Change Detection Method based on Block Similarity Measure

Yu Li, Tao Sun
Departments of Computer Science, Xi'an Shiyou University, Xi'an 710065, China
liyu@xsyu.edu.cn

Abstract. Due to the random distribution of speckle noises in SAR images, the method based on direct pixel contrast cannot correctly judge the change of the pixels. In this paper, Convolutional Neural Network (CNN) is used to describe the multitemporal image blocks. Then the learned image block features are input into the decision network for further learning. The change detection of the whole image is completed on the basis of comparing the image blocks. The innovation of this method is that the input of the network is not the difference image produced by traditional methods, but the corresponding multitemporal image blocks. The output of the network is the judgment of the change between blocks. In addition, CNN is adopted to describe the features, which can extract the main features of the image blocks and is more robust to coherent speckle noises. This method has excellent performance in the accuracy of change detection.

1. Introduction
With the development of space technology, a lot of SAR images are available in the military and civilian fields. And these data have become the main source of change detection. For synthetic aperture radar (SAR) images, the framework of change detection used in most literatures can be generalized as difference image (DI) analysis method. It can be generalized as following steps. First step is image preprocessing. Usually, geometric correction and registration are implemented to align the two multitemporal images in the same coordinate frame. A number of well-known despeckling filters are usually used to remove speckle noises in this step, e.g., the Lee [1], the Kuan [2], the Frost [3], and the Gamma-MAP filter [4]. Second step is generation of a DI from the multitemporal images. Ratio operator is often used to generate the DI. Also ratio image is expressed in a logarithmic or mean scale because of its robustness and non-sensitiveness to speckle noises [5]. For multispectral images, a comparable change vector analysis (CVA) technique is widely used [6]. The last and the most import step is analysis of the DI. It can be seen as an automatically segmentation problem. The most popular solution for image segmentation is thresholding [7] and clustering [8], [9], [10]. In some cases feature extraction on the DI is performed before the image segmentation, such as principal component analysis (PCA) [11], discrete wavelet transform (UDWT) [12], stationary wavelet transforms (SWT) [13], discrete wavelet transform (DWT) [14].

The advantage of DI analysis method is that it simplifies the research object. The original research object is two multitemporal images. And some operators are used to make the two images become one. In this process, the differences between the two images are enhanced. In other words, the contrast of changed pixels is enhanced in the DI. The gray of the changed areas and the unchanged areas becomes obviously. The most encouraging thing is that this step converts a change detection problem into a segmentation problem. A lot of segmentation methods can be referenced. However, due to the speckle effect of SAR, it is necessary to add local or non-local constraints to the segmentation algorithm. With these constraints, segmentation algorithms can be robust to the speckle noises. And a high accuracy of change detection result can be got.
The analysis based on DI simplifies the change detection problem. However, change detection also has its particularity which is different from image segmentation or classification. For example, the result category of change detection is generally less than that of segmentation. Usually the results fall into three or two categories. Three categories means exclude unchanged class, and changed class should be divided into positive and negative class. The increase of grey value corresponds to positive change, while decrease corresponds to negative change. Two categories problem is simpler. Only changed class and unchanged class should be distinguished. Therefore, for such simple category segmentation problem special algorithms should be taken into account.

The original data sets used in the change detection are two images. After being preprocessed, the two images have the same size and one-to-one pixel correspondence. The relationship between the pixels in the two images is simple. Therefore, change detection can be realized by directly measuring the difference between pixels. In the early stage of change detection, such simple idea has been used in processing optical remote sensing, such as difference method [15], ratio method [16], image regression method [17], vegetation index difference method [18], background subtraction method [19], and Change Vector Analysis (CVA) [20, 21], etc. However, since none of the above methods can obtain accurate change detection results, it is usually necessary to further combine them with threshold method. With the rapid development of pattern recognition and artificial intelligence in recent years, especially the revival of deep neural network, the old method just like direct comparison of two multitemporal images can be reused in change detection problem. In addition, unlike the early approaches, accurate detection results can be obtained at the same time.

Methods based on directly measuring the difference between pixels can be used to determine the change of pixels. However, the randomness of the speckle field often affects the correct judgment. Therefore, this paper puts forward a method based on block similarity measure. By building a deep Neural Network, unsupervised and supervised training methods are combined to complete change detection task.

2. Change Detection Method based on Block Similarity Measure

Let consider two co-registered intensity SAR images acquired over the same geographical area at two different times. They are expressed as \( I_1 = \{ I_1(i,j), 1 \leq i \leq H, 1 \leq j \leq W \} \) and \( I_2 = \{ I_2(i,j), 1 \leq i \leq H, 1 \leq j \leq W \} \) of size \( H \times W \). Through the change detection method, each pixel needs to be assigned a class label. In this paper, only changed class and unchanged class are considered. That is, there are only two classes of the result. Here, number 1 and number 0 are used to represent the class labels, where 1 corresponds to the changed pixel and 0 corresponds to the unchanged pixel.

2.1. Block-based representation

For a single image, the pixel-based method can be used to describe it directly, because pixel descriptor is not robust to noises. In addition, it does not have the ability to express the geometric structure. Block refers to a continuous region extracted from an image. And it can also be a descriptor of an image. Block-based description can be traced back to the Markov Random Field (MRF) model. The spatial constraint relation of a pixel is established through its neighborhood [22, 23]. Gu et al. defined the relationship between the spatial neighborhoods under the framework of MRF, which can be regarded as an effective usage of block-based description.

The most general and representative description of a block is the scale-invariant feature transform (SIFT) descriptor. In 2005, Mikolajczyk et al. applied SIFT to image matching and achieved good results [24]. The process of extracting SIFT can be regarded as a three-layer Convolutional Neural Network (CNN). In the first layer, gaussian kernel and image are executed a convolution to calculate gradient histogram. The second layer is a full connection layer, which uses gaussian function to distribute the weights of each gradient. In the third layer, the 128-dimensional features of \( 4 \times 4 \) window are extracted by pooling technique. Similar local descriptors include SURF[25], BRIEF[26], LIOP[27], etc.

Pooling technology, in fact, imitates human visual system. It generalizes the features of the adjacent block by the means of sum, max, average, or some statistical techniques. In this way, feature of the block
is associated with the relative position of the pixels, and has nothing to do with the absolute position. Therefore, pooling technology is robust to the input signal, which has been translated, curled, or rotated. It can be argued that pooling technology can generate a new kind of feature, which retains the main information of the signal and abandons irrelevant details. Such characteristic plays a vital role in judging the categories of the data in some application fields.

The representative of pooling technique in neural network is CNN. CNN can be regarded as a descriptor of image block. It can extract the main features of image block and ignore the secondary features. If CNN is used for image block comparison, it will be more robust to input errors, interference factors and so on. The convolution feature \( f \) is used to describe the image block, which can be defined as following:

\[
  f(x) = \gamma_k(\sigma_k(\omega_k \cdots \gamma_2(\sigma_2(\omega_2 \gamma_1(\sigma_1 \omega_1 x))) \cdots)),
\]

where \( x \) is a vector which represents the input image block. \( \omega_k \) is a weights matrix, where \( k \in \{1, 2, \ldots, K\} \), and \( K \) is the depth of the network. \( \sigma_k \) is a nonlinear function, such as sigmoid or logistic. \( \gamma_k \) is used to execute subsampling, which is also called pooling operation. Every combination of \( \gamma_k(\sigma_k(\omega_k \diamond \cdot)) \) corresponds to a layer in the network structure. Each layer of the network is made up of many neurons and is a representation of the input signal \( x \). The relationship of the adjacent layers is called mapping. For each input vector \( x \), the mapping can affect some neurons in the higher perceptual layer. In the construction of CNN, the number of layers \( K \) and the neurons in each layer should be fixed in advance, while the parameter \( \omega_k \) needs to be determined by learning method.

2.2. Network structure and target equation

To decide the image block is changed or unchanged do not depends on the absolute difference of pixels. Instead, the main features of the image block should be compared and various secondary features should be ignored. This task should be finished by automatically learning a change equation.

For each pair of input image blocks, the change equation completes the determination of the change. The framework is shown in figure 1. The function of the change equation can be approximated by a network architecture. The network needs to accomplish two main tasks. The first task is to describe the characteristics of the input image blocks. The second task is to make decision on the changes of the two image blocks. To complete the first task, two separate branches should be builded. The input of each branch is the grayscale vector of the image block. The output of each branch is the eigenvector. To accomplish the second task, the outputs of these two branches become inputs of the decision network. Decision network can decide if there has been a change between the two image blocks.

![Fig.1 Network architecture used to make decision on change occurred between the two input blocks](image-url)

The deep network used to extract image block features is composed of a series of CNN networks. Each CNN can be divided into two layers. One is the convolutional layer and the other is the pooling layer. The input of the first CNN is the grayscale vector of the image block. Except the first CNN, the inputs of other CNN networks are derived from the outputs of the previous CNN.
Similar to the simple neurons in the visual cortex, the convolutional layer extracts the local features of the image block by convolving the input signal with multiple filter kernels. Convolution enhances the input signal and is robust to noise. The operation of convolution is shown in figure 2. Assuming the size of the input matrix is \( n \times n \) and the size of the convolution kernel is \( m \times m \), then the output matrix is \( (n-m+1) \times (n-m+1) \).

\[
\begin{align*}
\text{Fig. 2 Convolution operation}
\end{align*}
\]

The input matrix and convolution kernel are pulled into column vectors. Each vector element is represented by a neuron. Assuming that each convolution layer has multiple convolution kernels, the \( i \)th convolution kernel corresponds to \( \omega_i \). For the input \( x \) of each branch, the convolution is defined as following:

\[
h^i = \sigma(x \ast \omega_i + b^i),
\]

where \( \ast \) represents convolution operator. \( \sigma \) is an activation function. Sigmoid function is use here, which is defined as \( \sigma(x) = (1 + e^{-x})^{-1} \). The output of each convolution layer will be the input of the adjacent pooling layer.

Pooling layer is similar to the complex neurons in the visual cortex. Its function is to obtain a basic and invariable feature by reducing the feature resolution. It can generate a low-dimensional feature representation by sampling from the neighborhood features through mean or max methods. Pooling can not only reduce dimensionality, but also preserve the key components. Max pooling is adopted in this paper, and the process is shown in figure 3. The left side is a 4x4 feature graph, and the non-overlapping 2x2 neighborhood is selected for max sampling. A 1/4 feature graph of the size of the original feature graph is obtained.

\[
\begin{align*}
\text{Fig. 3 Max pooling process}
\end{align*}
\]

For any input feature \( x \), the maximum value is taken in the non-overlapping region of size \( R \times R \). And the output feature \( h \) is obtained. The max pooling equation can be expressed as:

\[
h = \max(x).
\]

The two blocks generate the eigenvectors \( y_1 \) and \( y_2 \) respectively through their CNN networks. The next step is to analyze and compare the eigenvectors and make a decision on whether there is a change occurred. As the CNN acts as the task of feature extraction, the comparison of these features can be realized by any learning methods. In this paper, a fully connected neural network is used to implement the decision. In other words, the feature vector \( y = [y_1, y_2] \) generated by the method introduced above will be taken as the input of the decision network. By leaning, the decision network will be used to determine if there has been a change between the two image blocks.

The decision network is a full connected network. Since the output of the decision network is a neuron, a cross-entropy loss function is used here for its fast convergence speed. It can be shown as following:

\[
L(\theta_d) = -\frac{1}{N} \sum_{i=1}^{N} (z_i \ln(g_{\theta_d}(y_i)) + (1-z_i)\ln(1 - g_{\theta_d}(y_i))),
\]
where $\theta_d$ is the parameters of the decision network. $g_{\theta_d}(y_i)$ is the output of the $i$th training sample, and $N$ is the number of training samples. Parameter $z_i \in \{0,1\}$ represents the real output of the network. Symbols $\{0,1\}$ correspond to the changed and unchanged class.

The above network completes the comparison between image blocks. In practical operation, these image blocks overlap with each other. Each image block corresponds to the neighborhood of the center point $(i,j)$.

2.3. Method of parameter optimization

After the network structure is completed, the next step is to optimize the network parameters. Since the whole network is divided into two parts, these two parts can be solved separately. The first part is the deep CNN used to extract image block features. The second part is the network for decision-making, which adopts a supervised learning method. Therefore, some labeled samples are needed to complete the network decision training.

Unsupervised learning of deep CNN is realized by reference to autoencoder. For any input $x$ of an autoencoder, the equation of hidden layer $h$ is: $h = f_{\theta_h}(x) = \sigma(\omega_h x + b_h)$, with parameters $\theta_h = \{\omega_h, b_h\}$. This process is called encoding process. Reconstruction of input signal $x$ through hidden layer $h$ can be expressed as: $y = f_{\theta_y}(h) = \sigma(\omega_y h + b_y)$. The goal of reconstruction is to make $y$ as close as possible to the input signal, namely $y \rightarrow x$, which is called decoding process. The weights in these two equations have the following relation: $\omega_y = \omega_h^T$. That is, the same weights are used in the encoding and decoding process. Each training sample $x_i$ is mapped to $h_i$ through the network, then to $y_i$. Parameters are optimized by minimizing reconstruction errors:

$$\hat{\theta}_c = \arg \min_{\theta_c} \frac{1}{N} \sum_{i=1}^{N} E(x_i, y_i),$$

(5)

where $N$ is the number of training samples, and the loss equation $E$ is usually represented as $E(x, y) = \|x - y\|^2$.

The decoding process of convolution autoencoder includes the steps of deconvolution and unpooling. Generally the process of pooling is irreversible, so the unpooling is approximately completed. In the process of pooling, the maximum active location needs to be preserved, then the data is restored to the active location during the process of unpooling. And the rest of the locations are set to 0. This process can be illustrated in figure 4. Assume that the sample block is $2 \times 2$, and the size of the feature graph after pooling is $2 \times 2$, as shown in the left position of the figure. Unpooling is an upsampling process, which needs to expand the feature graph from size $2 \times 2$ to size $4 \times 4$. The value on the $2 \times 2$ feature graph should be filled in the corresponding $4 \times 4$ feature graph according to the maximum position reserved in the pooling process, while the other empty positions should be set to 0.

![Unpooling process](image)

The process of deconvolution is to convolve the feature graph $H$ with the transpose of the convolution kernel, then add the results. The formula is showed as follows:

$$y = \sigma(\sum_i h^* \omega_{xy}^T + c_y),$$

(6)

where $*$ represents convolution operator. $\sigma$ is an activation function.

Assuming that the input of the convolutional autoencoder is $x$ and output is $y$, the objective equation is:
\[ E(\theta) = \frac{1}{2N} \sum_{i=1}^{N} (x_i - y_i)^2, \]  
\[ (\theta_i')^{l+1} = (\theta_i')^{l+1} - \alpha \frac{\partial E(\theta)}{\partial \theta_i}, \]  

where  \( N \) is the number of training samples. The back propagation algorithm is used to calculate the gradient of the error function and update the parameters. The parameters of the convolution layer are updated. For the pooling and unpooling stages, residual is also pooled and unpoled. The formula for updating the parameters of the convolution layer is as follows:

\[ \frac{\partial E(\theta)}{\partial \theta_i} = x \delta_h + h' \delta_y, \]  

where \( \delta_h \) and \( \delta_y \) represent the residual of convolution layer and unconvolution layer, respectively. \( x \) is the input, and \( h' \) is the hidden layer node corresponding to the \( i \)th convolution kernel. The iterative equation is:

\[ (\theta_i')^{l+1} = (\theta_i')^{l+1} - \alpha \frac{\partial E(\theta)}{\partial \theta_i}, \]  

where symbol \( l \) represents the number of iterations, and \( \alpha \) represents learning rate.

The decision network is a typical fully connected neural network. The output is a neuron. Values 1 and 0 are used to represent changed and unchanged class, respectively. The classical back propagation algorithm is used iteratively to optimize the parameters of each layer until the entire network convergent.

### 3. Experimental Study

This section verifies the effectiveness of the method proposed in this paper through comparative experiments on the data set of the Yellow River delta. Compared with other four methods, they are GKI_LN[28], RFLICM[5], PCA_kmeans[11], and DNN [29].

The quantitative analysis of change detection results are based on the following values: false positives (FP), false negatives (FN), overall error (OE), and Kappa [30], which is a measure of accuracy or agreement based on the difference between the error matrix and chance agreement.

In this paper, two concatenated CNN networks are adopted. The learning rate of the deep CNN is set as 0.05, and the input is image block of 13×13 size. Through eight 4×4 convolution kernels, the maximum pool of size 2×2 is adopted in the first CNN. Then, the outputs of the first CNN of each channel are eight 5×5 feature matrices, which are convolved with sixteen 2×2 convolution kernels and through the maximum pool of size 2×2. The output of each deep CNN is 512 dimensional feature. The decision network uses two hidden layers, the number of hidden layer nodes is 100 and 10 respectively, and the learning rate is set as 0.6.

The data set includes two SAR images at the areas of Yellow River Estuary in China. The original two SAR images were acquired by Radarsat-2 in June 2008 and June 2009, respectively, with 7666×7692 size. The image acquired in 2008 is four-look, and the one acquired in 2009 is single-look. The influence of speckle noises on the later image is much greater than the former one. Two areas were selected from the Yellow River delta data set, namely coastal area and farmland area.

Table 1 lists the change detection results of the methods proposed in this paper and other methods in the coastal area. As it can be seen from the table, GKI_LN threshold algorithm makes a relatively high miss detection rate of 498 pixels. Clustering methods RFLICM and PCA_kmeans have higher false alarms, which are 651 pixels and 445 pixels, respectively. DNN algorithm cannot give a good result in this data set, which has the largest number of miss detection. The total number of errors reaches 548 pixels, and Kappa coefficient is lower than threshold algorithm and clustering algorithms. The proposed method gives the best result. The total number of errors in the proposed method is 299 pixels, and the Kappa coefficient is 0.8900, which is better than the other four algorithms.

| Method     | FP  | FN  | OE  | Kappa  |
|------------|-----|-----|-----|--------|
| proposed   | 175 | 124 | 299 | 0.8900 |
| GKI_LN     | 2   | 498 | 500 | 0.7708 |
Table 2 compares the test results of various methods in farmland area. It can be seen that threshold method GKI_LN still has the highest miss detection in this area, and the value of FN is 2301. The false alarm of PCA_kmeans method is relatively high, which is 1058. Both DNN and our proposed method show low false alarm and high miss detection. RFLICM has relatively higher KAPPA coefficient. Compared with RFLICM, FP of our proposed method has obvious advantage. Therefore, the proposed method increases the KAPPA coefficient from 0.8588 produced by RFLICM to 0.8780.

Table 2. Change detection results of the farmland area

| Method     | FP  | FN  | OE   | Kappa |
|------------|-----|-----|------|-------|
| proposed   | 92  | 1018| 1110 | 0.8780|
| GKI_LN     | 100 | 2301| 2401 | 0.6990|
| RFLICM     | 451 | 894 | 1345 | 0.8588|
| PCA_kmeans | 1058| 568 | 1626 | 0.8429|
| DNN        | 333 | 1229| 1562 | 0.8288|

Fig. 6 shows the effect of various methods of change detection in farmland area. It can be seen that the method proposed in this chapter, as shown in Fig. 6(d), has the least noises compared with other methods. It indicates that CNN is robust to noise in block feature extraction. The shortcoming of the proposed method is mainly reflected in the processing of detailed information. Some pixels are missed in the raster region in the middle of figure 6(d).
4. Conclusion

This paper proposes a change detection method based on block similarity. This method constructs a deep network to achieve the purpose of change detection. Due to the random distribution of speckle noises in SAR images, the method based on direct pixels contrast cannot correctly judge the category of the pixels. Therefore, the multitemporal image blocks are firstly described separately by deep CNN, and then the learned image block features are input into the decision network for further learning. The decision network realizes the judgment of the change between the blocks. The innovation of this method is that the input of the network is not the DI produced by the traditional methods, but the corresponding multitemporal graph blocks. The output of the network is the determination of the change between image blocks. In addition, the deep CNN is adopted to describe the features of the image blocks, which can extract the main features of the image block and is more robust to speckle noises. The experimental results demonstrate the effectiveness of the proposed method. As expected, it has better performance than the state-of-the-art algorithms, especially in region consistence, detail preserving and noise resistance.

Acknowledgements

This work was financially supported by Natural Science Basic Research Plan in Shaanxi Province of China (2018JM6093), and Scientific Research Plan of Shaanxi Committee of Education (17JK0595).

References

[1] Lee J S. Speckle suppression and analysis for synthetic aperture radar images[J]. Optical engineering, 1986, 25(5): 255636–255636.
[2] Kuan D T, Sawchuk A A, Strand T C, et al. Adaptive noise smoothing filter for images with signal-dependent noise[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,1985(2): 165–177.
[3] Frost V S, Stiles J A, Shanmugan S, et al. A model for radar images and its application to adaptive digital filtering of multiplicative noise[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1982(2): 157–166.
[4] Lopes A, Nezry E, Toutzi R, et al. Structure detection and statistical adaptive speckle filtering in SAR images[J]. International Journal of Remote Sensing, 1993, 14(9): 1735–1758.
[5] Gong M, Zhou Z, Ma J. Change detection in synthetic aperture radar images based on image
fusion and fuzzy clustering[J]. IEEE Transactions on Image Processing, 2012, 21(4): 2141–2151.

[6] Bruzzone L, Prieto D F. Automatic analysis of the difference image for unsupervised change detection[J]. IEEE Transactions on Geoscience and Remote Sensing, 2000, 38(3): 1171–1182.

[7] Hu M, Li M, Wang R G. Application of an improved Otsu algorithm in image segmentation[J]. Journal of Electronic Measurement & Instrument, 2010, 24(5):443-449.

[8] Coleman G B, Andrews H C. Image segmentation by clustering[J]. Proceedings of the IEEE, 2005, 67(5): 773-785.

[9] Sulaiman S N, Isa N A M. Adaptive fuzzy-K-means clustering algorithm for image segmentation[J]. IEEE Transactions on Consumer Electronics, 2011, 56(4):2661-2668.

[10] Tian X, Jiao L, Zhang X. A clustering algorithm with optimized multiscale spatial texture information: Application to SAR image segmentation[J]. International Journal of Remote Sensing, 2013, 34(4):1111-1126.

[11] Celik T. Unsupervised change detection in satellite images using principal component analysis and k-means clustering[J]. IEEE Geoscience and Remote Sensing Letters, 2009, 6(4):772–776.

[12] Celik T. Multiscale Change Detection in Multitemporal Satellite Images[J]. IEEE Geoscience and Remote Sensing Letters, 2009, 6(4):820-824.

[13] Bovolo F, Bruzzone L. A detail-preserving scale-driven approach to change detection in multitemporal SAR images[J]. IEEE Transactions on Geoscience and Remote Sensing, 2005, 43(12) : 2963 – 2972.

[14] Bovolo F, Bruzzone L. An adaptive multiscale approach to unsupervised change detection in multitemporal SAR images[C]// IEEE International Conference on Image Processing. IEEE, 2005.

[15] Muchoney D M, Haack B N. Change detection for monitoring forest defoliation[J]. Photogrammetric Engineering & Remote Sensing, 1994, 60(10): 1243–1251.

[16] Prakash A, Gupta R P. Land-use mapping and change detection in a coal mining area—a case study in the Jharia coalfield, India[J]. International Journal of Remote Sensing, 1998, 19(3): 391–410.

[17] Jha, C. S, Unni N V M. Digital change detection of forest conversion of a dry tropical Indian forest region[J]. International Journal of Remote Sensing, 1994, 15(13): 2543–2552.

[18] Townshend J R G, Justice C O. Spatial variability of images and the monitoring of changes in the Normalized Difference Vegetation Index[J]. International Journal of Remote Sensing, 1995, 16(16): 2187–2195.

[19] Singh A. Digital change detection techniques using remotely sensed data[J]. International Journal of Remote Sensing, 1988, 10(6): 989–1003.

[20] Lambin E F. Change detection at multiple temporal scales: seasonal and annual variations in landscape variables[J]. Photogrammetric Engineering & Remote Sensing, 1996, 62(8): 31–938.

[21] Allen T R, Kupfer J A. Application of Spherical Statistics to Change Vector Analysis of Landsat Data: Southern Appalachian Spruce–Fir Forests[J]. Remote Sensing of Environment, 2000, 74(3): 482–493.

[22] Kasetkasem T, Varshney P K. An image change detection algorithm based on Markov random field models[J]. IEEE Transactions on Geoscience & Remote Sensing, 2002, 40(8): 1815–1823.

[23] Yousif O, Ban Y. Improving SAR-Based Urban Change Detection by Combining MAP-MRF Classifier and Nonlocal Means Similarity Weights[J]. IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing, 2014, 7(10): 4288–4300.

[24] Mikołajczyk K, Tuyltelaeurs T, Schmid C, et al. A Comparison of Affine Region Detectors[J]. International Journal of Computer Vision, 2005, 65(1-2): 43–72.
[25] Bay H, Tuytelaars T, Gool L V. SURF: Speeded Up Robust Features[J]. Computer Vision & Image Understanding, 2006, 110(3): 404–417.
[26] Calonder M, Lepetit V, Strecha C, et al. BRIEF: Binary Robust Independent Elementary Features[C] // Computer Vision - ECCV 2010, European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings. 2010: 778–792.
[27] Wang Z, Fan B, Wu F. Local intensity order pattern for feature description[C] // IEEE International Conference on Computer Vision. 2011: 603–610
[28] Moser G, Serpico S B. Generalized minimum-error thresholding for unsupervised change detection from SAR amplitude imagery[J]. IEEE Transactions on Geoscience and Remote Sensing, 2006, 44(10): 2972–2982.
[29] Liu J, Gong M, Zhao J, et al. Difference representation learning using stacked restricted Boltzmann machines for change detection in SAR images[J]. Soft Computing, 2014: 1–13.
[30] Tanaka M, Torii A, Okutomi M. Fisher Vector based on Full-covariance Gaussian Mixture Model[J]. IPSJ Transactions on Computer Vision & Applications, 2013, 5: 50–54.