Journal of Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

Fuzzy autoencoder for multiple change detection in remote sensing images

Linzhi Su
Xin Cao

SPIE.

Linzhi Su, Xin Cao, "Fuzzy autoencoder for multiple change detection in remote sensing images," *J. Appl. Remote Sens.* **12**(3), 035014 (2018), doi: 10.1117/1.JRS.12.035014.

Fuzzy autoencoder for multiple change detection in remote sensing images

Linzhi Su and Xin Cao*
Northwest University, School of Information and Technology, Xuefu Avenue, Xi'an, China

Abstract. This paper establishes the fuzzy autoencoder (FAE) to detect multiple changes between two one-dimensional multitemporal images. Different from the traditional approaches based on the pixel intensity, FAE includes a multilayer structure through self-reconstruction to extract the feature from an image. Due to the existence of noise in the images, the raw data tend to be corrupted and fail to detect the real changes. Therefore, the fuzzy number is introduced to the autoencoder to establish the FAE which is able to suppress the noise and learn robust features. In this way, the information in the fuzzy domain is introduced into the input, and in practice the fuzzy domain is discretized to facilitate the calculation. In addition, the weighted Frobenius norm is used to establish the loss function which can be minimized to achieve the optimal parameters. The framework is highlighted by the newly designed FAE. As the fuzzy number is introduced into the autoencoder, more information concerning the fuzzy domain is taken into consideration and thus the impact brought by the noise is relieved to a large extent. Hence, the FAE can generate robust features, enhancing its performance on deep feature representation learning. Several tests on three datasets show us the proper parameter settings, and the experimental results from the FAE framework and the other compared approaches demonstrate its effectiveness and robustness in terms of accuracy and elapsed time. [DOI: 10.1117/1.JRS.12.035014]

Keywords: change detection; fuzzy number; fuzzy autoencoder; fuzzy neural networks; remote sensing.

Paper 180184 received Mar. 6, 2018; accepted for publication Aug. 16, 2018; published online Aug. 31, 2018.

1 Introduction

Researchers usually define the task of change detection as the process to detect and recognize the changes occurring in multitemporal images that cover the same area but are taken at different times.[1] In both scientific research and practical use, it is treated as a significant issue due to its wide applications in remote sensing,[1–6] urban studying,[7–11] and agricultural survey.[12]

In the literature, the research of change detection begins with the detection of changes from the unchanged areas. It is viewed as a classification problem and was originally researched through the utilization of the pixel intensity. For example, the Bayes formula is applied to the image histogram through the Kittler–Illingworth criterion[13,14] or the expectation-maximization iteration.[15] A more accurate approach is the fuzzy c-means clustering (FCM) algorithm due to the use of fuzzy membership. Based on FCM, many improved versions have been proposed, such as the reformulated fuzzy clustering with local information (RFLICM),[16] fuzzy clustering based on the Markov random field (MRFFCM),[17–19] and two-level clustering,[20] in which the spatial information is flexibly utilized. Recently, researchers have turned to the multiple change detection, a task where the changes are subdivided into several subclasses, and the change vector analysis (CVA) technique have been developed to deal with it.[2,21] Some other hierarchical frameworks based on CVA have also been applied to the hyperspectral images and very high-resolution images.[22,23] These typical intensity-based approaches, however, conduct the computations depending on fixed and complicated equations and thus have limited ability to realize the

*Address all correspondence to: Xin Cao, E-mail: xin_cao@163.com

1931-3195/2018/$25.00

knowledge learning and feature extraction. In fact, serving as a class-unbalanced problem, change detection can be formulated as an incremental learning problem, and the artificial neural networks (ANNs) are considered as an opportune tools to solve it.

Several ANN models have found their application to change detection. Gong et al.[24] proposed an approach by using the binary restricted Boltzmann machine (RBM) network. In the approach, several RBMs are arranged in series to extract the features from the two images, and such features are passed layer by layer. Zhang et al. then proposed a change-detection framework which incorporates unsupervised feature learning and mapping analysis. In the framework, the autoencoder is adopted in the feature learning process, and the features are mapped through another mapping network.[25] In the two literatures, the neighborhood pixels are involved in the input, i.e., the network incorporates the spatial information flexibly. Then based on such information, the whole network is optimized through the minimization of the reconstruction function. Due to the combination of the spatial information and deep learning techniques, these approaches are able to cope with some complicated tasks in change detection. Specifically, the technique in the work by Gong et al.[24] is capable of tackling the case of synthetic aperture radar (SAR) images which are usually severely corrupted by speckle noise,[26] while the framework by Zhang et al.[25] is especially designed for the images with different resolutions.

Despite the feature representation ability of ANNs, it is found that the units in every layer and the parameters that represent the relationships between layers are restricted to be constants. Chen et al.[27] have pointed out several disadvantages of a regular RBM due to such a structure. Actually, the similar structure can also be found in an autoencoder, so it has the similar problems. First, the feature extracted from an autoencoder is not so robust when the input itself is corrupted by noise. Second, the parameter learning process is confined in a restricted small space, which will limit its capability to represent the feature. In change detection, images are usually corrupted by noise, and the complicated terrain feature will make it more difficult to discern the changes. Particularly, the learning processes in Refs. 24 and 25 are based on determinate input and output values, and therefore these networks are quite sensitive to noise. In addition, these works aims to cope with the binary change detection task, and as for the multiple change detection, it is reasonable to develop some more robust ANNs which are able to tackle such a complicated task.

To improve the presentation ability of ANNs, the fuzzy neural networks (FNNs) are proposed and have been applied to many fields. By FNN is meant the ANN that incorporates fuzzy knowledge, such as the Takagi & Sugeno (TS) model, type-2 fuzzy method, fuzzy number, etc. Several techniques have been proposed oriented to the control system, reasoning, classification, and so on. For example, Lin et al.[28] embed the wavelet strategy after the rule layer in the FNN, designing a robust and intelligent approach and applying it to the electric power steering system. In Ref. 29, the authors combined the generalized ellipsoidal basis function and FNN by designing a proper TS model, realizing the dynamic tanker steering control system. In addition, in Ref. 30, the authors made an elaborate research of type-2 fuzzy systems and showd a modeling framework by ameliorating the fuzzification layer. Many other approaches based on FNN incorporating the type-2 system have been proposed. Some of them are focused for theoretical study[31,32] and some are of practical use.[33,34] In Ref. 35, the authors suggested utilizing the fuzzy number to improve the performance of regression analysis and a fuzzy support vector machine is used. Such an idea is also adopted to enhance the deep learning capability of the RBM as fuzzy RBM (FRBM),[27] and the theoretical analysis and the final test results demonstrate the excellent performance of the FRBM by incorporating fuzzy numbers.

This paper deals with the multiple change detection for the images with only one spectral channel, or one-dimenional (1-D) images as usually referred to. Compared with multichannel images (such as hyperspectral images and polarimetric SAR images), 1-D images are easy to visualize and occupy less storage space. Nevertheless, containing only one spectral channel, 1-D images are not as capable as the multichannel images to show more complicated surface features, which are also challenging to discern by using some ordinary methods. In Refs. 21 and 36, the authors have pointed out the CVA technique does not apply to the 1-D images. To overcome such a limitation, in Ref. 37 the authors provided an approach in which one 1-D image is expanded into a multichannel feature map by using the denoising autoencoder (DAE) and the CVA technique is applied to the feature map, which provides us a basic framework to the multiple change detection for 1-D images. Following the framework and enlightened by Refs. 27 and 35, we

propose a fuzzy autoencoder (FAE), an FNN that will be applied to detect multiple changes. In the FAE, fuzzy numbers are introduced to the autoencoder to create a fuzzy domain for the input data, and thus fuzzy information will be involved in its learning process. In this way, the network utilizes not only the spatial information but also the fuzzy information to alleviate the impact brought by the noise and increase the robustness.

The paper is organized into six sections. Section 2 gives out the corresponding background knowledge. Section 3 introduces the proposed FAE and its learning process as well as its application to multiple change detection. The datasets used in experiments and the experimental settings are introduced in Sec. 4 and the corresponding results along with their analysis are given in Sec. 5. Finally, the conclusions are drawn in Sec. 6.

2 Background

Let us consider $I_1 = \{I_1(i,j), 1 \le i \le A, 1 \le j \le B\}$ and $I_2 = \{I_2(i,j), 1 \le i \le A, 1 \le j \le B\}$, two multispectral images taken over the same geographical area at two different times, respectively. The task of change detection is to generate a map I_F that shows the unchanged class (U) and k types of changes $(C_1, C_2, \ldots, C_k)$. This section gives an introduction to the related background knowledge about the autoencoder and the fuzzy number.

2.1 *Autoencoder*

The autoencoder is a model used to obtain the feature representation of the given data.[38] Given a normalized input vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)^{\mathrm{T}} \in [0,1]^N$, the network first encodes it into the code vector $\mathbf{y} = (y_1, y_2, \ldots, y_M)^{\mathrm{T}} \in [0,1]^M$, which is then mapped back as $\mathbf{x}^{\dagger} = (x_1^{\dagger}, x_2^{\dagger}, \ldots, x_N^{\dagger})^{\mathrm{T}} \in [0,1]^N$. This is shown as

$$\begin{cases} \mathbf{y} = s(\mathbf{W}\mathbf{x} + \mathbf{b}) \\ \mathbf{x}^{\dagger} = s(\mathbf{W}'\mathbf{y} + \mathbf{c}) \end{cases}. \tag{1}$$

s is the sigmoid function usually defined as $s(t) = 1/(1 + \mathrm{e}^{-t})$. Usually the case of tied weight is considered, i.e., $\mathbf{W}' = \mathbf{W}^{\mathrm{T}}$, so there are actually three parameters to be determined. Here the symbol θ is used to denote the parameter set, i.e., $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$. A loss function $\mathcal{L}$ is then established between $\mathbf{x}$ and $\mathbf{x}^{\dagger}$, and several forms can be used to establish $\mathcal{L}$, such as the squared error and the average reconstruction cross-entropy that have been introduced in Ref. 38. By minimizing $\mathcal{L}$, θ can be determined. Such a minimization problem can be solved by the stochastic gradient descent algorithm, avoiding local minima of the training error.[39] In this way, $\mathbf{y}$ can be viewed as an appropriate representation of $\mathbf{x}$, or the feature as usually referred to. The autoencoder is shown in Fig. 1.

In Ref. 40, two modified versions of the autoencoder, the DAE, and the sparse autoencoder are summarized, and they have been demonstrated to be more robust than the regular autoencoder. From the introduction above, the optimal parameters are determined by $\mathcal{L}$, which is based on $\mathbf{x}$. When $\mathbf{x}$ is corrupted by noise, $\mathcal{L}$ will also be affected. Although the loss function of DAE

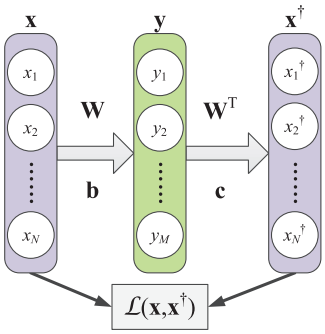


Fig. 1 Structure of the autoencoder.

can alleviate the impact of noise to some extent as reported in Ref. 38, $\mathcal{L}$ is still established on $\mathbf{x}$ which has been originally corrupted. In fact, to optimize $\mathcal{L}$ is to find an excellent fitting between the constant vector $\mathbf{x}$ and the reconstructed $\mathbf{x}^{\dagger}$. If such a fitting is based on fuzzy numbers, the learning space will not be confined in a restricted small space but extended to a fuzzy domain, and thus the network will become robust when extracting the features. The fuzzy number is first introduced here.

2.2 *Fuzzy Number*

2.2.1 *Basic concept*

The concept of fuzzy number was first proposed by Zadeh, and is well described in detail in Ref. 41. Given a real number x, a map $f:\mathbb{R} \to [0,1]$ is established such that

$$\begin{cases} f(t) = 1 \Leftrightarrow t = x \\ \lim\limits_{|t|\to\infty} f(t) = 0. \end{cases} \tag{2}$$

Such a mapping relation is called a fuzzy number, and the symbol $\overline{x}$ is used to represent it as in Ref. 27. $f(t)$ is the corresponding membership function. In practice, a symmetric and locally monotonic membership function is of practical significance. In this paper, the symmetric triangular fuzzy number will be adopted, and its membership function is

$$f(t) = \max\left\{1 - \frac{|t-x|}{a}, 0\right\}, \tag{3}$$

where a is a positive constant that controls the width of $\overline{x}$. Obviously, the center of $\overline{x}$ is x. Actually, $\overline{x}$ corresponds to a closed interval $[x^L, x^R]$, and we refer to x^L and x^R as the left bound and right bound of $\overline{x}$, respectively. We define $\overline{x}^L = [x^L, x]$ and $\overline{x}^R = [x, x^R]$ that are called the left interval and the right interval of $\overline{x}$, respectively. Figure 2 shows the symmetric triangular fuzzy number.

2.2.2 *α-cut*

Given a real constant $\alpha \in [0,1]$, a new membership function can be defined as

$$g(t) = \begin{cases} f(t), & \text{if } f(t) \geq \alpha \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

Thus, the new mapping relation is called the α-cut of $\overline{x}$ and the symbol $\overline{x}[\alpha]$ is used to represent it. $\overline{x}[\alpha]$ is also a fuzzy number which has the same center as $\overline{x}$ does. We can also define its left bound $x[\alpha]^L$ and right bound $x[\alpha]^R$. From Eq. (4), it is obvious that $\overline{x}[0] = \overline{x}$ and that $\overline{x}[1] = x$. In the latter case, the fuzzy number degrades to a common real number. Figure 3 shows the α-cut of the symmetric triangular fuzzy number.

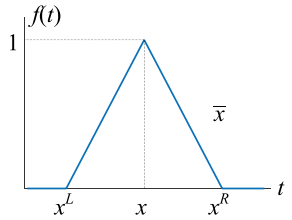


Fig. 2 Symmetric triangular fuzzy number.

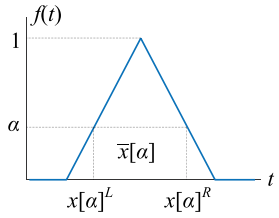


Fig. 3 Illustration of the α-cut of the fuzzy number.

2.2.3 *Operations*

The symmetric triangular fuzzy number actually corresponds to a closed interval, so its operation is just the interval arithmetic operation. Upon the determination of the membership function, we use the expression $\overline{x} = [x^L, x^R]$ as in Ref. 27 if no confusion is engendered.

In Ref. 42, the interval arithmetic is given. For two fuzzy numbers $\overline{x} = [x^L, x^R]$ and $\overline{y} = [y^L, y^R]$, the addition and subtraction operations are

$$H(\overline{x}) = \begin{cases} \overline{x} + \overline{y} = [x^L + y^L, x^R + y^R] \\ \overline{x} - \overline{y} = \overline{x} + (-\overline{y}) = [x^L - y^R, x^R - y^L] \end{cases}. \tag{5}$$

For a monotonic function $H(\cdot)$, the function operation is

$$H(\overline{x}) = \begin{cases} [H(x^L), H(x^R)], & \text{if } H(\cdot) \text{ is ascending} \\ [H(x^R), H(x^L)], & \text{if } H(\cdot) \text{ is descending} \end{cases}. \tag{6}$$

According to Eq. (6), we can define the scalar-multiplication. Given a constant $\lambda \in \mathbb{R}$, we have

$$\lambda\overline{x} = \begin{cases} [\lambda x^L, \lambda x^R], & \text{if } \lambda > 0 \\ 0, & \text{if } \lambda = 0 \\ [\lambda x^R, \lambda x^L], & \text{if } \lambda < 0 \end{cases}. \tag{7}$$

3 Design of FAE for Multiple Change Detection

3.1 *Basic Outline of FAE*

As in Refs. 24 and 25, a normalized input $\mathbf{x} = (x_1, x_2, \ldots, x_N)^{\mathrm{T}}$ includes a certain pixel and its neighborhood pixels, and thus the spatial information is taken into consideration. The FAE not only involves the spatial information but also fuzzifies the input. First, the fuzzy number transformation $\mathbf{x} \to \overline{\mathbf{x}} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_N)^{\mathrm{T}}$ is involved. Therefore, we have

$$\begin{cases} \mathbf{x}^L = (x_1^L, x_2^L, \ldots, x_N^L)^{\mathrm{T}} \\ \mathbf{x}^R = (x_1^R, x_2^R, \ldots, x_N^R)^{\mathrm{T}} \\ \overline{\mathbf{x}} = [\mathbf{x}^L, \mathbf{x}^R] \\ \overline{\mathbf{x}}^L = [\mathbf{x}^L, \mathbf{x}] \\ \overline{\mathbf{x}}^R = [\mathbf{x}, \mathbf{x}^R] \end{cases}. \tag{8}$$

Through the FAE, the reconstructed fuzzy vector $\overline{\mathbf{x}}^{\dagger}$ is generated. Thus, the loss function $\overline{\mathcal{L}} = \overline{\mathcal{L}}(\overline{\mathbf{x}}, \overline{\mathbf{x}}^{\dagger})$ in the FAE should reflect the proximity of two fuzzy vectors $\overline{\mathbf{x}}$ and $\overline{\mathbf{x}}^{\dagger}$. The aim of using the fuzzy number is to find a robust proximity by extending the range into the fuzzy domain, and to exhibit a robust feature representation, the left interval $\overline{x}^L$ and the right interval $\overline{x}^R$ are used to establish the reconstructed error, respectively. Therefore, two parameter sets, $\theta^L = \{\mathbf{W}^L, \mathbf{b}^L, \mathbf{c}^L\}$ and $\theta^R = \{\mathbf{W}^R, \mathbf{b}^R, \mathbf{c}^R\}$ are involved. Through the optimization of $\overline{x}^L$, the appropriate parameters can be obtained. Upon these ideas, the FAE can be designed as shown in Fig. 4.

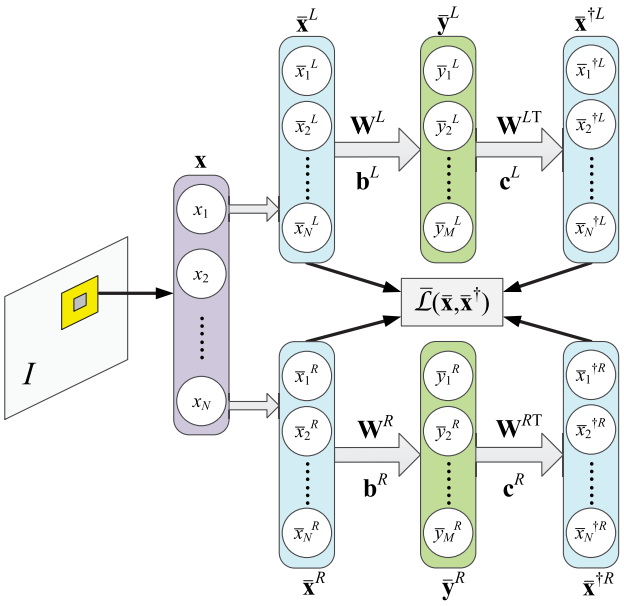


Fig. 4 Illustration of the FAE. The gray part in the image I represents the pixel (i,j) and the yellow area represents its neighborhood.

3.2 *Discretization Approximation*

The loss function $\overline{\mathcal{L}}$ is used to measure the proximity of $\overline{\mathbf{x}}$ and $\overline{\mathbf{x}}^\dagger$. However, a fuzzy number corresponds to an interval that contains infinitive points and we can not compute the reconstructed error for every point. So it is necessary to discretize the continuous fuzzy domain to facilitate the establishment of $\overline{\mathcal{L}}$, and the α-cut is used here.

First, the closed interval $[0,1]$ is divided into m equivalent subintervals and thus we have $m+1$ endpoints $\{0,1/m,2/m,\ldots,(m-1)/m,1\}$. And then we define

$$\alpha_r = \frac{r}{m}, \qquad r = 0,1,\ldots,m. \tag{9}$$

According to Sec. 2.2.2, each $\alpha \in [0,1)$ corresponds to two real numbers $x[\alpha]^L$ and $x[\alpha]^R$; in the case where $\alpha = 1$, there is only one corresponding value x. Therefore, the $m+1$ endpoints correspond to $2m+1$ real numbers ranging from x^L to x^R. Obviously, we have $x[\alpha_0]^L = x[0]^L = x^L$, $x[\alpha_0]^R = x[0]^R = x^R$, and $x[\alpha_m] = x[1] = x$. Thus, $\overline{x}$ can be expressed in vector form as

$$\overline{x} = (x^L, x[\alpha_1]^L, \ldots, x[\alpha_{m-1}]^L, x, x[\alpha_{m-1}]^R, \ldots, x[\alpha_1]^R, x^R). \tag{10}$$

The original input of the autoencoder is a column vector, and in Eq. (10), each element has been extended into a fuzzy row vector. Hence, the fuzzy input vector can be expressed as an $N \times (2m+1)$ matrix as

$$\begin{aligned}\overline{\mathbf{x}} &= (\mathbf{x}^L, \mathbf{x}[\alpha_1]^L, \ldots, \mathbf{x}[\alpha_{m-1}]^L, \mathbf{x}, \mathbf{x}[\alpha_{m-1}]^R, \ldots, \mathbf{x}[\alpha_1]^R, \mathbf{x}^R) \\ &= \begin{bmatrix} x_1^L & x[\alpha_1]_1^L & \ldots & x[\alpha_{m-1}]_1^L & x_1 & x[\alpha_{m-1}]_1^R & \ldots & x[\alpha_1]_1^R & x_1^R \\ x_2^L & x[\alpha_1]_2^L & \ldots & x[\alpha_{m-1}]_2^L & x_2 & x[\alpha_{m-1}]_2^R & \ldots & x[\alpha_1]_2^R & x_2^R \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ x_N^L & x[\alpha_1]_N^L & \ldots & x[\alpha_{m-1}]_N^L & x_N & x[\alpha_{m-1}]_N^R & \ldots & x[\alpha_1]_N^R & x_N^R \end{bmatrix}. \end{aligned} \tag{11}$$

In Eq. (11), the rows of $\overline{\mathbf{x}}$ denote all the pixels involved in a neighborhood, and the columns denote the fuzzy domain. Therefore, the input matrix $\overline{\mathbf{x}}$ represents not only the spatial information but also the fuzzy information, which serves as a great enhancement.

Similarly, $\overline{\mathbf{x}}^L$ and $\overline{\mathbf{x}}^R$ can be expressed in Eqs. (12) and (13), respectively:

$$\begin{aligned}\overline{\mathbf{x}}^L &= (\mathbf{x}^L, \mathbf{x}[\alpha_1]^L, \ldots, \mathbf{x}[\alpha_{m-1}]^L, \mathbf{x}) \\ &= \begin{bmatrix} x_1^L & x[\alpha_1]_1^L & \ldots & x[\alpha_{m-1}]_1^L & x_1 \\ x_2^L & x[\alpha_1]_2^L & \ldots & x[\alpha_{m-1}]_2^L & x_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_N^L & x[\alpha_1]_N^L & \ldots & x[\alpha_{m-1}]_N^L & x_N \end{bmatrix},\end{aligned} \tag{12}$$

$$\begin{aligned}\overline{\mathbf{x}}^R &= (\mathbf{x}, \mathbf{x}[\alpha_{m-1}]^R, \ldots, \mathbf{x}[\alpha_1]^R, \mathbf{x}^R) \\ &= \begin{bmatrix} x_1 & x[\alpha_{m-1}]_1^R & \ldots & x[\alpha_1]_1^R & x_1^R \\ x_2 & x[\alpha_{m-1}]_2^R & \ldots & x[\alpha_1]_2^R & x_2^R \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_N & x[\alpha_{m-1}]_N^R & \ldots & x[\alpha_1]_N^R & x_N^R \end{bmatrix}.\end{aligned} \tag{13}$$

Thus, we can use Eqs. (14) and (15) to describe the FAE:

$$\begin{cases} \overline{\mathbf{y}}^L = s(\mathbf{W}^L \overline{\mathbf{x}}^L + \mathbf{b}^L \mathbf{q}) \\ \overline{\mathbf{x}}^{\dagger L} = s(\mathbf{W}^{LT} \overline{\mathbf{y}}^L + \mathbf{c}^L \mathbf{q}) \end{cases}, \tag{14}$$

$$\begin{cases} \overline{\mathbf{y}}^R = s(\mathbf{W}^R \overline{\mathbf{x}}^R + \mathbf{b}^R \mathbf{q}) \\ \overline{\mathbf{x}}^{\dagger R} = s(\mathbf{W}^{RT} \overline{\mathbf{y}}^R + \mathbf{c}^R \mathbf{q}) \end{cases}. \tag{15}$$

It is worth noting that $\overline{\mathbf{x}}^L$ or $\overline{\mathbf{x}}^R$ is no longer a vector but a matrix. Therefore, the row vector $\mathbf{q} = (1,1,\ldots,1)$ is used to change the bias vectors into the matrix form. Thus, the hidden layer is also a fuzzy version, and thus the FAE is more robust in information transfer between layers than the ordinary autoencoder.

3.3 *Establishment of the Loss Function*

The loss function $\overline{\mathcal{L}}$ is used to measure the proximity of $\overline{\mathbf{x}}$ and $\overline{\mathbf{x}}^{\dagger}$, and its form will be more complicated due to the incorporation of fuzzy numbers. When two fuzzy numbers are equal, two corresponding intervals are totally coincident. Since the α-cut can represent any two symmetric points in the interval, the loss function can be expressed as an integral form as

$$\overline{\mathcal{L}}(\overline{\mathbf{x}}, \overline{\mathbf{x}}^{\dagger}) = \int_0^1 (\|\overline{\mathbf{x}}[\alpha]^L - \overline{\mathbf{x}}[\alpha]^{\dagger L}\| + \|\overline{\mathbf{x}}[\alpha]^R - \overline{\mathbf{x}}[\alpha]^{\dagger R}\|) \mathrm{d}\alpha. \tag{16}$$

However, this is difficult to realize due to the integral operation, so it is necessary that we use the discrete form to rewrite Eq. (16). In fact, the Euclidean norm for vectors used in the loss function means the sum of the square error. Since in Sec. 3.2 we have turn $\overline{\mathbf{x}}$ and $\overline{\mathbf{x}}^{\dagger}$ into matrices, the matrix Frobenius norm (F-norm), which possesses similar meaning to vector 2-norm, is adopted. In addition, as the fuzzy domain is involved, the F-norm will be modified in a weighted version.

If the ordinary F-norm is used, we have

$$\bar{\mathcal{L}}(\overline{\mathbf{x}}, \overline{\mathbf{x}}^{\dagger}) = \sqrt{\sum_{s=1}^{N} \sum_{r=0}^{m} (x[\alpha_r]_s^L - x[\alpha_r]_s^{\dagger L})^2} + \sqrt{\sum_{s=1}^{N} \sum_{r=0}^{m} (x[\alpha_r]_s^R - x[\alpha_r]_s^{\dagger R})^2}. \tag{17}$$

Equation (17) can be viewed as the discrete form of Eq. (16). However, each element is treated equally, which will result in a biased result because each element in the fuzzy domain does not function equally. If the membership is not taken into account, the loss function will lead to a biased result and also make the interval corresponding to the fuzzy number indiscriminative. To represent such inequality, a weight is allocated to each element. According to the definition of

the fuzzy number, the corresponding membership α_r to each element can serve as the weight. Thus, $\overline{\mathcal{L}}$ is expressed as

$$\overline{\mathcal{L}}(\overline{\mathbf{x}}, \overline{\mathbf{x}}^{\dagger}) = \sqrt{\sum_{s=1}^{N} \sum_{r=0}^{m} \alpha_r (x[\alpha_r]_s^L - x[\alpha_r]_s^{\dagger L})^2} + \sqrt{\sum_{s=1}^{N} \sum_{r=0}^{m} \alpha_r (x[\alpha_r]_s^R - x[\alpha_r]_s^{\dagger R})^2}. \quad (18)$$

By minimizing $\overline{\mathcal{L}}$ through the stochastic gradient descent algorithm, the parameters $\theta^L = \{\mathbf{W}^L, \mathbf{b}^L, \mathbf{c}^L\}$ and $\theta^R = \{\mathbf{W}^R, \mathbf{b}^R, \mathbf{c}^R\}$ can be determined.

3.4 *Parameter Sharing*

From the analysis above, each input corresponds to one basic block of FAE. For the entire n inputs in an image, we can arrange them to share a common set of parameters. For the input element $\mathbf{x}^{(t)}$, we can obtain the training result $\theta^{L(t)}$ and $\theta^{R(t)}$ by using the FAE with $\theta^{L(t-1)}$ and $\theta^{R(t-1)}$, $t = 1,2,\ldots,n$. Then, $\theta^{L(n)}$ and $\theta^{R(n)}$ are applied to all the input elements, and the objective function $\overline{J}$ is minimized:

$$\overline{J} = \frac{1}{n} \sum_{t=1}^{n} \overline{\mathcal{L}}(\overline{\mathbf{x}}^{(t)}, \overline{\mathbf{x}}^{\dagger(t)}). \quad (19)$$

Thus, the optimal parameters θ^{L*} and θ^{R*} can be obtained and we can compute the corresponding fuzzy feature layer $\overline{\mathbf{y}}^{(1)}, \overline{\mathbf{y}}^{(1)}, \ldots, \overline{\mathbf{y}}^{(n)}$. Figure 5 shows the FAE with respect to all the inputs clearly.

3.5 *Defuzzification*

The obtained feature to each pixel is a fuzzified one and it is necessary to defuzzify it from a matrix to a column vector. The weighted idea is employed again and we have

$$\mathbf{y} = \sum_{r=0}^{m} \alpha_r \mathbf{y}[\alpha_r]^L + \sum_{r=0}^{m} \alpha_r \mathbf{y}[\alpha_r]^R = \overline{\mathbf{y}}^L \alpha^L + \overline{\mathbf{y}}^R \alpha^R, \quad (20)$$

where

$$\begin{cases} \boldsymbol{\alpha}^L = \left(0, \frac{1}{m}, \frac{2}{m}, \ldots, \frac{m-1}{m}, 1\right)^{\mathrm{T}} \\ \boldsymbol{\alpha}^R = \left(1, \frac{m-1}{m}, \ldots, \frac{2}{m}, \frac{1}{m}, 0\right)^{\mathrm{T}}. \end{cases} \quad (21)$$

3.6 *Detection of Multiple Changes*

For the pixel (i, j), the corresponding feature vector is $\mathbf{y}(i, j)$, and thus an M-channel feature map $\mathbf{Y}$ is generated as shown in Fig. 6 intuitively.

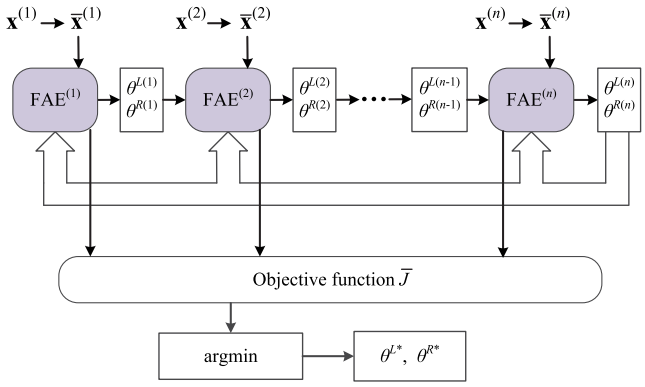


Fig. 5 Illustration of the FAE with respect to all the inputs.

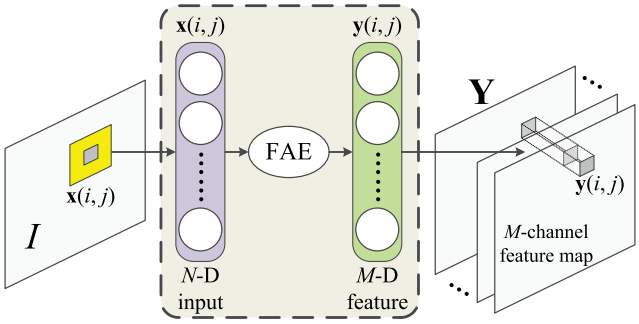


Fig. 6 Generation of the *M*-channel feature map **Y**.

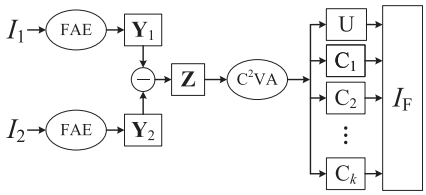


Fig. 7 Process to detect multiple changes by using the C^2VA technique after feature extraction.

In this way, the feature information in the original image is distributed into M channels, and due to the incorporation of fuzzy information, the feature is robust to noise interference. Actually, feature extraction is just like the process to generate a multispectral image. Therefore, the compressed CVA (C^2VA) technique[21] which is originally designed to detect multiple changes for multispectral images is used to generate the final map I_F.

We briefly recall the C^2VA technique here. First, the feature maps $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are generated from I_1 and I_2 through the FAE, respectively, and then the difference feature map $\mathbf{Z} = \mathbf{Y}_2 - \mathbf{Y}_1$ is calculated. According to $\mathbf{Z}$, the multiple changes can be detected by using the magnitude and the direction angle which can be computed through the related formulae in Ref. 21. This process is shown in Fig. 7.

4 Dataset Description and Experimental Settings

Three datasets are used to demonstrate the effectiveness of the FAE-based framework and the corresponding experimental settings will also be given in this section.

4.1 *Datasets*

The first dataset consists of two SAR images (290×260, resolution $8\ \text{m} \times 8\ \text{m}$), reflecting the changes of the Yellow River near the Dongying City, China, and hence the name Yellow River (YR for short) dataset. Figures 8(a) and 8(b) show the two images taken in 2008 and 2009, respectively. It can be seen that some area is flooded, whereas that some new lands emerged. The reference map, shown in Fig. 8(c), is generated by artificial marking through a combination of on-the-spot investigation and expert knowledge.

The second dataset is called the Irving dataset (591×491, resolution $12\ \text{m} \times 12\ \text{m}$) from optical remote sensing radars, reflecting the changes happening in the Irving City. Figures 9(a) and 9(b) are taken in 2001 and 2015, respectively. Three kinds of changes can be found: the new artificial constructions (such as the square and the high way), the disappearance of lands, and the emergence of wood (newly planted trees). Figure 9(c) is the reference map generated by an on-the-spot investigation and expert knowledge.

The third dataset, the Tunzhuang dataset (522×383, resolution $10\ \text{m} \times 10\ \text{m}$) also from optical remote sensing radars, shows the changes of the Tunzhuang Village in the Puyang

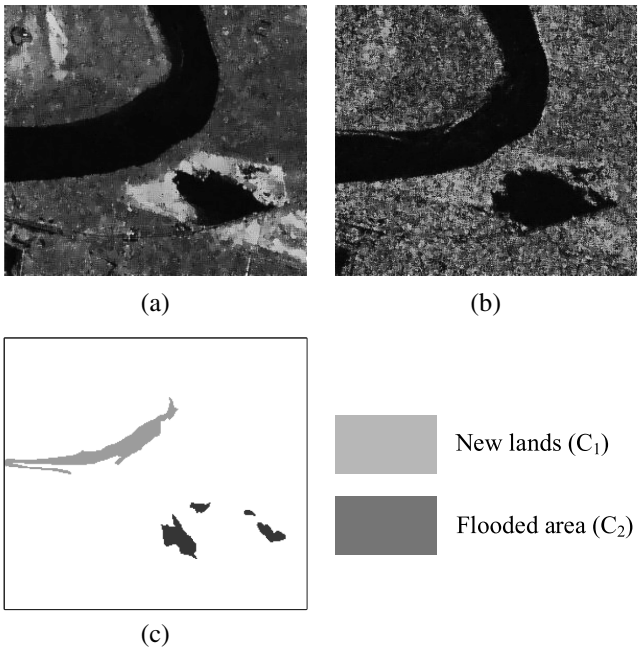


Fig. 8 YR dataset. (a) The image taken in 2008. (b) The image taken in 2009. (c) The reference map.

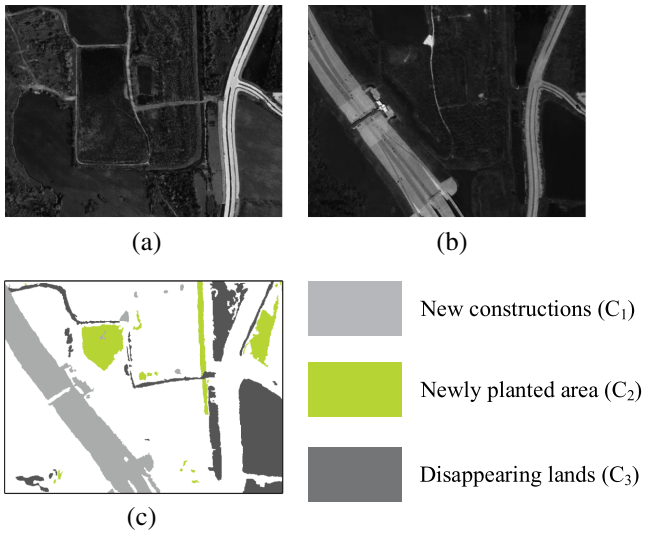


Fig. 9 Irving dataset. (a) The image taken in 2001. (b) The image taken in 2015. (c) The reference map.

City, China, between September and October, 2014. From Figs. 10(a) and 10(b), the crops (mainly the wheats) were harvested and some new crops were sown. In addition, many lake algae bred in the pond and some of the lands were flooded. Therefore, there are four kinds of changes, and the reference map which is based on an on-the-spot investigation is shown in Fig. 10(c).

4.2 *Experimental Settings*

Four experiments are designed here. The first one is the test of parameters including the input size N and the hidden feature size M. As stated in Sec. 3, N depends on the size of the

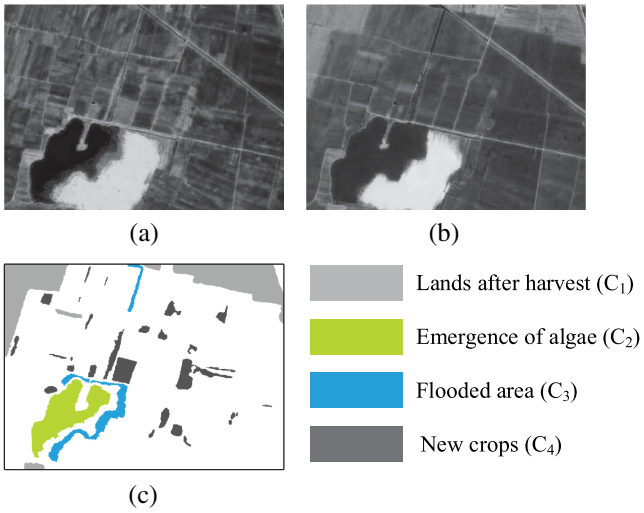


Fig. 10 Tunzhuang dataset. (a) The image taken in September, 2014. (b) The image taken in October, 2014. (c) The reference map.

neighborhood, and here three neighborhoods with their sizes 3×3, 5×5, and 7×7 are chosen. Thus, N ranges among 9, 25, and 49. M is set artificially from 10 to 50 at intervals of 10. We will test their impact on both accuracy and elapsed time.

The second experiment is the test of the integer parameter m which denotes the number of the subintervals that [0,1] is divided into. m will be set ranging from 1 to 20 at intervals of 1. It is worth noting that the case where $m = 1$ corresponds to the original autoencoder, so this experiment can also show the comparison of the performance of the autoencoder and the FAE.

The third experiment is to show the comparison of the final maps from different approaches. Two available classification approaches for image segmentation, FCM and KWFLICM,[43] will be adopted as the intensity-based comparison algorithm. FCM is the basic fuzzy clustering method and KWFLICM is the kernel weighted version by fully considering the local information. They are both pixel intensity-based algorithms. In addition, we will also show the results from the DAE-based approach where the FAE is replaced by the DAE in Fig. 7.

The fourth experiment is a comparison of the results from the weighted F-norm and the ordinary F-norm. This is to show the superiority the weighted F-norm to the ordinary F-norm when the loss function is established.

For the quantitative evaluations, the classification performance table (the data in which comprise a matrix) is given as in Ref. 21. According to the table, the percentage correct classification (PCC) and the Kappa coefficient (KC) can be calculated as introduced in Ref. 44. Since more classification information is incorporated, KC is a more cogent criterion that can reflect the overall performance than PCC.

5 Experimental Results and Analysis

5.1 *Test of N and M*

Figures 11 and 12 show the test results from the three datasets, and they include the impact on KC (which is cogent in accuracy) and elapsed time (in seconds). Their respective analyses are given as follows.

5.1.1 *Influence of N*

N depends on the size of the selected neighborhood. From Fig. 11, the 5×5 neighborhood generates the highest accuracy. In fact, the utilization of the neighbors as the input is based on the hypnosis that the neighbors share the same property of the central pixel. This is thought to be true if a moderate neighborhood is considered. As the image is usually corrupted by

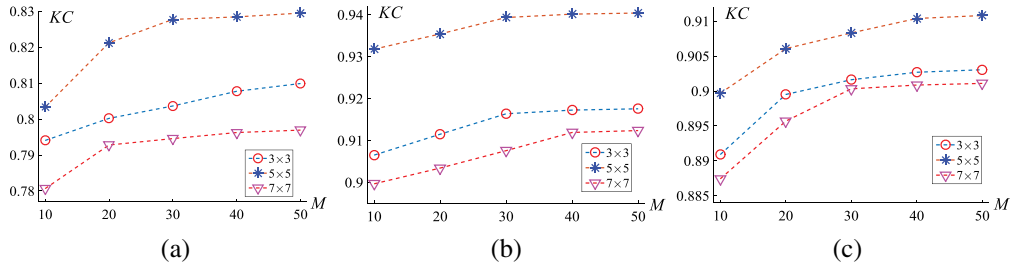


Fig. 11 Impact of N and M on KC. (a) The YR dataset. (b) The Irving dataset. (c) The Tunzhuang dataset.

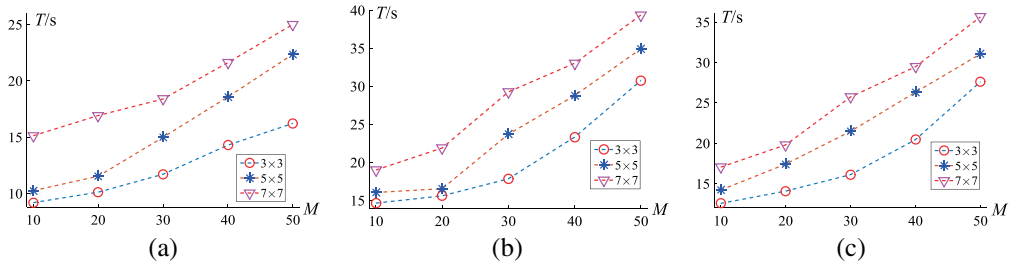


Fig. 12 Impact of N and M on the elapsed time (in seconds). (a) The YR dataset. (b) The Irving dataset. (c) The Tunzhuang dataset.

random noise, a small neighborhood (say 3×3) does not involve enough information to balance the noise influence, and a larger one will correspond to a better feature representation from this point of view. However, when the neighborhood is so large (say 7×7), the relevance between the central pixel and a far neighbor will be quite low, especially in the heterogeneous edge area. Therefore, too large a neighborhood size may fail to represent the feature of the central pixel and thus generate a low accuracy instead. In addition, according to Fig. 12, it takes more time for the algorithm concerning a larger neighborhood to run, just as anticipated. Therefore, in the following experiments, the 5×5 neighborhood is selected and N is set as 25.

5.1.2 *Influence of M*

Obviously, the accuracy and the time are both positively associated with M. By extracting the features, the information of a certain type of change can be expressed in one or a few channels. A small value of M will lead to the overlap of some information, leading to inaccuracy. As M increases, the feature information will be detailed and exhibited into more channels, facilitating the classification process. On the other hand, too large a value of M may lead to information redundancy, a similar phenomenon occurring in hyperspectral images. In this case, more time is required, but the accuracy does not increase so much. An integration of the KC values and the elapsed time suggests that it is appropriate to set M as 30.

5.2 *Test of m*

The test results of m on the accuracy and elapsed time are shown in Figs. 13 and 14. When $m = 1$, the result corresponds to an ordinary autoencoder.

From Fig. 13, the accuracy becomes high when $m > 1$, which demonstrates the superiority of the fuzzy number optimization to real number optimization. When optimizing a fuzzy number, the process involves fuzzy extension and thus the features will be affected less by the corrupted inputs. Especially when there is a severe noise corruption (such as the YR dataset), the accuracy increases remarkably at first. In addition, after m increases to a certain value, the accuracy almost remains stable. In addition, from Fig. 14, the time almost linearly increases, and this is due to the fact that Eqs. (18) and (20) incorporate the sum operation which linearly depends on the value of m. So after an integrative observation and analysis, m is set as 12 in the following experiments.

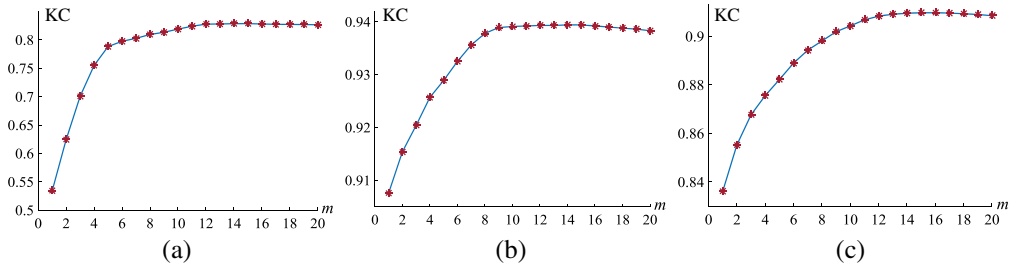


Fig. 13 Impact of *m* on KC. (a) The YR dataset. (b) The Irving dataset. (c) The Tunzhuang dataset.

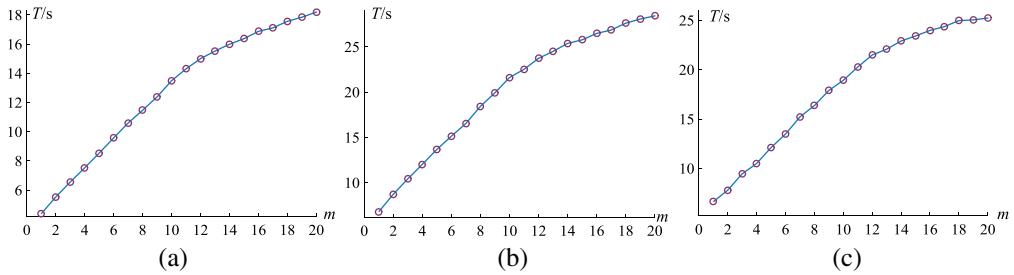


Fig. 14 Impact of *m* on the elapsed time (in seconds). (a) The YR dataset. (b) The Irving dataset. (c) The Tunzhuang dataset.

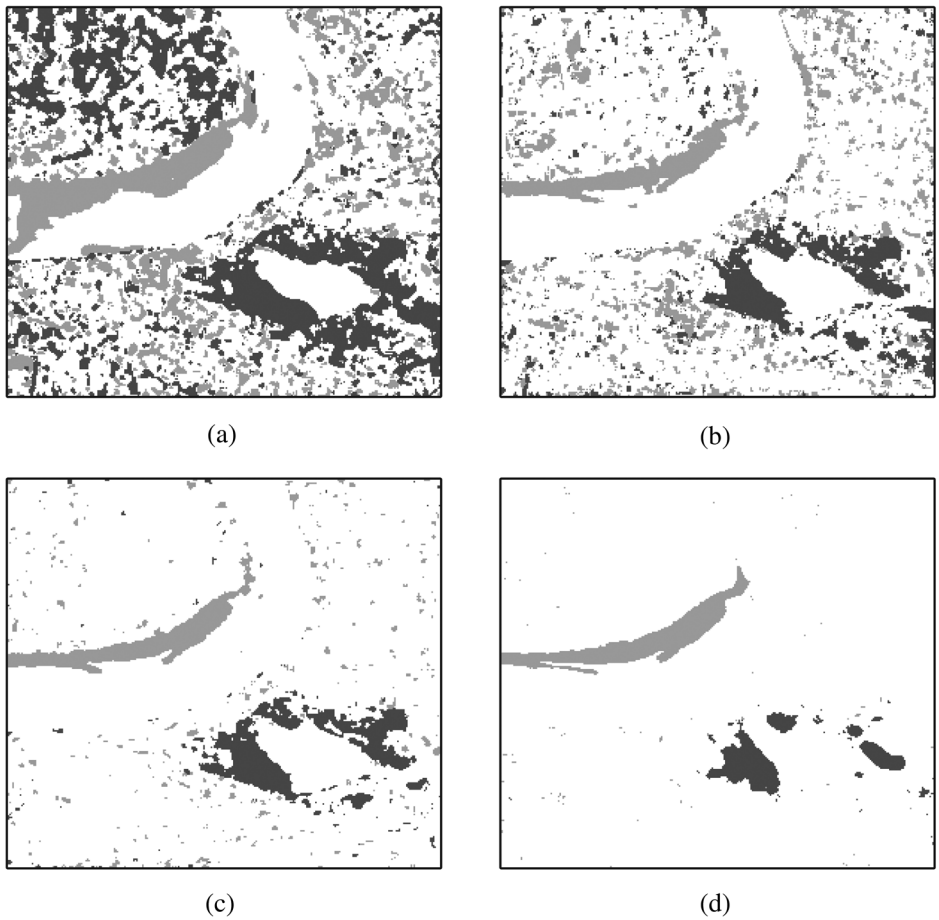


Fig. 15 Final maps from the YR dataset by: (a) FCM, (b) KWFLICM, (c) the DAE-based framework, and (d) the FAE-based framework.

5.3 *Final Maps from Different Approaches*

According to the first two experiments, we set N, M, and m as 25, 30, and 12, respectively. This setting will generate both satisfactory accuracy and moderate elapsed time.

5.3.1 *YR dataset*

Figure 15 and Table 1 show the results from the YR dataset by different approaches.

Table 1 Values of the evaluation criteria on the YR dataset.

			Estimated class		
FCM			U	C_1	C_2
True class		U	0.6116	0.0006	0.0005
		C_1	0.1345	0.0262	0
		C_2	0.2125	0	0.0140
	PCC			0.6518	
	KC			0.1403	
			Estimated class		
KWFLICM			U	C_1	C_2
True class		U	0.7887	0.0022	0.0005
		C_1	0.1010	0.0246	0.0001
		C_2	0.0690	0	0.0139
	PCC			0.8272	
	KC			0.2703	
			Estimated class		
DAE			U	C_1	C_2
True class		U	0.8992	0.0019	0.0007
		C_1	0.0247	0.0249	0
		C_2	0.0347	0	0.0138
	PCC			0.9379	
	KC			0.5340	
			Estimated class		
FAE			U	C_1	C_2
True class		U	0.9432	0.0004	0.0003
		C_1	0.0061	0.0264	0
		C_2	0.0094	0	0.0142
	PCC			**0.9838**	
	KC			**0.8277**	

Note: The bold characters are used to emphasize the overall evaluation criteria (PCC and KC) from the proposed FAE technique.

The YR dataset consists of two SAR images with severe speckle noise corruption, and therefore the results by the two intensity-based approaches are seriously affected by noise. From Figs. 15(a) and 15(b), the use of spatial information can reduce the impact of the noise to some extent, so the result by KWFLICM is better than that by FCM in which no spatial information is utilized. But such improvement is limited as it is still based on pixel intensity. Another reason is that the two approaches are not hierarchical-based, and the noise corruption from the unchanged class will severely affect the classification of the two changed classes. Taking these two factors into consideration, from Fig. 15(c) DAE generates a much better map because it suppresses the noise by extracting the inner feature. In addition, FAE further improves the detection accuracy from Fig. 15(d), and this results from the fact that the FAE involves the fuzzy domain, providing a wider range. Thus, the corrupted pixels may be ameliorated in the optimization process. From Table 1, the PCC and KC from the FAE are 0.9838 and 0.8277, respectively, much higher than those from the other three approaches.

5.3.2 *Irving dataset*

Figure 16 and Table 2 show the final maps and the quantitative results, respectively.

A comparison between Fig. 16(a) and the reference map suggests many wrongly detected regions as well as some vital undetected areas. Especially, some newly planted regions that show the vegetation status fails to be detected by FCM. KWFLICM performs better than FCM from Fig. 16(b), but it is still based on the intensity information without utilizing its deep feature, so some errors can also be found (e.g., the absent detection of the highway). The DAE improves the accuracy by dispersing the changed information into different feature bands, and for the FAE, the flexible fuzzy domain makes it possible to detect the real multiple changes. From the quantitative results, the KC value for the FAE is 0.9393, the highest of the four.

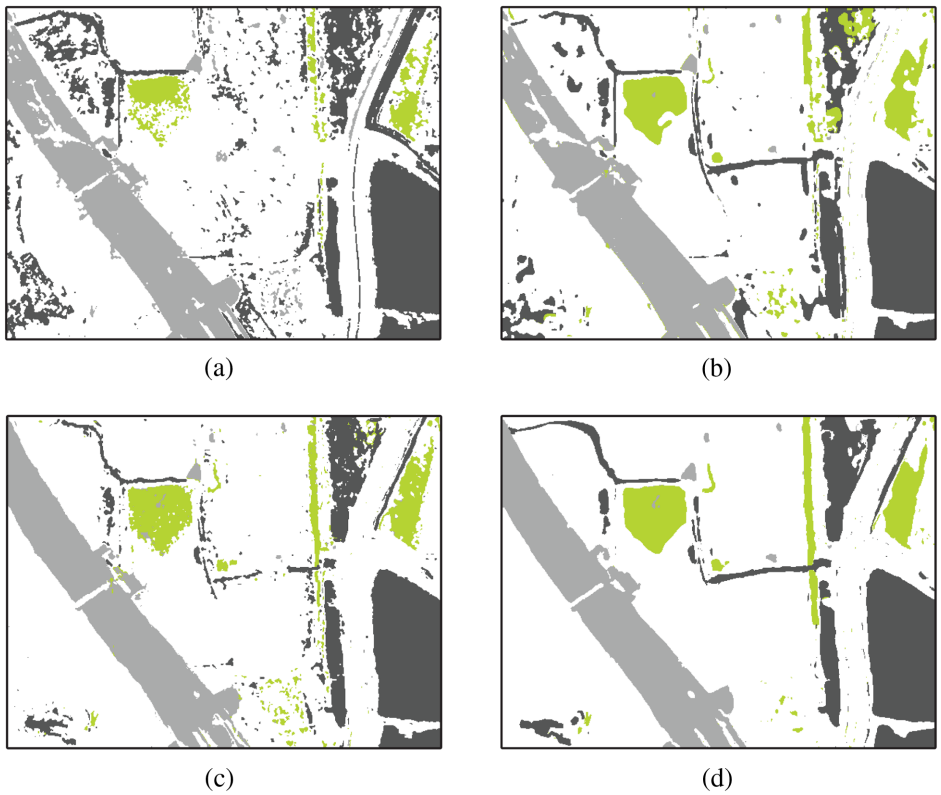


Fig. 16 Final maps from the Irving dataset by: (a) FCM, (b) KWFLICM, (c) the DAE-based framework, and (d) the FAE-based framework.

Table 2 Values of the evaluation criteria on the Irving dataset.

		Estimated class			
FCM		U	C_1	C_2	C_3
True class	U	0.5957	0.0084	0.0238	0.0193
	C_1	0.0112	0.1181	0.0013	0
	C_2	0.0018	0.0003	0.0261	0
	C_3	0.0797	0.0001	0.0017	0.1126
PCC			0.8524		
KC			0.7111		

		Estimated class			
KWFLICM		U	C_1	C_2	C_3
True class	U	0.6313	0.0079	0.0120	0.0136
	C_1	0.0085	0.1181	0.0001	0
	C_2	0.0083	0.0008	0.0403	0.0058
	C_3	0.0404	0	0.0006	0.1124
PCC			0.9022		
KC			0.8056		

		Estimated class			
DAE		U	C_1	C_2	C_3
True class	U	0.6642	0.0038	0.0059	0.0092
	C_1	0.0021	0.1224	0.0010	0
	C_2	0.0068	0.0007	0.0460	0.0008
	C_3	0.0152	0	0.0001	0.1219
PCC			0.9545		
KC			0.9076		

		Estimated class			
FAE		U	C_1	C_2	C_3
True class	U	0.6702	0.0018	0.0039	0.0055
	C_1	0.0046	0.1249	0.0001	0
	C_2	0.0033	0.0001	0.0489	0.0002
	C_3	0.0103	0	0	0.1261
PCC			**0.9700**		
KC			**0.9393**		

Note: The bold characters are used to emphasize the overall evaluation criteria (PCC and KC) from the proposed FAE technique.

5.3.3 *Tunzhuang dataset*

Figure 17 shows the final maps from different approaches and Table 3 exhibits the corresponding quantitative results.

Four types of changes are involved in the 1-D images in the Tunzhuang dataset, and therefore their detection appears more difficult than those of the YR and Irving datasets. As anticipated, FCM and KWFLICM generate many wrongly detected areas, and this is shown in two aspects. One is some unchanged areas are detected as changed areas, and the other is one type of change is wrongly detected as another. In Figs. 17(a) and 17(b), many areas are falsely detected as the areas with new crops, emergence of algae and flooded areas, which will mislead the judgement of the agricultural survey. In Fig. 17(c) from DAE, the classification performance becomes much better because of the utilization of the inner feature, but on the white background, some tiny multicolor spots can be found due to the influence of noise. Just as expected, the FAE eliminates the spots to a great extent and thus further ameliorates the performance. The values of evaluation criteria in Table 3 are consistent with the final maps visually, demonstrating the robust ability of the FAE to detect multiple changes from 1-D images.

5.4 *Results from Different Norms for Establishing the Loss Function*

In this experiment, the performance of two loss functions that adopt the ordinary F-norm and the weighted F-norm, as in Eqs. (17) and (18), respectively, are compared. Figures 18–20 show the final maps based on two loss functions with the corresponding PCC and KC values shown in Table 4, which also shows the elapsed time.

The results show the superiority of using the weighted F-norm to the ordinary F-norm when the loss function is established. Since the fuzzy number has a vague expression, every element functions differently in the network, and therefore it is inappropriate to treat each element equally in the fuzzy domain. If the ordinary F-norm is adopted, the robustness of the features will decrease. From the maps based on the ordinary F-norm, some falsely detected areas (such as many colored spots on the white background) can be found, and many of these areas vanished

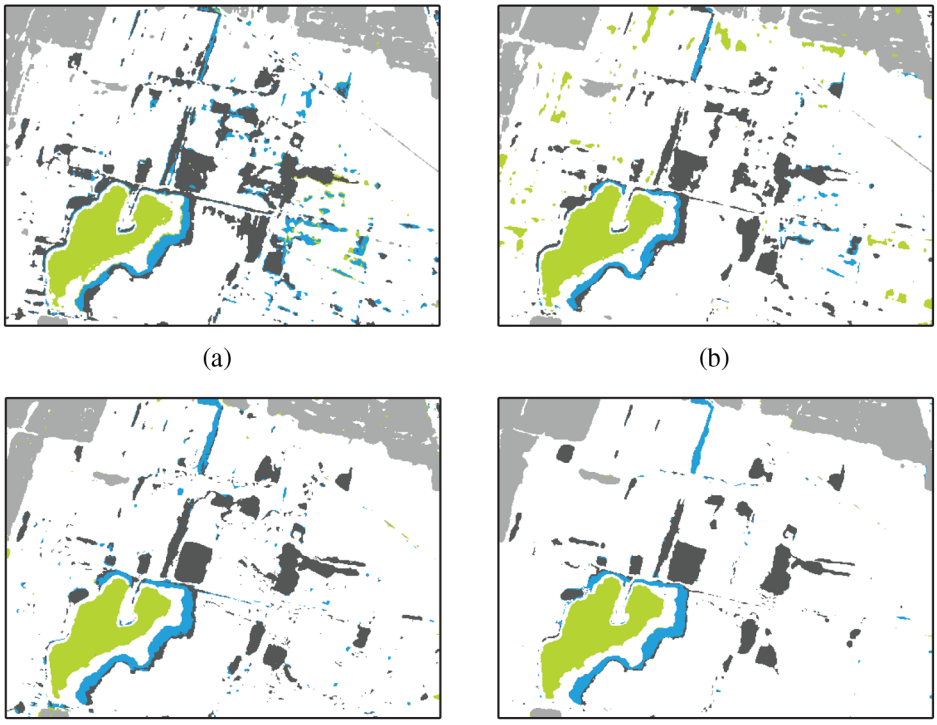


Fig. 17 Final maps from the Tunzhuang dataset by: (a) FCM, (b) KWFLICM, (c) the DAE-based framework, and (d) the FAE-based framework.

Table 3 Values of the evaluation criteria on the Tunzhuang dataset.

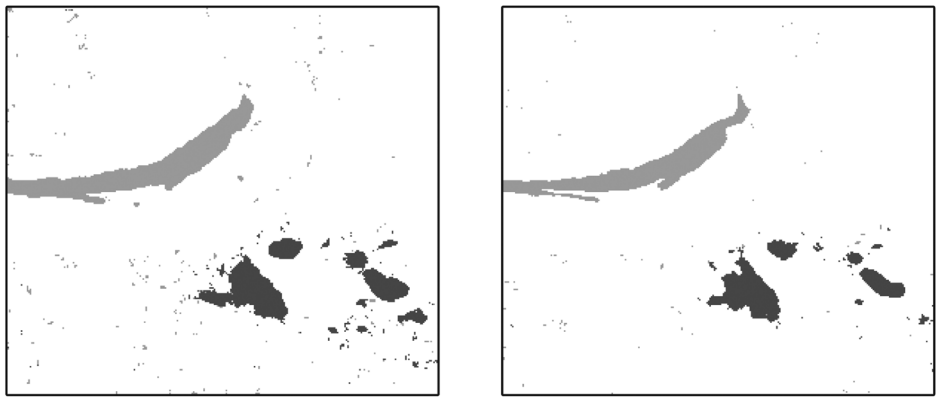
		Estimated class				
FCM		U	C_1	C_2	C_3	C_4
True class	U	0.7119	0.0155	0.0011	0.0002	0.0035
	C_1	0.0061	0.0816	0	0	0
	C_2	0.0048	0.0001	0.0401	0.0009	0.0002
	C_3	0.0215	0	0	0.0109	0.0030
	C_4	0.0527	0	0	0.0100	0.0360
PCC				0.8805		
KC				0.7020		

		Estimated class				
KWFLICM		U	C_1	C_2	C_3	C_4
True class	U	0.7347	0.0132	0.0002	0.0020	0.0096
	C_1	0.0079	0.0836	0	0	0
	C_2	0.0174	0.0003	0.0410	0	0
	C_3	0.0062	0	0	0.0109	0.0005
	C_4	0.0308	0	0	0.0090	0.0325
PCC				0.9028		
KC				0.7441		

		Estimated class				
DAE		U	C_1	C_2	C_3	C_4
True class	U	0.7544	0.0075	0.0018	0.0004	0.0026
	C_1	0.0043	0.0889	0.0001	0	0
	C_2	0.0011	0.0007	0.0393	0	0
	C_3	0.0077	0.0001	0	0.0167	0.0008
	C_4	0.0295	0	0	0.0049	0.0393
PCC				0.9387		
KC				0.8362		

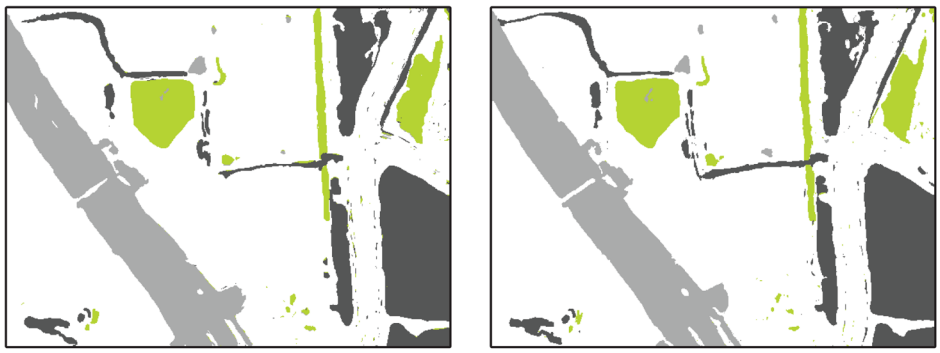
		Estimated class				
FAE		U	C_1	C_2	C_3	C_4
True class	U	0.7762	0.0041	0	0.0023	0.0029
	C_1	0.0030	0.0930	0	0	0
	C_2	0.0022	0.0001	0.0412	0.0001	0
	C_3	0.0031	0	0	0.0174	0.0005
	C_4	0.0126	0	0	0.0022	0.0393
PCC				**0.9670**		
KC				**0.9083**		

Note: The bold characters are used to emphasize the overall evaluation criteria (PCC and KC) from the proposed FAE technique.



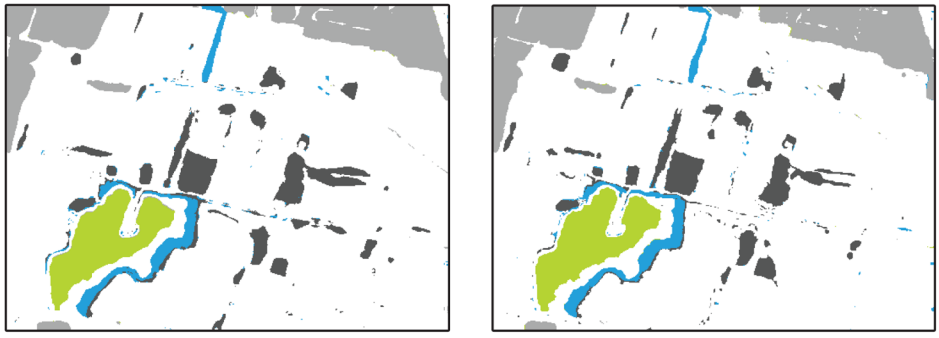
(a) (b)

Fig. 18 Final maps from the YR dataset based on the loss functions by using: (a) ordinary F-norm and (b) weighted F-norm.



(a) (b)

Fig. 19 Final maps from the Irving dataset based on the loss functions by using: (a) ordinary F-norm and (b) weighted F-norm.



(a) (b)

Fig. 20 Final maps from the Tunzhuang dataset based on the loss functions by using: (a) ordinary F-norm and (b) weighted F-norm.

in the maps based on the weighted F-norm. From Table 4, the detection accuracy is high when the weighted F-norm is used, which also demonstrates its effectiveness. Moreover, the weighted F-norm leads to a little more time, and this is due to the additional weighted operation. But this time is acceptable when we consider its high accuracy.

Table 4 Quantitative results based on different loss functions.

		PCC	KC	Time(s)
YR	F-norm	0.9672	0.7028	10.8
	Weighted F-norm	0.9838	0.8277	11.0
Irving	F-norm	0.9581	0.9154	16.4
	Weighted F-norm	0.9700	0.9393	18.7
Tunzhuang	F-norm	0.9543	0.8734	14.9
	Weighted F-norm	0.9670	0.9083	16.2

6 Concluding Remarks

This paper introduces the FAE-based framework for coping with the multiple change detection task for two 1-D remote sensing images. Since the intensity-based approaches fail to discover the inner feature of the images and thus lead to inaccuracy, we turn to the deep neural networks and design the FAE in which the fuzzy number is used to improve the performance of the autoencoder. In the FAE, the input vector is modified into a fuzzy number vector, utilizing both the spatial information and fuzzy domain information. To further facilitate the calculation, the α-cut is used, discretizing the continuous fuzzy domain and turning each input into a matrix. Based on the discretized fuzzy number, the corresponding loss function is established by using the weighted F-norm. Finally, after the defuzzification process, the C^2VA technique is applied to the multichannel feature map to detect multiple changes. Through the experiments on three datasets, its advantages over the intensity-based approaches are demonstrated and the most appropriate parameter settings are also determined.

In general, the FAE-based framework exhibits its unique superiority to traditional intensity-based approaches in terms of the multiple change detection for 1-D images. The FAE modifies the autoencoder into an FNN by using the fuzzy number. In this way, the features extracted by the FAE retain more information due to the incorporation of the fuzzy domain, reducing the impact brought by the corruptive noise. In the recent literatures, the networks with some deep structures, or the deep neural networks as usually referred to, have also been focused on due to their outstanding representation ability.[39,45,46] Hence, future work will focus more on the further theoretical study of the combination of FAE and deep learning, and it is also worthwhile to promote its applications to the other fields in remote sensing image processing.

Disclosures

The authors declare no competing financial interest.

Acknowledgments

This work was supported by the Program of the National Natural Science Foundation of China under Grant No. 61701403 and the Natural Science Basic Research Plan in Shaanxi Province of China under Grant No. 2017JQ6006.

References

1. L. Bruzzone and D. F. Prieto, "An adaptive semiparametric and context-based approach tounsupervised change detection in multitemporal remote-sensing images," *IEEE Trans. Image Process.* **11**, 452–466 (2002).
2. 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.* **45**, 218–236 (2007).

3. F. Bovolo, C. Marin, and L. Bruzzone, "A hierarchical approach to change detection in very high resolution SAR images for surveillance supplications," *IEEE Trans. Geosci. Remote Sens.* **51**, 2042–2054 (2013).
4. V. Neagoe et al., "Concurrent self-organizing maps for supervised/unsupervised change detection in remote sensing images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **7**, 3525–3533 (2014).
5. C. Huo et al., "Learning relationship for very high resolution image change detection," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **9**, 3384–3394 (2016).
6. H. C. Li et al., "Robust semi-NMF with total variation for unsupervised SAR image change detection," *IET Electron. Lett.* **54**(14), 892–894 (2018).
7. O. Yousif and Y. Ban, "Improving urban change detection from multitemporal SAR images using PCA-NLM," *IEEE Trans. Geosci. Remote Sens.* **51**, 2032–2041 (2013).
8. Y. Ban and O. A. Yousif, "Multitemporal spaceborne SAR data for urban change detection in China," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **5**, 1087–1094 (2012).
9. O. Yousif and Y. Ban, "Improving SAR-based urban change detection by combining MAP-MRF classifier and nonlocal means similarity weights," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **7**, 4288–4300 (2014).
10. M. R. Sahebi, "Evaluating coverage changes in national parks using a hybrid change detection algorithm and remote sensing," *J. Appl. Remote Sens.* **13**(8), 083646 (2014).
11. L. C. Chen and L. J. Lin, "Detection of building changes from aerial images and light detection and ranging (LIDAR) data," *J. Appl. Remote Sens.* **4**(1), 041870 (2010).
12. 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.* **35**, 858–867 (1997).
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.* **43**, 874–887 (2005).
14. G. Moser and S. B. Serpico, "Generalized minimum-error thresholding for unsupervised change detection from SAR amplitude imagery," *IEEE Trans. Geosci. Remote Sens.* **44**, 2972–2982 (2006).
15. Y. Bazi, L. Bruzzone, and F. Melgani, "Image thresholding based on the EM algorithm and the generalized Gaussian distribution," *Pattern Recognit.* **40**, 619–634 (2007).
16. 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.* **21**, 2141–2151 (2012).
17. M. Gong et al., "Fuzzy clustering with a modified MRF energy function for change detection in synthetic aperture radar images," *IEEE Trans. Fuzzy Syst.* **22**, 98–109 (2014).
18. B. N. Subudhi et al., "Spatio-contextual fuzzy clustering with Markov random field model for change detection in remotely sensed images," *Opt. Laser Technol.* **57**, 284–292 (2014).
19. F. Malfait et al., "Unsupervised change detection based on improved Markov random field technique using multichannel synthetic aperture radar images," *J. Appl. Remote Sens.* **8**(1), 083591 (2014).
20. H. C. Li et al., "Gabor feature based unsupervised change detection of multitemporal SAR images based on two-level clustering," *IEEE Geosci. Remote Sens. Lett.* **12**(12), 2458–2462 (2015).
21. 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.* **50**, 2196–2212 (2012).
22. S. Liu et al., "Hierarchical unsupervised change detection in multitemporal hyperspectral images," *IEEE Trans. Geosci. Remote Sens.* **53**, 244–260 (2015).
23. S. Marchesi, F. Bovolo, and L. Bruzzone, "A context-sensitive technique robust to registration noise for change detection in VHR multispectral images," *IEEE Trans. Image Process.* **19**, 1877–1889 (2010).
24. M. Gong et al., "Change detection in synthetic aperture radar images based on deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.* **27**, 125–138 (2016).

25. P. Zhang et al., "Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images," *ISPRS J. Photogramm. Remote Sens.* **116**, 24–41 (2016).
26. J.-S. Lee and E. Pottier, *Polarimetric Radar Imaging: From Basics to Applications*, CRC Press, Boca Raton, Florida (2009).
27. C. Chen et al., "Fuzzy restricted Boltzmann machine for the enhancement of deep learning," *IEEE Trans. Fuzzy Syst.* **23**, 2163–2173 (2015).
28. F.-J. Lin, Y.-C. Hung, and K.-C. Ruan, "An intelligent second-order sliding-mode control for an electric power steering system using a wavelet fuzzy neural network," *IEEE Trans. Fuzzy Syst.* **22**, 1598–1611 (2014).
29. N. Wang, M. J. Er, and M. Han, "Dynamic tanker steering control using generalized ellipsoidal-basis-function-based fuzzy neural networks," *IEEE Trans. Fuzzy Syst.* **23**, 1414–1427 (2015).
30. A. Rubio-Solis and G. Panoutsos, "Interval type-2 radial basis function neural network: a modeling framework," *IEEE Trans. Fuzzy Syst.* **23**, 457–473 (2015).
31. C.-F. Juang and Y.-W. Tsao, "A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning," *IEEE Trans. Fuzzy Syst.* **16**, 1411–1424 (2008).
32. A. Mohammadzadeh, O. Kaynak, and M. Teshnehlab, "Two-mode indirect adaptive control approach for the synchronization of uncertain chaotic systems by the use of a hierarchical interval type-2 fuzzy neural network," *IEEE Trans. Fuzzy Syst.* **22**, 1301–1312 (2014).
33. C.-J. Kim and D. Chwa, "Obstacle avoidance method for wheeled mobile robots using interval type-2 fuzzy neural network," *IEEE Trans. Fuzzy Syst.* **23**, 677–687 (2015).
34. C.-F. Juang and P.-H. Wang, "An interval type-2 neural fuzzy classifier learned through soft margin minimization and its human posture classification application," *IEEE Trans. Fuzzy Syst.* **23**, 1474–1487 (2015).
35. P. Y. Hao and J. H. Chiang, "Fuzzy regression analysis by support vector learning approach," *IEEE Trans. Fuzzy Syst.* **16**, 428–441 (2008).
36. L. Su et al., "Deep learning and mapping based ternary change detection for information unbalanced images," *Pattern Recognit.* **66**(C), 213–228 (2017).
37. L. Su et al., "Detecting multiple changes from multi-temporal images by using stacked denosing autoencoder based change vector analysis," in *Int. Joint Conf. on Neural Networks*, Vancouver, Canada, pp. 1269–1276 (2016).
38. P. Vincent et al., "Extracting and composing robust features with denoising autoencoders," in *Proc. of the Twenty-Fifth Int. Conf. on Machine Learning*, Helsinki, Finland, pp. 1096–1103 (2008).
39. J. Tang et al., "Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine," *IEEE Trans. Geosci. Remote Sens.* **53**, 1174–1185 (2015).
40. Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1798–1828 (2013).
41. J. G. Dijkman, H. Van Haeringen, and S. De Lange, "Fuzzy numbers," *J. Math. Anal. Appl.* **92**(2), 301–341 (1983).
42. J. J. Buckley, *Fuzzy Probabilities: New Approach and Applications*, Physica-Verlag, Heidelberg (2003).
43. M. Gong et al., "Fuzzy c-means clustering with local information and kernel metric for image segmentation," *IEEE Trans. Image Process.* **22**, 573–584 (2013).
44. J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.* **20**(1), 37–46 (1960).
45. Y. Bengio and S. Bengio, "Modeling high-dimensional discrete data with multi-layer neural networks," in *Proc. Advances in Neural Information Processing Systems*, Vol. 12, pp. 400–406 (2000).
46. C.-H. Chang, "Deep and shallow architecture of multilayer neural networks," *IEEE Trans. Neural Netw. Learn. Syst.* **26**, 2477–2486 (2015).

Linzhi Su is an assistant professor at Northwest University. He received his BS degree in artificial intelligence from Xidian University in 2011, and his PhD in remote sensing image processing from the Xidian University in 2016. His current research interests include image processing and intelligent system analysis.

Xin Cao is an assistant professor at Northwest University. He received his BS degree in detection, guidance, and control technology from Xidian University in 2011, and his PhD in pattern recognition and intelligent system from Xidian University in 2016. His current research interests include medical image processing, optical molecular imaging, and intelligent system analysis.