the subtraction operator, albeit not so accurate, is feasible in the sample selection process. The selection strategy given above is shown as follows:

$$\begin{cases} [\mathbf{R}_\mathrm{U}, \mathbf{R}_{\mathrm{C}+}, \mathbf{R}_{\mathrm{C}-}] = \mathrm{FCM}(\mathbf{Y}) \\ \eta = \underset{\{\mathrm{U},\mathrm{C}+,\mathrm{C}-\}}{\operatorname{argmax}} \{\mathbf{R}_\mathrm{U}(i, j), \mathbf{R}_{\mathrm{C}+}(i, j), \mathbf{R}_{\mathrm{C}-}(i, j)\} \\ \mathbf{ys}_\eta^1(i, j) = \mathbf{y}^1(i, j), \quad \mathbf{ys}_\eta^2(i, j) = \mathbf{y}^2(i, j). \end{cases} \tag{17}$$

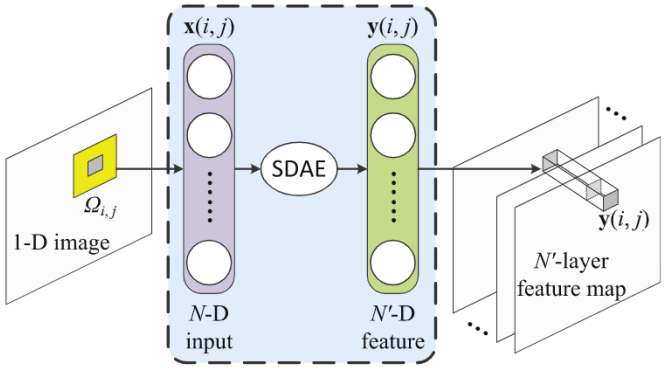

Fig. 7. The pixels in the neighborhood serve as the input of SDAE. The gray part in the 1-D image represents the central pixel (i,j) and the yellow part represents its corresponding neighborhood. The final output is an $N'-$layer feature map. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

4.3. Establishment of mapping functions through SMN

In Section 3.2, the SMN technique has been discussed in detail. After the selection of feature training samples, three mapping functions $\{F_\eta\}$ between $\{\mathbf{ys}_\eta^1\}$ and $\{\mathbf{ys}_\eta^2\}$ can be established separately as shown in Fig. 8, in which m_η denotes the numbers of the training samples belonging to η, $\eta \in \{\mathrm{U}, \mathrm{C}+, \mathrm{C}-\}$.

Hence, we have

$$\mathbf{ys}_\eta^2 = F_\eta(\mathbf{ys}_\eta^1), \quad \eta \in \{\mathrm{U}, \mathrm{C}+, \mathrm{C}-\}. \tag{18}$$

In Fig. 4, the dimension of the input is M, and obviously we have $M = N'$. Besides, since the three functions are generated separately, the three SMNs can be run in parallel to reduce the elapsed time.

4.4. Generation of the change feature channels

In this step, three kinds of estimated features $\{\hat{\mathbf{y}}_\eta^1\}$ are generated according to the three mapping functions obtained from the previous stage:

$$\hat{\mathbf{y}}_\eta^1 = F_\eta(\mathbf{y}^1), \quad \eta \in \{\mathrm{U}, \mathrm{C}+, \mathrm{C}-\}. \tag{19}$$

Let us make a comparison between Eqs. (18) and (19). In Eq. (18), F_η is generated through the training samples $\mathbf{ys}_\eta^1$ and $\mathbf{ys}_\eta^2$, and the functions are then applied to the entire extracted features $\mathbf{y}^1$ and $\mathbf{y}^2$ in Eq. (19). If a pixel $(i, j) \in \eta$, $\hat{\mathbf{y}}_\eta^1(i, j)$ and $\mathbf{y}^2(i, j)$ will be of little difference because the mapping function F_η is obtained by training on the samples belonging to η, which makes $\hat{\mathbf{y}}_\eta^1(i, j)$ approximately equal to $\mathbf{y}^2(i, j)$. In fact, F_η is based on the inputs mostly belonging to η, and for those not belonging to η, there is not such a mapping relationship between $\hat{\mathbf{y}}_\eta^1$ and $\mathbf{y}^2$, i.e. they are of large difference.

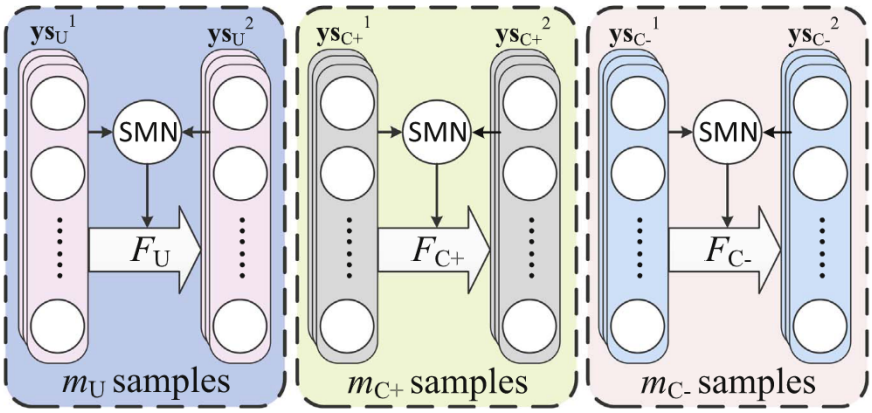

Fig. 8. Three mapping functions generated by using three SMNs separately.

Thus, the proximity of $\hat{\mathbf{y}}_{\eta}^{1}(i,j)$ to $\mathbf{y}^{2}(i,j)$ can be measured as:

$$\mathbf{z}_{\eta}(i,j) = |\mathbf{y}^{2}(i,j) - \hat{\mathbf{y}}_{\eta}^{1}(i,j)|, \tag{20}$$

where the function $|\cdot|$ denotes the absolute value of each dimensional component (rather than the vector norm). According to the analysis above, if $(i,j) \in \eta$, $\mathbf{z}_{\eta}(i,j)$ contains some small component values; if $(i,j) \notin \eta$, $\mathbf{z}_{\eta}(i,j)$ contains some large component values. So $\mathbf{z}_{\eta}$ itself can be viewed as a change feature channel that is able to highlight the area containing the pixels belonging to η, and there are $A \times B$ $N'-$D elements (A and B denote the length and the width of the image) in each channel. Fig. 9(a) shows the process to generate the change feature channels clearly. It is worth noting that Fig. 9(a) is a schematic diagram and that a channel is not arranged like that but exhibited as an $N'-$layer map (shown in Fig. 9(b)).

4.5. Classification of the change feature index

The final step is to classify a change feature index into three categories. Having acquired the three change feature channels, we put the three channels in series to get a $3N'-$layer change feature index $\mathbf{Z}$:

$$\mathbf{Z} = [\mathbf{z}_{\mathrm{U}}; \mathbf{z}_{\mathrm{C+}}; \mathbf{z}_{\mathrm{C-}}]. \tag{21}$$

And then a clustering method can be adopted to get a ternary map I_{F}. This process is shown in Fig. 9(c).

Here a discussion is made on the final clustering method. As is known to all, being capable of retaining more information from the data, FCM has robust characteristics for ambiguity due to the use of fuzzy strategy [16]. However, when it comes to the very large image data (or big data as is often referred to), i.e. the input images are in quite huge size in both pixel number and dimension, it is necessary to speed up the algorithm and at the same time to guarantee the accuracy. In [42], the authors have undertaken a deep research on the FCM for very large data and made a summary of several improved FCM algorithms specially designed to cope with very large data, and four main techniques are adopted to improve the basic FCM approach: sampling, weighting, blocking and kernel inducing. These improved methods includes the weighted FCM (wFCM), single-pass FCM (spFCM, first appearing in [43]) and online-FCM (oFCM, first appearing in [44]), etc. The authors also proposed the kernel-based versions such as wkFCM, spkFCM and okFCM, which classify the data in the high-dimensional reproducing kernel Hilbert space. According to the property of data and the hardware condition available, we can choose an appropriate method as recommended at the end of [42].

5. Dataset descriptions and experimental settings

In this section, the descriptions to three datasets and the experimental settings will be given. It also includes an introduction to the involved evaluation criteria.

5.1. Datasets

This subsection gives an introduction to the three datasets, each of which contains two remote sensing images and a ground truth image used as a reference map. In the ground truth image, the gray, white and black parts denote the U, C+ and C− classes, respectively.

The first dataset is an optical one called the Chanba dataset because it reflect the changes happening at the Chanba Ecological District in the Xi'an City (shown in Figs. 10(a) and (b)). During the three years from June 2002 to May 2005, some of the aquaculture regions vanished and some temporal reservoirs were built to facilitate the urban construction. Both images are in the size 500×385, and there is an update of image device between the two times, so they exhibit mainly as different levels of Gaussian noise. It can be seen that they also differ a little in brightness and contrast. The ground truth image Fig. 10(c) is generated through the on-the-spot investigation.

The second dataset, the farmland dataset, consists of two images (921×484) from a SAR sensor in June 2008 and an optical sensor in September 2012, respectively. The changes occurring in Shuguang Village in the Dongying City is shown in Figs. 11(a) and (b). During the time between 2008 and 2012, some houses were built on one of the farmland regions and a new aqueduct was dug. Since the two images are from different sensors, they are quite different in the noise distribution. The one in 2008 is corrupted by the multiplicative speckle noise and the one in 2012 is mainly affected by the additive Gaussian noise. The reference map is generated through artificial marking by a combination of expert knowledge and surface prior information.

The third dataset consists of a pair of SAR images shown in Fig. 12, which reflects the changes about the estuary area of the Yellow River, and hence the name the Yellow River (YR for short) dataset. Figs. 12(a) and (b) show the images taken in June 2008 and June 2009, respectively, and their size is 290×260. It is worth noting that they are a four-look image and a single-look image, which means that the speckle noise affects the image acquired in 2009 much greater than the one acquired in 2008. Such noise level disparity leads to the inconsonant manifestation of both the homogeneous and heterogeneous regions, aggravating the difficulty met in the task. The reference map is generated through artificial marking by a combination of expert knowledge and on-the-spot investigation.

5.2. Experimental settings

Three main experiments will be carried out on the three datasets. The first one is a test concerning the size of the input vector $\mathbf{x}$ (N), which depends on the size of the neighborhood as described in Section 4.1. Here we choose three kinds of square neighborhoods with their sizes 3×3, 5×5 and 7×7, and thus N ranges among 9, 25 and 49. We will test its impact on both accuracy and elapsed time. It is worth noting that since the input of SDAE should be a normalized one [20], so

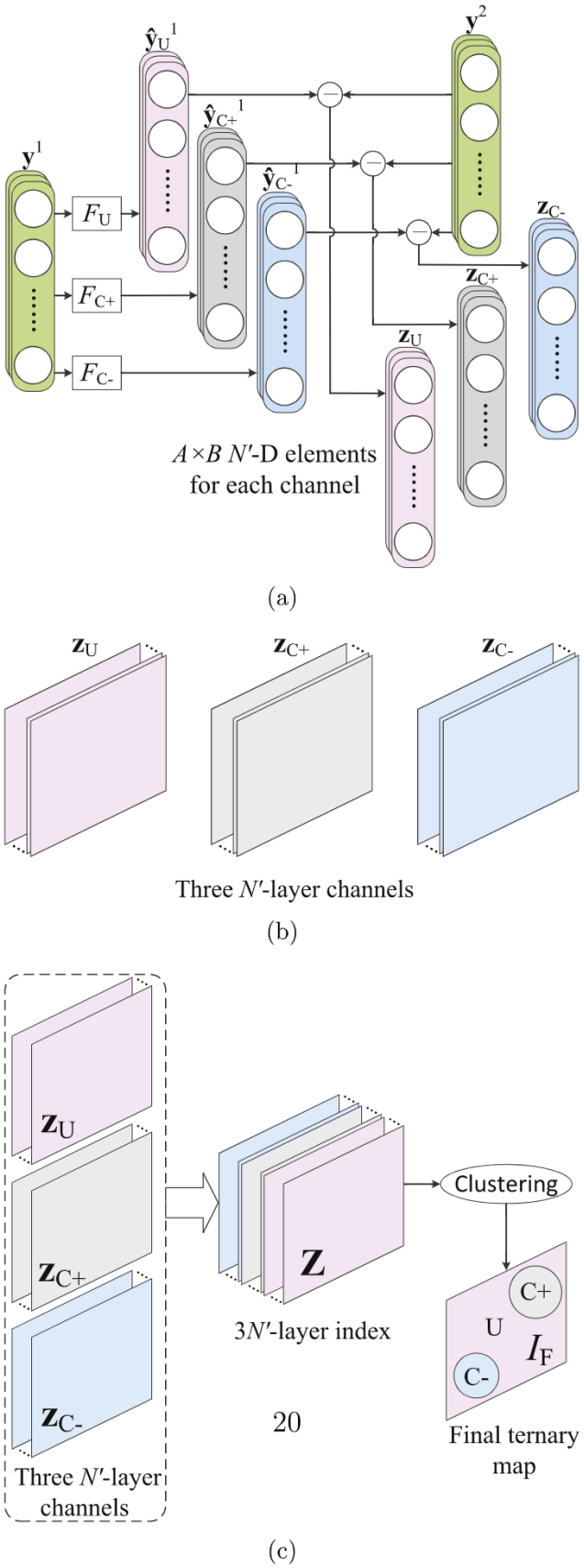

Fig. 9. Generation of the change feature channels and the final ternary map. (a) shows the process of the generation; (b) is the actual exhibition of the channels; and (c) $3N'$– D change feature index and the generation of the final ternary map.

it is essential to normalize the intensities of the entire pixels in the image at the beginning.

In the second experiment, another parameter test will be carried out with respect to the feature size N' and the size M' of the hidden layer $\mathbf{h}$ in DLM. Both N' and M' are set ranging from 10 to 100 at intervals of 10. In addition to accuracy, the elapsed time will also be taken into much consideration. It is well known that the time complexity will be high when the there are many neurons in a layer, so it is necessary to find a good setting of parameters that will engender both satisfactory accuracy and acceptable time complexity.

The third experiment will show the comparison of the results by the DLM framework and three other techniques available: FCM, KI [24] and C^{2}VA [8]. We will employ the parameters that the first two experiments have selected in the DLM framework. In the comparison methods, we use the popular log-ratio operator to generate the CI image. It is worth noting that the C^{2}VA technique is based on multi-spectral images and that for the 1-D images, the calculation will be largely simplified.

5.3. Evaluation criteria

The evaluation criteria in ternary change detection include the statistical classification matrix $\mathbf{P}$, the percentage correct classification (*PCC*), the Kappa coefficient (*KC*) and the F_1-score (F_1).

First, a statistical table is made to show the classification result in percentage (Table 1).

$\mathbf{P}$ is just the data in Table 1:

$$\mathbf{P} = \begin{bmatrix} P_{UU} & P_{UC+} & P_{UC-} \\ P_{C+U} & P_{C+C+} & P_{C+C-} \\ P_{C-U} & P_{C-C+} & P_{C-C-} \end{bmatrix} \tag{22}$$

PCC, which reflects the overall accuracy roughly, is calculated as:

$$PCC = P_{UU} + P_{C+C+} + P_{C-C-} = \mathrm{tr}(\mathbf{P}). \tag{23}$$

KC is a criterion that involves more detailed classification information and is calculated through matrix operations:

$$KC = \frac{\mathrm{tr}(\mathbf{P}) - \mathbf{q}^{\mathrm{T}}\mathbf{P}^2\mathbf{q}}{1 - \mathbf{q}^{\mathrm{T}}\mathbf{P}^2\mathbf{q}}, \tag{24}$$

where $\mathbf{q} = [1, 1, 1]^{\mathrm{T}}$. For F_1, the calculation is as follows:

$$F_1(\eta) = \frac{2\mathbf{e}_\eta^{\mathrm{T}}\mathbf{P}\mathbf{e}_\eta}{\mathbf{e}_\eta^{\mathrm{T}}\mathbf{P}\mathbf{q} + \mathbf{q}^{\mathrm{T}}\mathbf{P}\mathbf{e}_\eta}, \quad \eta \in \{\mathrm{U}, \mathrm{C}+, \mathrm{C}-\}, \tag{25}$$

where $\mathbf{e}_U = [1, 0, 0]^{\mathrm{T}}$, $\mathbf{e}_{C+} = [0, 1, 0]^{\mathrm{T}}$ and $\mathbf{e}_{C-} = [0, 0, 1]^{\mathrm{T}}$. It is worth noting that F_1 is used to show the classification performance to a certain class rather than the overall performance like *PCC* and *KC*, so we have to calculate three F_1 values as $F_1(\mathrm{U})$, $F_1(\mathrm{C}+)$ and $F_1(\mathrm{C}-)$. For the detailed derivations of Eqs. (24) and (25), refer to Appendix.

6. Experimental results

6.1. Test of the parameter N

This subsection is about the test of N, which stands for the size of the input. Two aspects are involved here: its impact on *KC* (which is cogent in accuracy) and the elapsed time. The experimental results are based on three cases where $N = 9$ (3×3), 25 (5×5) and 49 (7×7). As for the other parameters, (N', M') is set as (20, 60), (50, 80), (60, 90) and (80, 100), four typical groups of values.

The impact of N on *KC* is shown in Fig. 13.

For the three datasets, the input size 3×3 leads to the highest accuracy. In fact, the use of neighbors as the input is based on the hypnosis that neighborhood pixels share the same property of the central pixel. This is thought to be true when a small neighborhood is considered. However, when the neighborhood is quite large (say 7×7),

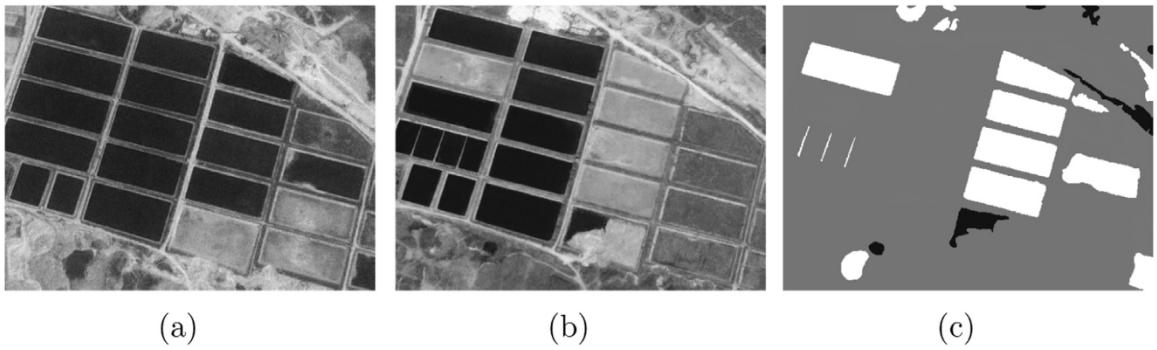

Fig. 10. Chanba dataset. (a) The image taken on June 2002. (b) The image taken on May 2005. (c) The reference ternary map.

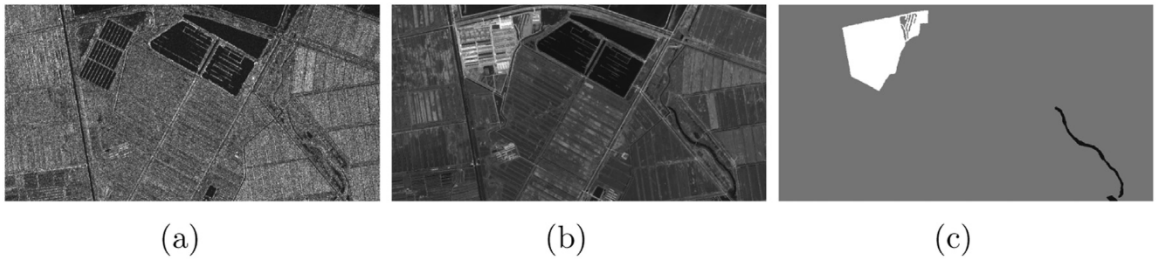

Fig. 11. Farmland dataset. (a) The image taken on June 2008. (b) The image taken on September 2012. (c) The reference ternary map.

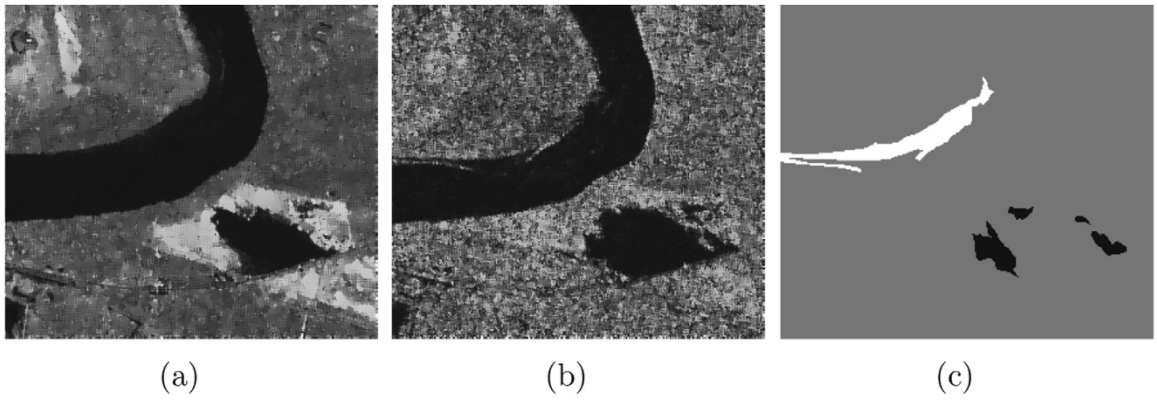

Fig. 12. Farmland dataset. (a) The image taken on June 2008. (b) The image taken on June 2009. (c) The reference ternary map.

Table 1
A general statistical table to show the classification result.

Classes		Estimated class		
		U	C+	C−
True class	U	P_{UU}	P_{UC+}	P_{UC-}
	C+	P_{C+U}	P_{C+C+}	P_{C+C-}
	C−	P_{C-U}	P_{C-C+}	P_{C-C-}

the relevance between a far neighbor and the central pixel is quite low, which is especially obvious in the heterogeneous edge area. Therefore, too large a size of N is not capable of representing the property of the central pixel and thus will not engender the satisfactory accuracy. For the Chanba dataset, the 3×3 and 5×5 neighbors lead to approximate accuracy because of its apparent region changes and clear edges. However, when it comes to the farmland and the YR datasets where more complicated ground information is shown, there is an obvious influence on the corresponding accuracy. In addition, according to

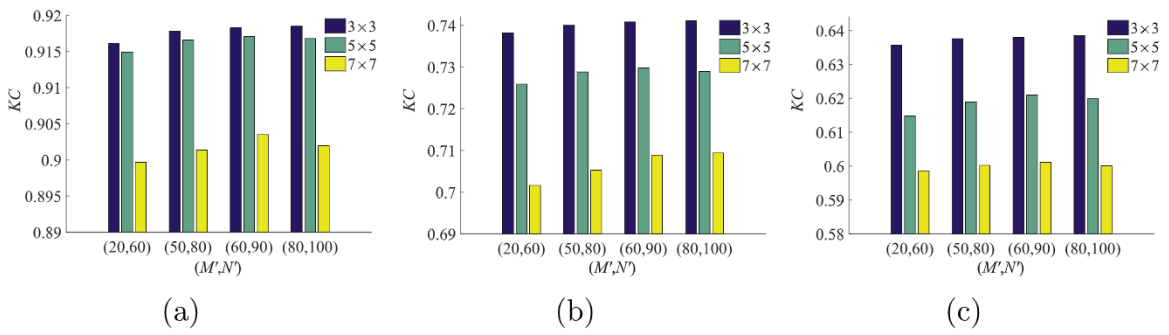

Fig. 13. Impact of N on KC. (a) The Chanba dataset. (b) The farmland dataset. (c) The YR dataset.

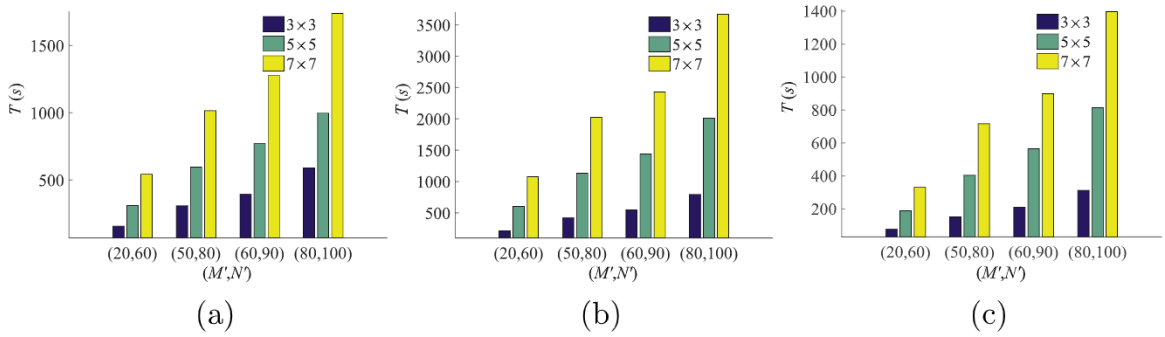

Fig. 14. Impact of N on the elapsed time (in seconds). (a) The Chanba dataset. (b) The farmland dataset. (c) The YR dataset.

Table 2
Impact of N' and M' on KC.

N'	M'									
	10	20	30	40	50	60	70	80	90	100
Chanba dataset										
10	0.9045	0.9070	0.9084	0.9101	0.9108	0.9114	0.9118	0.9120	0.9122	0.9125
20	0.9115	0.9126	0.9134	0.9143	0.9152	0.9161	0.9164	0.9165	0.9165	0.9166
30	0.9129	0.9138	0.9144	0.9150	0.9161	0.9169	0.9172	0.9173	0.9174	0.9176
40	0.9134	0.9145	0.9150	0.9156	0.9166	0.9173	0.9177	0.9176	0.9178	0.9180
50	0.9142	0.9151	0.9157	0.9161	0.9170	0.9175	0.9179	0.9179	0.9180	0.9182
60	0.9148	0.9157	0.9160	0.9168	0.9172	0.9176	0.9179	0.9181	0.9183	0.9183
70	0.9150	0.9158	0.9162	0.9170	0.9173	0.9176	0.9179	0.9182	0.9184	0.9183
80	0.9151	0.9159	0.9163	0.9170	0.9174	0.9177	0.9180	0.9182	0.9184	0.9185
90	0.9151	0.9159	0.9163	0.9171	0.9174	0.9177	0.9180	0.9182	0.9185	0.9186
100	0.9152	0.9160	0.9164	0.9172	0.9175	0.9178	0.9180	0.9183	0.9185	0.9186
Farmland dataset										
10	0.7302	0.7318	0.7320	0.7324	0.7331	0.7335	0.7340	0.7346	0.7351	0.7354
20	0.7341	0.7360	0.7363	0.7369	0.7375	0.7379	0.7382	0.7385	0.7390	0.7392
30	0.7355	0.7364	0.7369	0.7376	0.7379	0.7382	0.7389	0.7390	0.7392	0.7399
40	0.7364	0.7371	0.7374	0.7380	0.7383	0.7388	0.7395	0.7396	0.7400	0.7403
50	0.7368	0.7378	0.7381	0.7385	0.7389	0.7393	0.7398	0.7400	0.7405	0.7407
60	0.7369	0.7379	0.7383	0.7389	0.7392	0.7397	0.7401	0.7403	0.7408	0.7409
70	0.7370	0.7379	0.7384	0.7392	0.7396	0.7399	0.7402	0.7404	0.7410	0.7410
80	0.7370	0.7381	0.7385	0.7394	0.7397	0.7401	0.7402	0.7405	0.7410	0.7411
90	0.7370	0.7381	0.7385	0.7395	0.7397	0.7401	0.7402	0.7405	0.7411	0.7411
100	0.7372	0.7381	0.7386	0.7395	0.7399	0.7402	0.7403	0.7406	0.7411	0.7413
YR dataset										
10	0.6104	0.6249	0.6270	0.6283	0.6297	0.6304	0.6306	0.6308	0.6309	0.6310
20	0.6274	0.6298	0.6329	0.6337	0.6351	0.6357	0.6357	0.6357	0.6359	0.6363
30	0.6297	0.6315	0.6338	0.6343	0.6359	0.6363	0.6364	0.6368	0.6368	0.6374
40	0.6319	0.6334	0.6342	0.6348	0.6360	0.6366	0.6370	0.6370	0.6373	0.6380
50	0.6327	0.6339	0.6348	0.6352	0.6364	0.6370	0.6372	0.6375	0.6376	0.6382
60	0.6330	0.6345	0.6352	0.6354	0.6366	0.6371	0.6375	0.6377	0.6380	0.6383
70	0.6332	0.6348	0.6353	0.6355	0.6367	0.6371	0.6376	0.6378	0.6382	0.6384
80	0.6338	0.6349	0.6355	0.6355	0.6368	0.6371	0.6377	0.6378	0.6383	0.6385
90	0.6339	0.6350	0.6355	0.6356	0.6369	0.6373	0.6377	0.6378	0.6383	0.6385
100	0.6339	0.6351	0.6355	0.6357	0.6370	0.6374	0.6377	0.6380	0.6384	0.6386

Fig. 13, the 7×7 neighbor engenders a worse performance than the other two smaller neighborhoods, and it can be deduced that a smaller neighborhood corresponds to a more accurate result.

Fig. 14 shows the effect of N on elapsed time (in seconds).

As anticipated, a larger neighborhood requires more time than a smaller neighborhood. A combination of Figs. 13 and 14 will lead to the conclusion that the 3×3 neighborhood is the best in both accuracy and time complexity, and hence, the following experiments are all based on the 3×3 neighbor (i.e. N=9).

6.2. Test of the parameters N' and M'

The second experiment aims at testing the influence of the hidden layer size on accuracy and time. Table 2 shows the impact of the other two parameters N' and M' on KC when N=9. Clearly, the overall tendency indicates that a large size of the hidden layer will generate high accuracy. In fact, the larger N' is, the more detailed information to represent the feature there will be. And the larger M' is, the more elaborate calculations will be involved and thus the more accurate mapping function will be established. When N' and M' range among large values, the representation will be elaborate enough, and this is also shown in Table 2 where when $N' > 20$ or $M' > 60$, KC does not increase much.

The impact of N' and M' on elapsed time is shown in Table 3, where much more time is required when they are in quite large values. An integrating analysis of Tables 2 and 3 indicates that it is not appropriate to set them as too large values because the accuracy varies little while the corresponding time increases much.

Table 3
Impact of N' and M' on elapsed time (in seconds).

N'	M'									
	10	20	30	40	50	60	70	80	90	100
Chanba dataset										
10	62.7	68.5	75.6	106.8	132.1	140.9	167.3	212.6	242.8	274.8
20	79.5	84.2	90.9	116.5	113.7	157.2	185.9	239.1	259.2	306.0
30	80.1	91.0	118.0	122.5	162.0	188.8	231.2	249.7	295.5	355.7
40	89.8	100.8	102.6	130.5	165.0	195.7	244.0	294.1	310.1	386.3
50	97.5	111.6	134.9	161.6	197.0	232.0	255.9	308.6	354.7	431.6
60	111.4	131.7	149.3	162.4	213.7	239.5	295.3	328.5	394.0	483.4
70	119.8	144.0	187.7	184.8	228.1	281.5	317.0	356.4	451.6	528.6
80	159.7	163.1	182.9	229.9	246.9	296.4	363.0	416.6	495.4	590.2
90	134.7	190.5	198.7	236.3	273.3	336.6	409.2	473.3	538.5	655.9
100	143.9	203.9	228.5	265.5	312.2	381.3	422.5	520.6	610.5	732.2
Farmland dataset										
10	91.5	113.0	131.2	146.0	170.2	191.3	221.7	251.8	291.9	337.9
20	111.6	125.0	140.8	162.2	190.5	215.8	249.1	288.5	329.6	378.9
30	120.3	136.1	162.2	181.2	213.7	247.6	282.2	327.1	371.4	425.5
40	144.8	152.0	182.1	210.7	247.9	284.3	316.4	365.6	417.5	487.2
50	149.9	179.2	207.5	238.7	275.6	313.9	359.5	419.7	480.5	547.0
60	178.2	206.8	228.1	272.1	309.0	352.7	404.9	465.7	547.1	616.5
70	199.9	230.4	268.9	301.1	350.4	401.3	460.8	530.5	618.1	710.2
80	223.6	259.9	300.3	345.9	395.3	452.8	519.2	596.3	689.4	793.8
90	256.4	295.6	343.6	388.4	446.9	512.3	588.2	677.8	780.6	899.0
100	286.0	327.5	385.6	436.3	505.8	581.5	666.2	763.2	880.8	998.1
YR dataset										
10	30.6	36.0	40.8	53.4	59.7	73.4	84.9	94.4	109.8	132.7
20	36.3	42.8	48.0	57.7	68.0	76.6	94.3	109.0	131.0	149.1
30	40.5	42.5	52.6	61.7	74.6	89.5	104.4	120.6	143.8	170.2
40	42.6	51.1	60.1	72.2	87.8	100.7	117.0	135.2	165.2	190.3
50	48.7	57.8	71.3	79.6	96.2	115.1	129.0	152.3	184.9	217.5
60	56.7	60.5	75.3	94.8	106.9	127.5	152.9	175.3	209.6	246.4
70	62.7	76.2	85.8	105.9	121.8	138.0	168.8	197.7	232.5	277.5
80	72.9	88.9	98.1	117.1	139.0	159.8	187.4	225.7	267.8	312.8
90	78.4	93.6	110.8	132.2	153.8	183.5	216.2	251.1	302.7	351.2
100	91.1	106.9	128.4	147.4	174.6	206.7	243.5	286.7	334.6	399.8

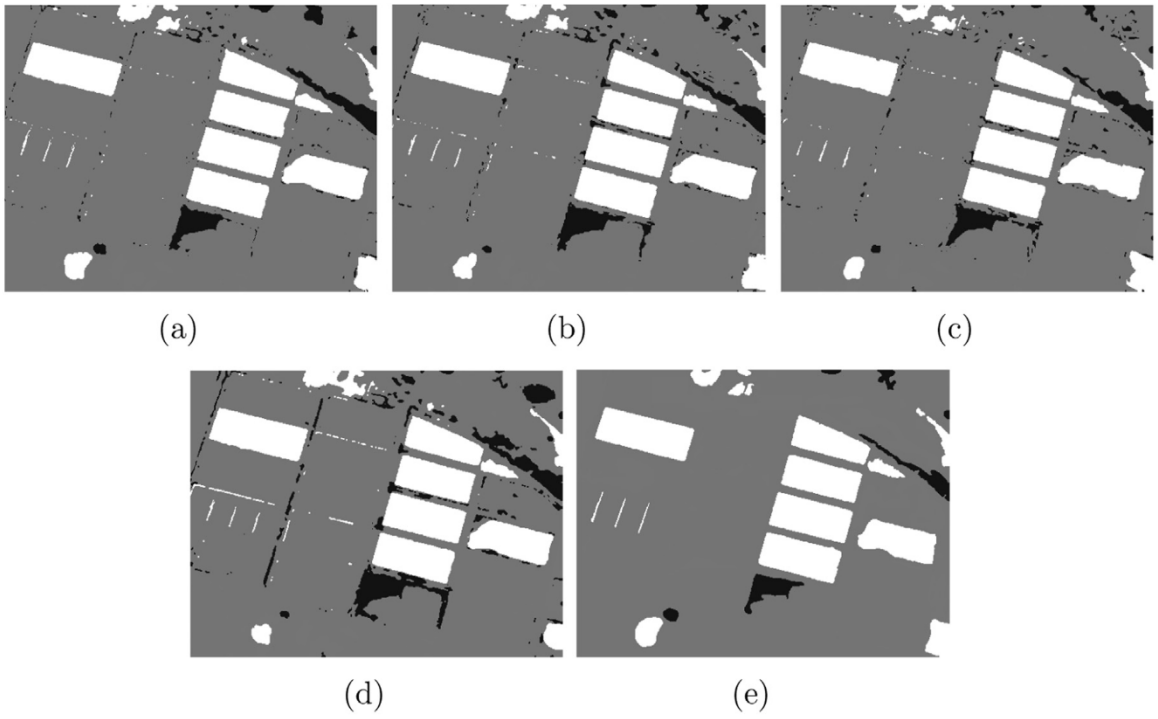

Fig. 15. Ternary maps from the Chanba dataset by (a) DLM, (b) C^2VA, (c) FCM, and (d) KI. (e) is the reference map.

Table 4
Values of the evaluation criteria on the Chanba dataset.

DLM		Estimated class			F_1
		U	C+	C−	
True class	U	0.7928	0.0045	0.0014	**0.9833**
	C+	0.0072	0.1616	0	**0.9651**
	C−	0.0139	0.0001	0.0186	**0.7086**
PCC		**0.9730**			
KC		**0.9161**			
C^2VA		Estimated class			F_1
		U	C+	C−	
True class	U	0.7855	0.0057	0.0028	0.9770
	C+	0.0067	0.1604	0	0.9626
	C−	0.0217	0	0.0172	0.5830
PCC		0.9630			
KC		0.8863			
FCM		Estimated class			F_1
		U	C+	C−	
True class	U	0.7901	0.0073	0.0036	0.9785
	C+	0.0067	0.1588	0	0.9578
	C−	0.0171	0	0.0164	0.6122
PCC		0.9654			
KC		0.8915			
KI		Estimated class			F_1
		U	C+	C−	
True class	U	0.7646	0.0076	0.0030	0.9623
	C+	0.0101	0.1583	0	0.9465
	C−	0.0391	0.0002	0.0170	0.4451
PCC		0.9399			
KC		0.8233			

The results in bold are the best ones among the results obtained from all the comparing methods.

Table 5
Values of the evaluation criteria on the farmland dataset.

DLM		Estimated class			F_1
		U	C+	C−	
True class	U	0.9190	0.0055	0.0005	**0.9828**
	C+	0.0078	0.0442	0	**0.8680**
	C−	0.0182	0.0001	0.0045	**0.3248**
PCC		**0.9678**			
KC		**0.7382**			
C^2VA		Estimated class			F_1
		U	C+	C−	
True class	U	0.8115	0.0019	0.0015	0.9222
	C+	0.1260	0.0479	0	0.4282
	C−	0.0075	0	0.0036	0.4444
PCC		0.8631			
KC		0.3807			
FCM		Estimated class			F_1
		U	C+	C−	
True class	U	0.7682	0.0028	0.0013	0.8947
	C+	0.1711	0.0471	0.0001	0.3513
	C−	0.0057	0	0.0037	0.5125
PCC		0.8191			
KC		0.3019			
KI		Estimated class			F_1
		U	C+	C−	
True class	U	0.5453	0.0053	0.0002	0.7291
	C+	0.0244	0.0443	0	0.7479
	C−	0.3754	0.0002	0.0048	0.0251
PCC		0.5945			
KC		0.1446			

The results in bold are the best ones among the results obtained from all the comparing methods.

6.3. Results and comparisons

According to the first two experiments, we set N, N' and M' as 9, 20 and 60, respectively. This group of parameters will generate both satisfactory accuracy and moderate elapsed time.

6.3.1. Chanba dataset

The final results by different methods are shown in Fig. 15 with the corresponding values of evaluation criteria listed in Table 4.

Clearly, all of the four methods can detect the obvious changes (both C+ and C−). However, the DLM framework performs better in

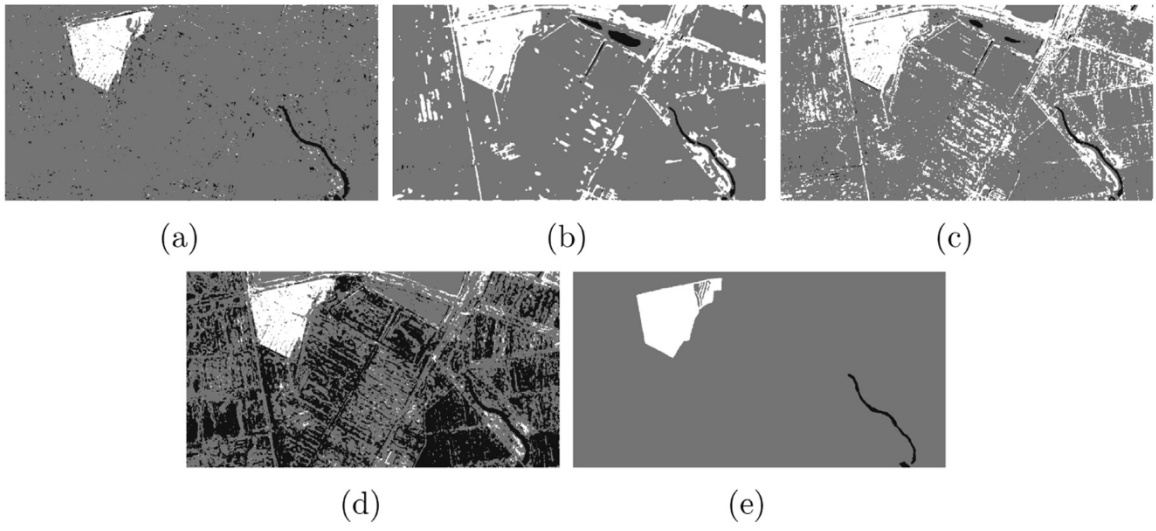

Fig. 16. Ternary maps from the farmland dataset by (a) DLM, (b) C^2VA, (c) FCM, and (d) KI. (e) is the reference map.

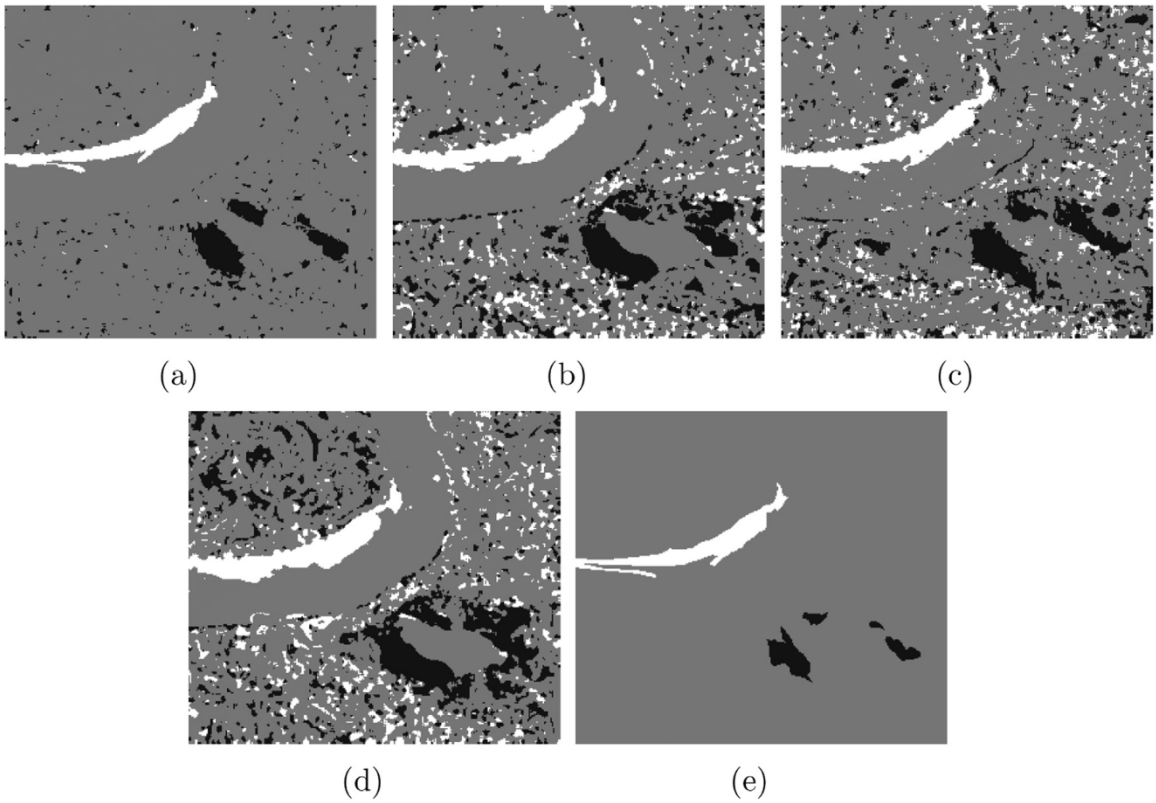

Fig. 17. Ternary maps from the YR dataset by (a) DLM, (b) C^2VA, (c) FCM, and (d) KI. (e) is the reference map.

details such as edges and small changed regions than the other state-of-the-art intensity-based methods. Due to the fact that these three methods do not incorporate the utilization of the spatial information, they are quite sensitive to noise. Besides, they are based on the operation directly on pixel intensity without revealing its detailed feature information, so the detection result will also be influenced by the discrepancy in noise level. From Table 4, the *PCC* and *KC* values from DLM are 0.9730 and 0.9161, respectively, higher than those obtained from the other methods. Moreover, the highest values of $F_l(U)$, $F_l(C+)$ and $F_l(C-)$ demonstrate its high detection accuracy to every class.

6.3.2. Farmland dataset

The final ternary maps and the values of evaluation criteria are shown and listed in Fig. 16 and Table 5, respectively.

Obviously, four methods lead to quite different results. The three intensity-based methods generate many wrongly classified pixels due to the disparity of noise types existing in the multi-sensor images. In general, C^2VA performs better than FCM and KI since it is capable of coping with multiple changes, but its ability is limited (seen from the scattered wrongly classified regions). The DLM framework outstands among these traditional methods because it extracts the inner feature from the pixel intensity, reducing the influence brought by the unbalanced corruption of noise to a large extent. The data in Table 5 shows the superiority of DLM to the other methods, demonstrating that DLM is robust when tackling the change-detection task from multi-sensor images.

6.3.3. YR dataset

The final ternary maps are shown in Fig. 17, and the values of evaluation criteria are given in Table 6.

The YR dataset is characterized by the large disparity in noise level. From both Fig. 17 and Table 6, the results from the intensity-based methods are severely affected by speckle noise, and some of the outliers even occupy some regions the size comparable to these small real changed regions. In fact, when the noise corruption is quite unbalanced, the multiplied noises do not follow the same distribution and hence it is not so feasible just to use the intensity information here. The C^2VA and KI methods include the Bayesian estimation which incorporates the use of prior knowledge from the pixel intensity, and such corrupted intensity may lead to quite inaccurate estimation. FCM, which is sensitive to noise, is also highly affected by unbalanced noise level. The DLM framework still keeps high accuracy due to its use of SDAE and SMN to extract the feature and to establish the flexible mapping functions. In Fig. 17(a), despite the noise disparity, DLM performs well from far fewer white or black spots appearing in the gray background, and the corresponding values of evaluation criteria also indicates its robustness in accuracy.

7. Concluding remarks

This paper introduces the DLM framework oriented to deal with the ternary change-detection task for two 1-D IU images. Different from the traditional methods which mainly consider the intensity of pixels, the DLM concerns more on the inner features of the pixels, avoiding the impact of unbalanced information to a large extent. Two pivotal networks, the SDAE used to extract the features and the SMN used to establish the mapping functions are employed. The features are extracted from the images first, and then three kinds of samples are selected, which are thereafter used to establish three corresponding mapping functions. A comparison of original feature and the mapping features is made to generate three feature channels, which are then arranged in series to comprise a change feature index. Finally, by classifying the index into three classes, we can obtain the final ternary

Table 6
Values of the evaluation criteria on the YR dataset.

DLM		Estimated class			F_1
		U	C+	C−	
True class	U	0.9194	0.0021	0.0005	**0.9778**
	C+	0.0046	0.247	0	**0.8808**
	C−	0.3460	0	0.0141	**0.4450**
PCC		**0.9582**			
KC		**0.6357**			
C^2VA		Estimated class			F_1
		U	C+	C−	
True class	U	0.8044	0.0008	0.0001	0.9121
	C+	0.0587	0.0261	0	0.4672
	C−	0.0956	0	0.0144	0.2317
PCC		0.8445			
KC		0.3081			
FCM		Estimated class			F_1
		U	C+	C−	
True class	U	0.8025	0.0004	0.0003	0.9110
	C+	0.0669	0.0264	0	0.4402
	C−	0.0893	0	0.0142	0.2410
PCC		0.8432			
KC		0.3063			
KI		Estimated class			F_1
		U	C+	C−	
True class	U	0.7568	0.0008	0.0005	0.8816
	C+	0.0729	0.0260	0	0.4135
	C−	0.1290	0	0.0140	0.1780
PCC		0.7968			
KC		0.2434			

The results in bold are the best ones among the results obtained from all the comparing methods.

map. The experimental results demonstrate its effectiveness from its high accuracy and the selected parameters from the test will lead to moderate and acceptable time complexity.

In general, the technique of DNN exhibits its unique superiority to traditional methods in terms of the change detection for IU images. The main contributions of the work can be summarized as follows.

First, it puts forward a new and significant change detection task, which is quite different from the previous works which focus much on the images with similar properties. Actually, since several aspects may affect the property of an image, IU images may show us more uncertainties than two similar images, so the analysis of the task is based on the deep feature domain rather than the traditional intensity domain. Second, it proposes a novel framework based on DNNs to tackle the new task. This framework involves the feature extraction and feature mapping strategies, which is superior to the traditional framework which is quite sensitive to the unbalanced information. Serving as the tools, two DNNs used to extract and map the features are applied to this framework. In this way, the images are processed beyond the superficial intensity information, reducing the impact of unbalanced information to a large extent. Moreover, the idea is also applicable to the detection of multiple changes (over three classes), and especially we can combine it with the C^2VA technique. So for the future work, we plan to design some adaptive version of DLM, focusing much on the way to cope with robust feature representations for a further task by using some other newly developed robust networks.

Acknowledgments

The authors wish to thank the editors and anonymous reviewers for their valuable comments and helpful suggestions which greatly improved the paper's quality. This work was supported by the National Natural Science Foundation of China (Grant no. 61422209), the National Program for Support of Top-notch Young Professionals of China and the Specialized Research Fund for the Doctoral Program of Higher Education (Grant no. 20130203110011).

Appendix

In this appendix, we will derive Eqs. (24) and (25). In the early literature [45], the *KC* was proposed and we can summarize its calculation. Let us suppose the square matrix $\mathbf{P} = \mathbf{P}_{3\times 3} = \mathbf{P}(i, j),\ 1 \le i \le 3,\ 1 \le j \le 3\}$, and *KC* is calculated as Eq. (26) originally:

$$KC = \frac{PCC - PRE}{1 - PRE}. \tag{26}$$

To calculate *PRE*, we turn to the steps in [45]. Two vectors $\mathbf{p}_i$ and $\mathbf{p}_j$ can be defined as follows:

$$\begin{cases} \mathbf{p}_i = \sum_{i=1}^{3} \mathbf{P}(i, j) = \mathbf{q}^{\mathrm{T}}\mathbf{P} \\ \mathbf{p}_j = \sum_{j=1}^{3} \mathbf{P}(i, j) = \mathbf{P}\mathbf{q}, \end{cases} \tag{27}$$

where $\mathbf{q} = [1, 1, 1]^{\mathrm{T}}$. Obviously, $\mathbf{p}_i$ is a row vector and $\mathbf{p}_j$ is a column vector. *PRE* is defined as their matrix product:

$$PRE = (\mathbf{p}_i)(\mathbf{p}_j) = (\mathbf{q}^{\mathrm{T}}\mathbf{P})(\mathbf{P}\mathbf{q}) = \mathbf{q}^{\mathrm{T}}\mathbf{P}^2\mathbf{q}. \tag{28}$$

Upon substituting Eqs. (23) and (28) into Eq. (26), we will be able to obtain the final form as shown in Eq. (24).

For F_1, according to the related formulae in [46], the precision (*pr*) and the recall (*re*) are first defined as:

$$\begin{cases} pr_\eta = \dfrac{P_{\eta\eta}}{P_{\mathrm{U}\eta} + P_{\mathrm{C}+\eta} + P_{\mathrm{C}-\eta}} = \dfrac{\mathbf{e}_\eta^{\mathrm{T}}\mathbf{P}\mathbf{e}_\eta}{\mathbf{e}_\eta^{\mathrm{T}}\mathbf{P}\mathbf{q}} \\ re_\eta = \dfrac{P_{\eta\eta}}{P_{\eta\mathrm{U}} + P_{\eta\mathrm{C}+} + P_{\eta\mathrm{C}-}} = \dfrac{\mathbf{e}_\eta^{\mathrm{T}}\mathbf{P}\mathbf{e}_\eta}{\mathbf{q}^{\mathrm{T}}\mathbf{P}\mathbf{e}_\eta}, \end{cases} \tag{29}$$

where $\eta \in \{U, C+, C-\}$. $\mathbf{e}_U = [1, 0, 0]^T$, $\mathbf{e}_{C+} = [0, 1, 0]^T$ and $\mathbf{e}_{C-} = [0, 0, 1]^T$. Then F_1 is defined as the harmonic mean of pr and re:

$$F_1(\eta) = \frac{2}{\frac{1}{pr_\eta} + \frac{1}{re_\eta}}. \tag{30}$$

Upon substituting Eq. (29) into Eq. (30), we can get Eq. (25).

References

[1] L. Bruzzone, D.F. Prieto, An adaptive semiparametric and context-based approach tounsupervised change detection in multitemporal remote-sensing images, IEEE Trans. Image Process. 11 (4) (2002) 452–466.

[2] C. Wu, B. Du, L. Zhang, A subspace-based change detection method for hyperspectral images, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 6 (2) (2013) 815–830.

[3] O. Yousif, Y. Ban, Improving urban change detection from multitemporal SAR images using PCA-NLM, IEEE Trans. Geosci. Remote Sens. 51 (4) (2013) 2032–2041.

[4] Y. Tang, X. Huang, L. Zhang, Fault-tolerant building change detection from urban high-resolution remote sensing imagery, IEEE Geosci. Remote Sens. Lett. 10 (5) (2013) 1060–1064.

[5] H. Hu, Y. Ban, Unsupervised change detection in multitemporal SAR images over large urban areas, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 7 (8) (2014) 3248–3261.

[6] Y. Ban, O.A. Yousif, Multitemporal spaceborne SAR data for urban change detection in China, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 5 (4) (2012) 1087–1094.

[7] F. Bovolo, L. Bruzzone, A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain, IEEE Trans. Geosci. Remote Sens. 45 (1) (2007) 218–236.

[8] F. Bovolo, S. Marchesi, L. Bruzzone, A framework for automatic and unsupervised detection of multiple changes in multitemporal images, IEEE Trans. Geosci. Remote Sens. 50 (6) (2012) 2196–2212.

[9] Y. Bazi, L. Bruzzone, F. Melgani, An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images, IEEE Trans. Geosci. Remote Sens. 43 (4) (2005) 874–887.

[10] F. Bujor, E. Trouvé, L. Valet, J.-M. Nicolas, J.-P. Rudant, Application of log-cumulants to the detection of spatiotemporal discontinuities in multitemporal SAR images, IEEE Trans. Geosci. Remote Sens. 42 (10) (2004) 2073–2084.

[11] J. Kittler, J. Illingworth, Minimum error thresholding, Pattern Recognit. 19 (1) (1986) 41–47.

[12] G. Moser, S.B. Serpico, Generalized minimum-error thresholding for unsupervised change detection from SAR amplitude imagery, IEEE Trans. Geosci. Remote Sens. 44 (10) (2006) 2972–2982.

[13] Y. Bazi, L. Bruzzone, F. Melgani, Image thresholding based on the EM algorithm and the generalized Gaussian distribution, Pattern Recognit. 40 (2) (2007) 619–634.

[14] W. Cai, S. Chen, D. Zhang, Fast and robust fuzzy *c*-means clustering algorithms incorporating local information for image segmentation, Pattern Recognit. 40 (3) (2007) 825–838.

[15] M. Gong, Z. Zhou, J. Ma, Change detection in synthetic aperture radar images based on image fusion and fuzzy clustering, IEEE Trans. Image Process. 21 (4) (2012) 2141–2151.

[16] M. Gong, L. Su, M. Jia, W. Chen, Fuzzy clustering with a modified MRF energy function for change detection in synthetic aperture radar images, IEEE Trans. Fuzzy Syst. 22 (1) (2014) 98–109.

[17] S. Liu, L. Bruzzone, F. Bovolo, P. Du, Hierarchical unsupervised change detection in multitemporal hyperspectral images, IEEE Trans. Geosci. Remote Sens. 53 (1) (2015) 244–260.

[18] S. Liu, L. Bruzzone, F. Bovolo, M. Zanetti, P. Du, Sequential spectral change vector analysis for iteratively discovering and detecting multiple changes in hyperspectral images, IEEE Trans. Geosci. Remote Sens. 53 (8) (2015) 4363–4378.

[19] S. Marchesi, F. Bovolo, L. Bruzzone, A context-sensitive technique robust to registration noise for change detection in VHR multispectral images, IEEE Trans. Image Process. 19 (7) (2010) 1877–1889.

[20] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: International Conference on Machine Learning, Helsinki, Finland, 2008, pp. 1096–1103.

[21] M. Gong, Y. Cao, Q. Wu, A neighborhood-based ratio approach for change detection in SAR images, IEEE Geosci. Remote Sens. Lett. 9 (2) (2012) 307–311.

[22] J. Ma, M. Gong, Z. Zhou, Wavelet fusion on ratio images for change detection in SAR images, IEEE Geosci. Remote Sens. Lett. 9 (6) (2012) 1122–1126.

[23] Y. Zheng, X. Zhang, B. Hou, G. Liu, Using combined difference image and *k*-means clustering for SAR image change detection, IEEE Geosci. Remote Sens. Lett. 11 (3) (2014) 691–695.

[24] Y. Bazi, L. Bruzzone, F. Melgani, Automatic identification of the number and values of decision thresholds in the log-ratio image for change detection in SAR images, IEEE Geosci. Remote Sens. Lett. 3 (3) (2006) 349–353.

[25] J.-S. Lee, E. Pottier, Polarimetric Radar Imaging: From Basics to Applications, CRC Press, Boca Raton, FL, 2009.

[26] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[27] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.

[28] J. Tang, C. Deng, G.-B. Huang, B. Zhao, Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine, IEEE Trans. Geosci. Remote Sens. 53 (3) (2015) 1174–1185.

[29] C.-H. Chang, Deep and shallow architecture of multilayer neural networks, IEEE Trans. Neural Netw. Learn. Syst. 26 (10) (2015) 2477–2486.

[30] L. Szymanski, B. McCane, Deep networks are effective encoders of periodicity, IEEE Trans. Neural Netw. Learn. Syst. 25 (10) (2014) 1816–1827.

[31] X. Chen, S. Xiang, C.-L. Liu, C.-H. Pan, Vehicle detection in satellite images by hybrid deep convolutional neural networks, IEEE Geosci. Remote Sens. Lett. 11 (10) (2014) 1797–1801.

[32] H. Goh, N. Thome, M. Cord, J.-H. Lim, Learning deep hierarchical visual feature coding, IEEE Trans. Neural Netw. Learn. Syst. 25 (12) (2014) 2212–2225.

[33] W. Hou, X. Gao, D. Tao, X. Li, Blind image quality assessment via deep learning, IEEE Trans. Neural Netw. Learn. Syst. 26 (6) (2015) 1275–1286.

[34] K. Zeng, J. Yu, R. Wang, C. Li, D. Tao, Coupled deep autoencoder for single image super-resolution, IEEE Trans. Cybern. 47 (1) (2017) 27–37. http://dx.doi.org/10.1109/TCYB.2015.2501373.

[35] R. Wang, D. Tao, Non-local auto-encoder with collaborative stabilization for image restoration, IEEE Trans. Image Process. 25 (5) (2016) 2117–2129.

[36] M. Gong, J. Zhao, J. Liu, Q. Miao, L. Jiao, Change detection in synthetic aperture radar images based on deep neural networks, IEEE Trans. Neural Netw. Learn. Syst. 27 (1) (2016) 125–138.

[37] P. Zhang, M. Gong, L. Su, J. Liu, Z. Li, Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images, ISPRS J. Photogramm. Remote Sens. 116 (2016) 24–41.

[38] Q.V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, A.Y. Ng, On optimization methods for deep learning, in: International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28–July, 2011, pp. 67–105.

[39] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep work with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.

[40] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670.

[41] S. Krinidis, V. Chatzis, A robust fuzzy local information *c*-means clustering algorithm, IEEE Trans. Image Process. 19 (5) (2010) 1328–1337.

[42] T.C. Havens, J.C. Bezdek, C. Leckie, L.O. Hall, M. Palaniswami, Fuzzy *c*-means algorithms for very large data, IEEE Trans. Fuzzy Syst. 20 (6) (2012) 1130–1146.

[43] P. Hore, L.O. Hall, D.B. Goldgof, Single pass fuzzy *c*-means, in: IEEE International Conference on Fuzzy Systems, London, UK, 2007, pp. 1–7.

[44] P. Hore, L.O. Hall, D.B. Goldgof, Y. Gu, A.A. Maudsley, A. Darkazanli, A scalable framework for segmenting magnetic resonance images, J. Signal Process. Syst. 54 (1–3) (2009) 183–203.

[45] J. Cohen, A coefficient of agreement for nominal scales, Educ. Psychol. Meas. 20 (1) (1960) 37–46.

[46] H. Huang, H. Xu, X. Wang, W. Silamu, Maximum F1-score discriminative training criterion for automatic mispronunciation detection, IEEE Trans. Audio Speech Lang. Process. 23 (4) (2015) 787–797.

Linzhi Su was born in 1989. He received the B.S. degree in intelligence science and technology from Xidian University, Xi'an, China, in 2011, where he is currently working toward the Ph.D. degree. His research interests include computational intelligence and image understanding.

Maoguo Gong was born in 1979. He received the B.S. degree in electronic engineering and the Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2003 and 2009, respectively. He is currently a Full Professor with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education. His research interests include computational intelligence with applications, image segmentation and change detection.

Puzhao Zhang was born in 1989. He received the B.S. degree in intelligence science and technology from Xidian University, Xi'an, China, in 2013, where he is currently working toward the Ph.D. degree. His research interests include computational intelligence and image understanding.

Mingyang Zhang was born in 1988. He received the B.S. degree in intelligence science and technology from Xidian University, Xi'an, China, in 2012, where he is currently working toward the Ph.D. degree. His research interests include computational intelligence and image understanding.

Jia Liu was born in 1991. He received the B.S. degree in intelligence science and technology from Xidian University, Xi'an, China, in 2013, where he is currently working toward the Ph.D. degree. His research interests include computational intelligence and image understanding.

Hailun Yang was born in 1992. He received the B.S. degree in intelligence science and technology from Qingdao Technological University, Qingdao, China, in 2014, where he is currently working toward the M.S. degree. His research interests include computational intelligence and image understanding.