2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Deep Learning to Classify Difference Image for Image Change Detection

Jiaojiao Zhao, Maoguo Gong, *Member*, *IEEE*, Jia Liu, Licheng Jiao, *Senior Member*, *IEEE*

***Abstract*—Image change detection is a process to analyze multi-temproal images of the same scene for identifying the changes that have occurred. In this paper, we propose a novel difference image analysis approach based on deep neural networks for image change detection problems. The deep neural network learning algorithm for classification includes unsupervised feature learning and supervised fine-tuning. Some samples with the labels of high accuracy obtained by a pre-classification are used for fine-tuning. Since a deep neural network can learn complicated functions that can represent high-level abstractions, it can obtain satisfactory results. Theoretical analysis and experiment results on real datasets show that the proposed method outperforms some other methods.**

I. Introduction

IMAGE change detection is more and more important in image processing field. It is a process to analyze two images of the same scene but taken at different times for identifying the changes that have occurred between the two images. It has been used widely in diverse disciplines such as remote sensing, medical diagnosis, and video surveillance [1]. Since synthetic aperture radar (SAR) images are independent of atmospheric and sunlight conditions, they have become valuable sources of information in change detection. However, SAR images exhibit more difficulties than optical ones because of the presence of the speckle noise [2].

Unsupervised change detection in SAR images can be divided into three steps [3]: (1) Preprocessing. (2) Producing a difference image (DI) between the multi-temporal images. (3) Analyzing the DI. Firstly, the preprocessing step often includes coregistration, geometric correction and denosing. The task of the second step is to generate a difference image by comparing two coregistered images pixel by pixel. Differencing and rationing (R) are the well-known methods for producing a DI. But because of the multiplicative nature of speckles, the ratio image is usually expressed in a logarithmic [4, 5]. In addition, in order to further suppress the effect of speckles, we proposed a neighborhood-based ratio (NR) approach to generate a DI [6]. In the third step, DI-analysis is actually taken as a process to segment an image. The thresholding method is a conventional method which applies a decision threshold to the histogram of the DI. Some classical thresholding methods such as Otsu [7] and the Kittler and Illingworth (KI) [8] are applied to determine the threshold in an unsupervised manner. For taking the non-Gaussian distribution of the amplitude values of SAR images into account, a KI-based method (GKI) was generalized [9]. And a generalized Gaussian model which is used in a KI approach was presented in [4]. Another kind of methods for segmenting images is the clustering method. The fuzzy c-means (FCM) algorithm which can retain more information than hard clustering in some cases is one of the most popular clustering methods. Some excellent algorithms were developed based on the FCM. We modified the objective function of the robust fuzzy local information c-means clustering algorithm (FLICM) [10] and proposed the reformulated FLICM (RFLICM) in [11].

As mentioned above, change detection technology in SAR images has developed well. But the involved algorithms can not satisfy requirements because of the data acquisition channels and the scope of applications increasing. The available thresholding methods need to establish suitable models and the clustering methods are sensitive to noise. However, deep neural network which has a powerful ability to learn has many successes in image processing.

Deep learning [12], as a new kind of machine learning method, has been paid more and more attention in recent years. In deep learning, we input the signals directly without any preprocessing, and the features can be learned layer by layer unsupervisedly. The output of one layer is the input of the next layer. And the layers form a deep network architecture which can learn the representations of the input signals. In image processing, the hierarchical structure can learn different features at different layers. And the deeper the network is, the more abstract information can be learned. Deep learning has attracted widespread interest due to a larger number of applications in traditional artificial intelligence (AI), such as natural language processing [13], modeling textures [14], handwriting recognition [15, 16] and image processing [17, 18], especially classification tasks [19-21]. There are some deep learning methods including convolutional neural networks (CNNs) [22], Deep Belief Networks (DBNs) [23] and auto-encoders [24]. CNNs which are inspired by biological processes are widely used models for image-recognition. DBNs exploit an unsupervised learning algorithm, the restricted Boltzmann machine (RBM) [25], for each layer. And auto-encoders apparently exploit the same principle in [24]. Auto-encoders, RBMs and DBNs can be

This work was supported by the National Nature Science Foundation of China (Grant no. 61273317), the National Top Youth Talents Program of China, the Specialized Research Fund for the Doctoral Program of Higher Education (Grant no.20130203110011) and the Fundamental Research Funds for the Central Universities (Grant No. K5051202053).

The authors are with Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, No.2 South TaiBai Road, Xi'an 710071, China. E-mail: gong@ieee.org.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

trained with unlabeled data. Deep learning has the property of unsupervised feature learning. It can learn from datasets with a few labeled data and many unlabeled data, which is important in our unsupervised change detection method.

In order to learn complicated functions that can represent high-level abstractions, a deep architecture is needed [26]. Deep architectures are composed of multiple levels of non-linear operations, such as in neural nets with many hidden layers. Automatically learning features at multiple levels of abstraction allows a system to learn complex functions mapping the input to the output directly from data. Compared with a shallow network, a deep network can describes images more sufficiently. In this paper, we propose a deep architecture for change detection. Firstly, we make a pre-classification for obtaining some samples; secondly, we use RBMs to pre-train a deep neural network and then fine-tune it; thirdly, we use the trained deep neural network to classify a DI.

This paper is organized into four sections. In Section II, the proposed method will be described in details. Section III will present experimental results on real multi-temporal SAR images to verify the feasibility of the method. Finally, we will make a conclusion in the last section.

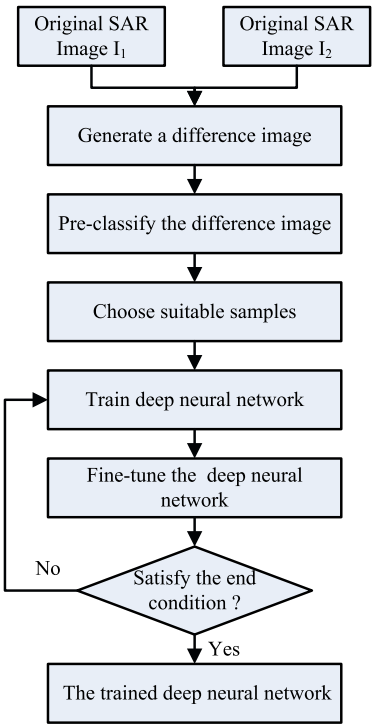


Fig.1. Flowchart of generating the deep neural network

II. METHODOLOGY

We consider two co-registered intensity SAR images $I_1 = \{I_1(i,j), 1 \le i \le A, 1 \le j \le B\}$ and $I_2 = \{I_2(i,j), 1 \le i \le A, 1 \le j \le B\}$. Both of them have the same size and are acquired over the same geographical area at two different times t_1 and t_2 respectively. We use the log-ratio (LR) operator to generate a difference image between the two original images.

$$DI(i,j) = \left| \log \left[\frac{I_1(i,j)}{I_2(i,j)} \right] \right| \tag{1}$$

The neighborhood features of each pixel on the difference image are taken as the inputs and we use a logistic output unit to indicate whether the pixel is changed or unchanged. There are three main phases in the proposed approach based on a deep architecture: 1) Pre-classify a difference image and choose suitable samples; 2) design, train and fine-tune a deep neural network; 3) use the trained deep neural network to classify the DI. The flowchart of generating the deep neural network is shown in Fig.1.

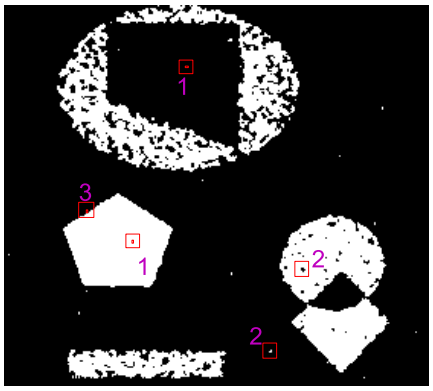


Fig.2. Illustrate how to choose suitable samples

A. Choose suitable samples

We use a simple thresholding method (e.g. KI algorithm) to pre-classify the difference image. The results of pre-classification are not entirely correct. The pixels which have high possibility of being correctly classified are chosen to train the networks. Suppose that the pixel p_{ij} at the position (i, j) in the pre-classification result map has the label R_{ij}. Let N_{ij} be a neighborhood with a center at the position (i, j) and of size $n \times n$. A simulated image as an example is shown in Fig.2, and there are three kinds of points should be considered. A point in the changed areas or unchanged areas, like Point 1 in Fig.2, has a neighborhood in which almost all pixels have the same labels with it. A point on the edge, like Point 3 in Fig.2, has a neighborhood in which about half all pixels have the same labels with it. These two kinds of point can be chosen as samples. A point that is wrongly classified (also called noise spot), like Point 2 in Fig.2, has a neighborhood in which a few or none of the pixels have the same labels with it. This kind of point should be eliminated. In conclusion, if the point p_{ij} has a neighborhood N_{ij} which satisfies Eq. (2), the point p_{ij} can be chosen as a sample.

$$\frac{Q(p_{\xi\eta} \in N_{ij} \wedge R_{\xi\eta} = R_{ij})}{n \times n} > \alpha \tag{2}$$

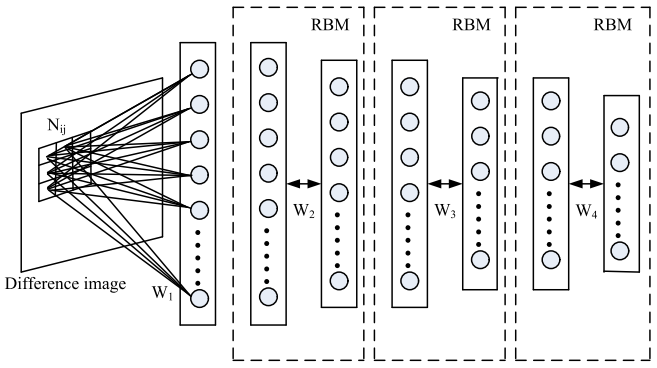
where the point $p_{\xi\eta}$ is in the neighborhood N_{ij}, and $Q(p_{\xi\eta} \in N_{ij} \wedge R_{\xi\eta} = R_{ij})$ means the number of pixels with the label equal to R_{ij} in the neighborhood N_{ij}. The parameter α which decides whether the p_{ij} is chosen as a sample is very important. α can not be set too small or too large. If α is set too

small, the result will be not robust to noise; and if α is set too large, the diversity of samples will decrease, which results in many missed alarms.

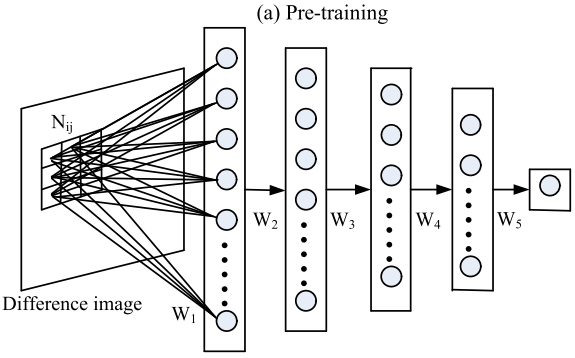
B. Using Deep Learning for Change Detection

Training a deep neural network is the core part of the algorithm. It is difficult to optimize the weights and biases in nonlinear networks which have multiple hidden layers. Starting with random weights, multiple BP networks can not always find a satisfactory result. If the initial weights are large, the result typically traps into local optimization. But small initial weights lead the gradients in the early layers to be tiny, thus making it infeasible to train networks with many hidden layers. The initial weights close to a good solution can make gradient descent work well, but finding such initial weights requires a very different type of algorithm that learns one layer of features at a time. The restricted Boltzmann machine (RBM) [25] can help to solve the problem.

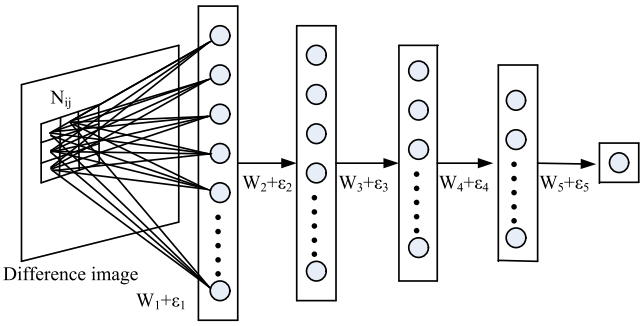
The process of the proposed method is illustrated in Fig.3. Firstly, inputting neighborhood features of one pixel on the difference image. Secondly, a stack of RBM is learned for pre-training. Next, the RBM's are "unrolled" to create a deep neural network for training. And then, the deep neural network is fine-tuned using BP of error derivatives [27].



(a) Pre-training



(b) Unrolling



(c) Fine-tuning

Fig.3. (a) RBM's are used for pre-training; (b) After pre-training, RBM's are "unrolled" to create a deep neural network; (c) Fine-tuning using BP.

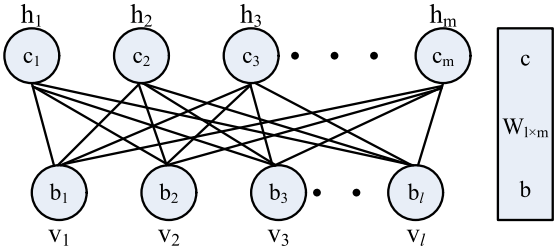


Fig.4. Structure of RBM

An ordinary structure of RBM network is shown in Fig.4. RBM has l visible units (v_1, v_2,…, v_l) corresponding to features of its inputs and m hidden units (h_1, h_2,…, h_m) that are trained. And each connection in an RBM must connect a visible unit to a hidden unit. $W_{l\times m}$ represents a weight matrix between visible layer and hidden layer; b=(b_1, b_2,… b_l) are biases of visible units and c=(c_1, c_2,…c_m) are biases of hidden units. A joint configuration (v, h) of the visible and hidden units has an energy [27, 28] given by

$$E(v,h) = -\sum_{i\in pixels} b_i v_i - \sum_{j\in features} c_j h_j - \sum_{i,j} v_i h_j W_{ij} \tag{3}$$

Suppose that $\forall i, j, v_i \in \{0,1\}, h_j \in \{0,1\}$. For a given v , the binary state h_j , of each hidden unit j, is set to 1 with probability:

$$P(h_j = 1 \mid v) = \sigma(\sum_{i=1}^{l} W_{ij} \times v_i + c_j) \tag{4}$$

where $\sigma(x) = 1/(1+e^{-x})$ is a sigmoid function. Once binary states have been chosen for the hidden units, a reconstruction is produced by setting each v_i to 1 with the probability:

$$P(v_i = 1 \mid h) = \sigma(\sum_{j=1}^{m} W_{ji} \times h_j + b_i) \tag{5}$$

The states of the hidden units are then updated once more so that they represent features of the confabulation. The change in a weight is given by

$$\Delta W_{ij} = \varepsilon(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon}) \tag{6}$$

where ε is a learning rate, $\langle v_i h_j \rangle_{data}$ is the fraction of times that the feature i and feature detector j are on together when the feature detectors are being driven by data, and $\langle v_i h_j \rangle_{recon}$

is the corresponding fraction for reconstruction. A simplified version of the same learning rule is used for the biases.

A two-layer RBM network in which stochastic, binary features are connected to stochastic, binary feature detectors using symmetrically weighted connections can be used to model pixels. The features correspond to visible units of the RBM because their states are observed; the feature detectors correspond to hidden units. The network assigns a probability to every possible pixel via this energy function in Eq. (3). Given the set of training samples which are chosen previously, RBM's are used to pre-train according to the rules described above. The pre-training does not use any information of the class labels. Here, a layer-by-layer learning algorithm is applied.

After pre-training, the RBM model is "unfolded" to produce a deep neural network that initially uses the same weights and biases. A BP algorithm based on minimizing the cross-entropy error is used through the whole network to fine-tune the weights for optimal classification. The cross-entropy error is represented by

$$E = -\sum_i e_i \log\hat{e}_i - \sum_i (1-e_i)\log(1-\hat{e}_i) \qquad (7)$$

where e_i is the label of the sample i and $\hat{e}_i$ is the classification result.

By training and fine-tuning the network, the final deep neural network is obtained. The neighborhood features of each pixel on the difference image are inputted into the deep neural network and the network outputs the class label of the pixel. Because a logistic output unit is used, the class label 0 represents the pixel being changed and the class label 1 represents the pixel being unchanged.

III. Experimental Study

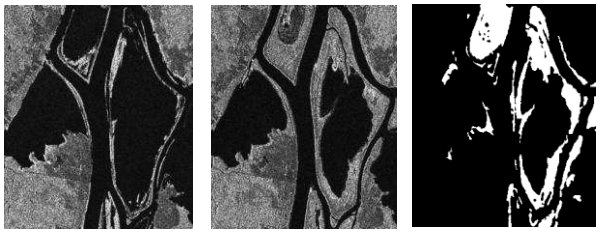
A. Instruction to Datasets

We adopt three datasets to test the proposed algorithm. The first dataset is the Ottawa dataset, which is a section (290×350 pixels) of two SAR images over the city of Ottawa acquired by RADARSAT SAR sensor. These images were registered by the automatic registration algorithm from A.U.G. Signals Ltd that is available through the distributed computing at www.signalfusion.com.

The second dataset is the Bern dataset, which is a section (301×301 pixels) of two SAR images acquired by the European Remote Sensing 2 satellite SAR sensor over an area near the city of Bern, Switzerland, in April and May 1999, respectively. Between the two dates, the river Aare flooded parts of the cities of Thun and Bern and the airport of Bern entirely. Therefore, the Aare valley between Bern and Thun was selected as a test site for detecting flooded areas [4].

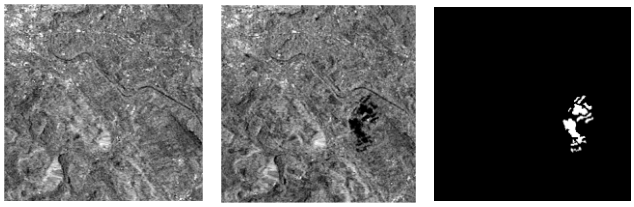
The third dataset is the Yellow River dataset, which is used in the experiments consists of two SAR images acquired by Radarsat-2 at the region of Yellow River Estuary in China in June 2008 and June 2009, as shown in Fig.7. It is worth noting that the two images are single-look image and four-look image, respectively. This means that the influence of speckle noise on the image acquired in 2009 is much greater than that of the one acquired in 2008. The huge difference of speckle noise level between the two images used may complicate the processing of change detection. The original size of these two SAR images acquired by Radarsat-2 is 7666×7692. They are too huge to show the detail information in such small pages. As shown in Fig.8, we select one typical area (280×450 pixels) for the experiment.

The available ground truths (reference images) in the three datasets were all created by integrating prior information with photo interpretation based on the input images.



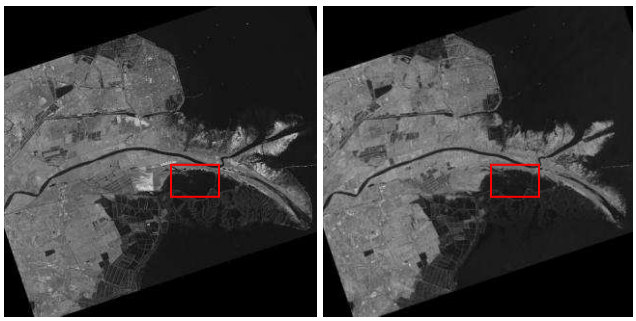
(a) (b) (c)

Fig.5. Multi-temporal images relating to Ottawa used in the experiments. (a) is the image acquired in July 1997 during the summer flooding, (b) is the image acquired in August 1997 after the summer flooding, (c) is the ground truth.



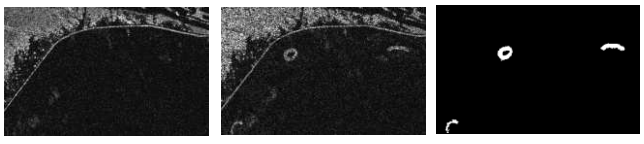
(a) (b) (c)

Fig.6. Multi-temporal images relating to the city of Bern used in the experiments. (a) is the image acquired in April 1999 before the flooding, (b) is the image acquired in May 1999 after the flooding, (c) is the ground truth.



(a) (b)

Fig.7. Multi-temporal images relating to Yellow River Estuary used in the experiments. (a) is the image acquired in June, 2008, (b) is the image acquired in June, 2009.



(a) (b) (c)

Fig.8. Multi-temporal images relating to one area of the Yellow River Estuary used in the experiments. (a) is the image acquired in June, 2008, (b) is the image acquired in June, 2009, (c) is the ground truth.

B. Evaluation Criteria and Experimental Settings

In this paper, we use some criteria to evaluate the change detection results. *FN* and *FP* represent the false negatives (changed pixels that undetected) and the false positives (unchanged pixels wrongly detected as changed), respectively. The overall error (*OE*) is defined as:

$$OE = FP + FN \tag{8}$$

Moreover, true positive represents the number of pixels that are detected as the changed area in both the reference image and the result. It is short for *TP*. And true negative is short for *TN*, which represents the number of pixels that are detected as the unchanged area in both the reference image and the result. The percentage correct classification (*PCC*) is given by

$$PCC = (TP + TN) \big/ (TP + FP + TN + FN) \tag{9}$$

As an overall evaluation criterion, *Kappa* statistic [29] is used to evaluate the effect of the image segmentation results. It is calculated as:

$$Kappa = \frac{PCC - PRE}{1 - PRE} \tag{10}$$

where

$$PRE = \frac{(TP + FP) \cdot (TP + FN)}{(TP + TN + FP + FN)^2} + \frac{(FN + TN) \cdot (FP + TN)}{(TP + TN + FP + FN)^2} \tag{11}$$

α is an important parameter in the process of choosing samples (seen as Section III.*A*.). We set α to 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9 and 0.95 for testing it. The relationship between α and the criteria on the Ottawa dataset is shown in Fig.9, Fig.10 and Fig.11. It is seen that the results are stable when α is between 0.45 and 0.7. When α is too small, the result is sensitive to noise which yields a high *FP*; when α is too large, the diversity of sample is decreased which yields a high *FN*. Furthermore, when α is set suitably, the proposed method can keep a balance between *FP* and *FN*.

For all datasets, every hidden layer is pre-trained 50 passes through the entire training set and a 25-250-200-100-1 network is used. In addition, a clustering algorithm RFLICM and a threshold algorithm GKI respectively used on difference images generated by LR operator are presented as comparative methods. DNN is short for the proposed method based on a deep neural network. It is noted that all images are not filtered.

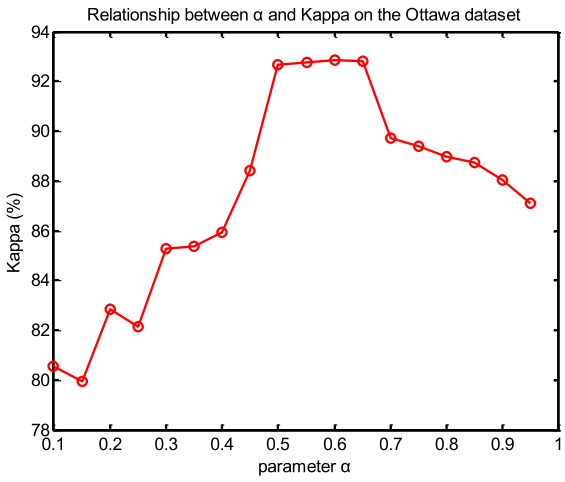


Fig.9. Relationship between α and *Kappa* on the Ottawa dataset

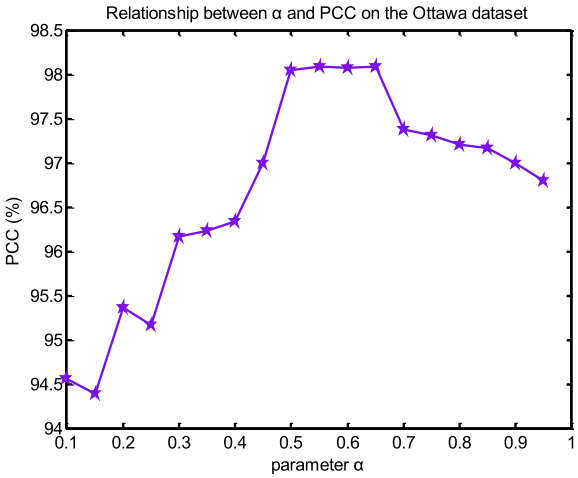


Fig.10. Relationship between α and *PCC* on the Ottawa dataset

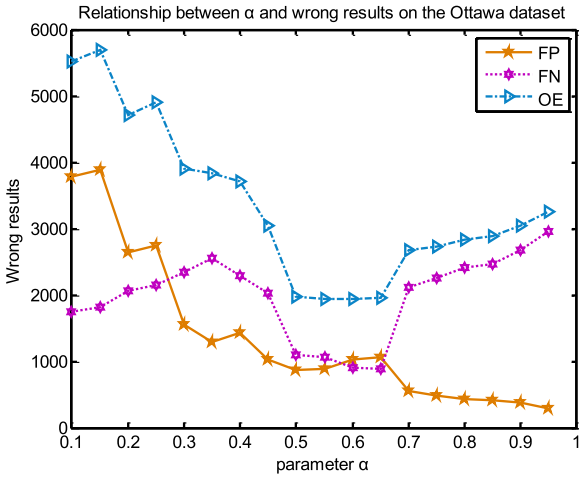


Fig.11. Relationship between α and the wrong results on the Ottawa dataset

C. Experimental Results and Analysis

In this section, we will exhibit the results of the main experiment and the two comparison experiments in two ways: the final maps in figure form and the criteria in tabular form. In

the table, the results of our proposed method are written in bold.

a. Results on the Ottawa Dataset

The change detection results generated by the proposed method and the two comparative methods on the Ottawa dataset are presented in Fig.12. As shown in Fig.12 (a), there are many white noise spots emerging on the black background. It is due to the necessity to search an optimal threshold by modeling data for the GKI method. Any little error in threshold may result in the presence of noise on the final map. As for RFLICM, the final map in Fig.12 (b) seems better but there are many missed detections. RFLICM which incorporates both local spatial and gray information is relatively insensitive to noise. By contrast, the proposed DNN algorithm applying deep learning has the best performance. This method has no requirements for the model or the data distribution. On the contrary, it has a strong ability to learn complicated functions. In TABLE I, the performances of the three methods are given intuitively. The *OE* yielded by DNN is the lowest and the *Kappa* yielded by DNN is the highest. Although DNN dose not result in the lowest *FP* and *FN*, it can balance the two criteria better. This point can be seen from the final map.

b. Results on the Bern Dataset

The results of the experiments on the Bern Dataset are illustrated in Fig.13 and listed in TABLE II. There are many noise spots on the final map generated by GKI (seen from Fig.13 (a)). The method can not get a good result when the considered images are not filtered. Although the sensitivity to noise of RFLICM is decreased, many changed areas are detected as the unchanged by RFLICM. Compared with the two above methods, DNN performs well seen form both the final map and the table. The *OE* yielded by DNN is equal to 290, lower than those by GKI and RFLICM. Moreover, as the overall evaluation, both the PCC and Kappa of DNN are among the best.

c. Results on the Yellow River Dataset

For the Yellow River dataset, the changed areas are relatively small. Fig.14 illustrates the final maps of the three methods on the dataset. GKI presents the worst performance. There are many noise spots on the final map. In Fig.14 (b), the final map obtained by RFLICM has many false alarms because of the existence of noise. This is due to that the influence of noise is great on the Yellow River dataset. Seen from the figure, the proposed method DNN is the best to complete the detection task. Moreover, quantitative analysis in TABLE III also declares this point. The *Kappa* yielded by DNN equal to 76.51% is much higher than that of 59.98% by RFLICM and that of 30.25% by GKI.

To sum up, the proposed approach fits the three situations where the changed areas appear large and scattered (the Ottawa dataset), centralized (the Bern dataset) and small (the Yellow River dataset). It also verifies its relatively far-ranging applicability.

(a) (b) (c)

Fig.12. Change detection results of Ottawa dataset achieved by (a) GKI, (b) RFLICM, (c) DNN.

TABLE I
CHANGE DETECTION RESULTS OF OTTAWA DATASET OBTAINED BY GKI, RFLICM AND DNN METHODS

Method	*FP*	*FN*	*OE*	*PCC*	*Kappa*
GKI	2811	214	3025	97.02%	89.49%
RFLICM	170	2535	2605	97.33%	89.35%
DNN	**883**	**1059**	**1942**	**98.09%**	**92.78%**

(a) (b) (c)

Fig.13. Change detection results of Bern dataset achieved by (a) GKI, (b) RFLICM, (c) DNN.

TABLE II
CHANGE DETECTION RESULTS OF BERN DATASET OBTAINED BY GKI, RFLICM AND DNN METHODS

Method	*FP*	*FN*	*OE*	*PCC*	*Kappa*
GKI	1382	127	1509	98.33%	56.93%
RFLICM	31	297	328	99.64%	83.77%
DNN	**147**	**143**	**290**	**99.68%**	**87.31%**

(a) (b) (c)

Fig.14. Change detection results of Yellow River dataset achieved by (a) GKI, (b) RFLICM, (c) DNN.

TABLE III
CHANGE DETECTION RESULTS OF YELLOW RIVER DATASET OBTAINED BY GKI, RFLICM AND DNN METHODS

Method	*FP*	*FN*	*OE*	*PCC*	*Kappa*
GKI	5252	115	5367	95.74%	30.25%
RFLICM	1361	172	1533	98.78%	59.98%
DNN	**47**	**480**	**527**	**99.58%**	**76.51%**

IV. CONCLUDING REMARKS

This paper has presented a novel change detection algorithm specifically focusing on analyzing multi-temporal SAR images based on a deep neural network. Deep neural

network has a powerful ability to learn features, and to represent images in an abstract way. For a deep architecture used on change detection, it is not necessary to model data and the result is robust to noise. Compared with a clustering method RFLICM and a thresholding method GKI, the proposed method exhibits good performance.

The experiments on the datasets which have different features verify the effectiveness of the proposed methods. In the future, we will try to extract the change detection results directly from the two original images rather than a difference image using deep leaning. And we will do research on the application of deep learning to the change detection in two images which have not been registered or obtained by different sensors.

REFERENCES

[1] R. J. Radke, S.Andra, O.Al-Kofahi, and B. Roysam, "Image change detection algorithm: A systematic survey," *IEEE Trans. Image Processing*, vol. 14, no. 3, pp. 294 – 307, Mar. 2005.

[2] E. E. Kuruoglu, and J. Zerubia, "Modeling SAR images with a generalization of the Rayleigh distribution," *IEEE Trans. Image Processing,* vol. 13, no. 4, pp. 527–533, 2004.

[3] L. Bruzzone and D. F. Prieto, "An adaptive semiparametric and context-based approach to unsupervised change detection in multi-temporal remote-sensing images," *IEEE Trans. Image Processing*, vol. 11, no. 4, pp. 452 – 466, 2002.

[4] Y. Bzai, L. Bruzzone and F. Melgani, "An unsupervised Approach Based on the Generalized Gaussian Model to Automatic Change Detection in Multitemporal SAR Images," *IEEE Trans. Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 874 – 887, 2005.

[5] F. Bujor, E. Trouvé, L. Valet, J. M. Nicolas, and J. P. Rudant, "Application of log-cumulants to the detection of spatiotemporal discontinuities in multitemporal SAR images," IEEE Trans. Geosci. Remote Sens., vol. 42, no. 10, pp. 2073–2084, Oct. 2004.

[6] M. Gong, Y. Cao and Q. Wu, "A neighborhood-based ratio approach for change detection in SAR images," IEEE Geoscience and Remote Sensing Letters, vol. 9, no.2, Mar. 2012.

[7] P. S. Liao, T. S. Chen and P. C. Chung, "A fast algorithm for multilevel thresholding," *J. Inf. Sci. Eng.*17 (5): 713 – 727.

[8] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognit.*, vol. 19, pp. 41-47, 1986.

[9] G. Moser and S. B. Serpico, "Generalized minimum-error thresholding for unsupervised change detection from SAR amplitude imagery," *IEEE Trans. Geoscience and Remote Sensing*, vol. 44, no. 10, pp. 2972 – 2982, Oct. 2006.

[10] S.Krinidis and V.ChatzisF, "A robust fuzzy local information C-means clustering algorithm," *IEEE Trans. Image Processing*, vol. 19, no. 5 pp. 1328 – 1337, May. 2010.

[11] M. Gong, Z. Zhou and J. Ma, "Change detection in Synthetic Aperture Radar images based on image fusion and fuzzy clustering ," *IEEE Trans. Image Processing*, vol. 21, no. 4, Apr. 2012.

[12] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, Nov. 2010.

[13] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing," *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 127–135.

[14] S. Osindero and G. E. Hinton, "Modeling image patches with a directed hierarchy of Markov random field," *Neural Information Processing Systems* 20 (*NIPS*), pp. 1121–1128, 2008.

[15] A. Graves, M. Liwicki, S. Fern´andez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[16] D. C. Cires¸an, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[17] M. Ranzato, V. Mnih, J. Susskind, and G. Hinton, "Modeling natural images using gated MRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2206–2222, 2013.

[18] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[19] W. K. Wong and M. Sun, "Deep learning regularized fisher mappings," *IEEE Trans. Neural Network*, vol. 22, no. 10, pp. 1668–1675, Oct. 2011.

[20] A. Stuhlsatz, J. Lippel and T. Zielke, "Feature extraction with deep neural networks by a generalized discriminant analysis," *IEEE Trans. Neural Network and Learning Systems*, vol. 23, no. 4, pp. 596–608, 2012.

[21] D.Prokhorov, "A convolutional learning system for object classification in 3-D lidar data," *IEEE Trans. Neural Network*, vol. 21, no. 5, pp. 858–863, May. 2010.

[22] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex," *Journal of Physiology* (London), vol. 160, pp. 106–154, 1962.

[23] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[24] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Neural Information Processing Systems* 19 (*NIPS*), pp. 153–160, 2007.

[25] H. Larochelle, Y. Bengio, "Classification using discrimination Restricted Boltzmann Machines," ACM, pp. 536–543, 2008.

[26] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Leaning*, vol. 2, no.1, pp. 1–127, 2009.

[27] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504 – 507, Jul. 2006.

[28] G. E. Hinton, "A practical guide to training Restricted Boltzmann Machines," *Springer Link*, *Lecture Notes in Computer Science*, vol. 7700, pp. 599 – 619, 2012.

[29] G. H. Rosenfield, A. Fitzpatrick-Lins, "A coefficient of agreement as a measure of thematic classification accuracy," *Photogrammetric Engineering and Remote Sensing*, vol.52, pp. 223−227, 1986.