ISPRS Journal of Photogrammetry and Remote Sensing 129 (2017) 212–225

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

Feature learning and change feature classification based on deep learning for ternary change detection in SAR images

Maoguo Gong [*], Hailun Yang, Puzhao Zhang

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an, Shaanxi Province 710071, China

ARTICLE INFO

Article history:
Received 4 July 2016
Received in revised form 26 April 2017
Accepted 1 May 2017

Keywords:
Ternary change detection
Deep learning
Synthetic aperture radar
Representation learning
Sparse autoencoder
Convolutional neural networks

ABSTRACT

Ternary change detection aims to detect changes and group the changes into positive change and negative change. It is of great significance in the joint interpretation of spatial-temporal synthetic aperture radar images. In this study, sparse autoencoder, convolutional neural networks (CNN) and unsupervised clustering are combined to solve ternary change detection problem without any supervison. Firstly, sparse autoencoder is used to transform log-ratio difference image into a suitable feature space for extracting key changes and suppressing outliers and noise. And then the learned features are clustered into three classes, which are taken as the pseudo labels for training a CNN model as change feature classifier. The reliable training samples for CNN are selected from the feature maps learned by sparse autoencoder with certain selection rules. Having training samples and the corresponding pseudo labels, the CNN model can be trained by using back propagation with stochastic gradient descent. During its training procedure, CNN is driven to learn the concept of change, and more powerful model is established to distinguish different types of changes. Unlike the traditional methods, the proposed framework integrates the merits of sparse autoencoder and CNN to learn more robust difference representations and the concept of change for ternary change detection. Experimental results on real datasets validate the effectiveness and superiority of the proposed framework.

1. Introduction

Image change detection aims to highlight changed regions by analyzing two or more images over the same area acquired at different times. As is known, change detection has been widely used in urban change analysis (Bouziani et al., 2010; Ban and Yousif, 2012; Liu et al., 2014; Gong et al., 2016a), environmental monitoring (Mas, 1999; Jin et al., 2013), hazard assessment (Giustarini et al., 2013; Yang et al., 2014; Li et al., 2014; Brunner et al., 2010). In recent years, synthetic aperture radar (SAR) images gradually play an important role in the field of remote sensing and attract more and more attentions (Bazi et al., 2005, 2006; Gong et al., 2014; Parrilli et al., 2012; Yu et al., 2012; Jia et al., 2015; Hou et al., 2014; Li et al., 2016). This is mainly because of its independence on atmospheric and weather conditions. However, due to the fact that the imaging system of SAR is based on the coherent principle, the gray value of adjacent pixels would generate some random changes when signals are returned. This process would bring speckle noise, which makes it more difficult to detect meaningful changes.

In the literatures, many methods have been proposed for SAR images change detection (Bazi et al., 2005, 2006; Krinidis and Chatzis, 2010; Gong et al., 2014; Wu et al., 2014). Among them, threshold methods and clustering methods are the most popular two. In threshold methods, it is desired to find an optimal threshold used to classify every pixel into changed or unchanged categories properly, such as Otsu, the Kittler and Illingworth minimum-error thresholding algorithm (KI) and the expectation maximization algorithm (EM). On the other hand, the core idea of clustering methods is to maximize the similarity between data points belong to the same cluster and minimize the similarity across different clusters. Fuzzy c-means (FCM) (Bezdek, 2013) is one of the most popular clustering algorithms, and it retains more image information than the hard clustering methods by introducing the fuzziness relation into data distribution and partition. In one given image, every pixel is not independent in fact, and spatial relationship is an important priori. However, FCM does not consider the spatial priori among pixels, and therefore it is sensitive to noise. Later, spatial priori is introduced into some classical

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

http://dx.doi.org/10.1016/j.isprsjprs.2017.05.001
0924-2716/© 2017 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.

clustering methods to overcome this drawback (Krinidis and Chatzis, 2010; Gong et al., 2013).

SAR is a typical active imaging system which actively launches the microwave signals and forms the images by receiving the reflected signals. By using the pulse compression technique and the principle of synthetic aperture, SAR image possesses higher range resolution and azimuth resolution, and therefore, large format of remote sensing images can be acquired. Most of SAR images change detection methods focus on classifying the difference image into two parts, changed and unchanged ones (Bazi et al., 2005; Gong et al., 2014; Jia et al., 2015; Lu et al., 2015). However, as remote sensing techniques develop, it is not very appropriate for all change detection problems only to identify whether each pixel changes or not, due to the fact that it is too limited for practical applications. In most cases, different objects have different backscattering values in SAR images. And if one object changes, then the corresponding backscattering values would either increase or decrease between bi-temporal SAR images. Therefore, it is more meaningful to further group the changed areas into positive change and negative change in these cases. For example, in urban change analysis, some old buildings are dismantled, while some houses will be built in some other areas, it is necessary to distinguish these two different types of changes from bi-temporal SAR images. For these two completely different types of changes, we want to distinguish which type of change this areas belong to.

As far as we know, there has been a few articles about SAR ternary change detection. In Bazi et al. (2006), Bazi et al. proposed a Double-Thresholding KI algorithm, which classify difference image into three groups by automatically estimating two thresholds. In 2012, Bovolo et al. (2012) presented a framework to detect multiple changes from bi-temporal images, called as the compressed CVA (C^2VA). Lately, Su et al. (2017) proposed a deep learning and mapping framework oriented to the ternary change detection task with information unbalanced images. In this framework, different mapping networks are established to learn the relationship between bi-temporal features of each type of changes. Traditional methods such as KI and FCM also have been considered to solve the SAR ternary change detection problems, but the results are not very satisfactory. The reason may lie in the following two points: (1) SAR images are easy to be affected by speckle noise, and the traditional methods are unable to deal with speckle noise well. (2) The raw pixel and its neighborhood information are still too limited in representation ability. In recent years, deep learning has attracted increasing attentions due to its powerful representation learning ability. As a flexible feature learning tool, deep learning has the ability to transform the raw pixels into a suitable feature space to capture the key information for discrimination and suppress meaningless variations caused by environment.

Deep learning is an important branch of machine learning, and it tries to learn abstract concepts by simulating the cognitive mechanism of human brain and explore the latent pattern by establishing deep architecture (Arel et al., 2010). When data is fed into a deep network, the features can be learned layer by layer, and the output of one layer can be taken as the input of the next layer (Bengio et al., 2013). The deeper the network is, the more abstract features can be learned by the deep network. As is well known, deep learning has many classical models, such as CNN (Lawrence et al., 1997), deep belief networks (DBN) (Hinton et al., 2006), sparse autoencoder (SAE) (Zhang et al., 2015). CNN is inspired by the receptive fields in neural cortex, and it is a multilayer neural network suitable for processing 2-D data such as videos and images. DBN was proposed by Hinton in 2006, and it is composed of many layers of restricted boltzmann machines (RBM). The emergence of DBN addresses the gradient diffusion problems that one often encounters in applying backpropagation algorithm to deep neural networks (Arel et al., 2010). Autoencoder is another unsupervised model, and it tries to learn the features by minimizing the reconstruction error. SAE is a variant of the standard autoencoder, and it is generated by suppressing activation rate of hidden units. For their good performance in representation learning, these deep models have been widely applied to speech recognition (Hinton et al., 2012; Dahl et al., 2012), medical diagnosis (Shin et al., 2013), image classification (Zhang et al., 2015; Yuan et al., 2015). In recent years, it receives an enormous amount of attentions (Toshev and Szegedy, 2014; Liu et al., 2016; Goh et al., 2014; Hou et al., 2015; Gong et al., 2015; Shao et al., 2014; Lei et al., 2014; Glorot et al., 2011; Tang et al., 2015; Chen et al., 2015; Gong et al., 2016b; Zhang et al., 2016b; Zhang et al., 2016a). In Tang et al. (2015), Tang et al. used wavelet coefficients extracted from compressed domain combined with deep neural networks and extreme learning machine to build an architecture, which can detect the ship on spaceborne optical image. Based on DBN, Chen et al. (2015) proposed an novel architecture, which combines spectral-spatial feature extraction and classification together to obtain an accurate classification result for hyperspectral images. In Gong et al. (2016b), Gong et al. applied deep neural networks to SAR images change detection without the generation of difference image, in which change detection problem is transformed into a classical classification task.

In this paper, we propose a ternary change detection framework which combines SAE, CNN and unsupervised clustering for detecting changes from bi-temporal SAR images. In this framework, we firstly extract the features from the difference image via SAE, and these difference features are clustered into three classes as the pseudo labels for training CNN. Then, a certain number of samples are selected from the feature maps and taken as the input of CNN. Having training samples and the corresponding pseudo labels, the concept of changes can be learned by training a CNN model as change feature classifier. During its training, CNN is driven to learn more robust different representations for better distinguishing different types of changes. After training, all testing samples from the feature maps are fed into the learned CNN, and the final ternary change detection results can be obtained.

The rest of this paper is organized as follows: Section 2 makes a brief introduction to the related background knowledge of this study. Section 3 illustrates the proposed approach in details. The datasets and evaluation criteria are presented in Section 4. Experimental results and the corresponding analysis are shown in Section 5. Section 6 makes a conclusive remark on this study.

2. Background knowledge

2.1. Problem statements

Given two coregistered SAR images $\mathbf{X_1}$ and $\mathbf{X_2}$ of size $W \times H$, where W and H are the width and height of them, and they are acquired over the same area at different times. The difference image (**DI**) is computed by using the log-ratio operator shown in Eq. (1).

$$DI(i,j) = \log\left[\frac{X_2(i,j)}{X_1(i,j)}\right] \tag{1}$$

where $X_1(i,j)$ and $X_2(i,j)$ represent the pixels in the image $\mathbf{X_1}$ and $\mathbf{X_2}$ at position (i,j), respectively. $DI(i,j)$ is the pixel in **DI** at position (i,j).

In most literatures, **DI** is classified into two parts, changed and unchanged ones. However, in most cases, SAR images have either increased or decreased backscattering values in different times. Therefore, it makes sense to further classify the changed region into positive change and negative change. In other words, the final result should contain three classes, namely unchanged pixels, negative change and positive change. A simple but intuitional example for ternary change detection is shown in Fig. 1, in which the triangular objects vanish while rounded one appears, and they should be grouped into different types of changes.

2.2. Introduction to SAE

Autoencoder is a classical three-layer neural networks which can learn useful features by minimizing the reconstruction error between the input data and its reconstruction (Zhang et al., 2015), and its structure is shown in Fig. 2. Autoencoder is composed of an input layer $\mathbf{i} = \{i_1, i_2, \ldots, i_n\}^{\mathrm{T}}$, a hidden layer $\mathbf{a} = \{a_1, a_2, \ldots, a_m\}^{\mathrm{T}}$, and an output layer $\mathbf{o} = \{o_1, o_2, \ldots, o_n\}^{\mathrm{T}}$, where n and m are the number of visible and hidden units, respectively. Autoencoder consists of an encoder and a decoder. During the encoding step, the input data is mapped into a hidden representation with a linear mapping and nonlinear activation function as shown in Eq. (2).

$$\mathbf{a} = \mathrm{sigm}(\boldsymbol{\omega}_1 \mathbf{i} + \mathbf{ib}) \tag{2}$$

where $\boldsymbol{\omega}_1 = \{\omega_{1lj} | 1 \leqslant l \leqslant m, 1 \leqslant j \leqslant n\}$ is the weight matrix for encoding. $\mathbf{ib} = \{ib_1, ib_2, \ldots, ib_m\}^{\mathrm{T}}$ is the bias for encoding, and sigm represents logistic sigmoid function $\mathrm{sigm}(x) = 1/[1 + \exp(-x)]$. In the decoding step, the hidden representation is mapped into the reconstructed output with an inverse mapping as shown in Eq. (3).

$$\mathbf{o} = \mathrm{sigm}(\boldsymbol{\omega}_2 \mathbf{a} + \mathbf{ob}) \tag{3}$$

where $\boldsymbol{\omega}_2 = \{\omega_{2lj} | 1 \leqslant l \leqslant n, 1 \leqslant j \leqslant m\}$ is the weight matrix for decoding, and $\mathbf{ob} = \{ob_1, ob_2, \ldots, ob_n\}^{\mathrm{T}}$ is the bias for decoding.

As shown in Eq. (4), the cost function of autoencoder consists of average sum-of-squares error term and regularization term.

$$J(\boldsymbol{\omega}, \mathbf{ib}, \mathbf{ob}) = \frac{1}{2s} \sum_{k=1}^{s} \left\| \mathbf{i}^k - \mathbf{o}^k \right\|^2 + \frac{\lambda}{2} \|\boldsymbol{\omega}\|^2 \tag{4}$$

where s is the number of input samples. $\mathbf{i}^k$ is the k-th input sample, and $\mathbf{o}^k$ is the reconstructed output of k-th sample. λ is the parameter of weight decay, and it is used to control the relative importance between reconstruction error and regularization term. $\boldsymbol{\omega}$ is the weight matrix which includes the encoding matrix $\boldsymbol{\omega}_1$ and decoding matrix $\boldsymbol{\omega}_2$.

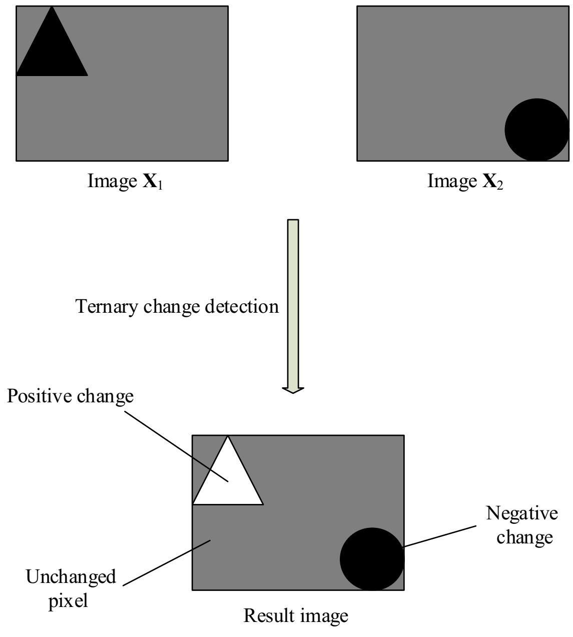


Fig. 1. An example of ternary change detection.

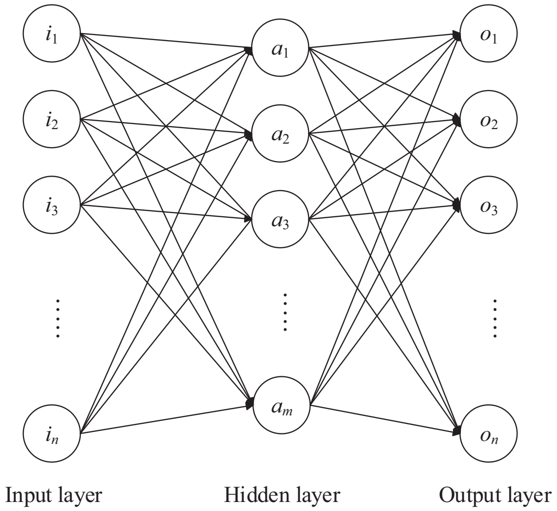


Fig. 2. Structure of SAE.

The core of SAE is to add a sparsity constraint on the activation of hidden layer in autoencoder, which makes most hidden units inactive. The cost function of SAE is shown in Eq. (5).

$$J_{SAE}(\boldsymbol{\omega}, \mathbf{ib}, \mathbf{ob}) = J(\boldsymbol{\omega}, \mathbf{ib}, \mathbf{ob}) + \beta \sum_{j=1}^{m} KL(\xi || \hat{\xi}_j) \tag{5}$$

where β is the weight of the sparsity penalty, and m is the number of the hidden units. $KL(x)$ is the Kullback Leibler divergence, whose expression is Eq. (6). ξ represents sparsity parameter, and $\hat{\xi}_j = \frac{1}{s}\sum_{k=1}^{s} a_j^k$ represents the average activation of the j-th hidden units. When $\hat{\xi}_j = \xi, KL(\xi || \hat{\xi}_j)$ achieves the minimum, and its value is 0. When $\hat{\xi}_j$ is close to 0 or $1, KL(\xi || \hat{\xi}_j)$ will be large.

$$KL(\xi || \hat{\xi}_j) = \xi \log\left(\frac{\xi}{\hat{\xi}_j}\right) + (1 - \xi) \log\left(\frac{1 - \xi}{1 - \hat{\xi}_j}\right) \tag{6}$$

SAE is trained by minimizing the cost function Eq. (5) with respect to $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$, $\mathbf{ib}, \mathbf{ob}$. The activation of hidden layer $\mathbf{a}$ is the desired features.

2.3. Introduction to CNN

CNN is a multilayer neural network which is mainly designed for 2-D data (such as videos and images). As a deep learning framework, CNN needn't any data preprocessing requirements, and it achieves some degree of shift and deformation invariance by introducing local receptive field, weights of shared and pooling together. The use of shared weights also greatly reduce the number of parameters to be optimized.

A typical structure of CNN is shown in Fig. 3. The network consists of several layers, and each of which contains multiple planes. In CNN, small portions of the images (called a local receptive field) enter at the input layer, and the weights of the receptive field for a plane are forced to be equal at all points in the plane (Lawrence et al., 1997). Each plane can be seen as a feature map which has a fixed feature detector that is convoluted with a local window which is scanned over the plane in the previous layer. In order to detect multiple features, many planes are used in each layer, and we call them convolutional layers.

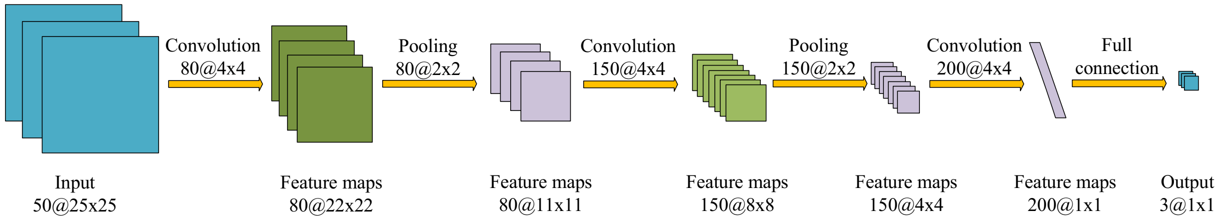


Fig. 3. Structure of CNN.

After features are extracted by the convolutional layer, the dimension of features is always high. The high dimension makes the features compute very inconvenient, and it may lead to over-fitting as well. In order to improve computing efficiency and avoid over-fitting, it is standard choice to use pooling to reduce the dimension of features. In the pooling process, the average or the maximum of the region which has size of $v \times v$ is used to replace the features of this region on the input feature map, in turn, and this could greatly reduce the dimension of feature maps. The pooling process is performed as follows:

$$y_j^{l+1} = \text{down}\left(y_j^l\right) \tag{7}$$

where y_j^{l+1} is the j-th feature map after pooling. y_j^l denotes the j-th input feature map, and the operator 'down' represents pooling function.

Finally, the features extracted by multilayer convolution and pooling are fed into the softmax layer for obtaining the final ternary change detection result.

3. The proposed framework

In this section, the proposed method is illustrated in detail. Firstly, the general flowchart of the proposed framework is shown in Fig. 4. Given two coregistered SAR images, we can get the **DI** by using the log-ratio operator shown in Eq. (1). Then, the change features are extracted by SAE from **DI**. For the features of each pixel, **DI** is classified into three categories as the pseudo labels for training CNN. These three classes include unchanged category, positive change and negative change. Then, selected patches from the feature maps are taken as the training data for CNN. Having the input data and the corresponding pseudo labels, a CNN model can be driven to learn the concept of changes and a power change feature classification model can be established. At the testing stage, all samples from the feature maps are fed into the learned CNN, and the final ternary change detection results can be acquired.

3.1. SAE for feature learning

Unlike the traditional methods such as KI and FCM, in which they only consider pixel values and the neighborhood information of the difference image for change detection, we desire to learn the relationship between neighborhood pixels and establish a robust representation. On the other hand, it also can reduce the impact of noise by transforming the difference image into a suitable feature space. Due to its powerful ability in learning abstract representation from data in their raw form, SAE is a good choice for unsupervised feature learning from difference map. For the properties of pixel are largely associated with the neighborhood (Krinidis and Chatzis, 2010; Gong et al., 2013), so we use the pixel with its neighborhood pixels as the input sample. Fig. 5 shows feature learning process. In Fig. 5, $\mathbf{D}_{ij}$ represents a center at position (i, j) with its neighborhood of size $l \times l$ in **DI**. The size $l \times l$ may influence the classification result, and the analysis of this parameter is made next. Convert $\mathbf{D}_{ij}$ to vector $\mathbf{V}_{ij}$, and put this vector to the input layer of SAE. Then in the encoding step, the vector $\mathbf{V}_{ij}$ is transformed into hidden representation $\mathbf{F}_{ij}$, and its expression is as follows:

$$\mathbf{F}_{ij} = \text{sigm}(\omega_1 \mathbf{V}_{ij} + \mathbf{ib}) \tag{8}$$

where ω_1 represents the weight matrix for encoding, and **ib** is the bias vector for encoding.

Then, as Eq. (9) shown, the hidden representation $\mathbf{F}_{ij}$ is processed by an inverse mapping to a reconstructed $\mathbf{V}'_{ij}$.

$$\mathbf{V}'_{ij} = \text{sigm}(\omega_2 \mathbf{F}_{ij} + \mathbf{ob}) \tag{9}$$

where ω_2 represents the weight matrix for decoding, and **ob** is the bias vector for decoding. By minimizing the cost function of SAE

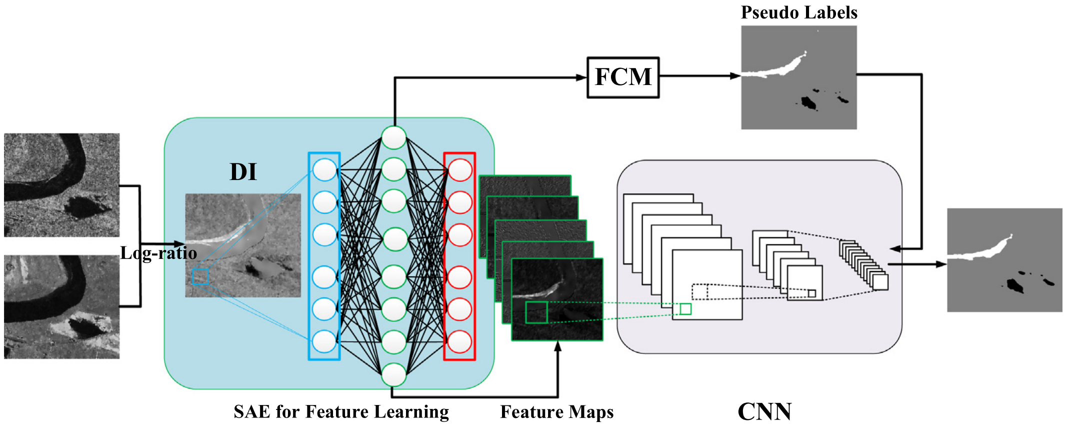


Fig. 4. Illustration on the proposed framework. Firstly, **DI** is computed by the log-ratio operator. Then, the features of **DI** are extracted by SAE. According to the features, **DI** is clustered into three classes as the pseudo labels for training CNN. Then, some patches selected from the feature maps are taken as the training data for CNN. After training, all samples from the feature maps are fed into the learned CNN, and the final ternary change detection results can be obtained.

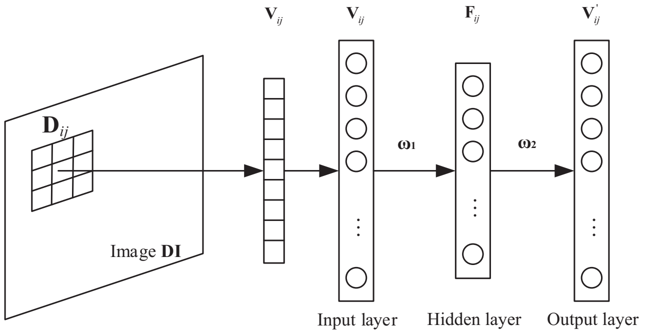


Fig. 5. The process of unsupervised feature learning.

between $\mathbf{V}_{ij}$ and $\mathbf{V}'_{ij}$, the data features $\mathbf{F}_{ij}$ can be extracted from the hidden layer.

3.2. Feature classification based on CNN

Having the features learned by SAE, the next step is to establish a set of pseudo labels by analyzing the features. Clustering method is a widely used classification algorithm, which groups similar structure into the same category. As is known, similar samples in **DI** will have similar change features. Therefore, clustering method is adopted to classify the image **DI** into three categories with the features of each pixel. In general, the performance of FCM is better than traditional hard clustering methods such as K-means in SAR change detection task. Fuzzy membership matrix is introduced to update the clustering centers so that it can maximize the intra-cluster similarity and minimize the inter-cluster similarity at the same time, which would be helpful to retain more image details and improve the accuracy of change detection. Furthermore, due to the fact that the imaging mechanism of SAR results in the adherence of serious speckle noise points in the images, which is difficult for conventional hard clustering methods to accurately obtain clustering centers and assign a suitable label for each pixel. Therefore, it is appropriate to adopt FCM to weaken this issue owing to its robustness to isolate noise point. In the proposed framework, FCM is used to establish a set of pseudo labels for successive processing by analyzing **DI**.

In order to improve the accuracy of change feature classification, the preclassification result (i.e., pseudo labels) is used to train a more power change feature classifier. In the proposed framework, CNN is selected as the change feature classifier due to its power ability in invariance feature learning. Under the guidance of the pseudo labels from the preclassification map, CNN is driven to learn higher-level and more abstract features from the SAE feature maps to re-classify all samples into three desired categories. A simple rule that the minority is subordinate to the majority is used to select reliable training samples in the proposed framework (Liu et al., 2016). Whether a pixel changes or not is closely related to whether the neighborhood pixels change or not. Suppose that $C(i,j)$ represents the pixel in the preclassification map at position (i,j), and its label is class β. N_{ij} represents a center at position (i,j) with its neighborhood of size $q \times q$. When most pixels in N_{ij} have the same labels with $C(i,j)$, the pixel $C(i,j)$ should be selected as the reliable samples of label β, such as Point 1 and Point 2 in Fig. 6. When a few or none of the pixels in N_{ij} have the same labels with $C(i,j)$, the pixel $C(i,j)$ is more likely to be wrongly classified, such as Point 3 in Fig. 6, and these points should be excluded from training samples. Mathematically, if a pixel (i,j) satisfies the condition in Eq. (10), the pixel can be taken as the reliable samples of class β.

$$\frac{M_{mn}[(m,n) \in N_{ij} \bigcap C(m,n) = \beta]}{q \times q} \geqslant \tau \qquad (10)$$

where $M_{mn}(x)$ is the number of pixels meeting the condition x. N_{ij} is the neighborhood of pixel (i,j). $C(m,n)$ represents the pixel at position (m,n), and it is in the neighborhood N_{ij}. τ is the parameter which is defined to control the condition of selecting samples. Considering the effect of noise, we set $0.5 \leqslant \tau < 1$ instead of 1.

Let S_{ij} represents a training sample which consists of the center pixel at position (i,j) and its neighborhood in the SAE feature maps. When S_{ij} is fed into CNN, the features will be learned by using the convolution kernels to convolute S_{ij}, and the expression is shown in follows:

$$CF_m = f(S_{ij} * k_m + b_m) \qquad (11)$$

where CF_m denotes the m-th change feature map in the first convolution layer. k_m is the m-th convolutional kernel which can be obtained by the back-propagation algorithm, and $*$ denotes convolution. b_m represents the m-th bias in the first convolution layer.

Because the dimension of CF_m is still high, the next step is to reduce the dimension with pooling, as is shown in Eq. (12).

$$SF_m = \mathrm{down}(CF_m) \qquad (12)$$

where SF_m is the m-th change feature map after pooling.

After layer-wise convolution and pooling, the features are fed to a softmax classifier finally, and we can get the classification error in CNN. And the CNN can be trained by the back propagation of classification error. Having CNN change feature classifier, all samples from the feature maps are fed into the learned CNN, and the final change detection map can be acquired directly.

4. Experimental settings

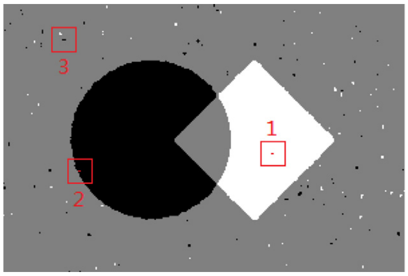
In this section, we firstly make a brief introduction to the datasets used in the experiments. And then the compared algorithms and evaluation criteria also are presented. What's more, the parameter setting used in our experiment is given.

4.1. Datasets

Fig. 7 shows the first dataset used in our experiments, and it is a simulated dataset of size 440 × 700. This dataset consists of five image pairs with different equivalent number of looks (ENL), and ENL = 1, 2, 3, 4, and 5, respectively. ENL denotes the noise level in the given images, and the lower of ENL value means the higher of the noise level. This dataset is used to analyze the influence of noise on the change detection accuracy.

The second and third datasets used in our experiments are two groups of SAR images acquired by the ESAERS-2 satellite over the city of San Francisco, in August 2003 and May 2004, respectively. The original images are so large that it is not easy to handle, therefore, two typical regions are selected to demonstrate the effectiveness of the proposed framework. The images shown in Fig. 8 is of size 390 × 370, and with the same spatial resolution 26 m, and it is named as San Francisco-1. The other dataset of San Francisco is shown in Fig. 9, whose size is 390 × 390, and it is named as San Francisco-2. The reference images in first and second datasets are ploted by integrating prior information with photo interpretation.

The forth dataset, the Yellow River dataset, was acquired by Radarsat-2 over the region of the Yellow River Estuary in China in June 2008 and June 2009, respectively. In Fig. 10, (a) and (b) are of the same pixel resolution 290 × 260, and with the same spatial resolution 8 m. From this dataset, it is observed that the image acquired in 2009 is seriously influenced by speckle noise. The bi-temporal images and the corresponding reference image, which



(a)

P	**P**	**P**	**P**	**P**
P	**P**	**P**	**P**	**P**
P	**P**	**P**	**P**	**P**
P	**P**	**P**	**P**	**P**
P	**P**	**P**	**P**	**P**

(b)

U	**N**	**N**	**N**	**N**
U	**U**	**N**	**N**	**N**
U	**U**	**N**	**N**	**N**
U	**U**	**U**	**N**	**N**
U	**U**	**U**	**N**	**N**

(c)

U	**U**	**U**	**U**	**U**
U	**P**	**P**	**U**	**U**
U	**P**	**N**	**N**	**U**
U	**U**	**U**	**U**	**U**
U	**U**	**U**	**U**	**U**

(d)

Fig. 6. Example to show how to select samples for fine tuning. P, U and N respectively represent positive changed, unchanged and negative change. (a) Simulated map. (b) Point 1. (c) Point 2. (d) Point 3.

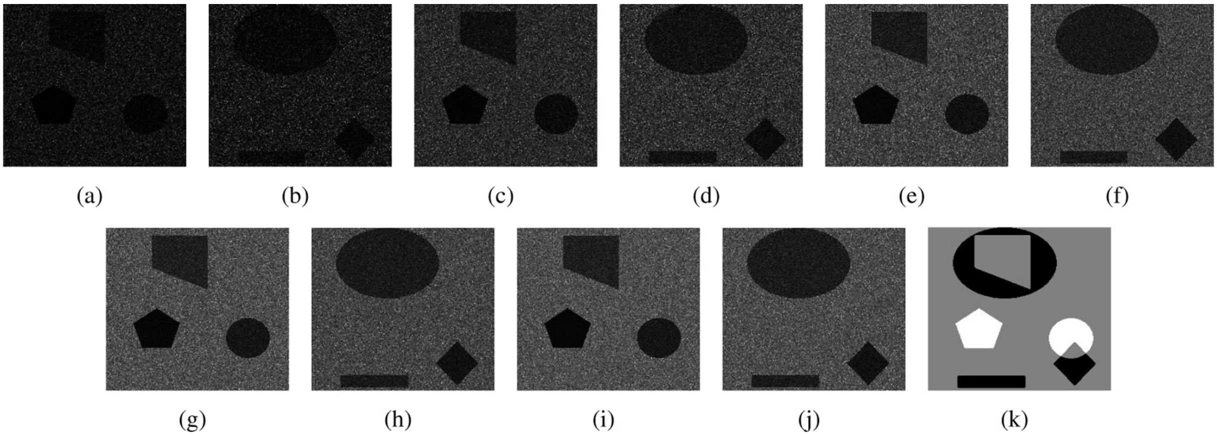


Fig. 7. The simulated dataset with five different ENLs. (a) and (b) are the images with ENL = 1. (c) and (d) are the images with ENL = 2. (e) and (f) are the images with ENL = 3. (g) and (h) are the images with ENL = 4. (i) and (j) are the images with ENL = 5. (k) is the reference image.

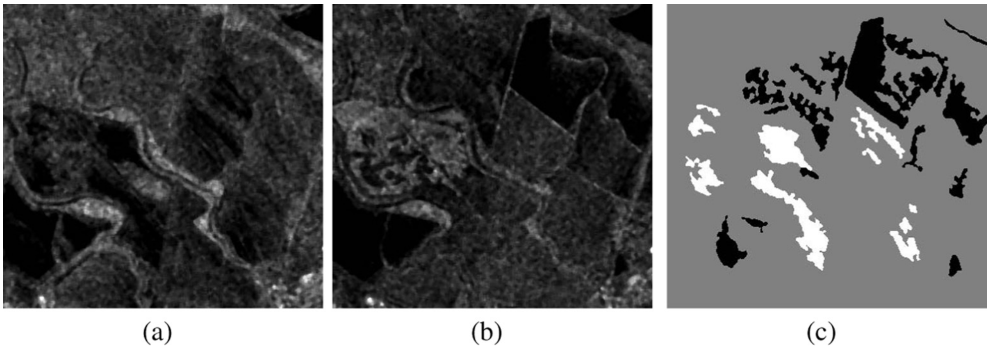


Fig. 8. San Francisco-1 dataset. (a) Image acquired in August 2003. (b) Image acquired in May 2004. (c) Reference image.

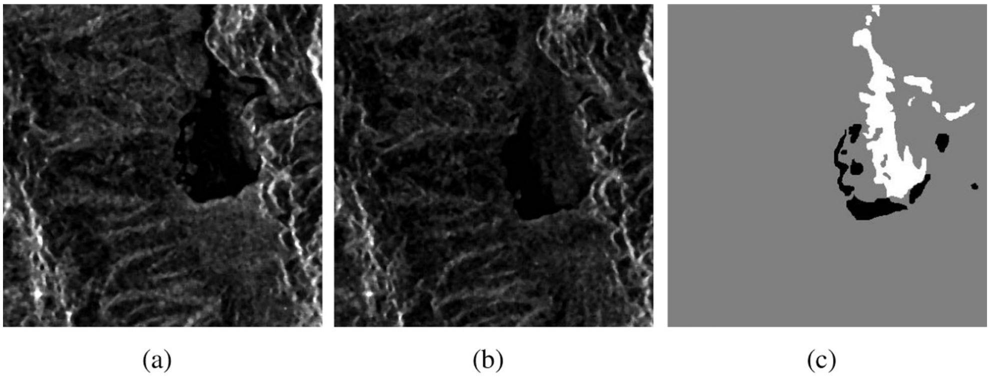


Fig. 9. San Francisco-2 dataset. (a) Image acquired in August 2003. (b) Image acquired in May 2004. (c) Reference image.

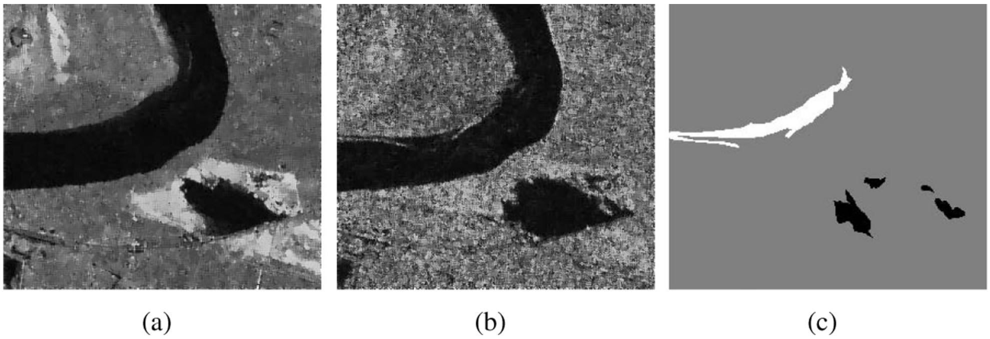


Fig. 10. Yellow River dataset. (a) Image acquired in June 2008. (b) Image acquired in June 2009. (c) Reference image.

is obtained by integrating prior information with photo interpretation, are shown in Fig. 10.

The Xi'an dataset is a section of size 220 × 200 from two SAR images over the city of Xi'an in Shaanxi Province, acquired by TerraSAR in February 2013 and March 2015, and its spatial resolution is 3.5 m. The reference image is created by integrating prior information with photo interpretation based on the bi-temporal images in Fig. 11(a) and (b).

4.2. Parameter testing and compared algorithms

In the proposed approach, there are two important parameters which have a great influence on the accuracy of change detection, namely the sliding window size for SAE training, and parameter τ that controls the condition of sample selection. These two parameters are tested to explore their influence on the accuracy. The testing on parameters will provide a better guidance for selecting appropriate parameters to improve the performance of the proposed method.

In order to demonstrate the effectiveness of the proposed approach, we compare our method with four traditional methods. The first compared algorithm is Double-Thresholding KI algorithm (DTKI) (Bazi et al., 2006), which is a modified KI algorithm designed for ternary change detection. The second compared algorithm is FCM. Fuzzy local information c-means algorithm (FLICM) (Krinidis and Chatzis, 2010) is taken as the third compared method,

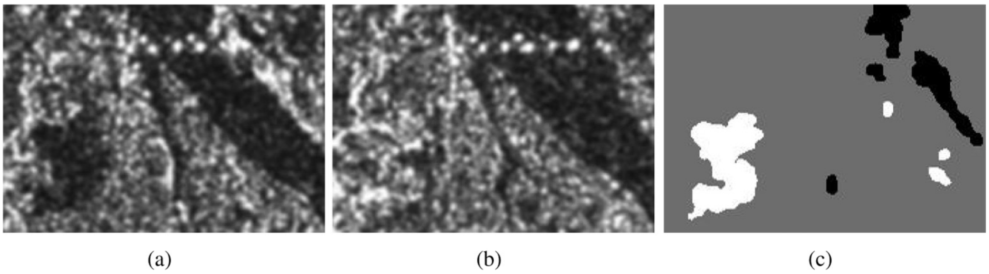


Fig. 11. Xi'an dataset. (a) Image acquired in February 2013. (b) Image acquired in March 2015. (c) Reference image.

Table 1
Confusion matrix.

Confusion matrix		Result image		
		NC	UC	PC
Reference image	NC	T_{NN}	F_{NU}	F_{NP}
	UC	F_{UN}	T_{UU}	F_{UP}
	PC	F_{PN}	F_{PU}	T_{PP}

while kernel weighted FLICM algorithm (KWFLICM) (Gong et al., 2013) is taken as the forth. The difference between KWFLICM and FLICM is that KWFLICM introduces a tradeoff weighted fuzzy factor and a kernel metric based on FLICM. These experiments are implemented to demonstrate the superiority of the proposed approach based on deep learning over the traditional methods.

4.3. Evaluation criteria

Having change detection results, it is necessary to use some criteria to evaluate the results quantitatively. In our experiments, four evaluation criteria are used to quantitatively evaluate the proposed framework, such as the overall error (OE), the percentage correct classification (PCC), kappa coefficient (KC) and normalized mutual information (NMI). In reference ground truth map, we first count the total number N of all pixels. Then, the results of each method are compared with the reference map to obtain quantitative statistics. According to Table 1, the pixels are classified into different classes, and they are put into a confusion matrix $\mathbf{D}$, which is shown in Eq. (13). Therefore, the overall error (OE) is calculated as Eq. (14).

$$\mathbf{D} = \begin{bmatrix} T_{NN} & F_{NU} & F_{NP} \\ F_{UN} & T_{UU} & F_{UP} \\ F_{PN} & F_{PU} & T_{PP} \end{bmatrix} \tag{13}$$

$$OE = N - T_{NN} - T_{UU} - T_{PP} = N - \mathrm{tr}(\mathbf{D}) \tag{14}$$

To evaluate the overall performance of different methods, the percentage correct classification (PCC) can be defined as follows:

$$PCC = \frac{T_{NN} + T_{UU} + T_{PP}}{N} = \frac{\mathrm{tr}(\mathbf{D})}{N} \tag{15}$$

What's more, kappa coefficient (KC) is another metric used to evaluate the classification accuracy (Cohen, 1960). The higher the value of KC, the better the classification result. KC is calculated as follows:

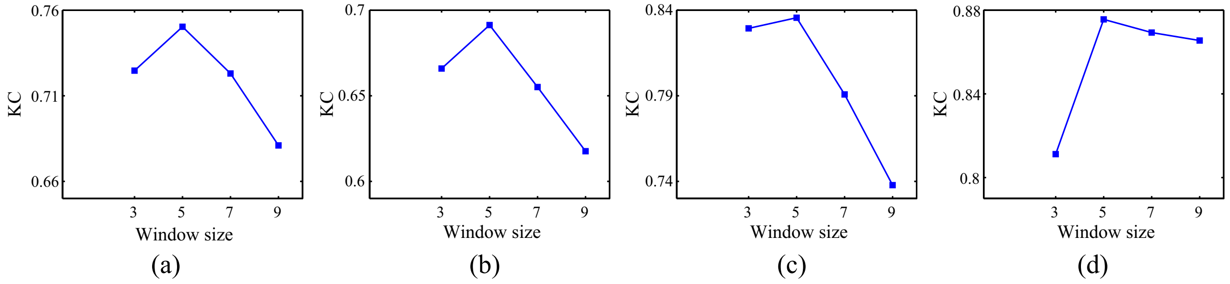


Fig. 12. Relationship between change detection results and the window size of SAE samples. (a) San Francisco-1 dataset. (b) San Francisco-2 dataset. (c) Yellow River dataset. (d) Xi'an dataset.

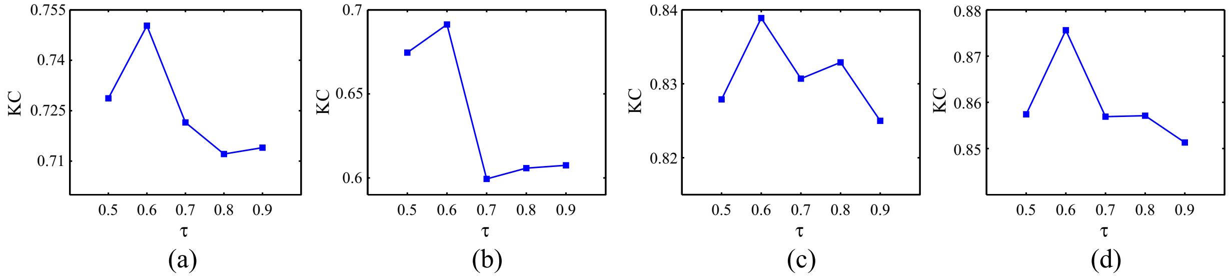


Fig. 13. Relationship between change detection results and parameter τ. (a) San Francisco-1 dataset. (b) San Francisco-2 dataset. (c) Yellow River dataset. (d) Xi'an dataset.

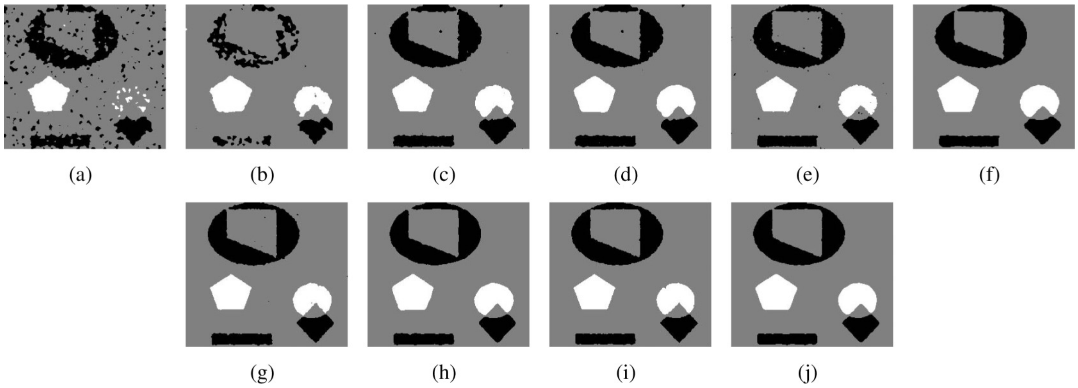


Fig. 14. Ternary change detection results of the simulated dataset. (a) The best result of the four traditional algorithms obtained by FLICM on the images with ENL = 1. (b) The result of the proposed method on the images with ENL = 1. (c) The best result of the four traditional algorithms obtained by FLICM on the images with ENL = 2. (d) The result of the proposed method on the images with ENL = 2. (e) The best result of the four traditional algorithms obtained by KWFLICM on the images with ENL = 3. (f) The result of the proposed method on the images with ENL = 3. (g) The best result of the four traditional algorithms obtained by KWFLICM on the images with ENL = 4. (h) The result of the proposed method on the images with ENL = 4. (i) The best result of the four traditional algorithms obtained by KWFLICM on the images with ENL = 5. (j) The result of the proposed method on the images with ENL = 5.

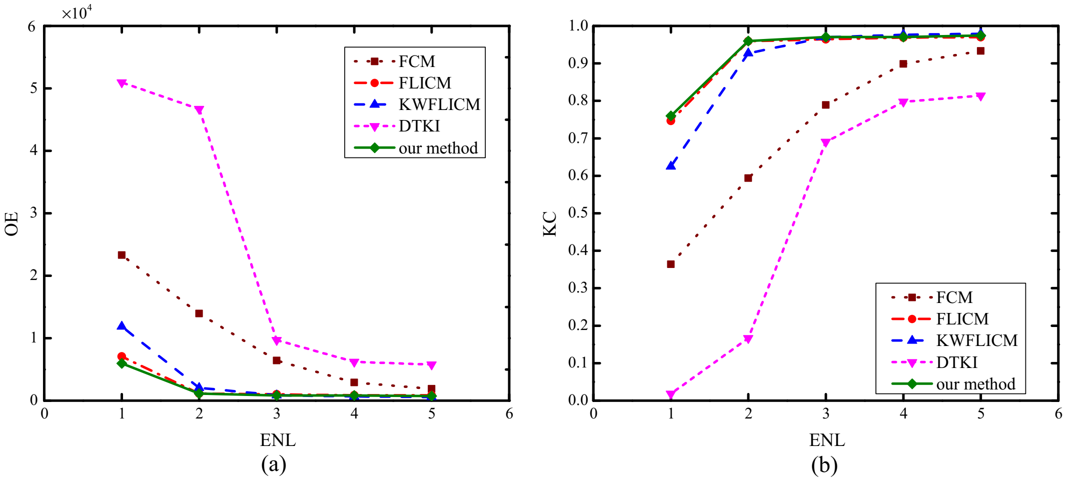
$$KC = \frac{N \cdot \mathrm{tr}(\mathbf{D}) - \mathbf{t}\mathbf{D}^2\mathbf{t}^{\mathrm{T}}}{N - \mathbf{t}\mathbf{D}^2\mathbf{t}^{\mathrm{T}}} \tag{16}$$

where $\mathbf{t} = [1, 1, 1]$. The derivation of the *KC* is described in Appendix A.

Normalized mutual information (*NMI*) can measure the agreement of experimental result and reference result (Wang et al., 2014). The higher the value of *NMI*, the more similar of two maps. *NMI* is calculated as follows:

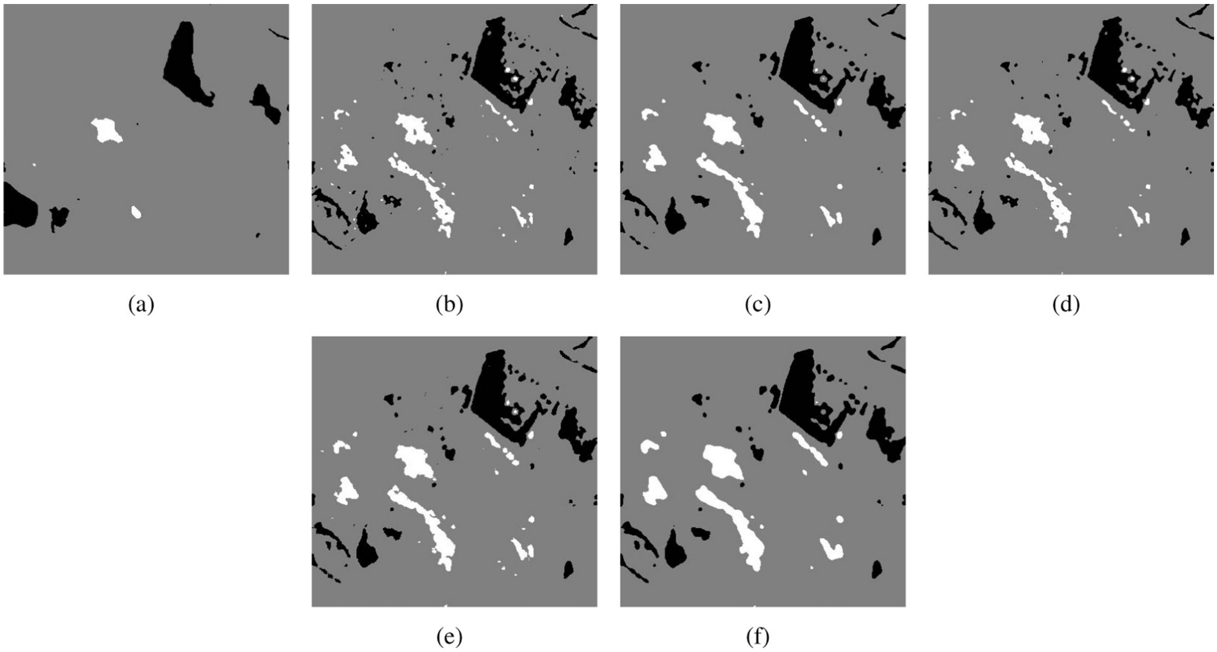
$$NMI = \frac{\sum_{r=1}^{3}\sum_{p=1}^{3} n_{rp} \cdot \log\left(\frac{N \cdot n_{rp}}{n_r \cdot n_p}\right)}{\sqrt{\left(\sum_{r=1}^{3} n_r \cdot \log\left(\frac{n_r}{N}\right)\right) \cdot \left(\sum_{p=1}^{3} n_p \cdot \log\left(\frac{n_p}{N}\right)\right)}} \tag{17}$$

where n_r is the number of pixels which belong to the r-th class in reference image. n_p is the number of pixels which belong to the



(a) (b)

Fig. 15. Relationship between ternary change detection results and ENL. (a) The value of OE. (b) The value of KC.



(a) (b) (c) (d) (e) (f)

Fig. 16. Ternary change detection results of the San Francisco-1 dataset. (a) DTKI. (b) FCM. (c) FLICM. (d) KWFLICM. (e) SAE + FCM. (f) SAE + FCM + CNN.

Table 2
Quantitative comparison among different methods on the San Francisco-1 dataset.

	OE	*PCC*	*KC*	*NMI*
DTKI	19,677	0.8636	0.3348	0.2033
FCM	12,971	0.9101	0.6169	0.4290
FLICM	10,887	0.9242	0.6852	0.5078
KWFLICM	12,504	0.9133	0.6399	0.4475
SAE + FCM	10,528	0.9270	0.7047	0.5169
SAE + FCM + CNN	**9624**	**0.9333**	**0.7503**	**0.5438**

p-th class in our result. n_{rp} is the number of pixels which belong to the r-th class in reference image and belong to the p-th class in result map at the same position.

4.4. Parameters setting

Except for the window size of SAE samples and the parameter τ, there are some other important parameters mentioned in our method. During the training of SAE, the learning rate is set to be 1.0. The epoch number is 100, and weight of the sparsity penalty β is set to be 0.05.

The structure of CNN used in our experiments is shown in Fig. 3. In our method, the training samples for CNN are selected from 50 feature maps acquired by SAE, and these patches are of size 25×25. We apply convolutions with a kernel size of 4×4 and 2×2 pooling. There are three convolution layers and two pooling layers. In the first convolution layer, 80 convolution kernels are used, and we get 80 feature maps whose size is 22×22. After pooling, the size of the feature maps becomes 11×11. In the second convolution layer, by using 150 convolution kernels, we get 150 feature maps whose size is 8×8. After the second pooling, the size of feature maps becomes 4×4. In the third convolution layer, 200 convolution kernels are used, and the size of feature maps is reduced to 1×1. Finally, the higher-level change features are fed to a softmax classifier for identifying the changes.

5. Experimental results and analysis

5.1. Analysis of parameters

In this section, we will test the window size for training SAE and parameter τ, and analyze the influence of these two parameters on the final change detection results.

5.1.1. Window size for training SAE

In the proposed method, the window size of SAE samples is a significant parameter which can directly influence the extracted features and the classification results. The experiments are conducted on the real datasets, and the window of SAE is set to be 3×3, 5×5, 7×7 and 9×9, respectively. The curves of KC with respect to the sliding window size are shown in Fig. 12.

In Fig. 12, it can be seen the proposed method achieves the highest KC value when the window size is set as 5×5. The reason lies in the fact that smaller window such as 3×3 is not enough large for SAE to see for extracting more robust representations. However, if the window size is too large, the samples would contain too much unrelated information, and it may weaken the representation power to the central pixels. In the pixel-wise sampling procedure, a larger window would cause higher repetition rate between the two adjacent pixels, it will lead to more similar change information between them, generating worse results with

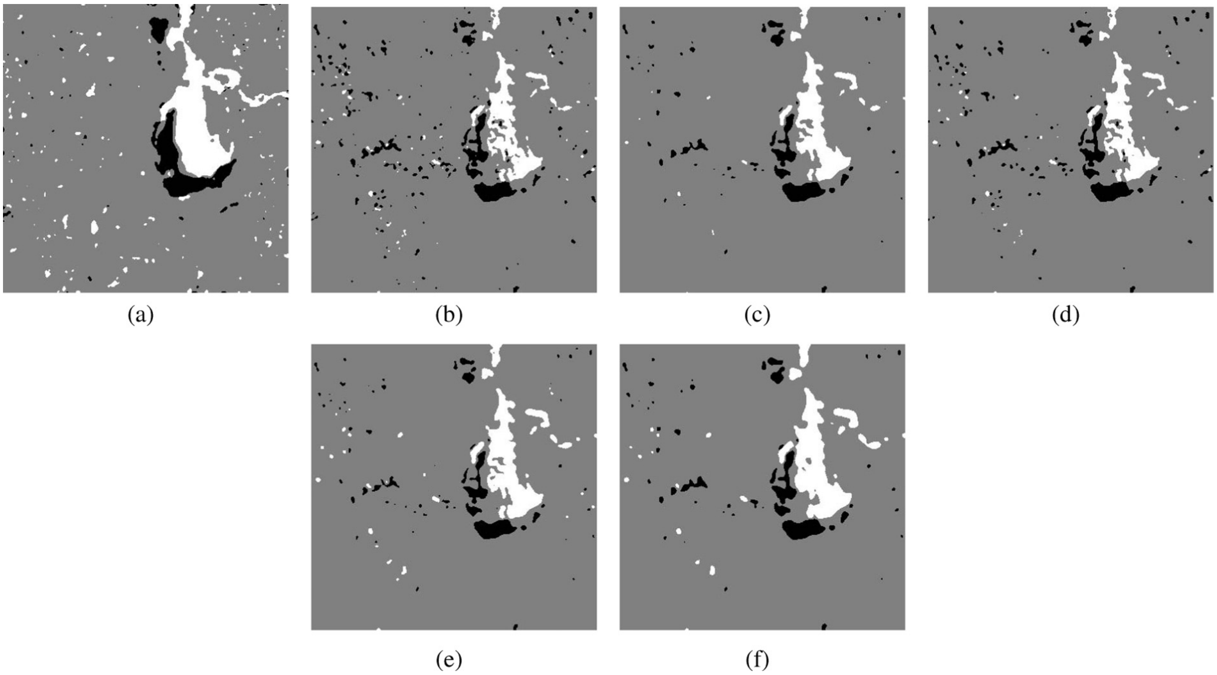


Fig. 17. Ternary change detection results of the San Francisco-2 dataset. (a) DTKI. (b) FCM. (c) FLICM. (d) KWFLICM. (e) SAE + FCM. (f) SAE + FCM + CNN.

Table 3
Quantitative comparison among different methods on the San Francisco-2 dataset.

	OE	*PCC*	*KC*	*NMI*
DTKI	9083	0.9403	0.6450	0.4536
FCM	9313	0.9388	0.5562	0.3515
FLICM	6594	0.9566	0.6612	0.4569
KWFLICM	8353	0.9451	0.5892	0.3833
SAE + FCM	6711	0.9559	0.6664	0.4528
SAE + FCM + CNN	**6657**	**0.9562**	**0.6913**	**0.4843**

blurry boundary. In the subsequent experiments, the window of SAE is set to be 5 × 5.

5.1.2. Influence of parameter τ

In this section, we analyze the influence of the parameter τ on the change detection accuracy. The parameter τ is selected as a testing one due to the fact that it can control the reliability of selected samples for training a CNN as the change feature classifier. In this set of experiments, τ is set to be different values such as 0.5, 0.6, 0.7, 0.8 and 0.9, each of which will be tested on the real datasets. Then, the measurement *KC* is used to quantitatively evaluate the corresponding change detection results, and the curves *KC*-τ are shown in Fig. 13.

As shown in Fig. 13, it is easy to find that the proposed method works much better when τ is set to be 0.6 or 0.7. When τ is set to be 0.5, the selected samples contain more noisy training samples that are not correctly classified by FCM, and these samples would mislead the learning process into a wrong way during training change feature classifier. When τ is set to be larger values such as 0.8 or 0.9, the selected samples would be not enough for training CNN. Therefore, τ is set to be 0.6 In the subsequent experiments.

5.2. Anti-noise performance of the proposed method

A series of experiments are conducted on the simulated datasets to evaluate the anti-noise performance of the proposed method, and they are shown in Figs. 14 and 15. The proposed method is compared with the other four algorithms. Due to the fact that there are so many result maps, we only exhibit our result maps and the best result of the four algorithms in Fig. 15. *OE* and *KC* are used to quantitatively evaluate the experimental results. In Fig. 15, when the noise level is high, the traditional methods perform not well. In the selected best result acquired by the traditional methods, it can roughly classify the image into three classes, but there are many evident black spots in the result map. This denotes that these traditional methods cannot deal with the noise well. Although the proposed method cannot guarantee a very accurate result as well, it performs much better than the other algorithms, and only a few noise spots emerge in our result. As the noise level decreases, these five methods can get much better results. When the noise level is low, the proposed method performs much better than most of compared algorithms. From these experimental results, it can be seen that the proposed approach performs better than the traditional methods in suppressing noise.

5.3. Performance of the proposed method

This set of experiments is conducted on four real datasets (including San Francisco-1 dataset, San Francisco-2 dataset, Yellow River dataset and Xi'an dataset) to demonstrate the effective of the proposed method. In addition, DTKI, FCM, FLICM and KWFLICM are taken as the comparative methods. In the table, the measurements on the proposed mehtod are written in bold.

5.3.1. Results on the San Francisco-1 dataset

The results of San Francisco-1 are shown in Fig. 16 and Table 2. As shown in Fig. 16, among DTKI, FCM, FLICM and KWFLICM, DTKI gets the worst result, and it fails to detect some obvious changed

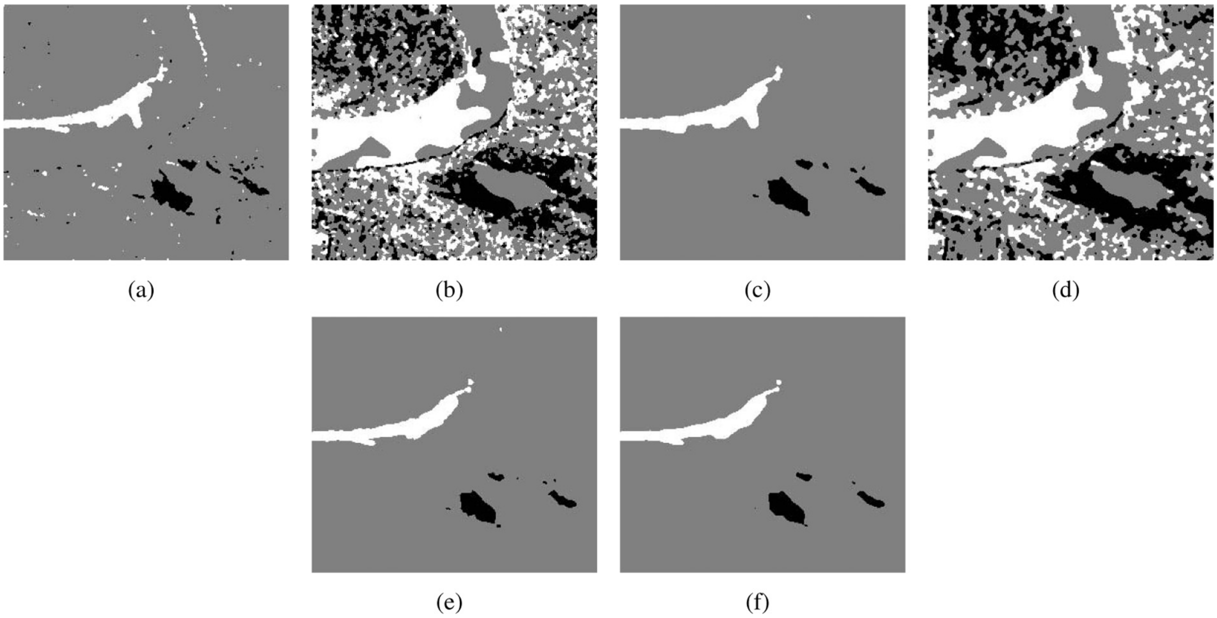


Fig. 18. Ternary change detection results of the Yellow River dataset. (a) DTKI. (b) FCM. (c) FLICM. (d) KWFLICM. (e) SAE + FCM. (f) SAE + FCM + CNN.

Table 4
Quantitative comparison among different methods on the Yellow River dataset.

	OE	*PCC*	*KC*	*NMI*
DTKI	2001	0.9735	0.7276	0.5693
FCM	32,399	0.5703	0.1069	0.1215
FLICM	1453	0.9807	0.7913	0.6474
KWFLICM	31,268	0.5853	0.1153	0.1364
SAE + FCM	1030	0.9863	0.8364	0.6941
SAE + FCM + CNN	**1013**	**0.9866**	**0.8389**	**0.6980**

areas. This is because DTKI tries to find two thresholds to classify the **DI** into three parts, and any little error in thresholds would lead to large variation in the final change detection result. By comparing the results obtained by these three clustering algorithms, we can find that FCM performs worse than the other two algorithms. Because FCM does not consider any information provided by the neighbors, and therefore, it is too sensitive to noise. FLICM and KWFLICM achieve much better results by introducing the context information. In contrast, SAE is able to learn the inherent relationships from the samples and suppress the noise of the image, and it can achieve much better results. CNN is adopted to learn more abstract change features from the SAE feature maps, and it can improve the accuracy of the change detection significantly.

5.3.2. Results on the San Francisco-2 dataset

Fig. 17 and Table 3 show the change detection results of the five different methods on the San Francisco-2 dataset. From Fig. 17, it is observed that some white and black spots emerge in all the result maps, this is because of the impact of speckle noise. Compared with the traditional methods results, the proposed method achieves fewer noise spots. This is because that the proposed method transforms **DI** into a suitable latent features space, and it can extract the key information and suppress the speckle noise, and therefore, the proposed method has better anti-noise performance. In Table 3, the proposed method does not achieve the best *OE* and *PCC*. However, the *KC* yielded by the proposed method equals to 0.6913 and the *NMI* equals to 0.4843, both of which are higher than the other methods. Undoubtedly, the quantitative comparison has demonstrated the superiority of the proposed method.

5.3.3. Results on the Yellow River dataset

Fig. 18 shows the visual comparison among different methods on the Yellow River dataset, while Table 4 summarizes the corresponding quantitative comparison. By visual comparison, we can find that the traditional approaches perform not well on this dataset. In FCM and KWFLICM, the detection results seem to be extremely poor, where three classes are not clearly separated. In DTKI, it could roughly group the map into three classes, but there are many noisy pixels in the result map. Compared with the other methods, FLICM can group them into the desired classes well. However, there are still some obvious mistakes. SAE + FCM achieves better values in metrics such as OE, PCC, KC and *NMI*. This is because SAE + FCM is mainly based on the features learned by SAE, and it is more robust to noise. Having the training samples and pseudo labels, CNN is driven to learn the concepts of change, and more power model can be established to distinguish different types of changes, and better result can be obtained.

5.3.4. Results on the Xi'an dataset

Fig. 19 and Table 5 show the change detection results of different methods on the Xi'an dataset. From visual comparison in Fig. 19, FLICM, SAE + FCM and SAE + FCM + CNN achieve much better change detection results, while FCM and KWFLICM fail to suppress unchanged pixels. And it is easy to find that SAE + FCM achieves more noisy results than the proposed framework. Through sample selection, more reliable training samples are selected for training a CNN as change feature classifier. Having reliable training samples and the corresponding pseudo labels, CNN is driven to learn the concept of changes and identify different types of changes. Therefore, the proposed method performs much better than the compared methods. As is proven, the proposed method wins both in the qualitative and quantitative comparisons. For both land and water areas, the proposed method can achieve satisfactory performance. In a word, the experimental results demonstrate the effectiveness and superiority of the proposed method.

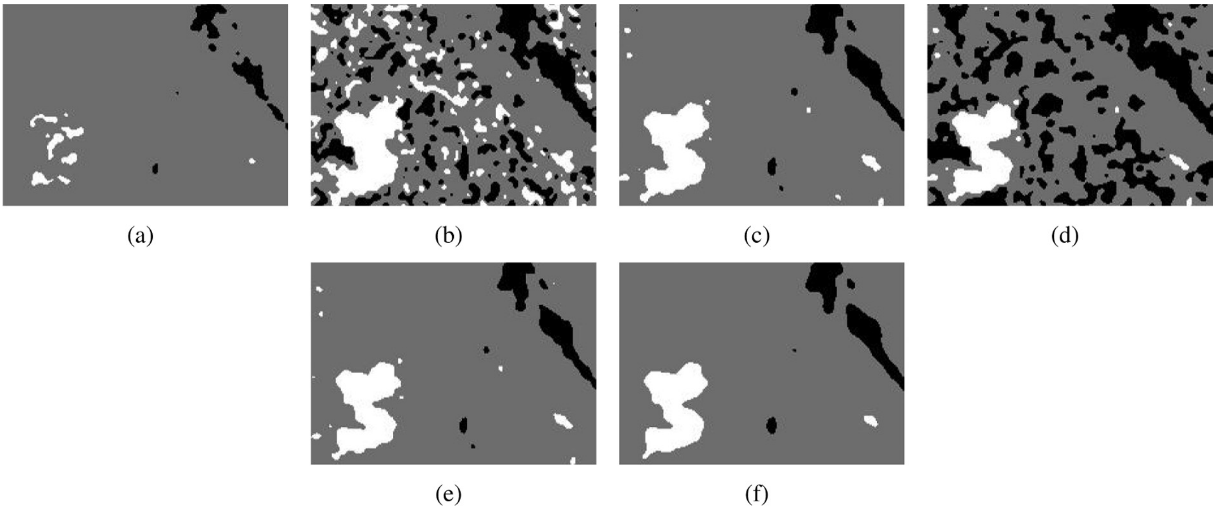


Fig. 19. Ternary change detection results of Xi'an dataset. (a) DTKI. (b) FCM. (c) FLICM. (d) KWFLICM. (e) SAE + FCM. (f) SAE + FCM + CNN.

Table 5
Quantitative comparison among different methods on the Xi'an dataset.

	OE	*PCC*	*KC*	*NMI*
DTKI	3251	0.9261	0.4581	0.3997
FCM	11,869	0.7303	0.3354	0.2379
FLICM	1389	0.9684	0.8396	0.6738
KWFLICM	12,188	0.723	0.324	0.3292
SAE + FCM	1286	0.9708	0.8442	0.6896
SAE + FCM + CNN	**1050**	**0.9769**	**0.8756**	**0.7462**

6. Conclusion

In this study, we have presented a novel ternary change detection framework for joint interpretation of spatial-temporal SAR images. In this framework, SAE, CNN and unsupervised clustering are successfully combined to achieve more precise ternary change detection without any supervision. SAE is used to transform difference image into a suitable feature space for suppressing noise and extracting key change information, and the feature maps learned by SAE are taken as the input of CNN. Then, unsupervised clustering is carried out on the feature representations learned by SAE to provide reliable pseudo labels for training a CNN as change feature classifier. Both qualitative and quantitative comparisons have proven the superiority of the proposed framework over the state-of-the-art methods.

Acknowledgements

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 A

In the appendix, we will derive Eq. (16), which represents the kappa coefficient. According to Cohen (1960), the formula of *KC* is as follows:

$$KC = \frac{PCC - PRE}{1 - PRE} \tag{18}$$

where *PCC* is the proportion of expected agreement, which expressed in Eq. (15). *PRE* represents the proportion of expected agreement. As defined in Cohen (1960), its expression is shown in Eq. (19).

$$\begin{aligned} PRE = & \frac{(T_{NN} + F_{NU} + F_{NP}) \cdot (T_{NN} + F_{UN} + F_{PN})}{N^2} \\ & + \frac{(F_{UN} + T_{UU} + F_{UP}) \cdot (F_{NU} + T_{UU} + F_{PU})}{N^2} \\ & + \frac{(F_{PN} + F_{PU} + T_{PP}) \cdot (F_{NP} + F_{UP} + T_{PP})}{N^2} \end{aligned} \tag{19}$$

Combined the matrix $\mathbf{D}$, *PRE* can be rewritten as follows:

$$PRE = \frac{\mathbf{t}\mathbf{D}^2\mathbf{t}^{\mathrm{T}}}{N^2} \tag{20}$$

where $\mathbf{t} = [1, 1, 1]$. Substituting Eqs. (15) and (20) in Eq. (18), we obtain the following final form of the kappa coefficient *KC*:

$$KC = \frac{N \cdot \mathrm{tr}(\mathbf{D}) - \mathbf{t}\mathbf{D}^2\mathbf{t}^{\mathrm{T}}}{N - \mathbf{t}\mathbf{D}^2\mathbf{t}^{\mathrm{T}}} \tag{21}$$

References

Arel, I., Rose, D.C., Karnowski, T.P., 2010. Deep machine learning - a new frontier in artificial intelligence research [research frontier]. IEEE Comput. Intelli. Mag. 5, 13–18.

Ban, Y., Yousif, O.A., 2012. Multitemporal spaceborne SAR data for urban change detection in China. IEEE J. Sel. Top. Appl. Earth Observ. Rem. Sens. 5, 1087–1094.

Bazi, Y., Bruzzone, L., Melgani, F., 2005. An unsupervised approach based on the generalized gaussian model to automatic change detection in multitemporal SAR images. IEEE Trans. Geosci. Rem. Sens. 43, 874–887.

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

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: a review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1798–1828.

Bezdek, J.C., 2013. Pattern Recognition with Fuzzy Objective Function Algorithms. Springer Science & Business Media.

Bouziani, M., GoÃŕta, K., He, D.C., 2010. Automatic change detection of buildings in urban environment from very high spatial resolution images using existing geodatabase and prior knowledge. ISPRS J. Photogram. Rem. Sens. 65, 143–153.

Bovolo, F., Marchesi, S., Bruzzone, L., 2012. A framework for automatic and unsupervised detection of multiple changes in multitemporal images. IEEE Trans. Geosci. Rem. Sens. 50, 2196–2212.

Brunner, D., Bruzzone, L., Lemoine, G., 2010. Change detection for earthquake damage assessment in built-up areas using very high resolution optical and sar imagery. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, pp. 3210–3213.

Chen, Y., Zhao, X., Jia, X., 2015. Spectral-spatial classification of hyperspectral data based on deep belief network. IEEE J. Sel. Top. Appl. Earth Observ. Rem. Sens. 8, 2381–2392.

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educ. Psychol. Meas. 20, 37–46.

Dahl, G.E., Yu, D., Deng, L., Acero, A., 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. Audio Speech Lang. Process. 20, 30–42.

Giustarini, L., Hostache, R., Matgen, P., Schumann, G.J.P., Bates, P.D., Mason, D.C., 2013. A change detection approach to flood mapping in urban areas using TerraSAR-X. IEEE Trans. Geosci. Rem. Sens. 51, 2417–2430.

Glorot, X., Bordes, A., Bengio, Y., 2011. Domain adaptation for large-scale sentiment classification: a deep learning approach. In: Proceedings of the 28-th International Conference on Machine Learning (ICML-11), pp. 513–520.

Goh, H., Thome, N., Cord, M., Lim, J.H., 2014. Learning deep hierarchical visual feature coding. IEEE Trans. Neural Netw. Learn. Syst. 25, 2212–2225.

Gong, M., Liang, Y., Shi, J., Ma, W., Ma, J., 2013. Fuzzy c-means clustering with local information and kernel metric for image segmentation. IEEE Trans. Image Process. 22, 573–584.

Gong, M., Liu, J., Li, H., Cai, Q., Su, L., 2015. A multiobjective sparse feature learning model for deep neural networks. IEEE Trans. Neural Netw. Learn. Syst. 26, 3263–3277.

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

Gong, M., Zhang, P., Su, L., Liu, J., 2016a. Coupled dictionary learning for change detection from multisource data. IEEE Trans. Geosci. Rem. Sens. 54, 7077–7091.

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

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., Kingsbury, B., 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. 29, 82–97.

Hinton, G.E., Osindero, S., Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18, 1527–1554.

Hou, B., Wei, Q., Zheng, Y., Wang, S., 2014. Unsupervised change detection in SAR image based on gauss-log ratio image fusion and compressed projection. IEEE J. Sel. Top. Appl. Earth Observ. Rem. Sens. 7, 3297–3317.

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

Jia, L., Li, M., Wu, Y., Zhang, P., Liu, G., Chen, H., An, L., 2015. SAR image change detection based on iterative label-information composite kernel supervised by anisotropic texture. IEEE Trans. Geosci. Rem. Sens. 53, 3960–3973.

Jin, S., Yang, L., Danielson, P., Homer, C., Fry, J., Xian, G., 2013. A comprehensive change detection method for updating the national land cover database to circa 2011. Rem. Sens. Environ. 132, 159–175.

Krinidis, S., Chatzis, V., 2010. A robust fuzzy local information c-means clustering algorithm. IEEE Trans. Image Process. 19, 1328–1337.

Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D., 1997. Face recognition: a convolutional neural-network approach. IEEE Trans. Neural Netw. 8, 98–113.

Lei, Y., Scheffer, N., Ferrer, L., McLaren, M., 2014. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1695–1699.

Li, H., Gong, M., Wang, Q., Liu, J., Su, L., 2016. A multiobjective fuzzy clustering method for change detection in SAR images. Appl. Soft Comput. 46, 767–777.

Li, N., Wang, R., Deng, Y., Chen, J., Liu, Y., Du, K., Lu, P., Zhang, Z., Zhao, F., 2014. Waterline mapping and change detection of tangjiashan dammed lake after wenchuan earthquake from multitemporal high-resolution airborne SAR imagery. IEEE J. Sel. Top. Appl. Earth Observ. Rem. Sens. 7, 3200–3209.

Liu, J., Gong, M., Zhao, J., Li, H., Jiao, L., 2016. Difference representation learning using stacked restricted boltzmann machines for change detection in SAR images. Soft Comput. 20, 4645–4657.

Liu, M., Zhang, H., Wang, C., Wu, F., 2014. Change detection of multilook polarimetric SAR images using heterogeneous clutter models. IEEE Trans. Geosci. Rem. Sens. 52, 7483–7494.

Lu, J., Li, J., Chen, G., Zhao, L., Xiong, B., Kuang, G., 2015. Improving pixel-based change detection accuracy using an object-based approach in multitemporal SAR flood images. IEEE J. Sel. Top. Appl. Earth Observ. Rem. Sens. 8, 3486–3496.

Mas, J.F., 1999. Monitoring land-cover changes: a comparison of change detection techniques. Int. J. Rem. Sens. 20, 139–152.

Parrilli, S., Poderico, M., Angelino, C.V., Verdoliva, L., 2012. A nonlocal SAR image denoising algorithm based on llmmse wavelet shrinkage. IEEE Trans. Geosci. Rem. Sens. 50, 606–616.

Shao, L., Wu, D., Li, X., 2014. Learning deep and wide: a spectral method for learning deep networks. IEEE Trans. Neural Netw. Learn. Syst. 25, 2303–2308.

Shin, H.C., Orton, M.R., Collins, D.J., Doran, S.J., Leach, M.O., 2013. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data. IEEE Trans. Pattern Anal. Mach. Intell. 35, 1930–1943.

Su, L., Gong, M., Zhang, P., Zhang, M., Liu, J., Yang, H., 2017. Deep learning and mapping based ternary change detection for information unbalanced images. Pattern Recogn. 66, 213–228.

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

Toshev, A., Szegedy, C., 2014. Deeppose: human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-14), pp. 1653–1660.

Wang, Y., Chen, L., Mei, J.P., 2014. Incremental fuzzy clustering with multiple medoids for large data. IEEE Trans. Fuzzy Syst. 22, 1557–1568.

Wu, C., Du, B., Zhang, L., 2014. Slow feature analysis for change detection in multispectral imagery. IEEE Trans. Geosci. Rem. Sens. 52, 2858–2874.

Yang, H., Dou, A., Zhang, W., Huang, S., 2014. Study on extraction of earthquake damage information based on regional optimizing change detection from remote sensing image. In: Proceedings of IEEE Geoscience and Remote Sensing Symposium, pp. 4272–4275.

Yu, P., Qin, A.K., Clausi, D.A., 2012. Unsupervised polarimetric SAR image segmentation and classification using region growing with edge penalty. IEEE Trans. Geosci. Rem. Sens. 50, 1302–1317.

Yuan, Y., Mou, L., Lu, X., 2015. Scene recognition by manifold regularized deep learning architecture. IEEE Trans. Neural Networks Learn. Syst. 26, 2222–2233.

Zhang, F., Du, B., Zhang, L., 2015. Saliency-guided unsupervised feature learning for scene classification. IEEE Trans. Geosci. Rem. Sens. 53, 2175–2184.

Zhang, H., Gong, M., Zhang, P., Su, L., Shi, J., 2016a. Feature-level change detection using deep representation and feature change analysis for multispectral imagery. IEEE Geosci. Rem. Sens. Lett. 13, 1666–1670.

Zhang, P., Gong, M., Su, L., Liu, J., Li, Z., 2016b. Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images. ISPRS J. Photogram. Rem. Sens. 116, 24–41.