Change detection in multi-temporal SAR images using dual-channel convolutional neural network

Liu, Tao; Li, Ying; Cao, Ying; Shen, Qiang

Published in:
Journal of Applied Remote Sensing

DOI:
10.1117/1.JRS.11.042615

Publication date:
2017

Citation for published version (APA):
Liu, T., Li, Y., Cao, Y., & Shen, Q. (2017). Change detection in multi-temporal SAR images using dual-channel convolutional neural network. *Journal of Applied Remote Sensing*, *11*(4), [042615]. https://doi.org/10.1117/1.JRS.11.042615

Change detection in multi-temporal SAR images using dual-channel convolutional neural network

Tao Liu,[a] Ying Li,[a,*] Ying Cao,[a] Qiang Shen[b]
[a]Northwestern Polytechnical University, School of Computer Science, 127 Youyixi road, Xi'an City, PRC, 710072
[b]Department of Computer Science, Institute of Mathematics, Physics and Computer Science, Aberystwyth University, SY23 3DB Aberystwyth, U.K

Abstract. This paper proposes a novel model of dual-channel convolutional neural network (CNN) that is designed for change detection in SAR images, in an effort to acquire higher detection accuracy and lower misclassification rate. This network model contains two parallel CNN channels, which can extract deep features from two multi-temporal SAR images. For comparison and validation, the proposed method is tested along with other change detection algorithms on both simulated SAR images and real-world SAR images captured by different sensors. Experimental results demonstrate that the presented method outperforms the state-of-the-art techniques by a considerable margin.

Keywords: change detection, SAR image, dual-channel CNN, deep learning.

***Ying Li, E-mail: lybyp@nwpu.edu.cn

1 Introduction

Change detection is a process to identify any changes that have occurred between two images of the same scene taken at different times. It has played an ever increasingly important role in both civil and military applications. Change detection in synthetic aperture radar (SAR) images has been widely used, including deforestation detecting[1], disasters monitoring[2], damage assessment[3], urban expansion[4], and environmental studies[5]. Generally, the goal of change detection is to divide samples into changed areas and unchanged areas. Most change detection methods can be grouped into three categories: *pixel-based change detection* (PBCD), *object-based change detection* (OBCD), *and hybrid change detection*[6] (HCD). Although many existing methods have helped boost the performance of change detection, there are still challenging tasks in automatic detection of changes in SAR images. Most conventional change detection methods are

commonly based on handcrafted features, relying heavily on specific domain knowledge. However, the design of handcrafted features can be tedious and is typically suboptimal[7].

Deep neural networks (DNNs) are designed to simulate the human's nervous system[8,9]. Generally speaking, there are five major types of DNN, including deep belief networks (DBN) [10], recurrent neural networks (RNN) [11], stacked auto-encoders (SAE) [12], sparse coding [13,14], and convolutional neural networks (CNN) [15,16]. A number of approaches to developing deep neural networks have recently been widely applied with improved performance in problem-solving[17,18], but CNNs have not yet been applied on change detection in SAR images since the original proposal[19]. A CNN is a trainable multilayer architecture composed of multiple feature-extraction stages. Each stage consists of two different layers: convolution layer and pooling layer. A typical CNN contains several such layers, followed by one or more traditional, fully connected layers and a final classification layer. The mechanism of CNNs can be explained as an application of the receptive field[20]. In a convolution layer, the input data is convolved with 2D kernels, which then go through the activation function to form the output data (features). Pooling can be used to: reduce the dimensionality of the features, offer invariance and increase the range of receptive field. Two main types of pooling operation are max-pooling and averaging-pooling, aiming to obtain the maximal value or the averaged value from a 2D image patch, respectively. Compared with fully-connected neural networks, one of the outstanding advantages of CNNs is that much less parameters are required to be trained in developing an applied CNN.

In this paper, CNNs are introduced to explore a new approach for change detection in SAR images with higher accuracy. A model made up with two parallel channels of a CNN (which is named: dual-channel convolutional neural network, DC-CNN) is proposed. It is experimentally

proven to be able to produce improved performance over state-of-the-art change detection methods.

The rest of this paper is organized as follows: In Section 2, the proposed method will be introduced, including the architecture of the new network model and its input and output, the data flowing path, and the training process associated with the model. We will describe the datasets and experimental setup and discuss the experimental results in Section 3, empirically comparing the proposed method with other approaches. In Section 4, we will summarize the work and discuss future work.

The specific contributions of this paper are listed below:

(1) A novel network model based on dual-channel convolutional neural network is designed as feature-extractor and classifier.

(2) The proposed model can be used in change detection, achieving promising results on both simulated SAR images and real-world SAR images.

(3) The proposed approach is more efficient than the state-of-the-art techniques, e.g., compared with traditional PBCD, the proposed approach does not require any pre-processing; and compared with traditional OBCD, this approach does not require pre-segmentation.

(4) A change detection map is computed as the output from two original images, without the procedure of any kind of pre-processing or post-processing[21]; this differs from most of the existing methods which are either a direct hybrid of certain existing techniques or an improvement on part of a certain algorithm.

2 Proposed Method

Two important challenges in designing an effective change detection model using a CNN are: 1) It needs to be sufficiently flexible to identify the underlying features in manifold forms hidden in

the images; 2) It needs to be sufficiently effective to determine the changes between two SAR images with a high accuracy. In the final detection outcome, the coincidence range between the detected result and actually changed area is of particular significance. To address these challenges the following approach is proposed.

2.1 Dual-Channel CNN (DC-CNN) for Change Detection

A DC-CNN detection model can be illustrated by the architecture shown in Fig. 1. As far as the applicable range is concerned, this model is specifically designed for change detection between two SAR images that have been geometrically rectified and registered. Registration is a process to match precisely two pictures taken at different times. A CNN model with two channels is designed to meet the corresponding pixels within the given image-pair.

As mentioned above, a classical CNN generally has two types of layer: convolution layers and pooling layers, which are arranged in an alternative order. In the proposed DC-CNN, the CNN in each channel consists of four convolution layers and two pooling layers as shown in Fig. 2. In particular, a max-pooling layer and an averaging-pooling layer work after the first and the fourth convolution layer, respectively. A large number of image-pair to be examined for change detection will be divided into many image-patch-pairs before being processed by a recognition model. After the two image patches of one image-patch-pair are fed to the two channels of DC-CNN simultaneously, both of them are convolved with learnt kernels and put through an activation function in sequence for feeding to the subsequent layer.

Note that the network adopted in our work consists of two channels, each of which is a convolutional network, as shown in Fig 2. The kernels used in the corresponding positions of the two channels are the same in size, but not the same in weights. The activation function adopted in this model is the Rectified Linear Units (ReLU) [22] owing to its popularity. The feature maps

generated from the final averaging-pooling layer of each channel are concatenated to construct a required feature vector.

In order to extract features from an original image-pair, a window sliding through each image selects a pixel patch that will be used as the input into one of the two channels. While the sliding window is moving from left-top to right-bottom, all the pixels belonging to each image will be selected one by one. To enable a reasonably accurate comparison between the corresponding pixels of two original images, in an effort to determine where the changes may have occurred, the size of each pixel patch should not be too large. Otherwise(where the pixel patch is large), a sizeable image area would be involved, leadings to the introduction of irrelevant information, thereby causing less accurate change detection. As shown in Fig. 2, a sliding window with the size of 5×5 is used in the following experimental studies. That is, a 5×5 pixel patch from each image is chosen to form an input image-patch pair at a time. After features are extracted by the use of the convolutional layer, the dimensionality of features is always high. This makes the computing of features rather expensive, and may lead to overfitting. Thus, it is a standard choice to use pooling to reduce feature dimensionality. In devising the pooling process, the average or the maximum of an image region that is of a size of $m\times n$ is used to replace the features of that region on the input feature map, thereby greatly reducing the dimensionality of the feature maps. In addition, this operator helps ensure that the same deep feature can be extracted, even if image involve small translation or rotation.

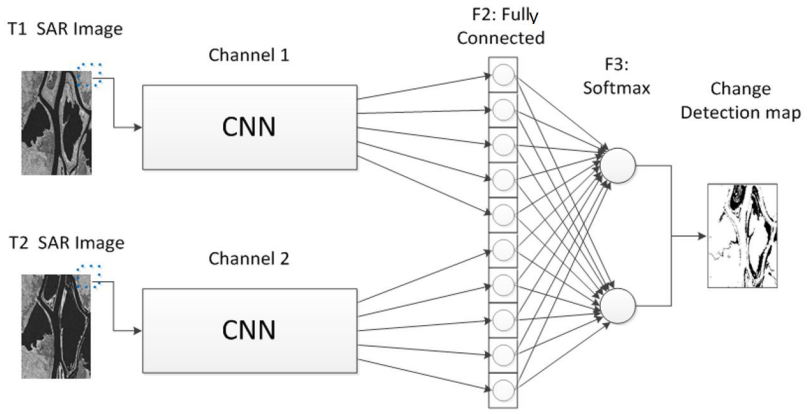

Fig. 1. Framework of dual-channel CNN model

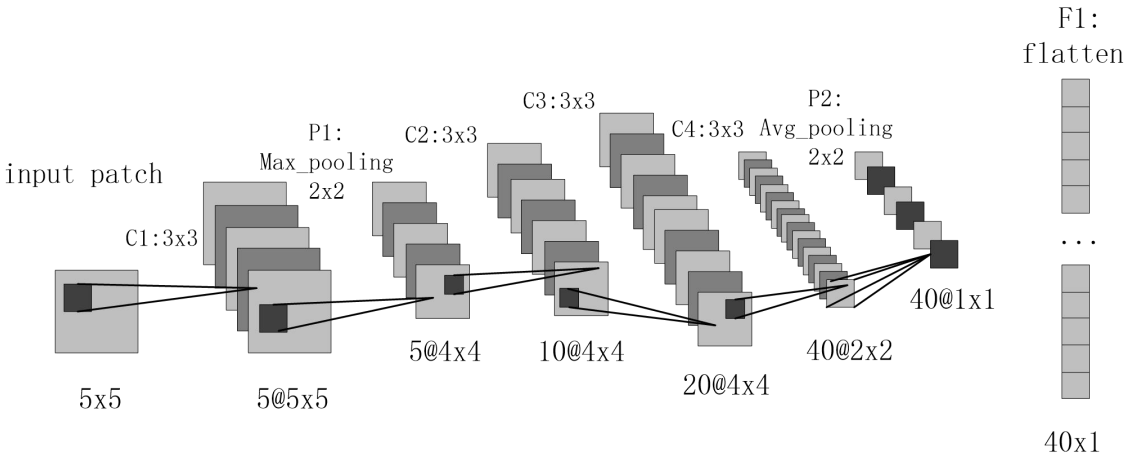

Fig. 2. Structure of the CNN in each dual-channel

As shown in Fig. 1, channel 1 takes the patch cut from the SAR image acquired at time T1, and channel 2 takes the corresponding patch with the same size at time T2. The sliding windows and the input patches will move from left-top to right-bottom throughout the two images synchronously. The CNN in each of the two channels consists of several layers, as shown in Fig.2 (given the size of an input image patch being 5×5). In particular, seven layers are involved in this work: four 3×3 convolution layers (C1-C4 as shown in Fig 2), two 2×2 pooling layer (P1-P2), and one flatten layer (F1) which will concatenate the feature maps resulting from P2 into one feature vector. (We use "3×3 convolution layer" to note a convolution layer whose kernel size is 3×3, use "2×2 pooling layer" to note a pooing layer whose kernel size is 2×2.)

After one pair of image-patches have been passed through the two CNN channels, two feature vectors are produced, which will be concatenated into one single feature vector and fed to a fully-connected layer (i.e.,F2), as shown in Fig 1. The output layer (F3) is a softmax layer with two neurons that are fully-connected to the F2 layer, where the probabilities given by the two neurons are utilised to determine whether the pixel which is in the centre of this input image patch changes or not. The change detection map is finally achieved by sliding the windows from left-top to right-bottom over the two images to obtain the required result over each image pixel.

2.2 Training of DC-CNN

As each pixel in an image has a neighborhood except the pixels located in image edges, an image-pair can lead to many "patch-pairs" if every neighborhood is regarded as one patch. Since the DC-CNN can be seen as a supervised model, it also needs certain image-pairs with ground truth for training. To constitute a training set, a certain part of the patch-pairs are randomly selected from the image-pairs which are extracted from the original images. Another part of the patch-pairs are randomly selected from the remainder to form a validation set. Usually, the proportion of the pixels in a training set is 0.5%-1% among the total pixels of these images, and the proportion of a validation set is about 1%. At the test stage, all samples are fed into the learned DC-CNN, and the final change detection results are then computed.

During training stage, firstly we feed one image patch (e.g., with a size of 5×5) of a labeled patch-pair to the first channel of DC-CNN, and the other image patch of the same patch-pair to the second channel synchronously. Secondly, after the output from the DC-CNN is achieved, we compare it with the central pixel's label of the patch-pair. Based on the result of comparison, the weights, biases, and other model parameters are revised by back-propagation algorithm[23]. The

DC-CNN will have to be trained for many epochs before it converges and remains stable. In order to improve the accuracy of change detection, a local response normalisation(LRN) procedure is employed during the training of DC-CNN. LRN aims to achieve topoinhibition by the use of lateral inhibition, and it works effectively when ReLU is adopted as the activation function. Then the stage of training is regarded complete.

3 Experiments and Results

3.1 Data Sets

To assess the capability of DC-CNN, a pair of synthetic images are produced to act as the simulated dataset, and two real SAR image data sets acquired by distinct sensors are chosen for the experiments.

We firstly use a pair of artificially synthesized images. The method of producing the images is similar to what is used in experiments described in [6]. The ground truth image (500×500 pixels) is shown in Fig. 3(c), and the two different backscatter intensities are chosen to generate two multi-temporal flood images (shown in Fig. 3(a) and Fig. 3(b)), one of which is relatively bright (i.e., -10dB) for the background and the other relatively dark (i.e., -22dB) for the six round flood objects. Before being used in the experiments, the background image and the flood image are degraded by speckle noise of a gamma distribution to derive the simulated image pairs (ENL = 5): Fig. 3(a) is used as the first-time noisy image (the background image) and Fig. 3(b) as the second-time noisy image (the flood image).

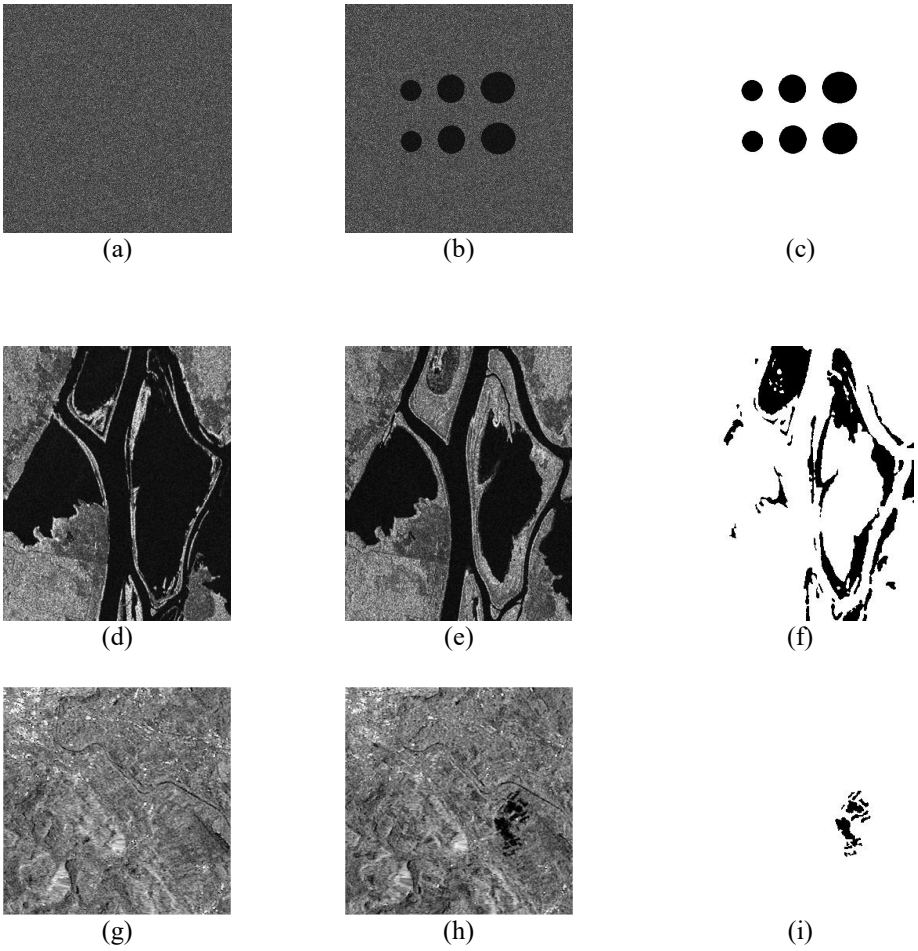

Fig. 3. Detection targets and their ground-truths used in the experiments

In our experiments, real multi-temporal SAR Data Sets are also used. Fig. 3(d) and (e) show a pair of SAR images (290×350 pixels) with C-band and HH polarization. They were acquired by Radarsat-1 satellite SAR sensor over the lakes near Ottawa, Canada, in July and August 1997, with the ENL being 12.5 and 12.6, respectively. Fig. 3(f) shows the ground truth obtained by integrating prior information with photograph interpretation. Fig. 3(g) and (h) show another SAR image-pair consisting of two C-band SAR images (301×301 pixels) acquired by the satellite of ERS-2 before and after floods over Berne, Switzerland, on April and May, 1999, respectively.

The ENL of the first image is 10.89 and that of the second is 9.26 [6]. Fig. 3(i) shows the ground truth of the change detection map that is manually created with a visual interpretation[6].

To facilitate comparison, by using notations in accordance with what is used in [6], we adopt the following abbreviations to present certain evaluation indexes hereafter: UP stands for the number of under-detected pixels; OP for the number of over-detected pixels; and OE for the number of overall error pixels.

3.2 Experimental Results

To illustrate the performance of DC-CNN, state-of-the-art algorithms are used in our experiments for comparison, including: the supervised manual trial-and-error procedure[24] (MTEP), the change detection threshold selection method based on the histogram ratio (named as "Xiong algorithm"[24]), the fully automated and time efficient extraction algorithm that can deal with non-bimodal histograms (named "M3 method"[25]), and the unsupervised algorithm-level fusion scheme of hybrid change detection (UAFS-HCD[6]). As mentioned above, Data sets used in the experiments include both simulated Data Sets and real SAR Data Sets acquired by various sensors. Each image-pair in the data sets include two multi-temporal SAR images, which are obtained in times T1 and T2, respectively.

3.2.1 Simulated images

We compare the proposed method quantitatively with other algorithms when dealing with the same simulated images given in Fig. 4(b) and (c) that are under Gamma noise. The results obtained from the various methods are listed in Table 1 and their change detection maps are shown respectively in Fig. 4(d), (e), (f), (g), and (h).

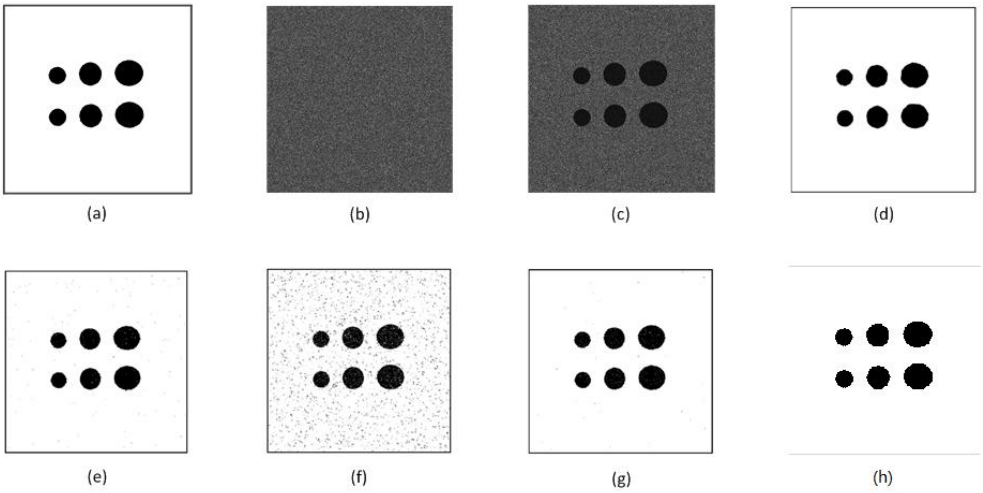

Fig. 4. Simulated images and results: (a) Ground truth. (b) Background image in Gamma noise. (c) Flood image in Gamma noise. (d) Change detection map with MTEP[24]. (e) Change detection map with Xiong algorithm[24]. (f) Change detection map with M3[25]. (g) Change detection map with UAFS-HCD[6]. (h) Change detection map with DC-CNN

Table 1 Experimental results on simulated SAR images.

Method	OP	UP	OE
MTEP[24]	185	**220**	405
Xiong algorithm[24]	46	898	944
M3[25]	3066	814	3880
UAFS-HCD[6]	**24**	696	720
DC-CNN	42	333	**375**

Fig. 4(h) visually shows that the proposed method DC-CNN achieves accurate detection results, which is also confirmed by the results of Table 1. The DC-CNN method is characterized by 42 over-detected pixels and 333 under-detected pixels, which is generally satisfactory, only having one incorrect detection overall more than the least error performer among all the methods compared.

3.2.2 Radarsat-1 SAR Images

In dealing with the real-world SAR images as given in Fig. 5(b) and (c) that were obtained from Radarsat-1, the results of applying the various methods are listed in Table 2 and their change detection maps are shown respectively in Fig. 5(d), (e), (f), (g), and (h).

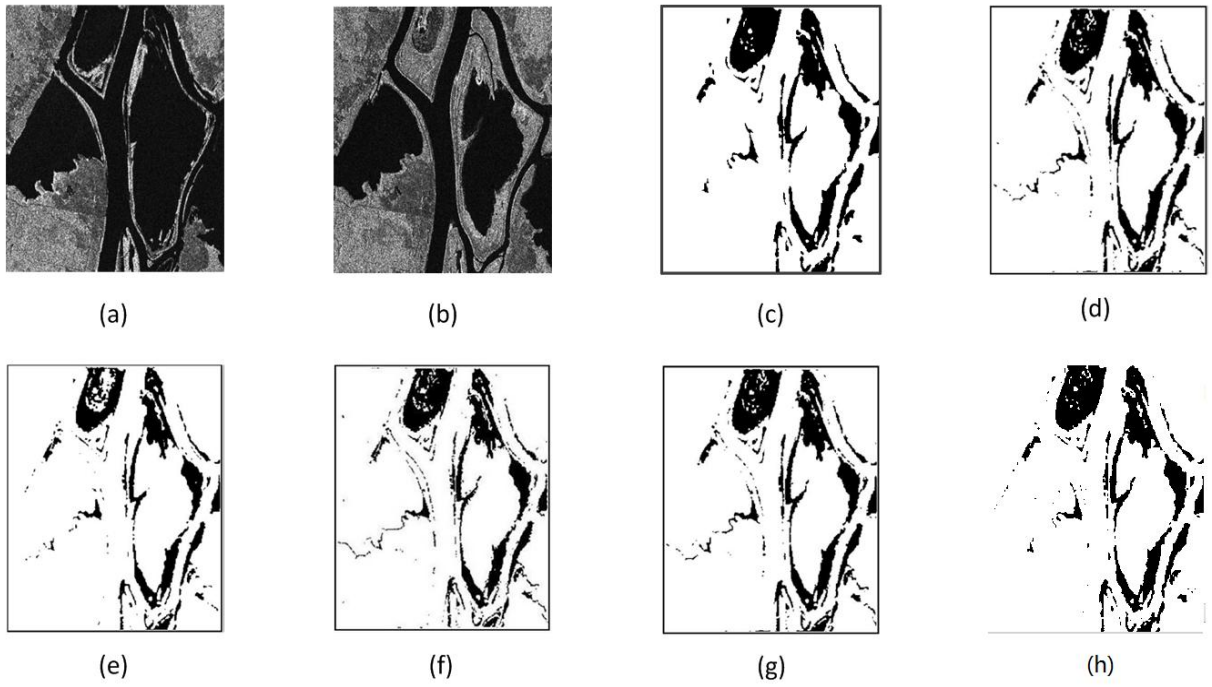

Fig. 5. Radarsat-1 SAR images and results: (a) Flood image. (b) Reference image. (c) Ground truth (the black pixels show the detected floods). (d) Change detection map with MTEP[24]. (e) Change detection map with Xiong algorithm[24]. (f) Change detection map with M3[25]. (g) Change detection map with UAFS-HCD[6]. (h) Change detection map with DC-CNN.

Table 2 Experimental results on Ottawa.

Method	OP	UP	OE
MTEP[24]	1222	1912	3134
Xiong algorithm[24]	665	3292	3957
M3[25]	1793	2847	4640
UAFS-HCD[6]	797	2726	3523
DC-CNN	**518**	**1081**	**1599**

From Fig 5(h) we can see that the difference map produced by DC-CNN is obviously clearer and more recognizable than other difference maps. Quantitative results of the changes detected using different methods are listed in Table 2, from which we can also see that the proposed method obtains the best result with the lowest under-detected pixels: 1081, the lowest over-detected pixels:518,and the lowest overall error pixels: 1599. Compared with the result of MTEP which has the second best performance: 1912 under-detected pixels and 3134 overall

error pixels, DC-CNN's error rate is only about half of them. This demonstrates that the proposed model offers a substantially improved performance.

3.2.3 ERS-2 SAR Images

We also carried out further experiments on the real-world SAR images given in Fig. 6(b) and (c) that were obtained from ERS-2. The results of using the various methods are listed in Table 3 and their change detection maps are shown respectively in Fig. 6(d), (e), (f), (g), and (h).

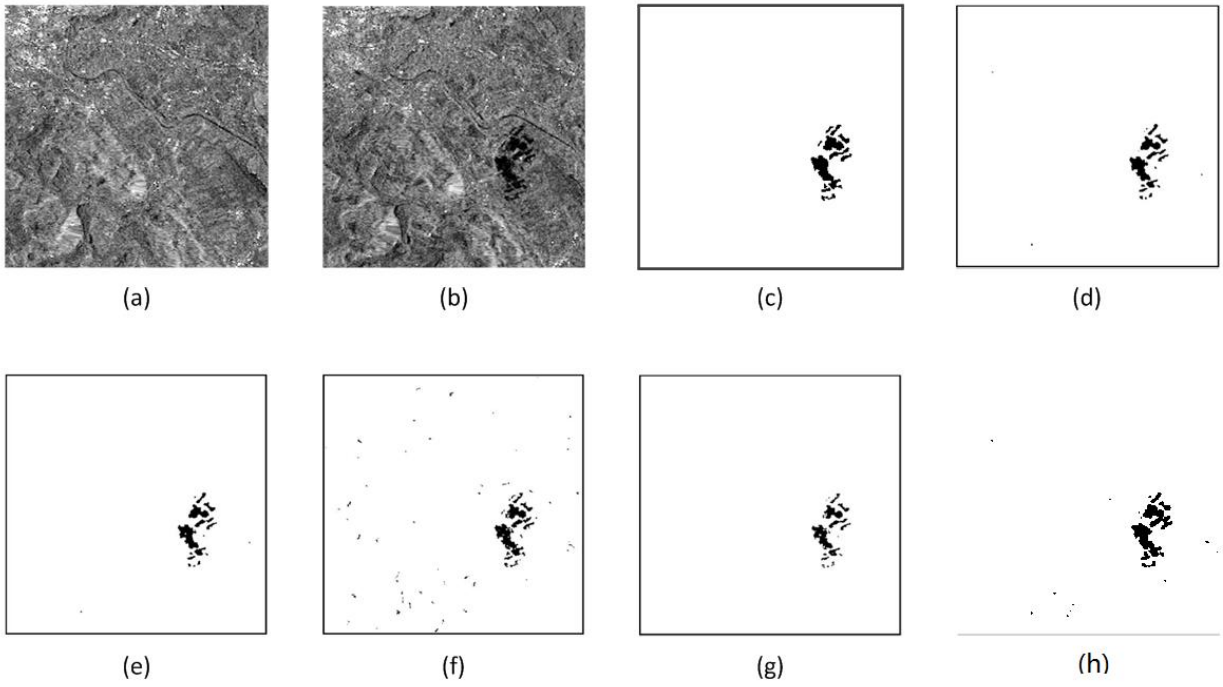

Fig. 6. ERS-2 SAR Images and results (the black pixels show the detected floods): (a) Reference image. (b) Flood image. (c) Ground truth. (d) Change detection map with MTEP[24]. (e) Change detection map with Xiong algorithm[24]. (f) Change detection map with M3[25]. (g) Change detection map with UAFS-HCD[6] (h) Change detection map with DC-CNN.

Table 3 Experimental results on Berne image.

Method	OP	UP	OE
MTEP[24]	140	272	412
Xiong algorithm[24]	111	317	428
M3[25]	464	155	619
UAFS-HCD[6]	**64**	322	386
DC-CNN	226	**102**	**328**

Fig 6(h) shows the prominent result obtained by the proposed DC-CNN. We can even recognize by naked eyes that the change detection map obtained by DC-CNN is by far better than other difference maps. This conclusion is also based on the evidence given by Table 3, from which we can see that the proposed method obtains the lowest under-detected pixels: 102, and the lowest overall error pixels: 328. In other words, the proposed approach outperforms the state-of-the-art techniques in real-world change detection.

Together, the above experiments have jointly demonstrated that the proposed method offers superior performance over many state-of-the-art algorithms, in the detection of changes in both simulated and real-world SAR images.

3.3 Influence of Parameters

3.3.1 Size of input patch

A comparative study is performed here on the effects of using a different size of the input patch. The patch size of 5×5 is empirically used in the proposed DC-CNN model to deal with the change detection. This is based on the results of relevant experimental investigations, where 7 alternative patch sizes are selected for conducting the comparison, which are 3×3, 7×7, 9×9, 11×11, 13×13, 15×15, 17×17, respectively. Accordingly, the DC-CNN model needs to be modified to work compatibly with this change of the input size. In particular, when the image patch size is 3×3, all of the four convolution layers (i.e., C1-C4) execute the convolution manipulation with padding, resulting in no change of the size of image patch after the convolution. The size of image patch will only be reduced after going through the two pooling layers (i.e., max-pooling of P1 and averaging-pooling of P2). The convolution manipulation in

convolution layers is slightly different as the size of image patch becoming larger. Convolution layers C1 and C2 work with padding of size 7×7, while C3 and C4 perform the convolution manipulations without padding. In size 9×9, only convolution layer C1 conducts convolution manipulation with padding while the remaining three do the opposite ones. As the size of image patch reaching 11×11, all of the four convolution layers do the convolution without padding, which is just in the opposite side of that with the image patch size of 3×3 at the beginning. Up to now, the size of feature map produced by the averaging-pooling (P2) is 1×1 when the size of image patch is increasing from 3×3 to 11×11. The CNN remains the same structure when the size is 13×13 as that of 11×11, but generates the feature map in the size of 3×3. As the size of image patch goes up to 15×15 or larger, the corresponding modification for the CNNs is implemented by setting the stride of the convolution layers and pooling layers. Fig.7 shows the results of this experiment with various image-patch sizes.

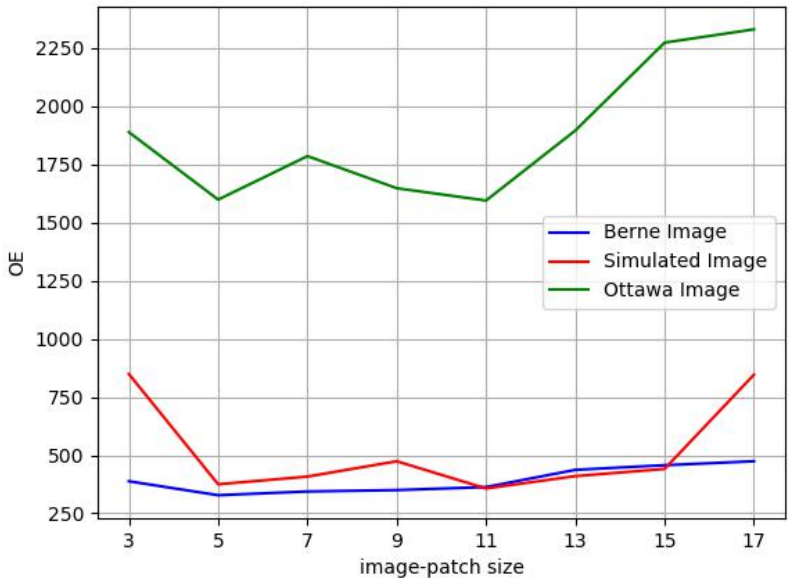

Fig. 7. Overall error obtained by varying the size of image patch on three different datasets

Table 4 presents the results of this experimental investigation by varying the size of image-patch on three different datasets, with the figures are plotted in Fig. 7. As can be seen, the relative better performance has been achieved when the size of image patch is 5×5 and 11×11 on the simulated dataset. This is also verified by the use of the SAR image obtained from Radarsat-1. The overall error collected using SAR image from ERS-2 indicates the best performance can be achieved when the size is 5×5. Moreover, the accuracy of change detection deteriorates as the size of image patch goes higher than 11 on all of the three datasets.

Table 4 Overall errors with respect to different image-patch sizes.

Method	**3×3**	**5×5**	**7×7**	**9×9**	**11×11**	**13×13**	**15×15**	**17×17**
Simulated Image	849	375	408	474	**357**	410	441	846
Ottawa Image	1889	**1599**	1786	1907	1648	1896	2274	2331
Berne Image	388	**328**	344	350	363	437	457	474

This makes logical sense because if the image-patch size is too large, the samples would contain too much unrelated information, thereby weakening 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, which will lead to more similar change information between them, therefore generating worse results with blurry boundary. Thus, the size of image patch will be set as 5×5 in the following experiments.

3.3.2 Number of neurons in the Fully-connected Layer

When designing a DC-CNN model, we use the powerful feature mapping ability contained in convolution layers to extract features. However, after many convolution layers and pooling layer have been used and many abstract features obtained, how to merge them together becomes a significant issue that requires careful consideration. One efficient method is to fuse these features with a fully-connected layer, so we examine the effects of using a different number (say 5,10,

50,100, 150, or 200) of neurons in a fully-connected layer. Fig. 8 and Table 5 show the corresponding experimental results.

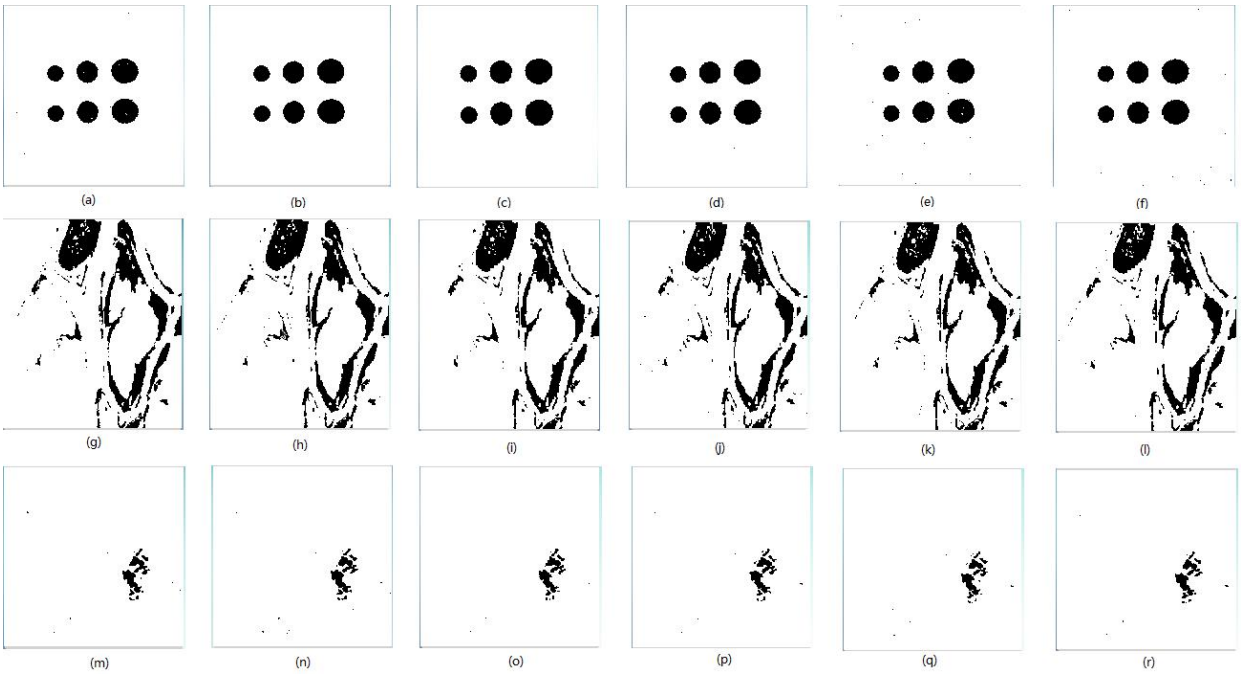

Fig. 8. Experimental results of change detection vs. different numbers of neurons in the fully-connected layer (a) Simulated images with 5 neurons. (b) Simulated images with 10 neurons. (c) Simulated images with 50 neurons. (d) Simulated images with 100 neurons. (e) Simulated images with 150 neurons. (f) Simulated images with 200 neurons. (g) Radarsat-1 SAR images with 5 neurons. (h) Radarsat-1 SAR images with 10 neurons. (i) Radarsat-1 SAR images with 50 neurons. (j) Radarsat-1 SAR images with 100 neurons. (k) Radarsat-1 SAR images with 150 neurons. (l) Radarsat-1 SAR images with 200 neurons. (m) ERS-2 SAR images with 5 neurons. (n) ERS-2 SAR images with 10 neurons. (o) ERS-2 SAR images with 50 neurons. (p) ERS-2 SAR images with 100 neurons. (q) ERS-2 SAR images with 150 neurons. (r) ERS-2 SAR images with 200 neurons.

Table 5 Overall errors with respect to different numbers of neurons in the fully-connected layer

number of neurons	**5**	**10**	**50**	**100**	**150**	**200**
simulated images	597	**375**	435	520	696	710
Radarsat-1 SAR Image	1799	**1599**	1686	1793	1754	1868
ERS-2 SAR image	330	328	**307**	316	325	311

The results showed in Fig. 8 and table 5 are acquired under the prerequisite that the two channels of the DC-CNN both have the same structure with the neuron numbers being “5-10-20-40”. This means that each channel has 4 convolution layers, with the first convolution layer of

each channel having 5 neurons leading to 5 feature maps, max-pooling and avg-pooling make no difference for the number of features maps. The second convolution layer having 10 neurons leading to 10 feature maps, and so on. The fully-connected layer may be viewed as implementing a dimensionality transformation from one dimensionality to another; the use of an appropriate number of its neurons can preserve useful information embedded in the input feature maps. As can be seen in Table 5, the best performance is achieved when the number of neurons is 10, regarding simulated images and Radarsat-1 SAR images. The overall error increases as the number of neurons becomes larger than 10. Although the overall error is lowest when the number of neurons is 50 regarding the ERS-2 SAR image, there is no significant changing of the performance overall.

3.3.3 Number of Feature Maps

We have discussed the impact of the neuron numbers in the fully-connected layer on the effectiveness of a DC-CNN model. The number of feature maps in every convolution layer may also have an impact upon the change detection results of DC-CNN. The two images or image-patches processed in each DC-CNN channel have a high similarity between each other, so the features to be extracted from both images may also possess certain correspondence. Based on this observation, we enforce the two DC-CNN channels to be of the identical size and shape (but not the same parameter values). In our experiment, we compare four types of typical model structure, namely "gradually-decrease", "stagger", "spindle", and "gradually-increase" arrangements. The feature map numbers in each of these types are configured in the following manner: "gradually-decrease" 50-40-30-20, "stagger" 5-10-20-10, "spindle" 20-30-30-20, and "gradually-increase" 5-10-20-30. For example, the model structure of "gradually-decrease" arrangement has the structure of "50-40-30-20", meaning that this DC-CNN model has 4

convolution layers in either of the two channels, where the number "50" in the notation "50-40-30-20" means that the first convolution layer produces 50 feature maps, the second convolution layer produces 40 feature maps, etc. The experimental results are shown in Fig. 9 and Table 6.

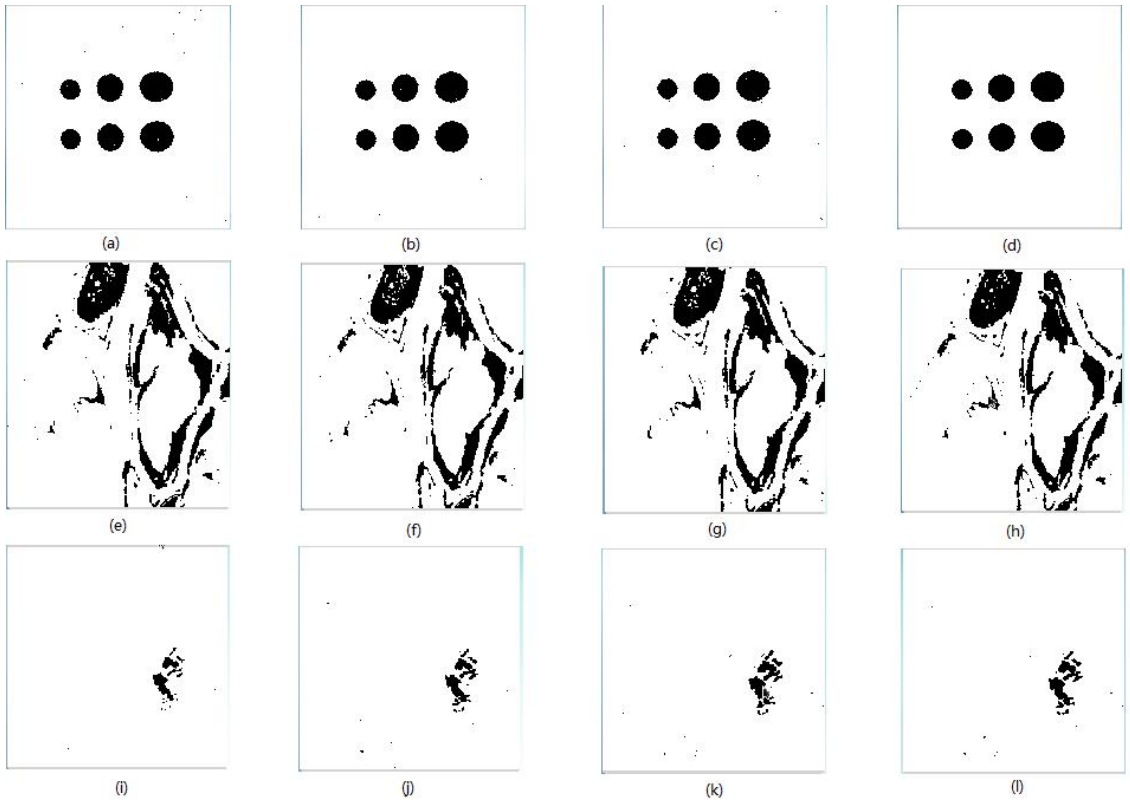

Fig. 9. Results of change detection vs. different numbers of feature maps in every convolution layer (a) Simulated images with a "gradually-decrease" arrangement. (b) Simulated images with a "stagger" arrangement. (c) Simulated images with a "spindle" arrangement. (d) Simulated images with a "gradually-increase" arrangement. (e) Radarsat-1 SAR images with a "gradually-decrease" arrangement. (f) Radarsat-1 SAR images with a "stagger" arrangement. (g) Radarsat-1 SAR images with a "spindle" arrangement. (h) Radarsat-1 SAR images with a "gradually-increase" arrangement. (i) ERS-2 SAR images with a "gradually-decrease" arrangement. (j) ERS-2 SAR images with a "stagger" arrangement. (k) ERS-2 SAR images with a "spindle" arrangement. (l) ERS-2 SAR images with a "gradually-increase" arrangement.

Table 6 Overall errors with respect to different numbers of feature maps in every convolution layer

Net Structure	gradually-decrease	stagger	spindle	gradually-increase
simulated images	767	581	541	**375**
Radarsat-1 SAR Image	2022	1934	1746	**1599**
ERS-2 SAR image	350	330	**318**	328

From Fig. 9 and Table 6 we can see that the DC-CNNs with different structural specifications all produce relatively good performance. The model with a "gradually-increase" arrangement structure gives the best outcome, while the model with a "gradually-decrease" arrangement structure has the worst effect. Such a phenomenon reflects an internal mechanism of CNNs that the features obtained from the first several layers are relatively concrete to represent the details of an input image, while the features achieved from the subsequent layers are usually more abstract in representing the essential characteristics of the given image.

4 Conclusion

This paper has presented a novel change detection model based on a dual-channel convolutional neural network. The new model is composed of two paralleled CNN channels followed by a full-connection layer and softmax layer, which can concatenate the output of the two channels before producing the final detection results. In order to demonstrate the potential of this work, in addressing the challenges of change detection in SAR images, we have trained the model with both synthesized images and real-world SAR images, and conducted a series of experimental studies using the trained model. The experimental investigations have been carried out in comparison with state-of-the-art approaches. The results of such a systematic comparison have collectively shown that the new model of DC-CNN offers a promising performance with high accuracy. In the future, we will apply this proposed method for change detection in images that come from different remote sensors, over scenes that are of practical, and public, interests in environmental protection.

Disclosures

All the authors have no relevant financial interests in the manuscript and no other potential conflicts of interest to disclose.

Acknowledgments

This work was supported by National Key Research and Development Program of China (Grant No. 2016YFB0502502) and Key Project of the National Natural Science Foundation of China (Grant No. 61231016). The authors are grateful to the Editor and reviewers for their constructive comments that have helped improve this work significantly.

Author Contributions

All the authors made significant contributions to this work. Prof. Ying Li and Tao Liu devised the approach and analyzed the data; Prof. Qiang Shen helped design the experiments and provided advice for the preparation and revision of the work; Ying Cao performed the experiments and provided detailed revisions for this work especially in the training of DC-CNN and revise.

References

1. R. Almeida-Filho et al., “Detecting deforestation with multi-temporal L-band SAR imagery: a case study in western Brazilian Amazonia,” *J. Remote Sensing* **28**(6), 1383-1390 (2007).
2. D. Singh et al. "Monitoring and change detection of natural disaster (like subsidence) using Synthetic Aperture Radar (SAR) data." *Proc.* International Conference on Microwave **08**, 419-421 (2008).
3. S. Plank, “Rapid damage assessment by means of multi-temporal SAR: a comprehensive review and outlook to Sentinel-1,” *J. Remote Sensing* **6**(6), 4870-4906 (2014).

4. N. A. Kuchay et al., “Analysis and simulation of urban expansion of Srinagar city,” *J. Institute of Indian Geographers* **36**(1), 110-121 (2014).
5. K. Topouzelis et al., “Oil spill detection: SAR multi-scale segmentation & object features evaluation,” in Remote Sensing of the Ocean and Sea Ice 2002, C. R. Bostater Ed., *Proc. SPIE* **4880**, 77-87 (2003).
6. J. Lu et al, “Improving pixel-based change detection accuracy using an object-based approach in multitemporal SAR flood images,” *J. Selected Topics in Applied Earth Observations & Remote Sensing* **8**(7), 3486-3496 (2015).
7. M. Hussain et al. “Change detection from remotely sensed images: From pixel-based to object-based approaches,” *J. Photogrammetry & Remote Sensing* **80**(2), 91-106 (2013).
8. J. Dicarlo et al., “How does the brain solve visual object recognition?” *J. Neuron* **73**(3), 415-434 (2012).
9. B. Leng et al., “3D object understanding with 3D Convolutional Neural Networks,” *J. Information Sciences An International Journal* **366**(C), 188-201 (2015).
10. G. Hinton et al., “A fast learning algorithm for deep belief nets,” *J. Neural Computation* **18**(7), 1527-1554 (1989).
11. T. Mikolov et al., “Recurrent neural network based language model,” *Proc. INTERSPEECH 2010*, 1045-1048. Makuhari, Chiba, Japan (2010).
12. M. Kang et al., “Synthetic aperture radar target recognition with feature fusion based on a Stacked Autoencoder,” *J. Sensors* **17**(1), 1-16 (2017).
13. J. Yang et al., “Linear spatial pyramid matching using sparse coding for image classification,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, 1794-1801 (2009).
14. S. Gao et al., “Laplacian sparse coding for image classification,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010* **23**(3), 3555-3561 (2010)

15. J. Bontar et al., "Stereo matching by training a convolutional neural network to compare image patches," *J. Machine Learning Research* **17**(1), 2287-2318 (2016).
16. Y. Li et al., "Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network," *J. Remote Sensing* **9**(1), 1-21. (2017).
17. O. Plchot et al., "Audio enhancing with DNN Autoencoder for speaker recognition," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* 2016, 5090-5094 (2016).
18. M. Gong et al., "Change detection in Synthetic Aperture Radar images based on Deep Neural Networks," *J. Trans. on Neural Networks & Learning Systems* **27**(1), 125-138 (2015).
19. Y. Lecun, "Generalization and network design strategies," in *Connectionism in Perspective* Ed., pp. 1-19, Elsevier (1989).
20. Y Guo et al., "Deep learning for visual understanding: A review," *J. Neurocomputing* **187**(C), 27-48 (2016).
21. Y. Lecun et al., "Off-road obstacle avoidance through end-to-end learning," *Proc. NIPS* 2005, 739-746. (2005)
22. M. D. Zeiler et al., "On rectified linear units for speech processing," *Proc.* IEEE International Conference on Acoustics, Speech and Signal Processing **32**(3), 3517-3521 (2013).
23. P. Zhang et al., "The algorithm study for using the back propagation neural network in CT image segmentation," in International Conference on Innovative Optical Health Science, X. Li, Ed., Proc. SPIE **10245**, 0B1-0B4 (2017).
24. B. Xiong et al., "A change detection measure based on a likelihood ratio and statistical properties of SAR intensity images," *J. Remote Sensing Letters* **3**(3), 267-275 (2012).
25. J. Lu et al., "Automated flood detection with improved robustness and efficiency using multi-temporal SAR data," *J. Remote Sensing Letters* **5**(3), 240-248 (2014).

Ying Li is a professor at the School of Computer Science, Northwestern Polytechnical University. She received the Ph. D degree from the Xidian University, Xi'an, China, in 2002. She was a Postdoctoral Researcher with the School of Computer Science, Northwestern Polytechnical University, Xi'an, China from Feb. 2003 to Apr. 2005. Her interests include image processing, harmonic analysis, computation intelligence, and signal processing. She is the author or co-author of more than 100 scientific papers.

List of Figure Captions:

Fig. 1. Framework of dual-channel CNN model

Fig. 2. Structure of the CNN in each dual-channel

Fig. 3. Detection targets and their ground-truths used in the experiments

Fig. 4. Simulated images and results: (a) Ground truth. (b) Background image in Gamma noise. (c) Flood image in Gamma noise. (d) Change detection map with MTEP24. (e) Change detection map with Xiong algorithm24. (f) Change detection map with M325. (g) Change detection map with UAFS-HCD6. (h) Change detection map with DC-CNN

Fig. 5. Radarsat-1 SAR images and results: (a) Flood image. (b) Reference image. (c) Ground truth (the black pixels show the detected floods). (d) Change detection map with MTEP24. (e) Change detection map with Xiong algorithm24. (f) Change detection map with M325. (g) Change detection map with UAFS-HCD6. (h) Change detection map with DC-CNN.

Fig. 6. ERS-2 SAR Images and results (the black pixels show the detected floods): (a) Reference image. (b) Flood image. (c) Ground truth. (d) Change detection map with MTEP24. (e) Change

detection map with Xiong algorithm24. (f) Change detection map with M325. (g) Change detection map with UAFS-HCD6. (h) Change detection map with DC-CNN.

Fig. 7. Overall error obtained by varying the size of image patch on three different datasets

Fig. 8. Experimental results of change detection vs. different numbers of neurons in the fully-connected layer (a) Simulated images with 5 neurons. (b) Simulated images with 10 neurons. (c) Simulated images with 50 neurons. (d) Simulated images with 100 neurons. (e) Simulated images with 150 neurons. (f) Simulated images with 200 neurons. (g) Radarsat-1 SAR images with 5 neurons. (h) Radarsat-1 SAR images with 10 neurons. (i) Radarsat-1 SAR images with 50 neurons. (j) Radarsat-1 SAR images with 100 neurons. (k) Radarsat-1 SAR images with 150 neurons. (l) Radarsat-1 SAR images with 200 neurons. (m) ERS-2 SAR images with 5 neurons. (n) ERS-2 SAR images with 10 neurons. (o) ERS-2 SAR images with 50 neurons. (p) ERS-2 SAR images with 100 neurons. (q) ERS-2 SAR images with 150 neurons. (r) ERS-2 SAR images with 200 neurons.

Fig. 9. Results of change detection vs. different numbers of feature maps in every convolution layer (a) Simulated images with a “gradually-decrease” arrangement. (b) Simulated images with a “stagger” arrangement. (c) Simulated images with a “spindle” arrangement. (d) Simulated images with a “gradually-increase” arrangement. (e) Radarsat-1 SAR images with a “gradually-decrease” arrangement. (f) Radarsat-1 SAR images with a “stagger” arrangement. (g) Radarsat-1 SAR images with a “spindle” arrangement. (h) Radarsat-1 SAR images with a “gradually-increase” arrangement. (i) ERS-2 SAR images with a “gradually-decrease” arrangement. (j) ERS-2 SAR images with a “stagger” arrangement. (k) ERS-2 SAR images with a “spindle” arrangement. (l) ERS-2 SAR images with a “gradually-increase” arrangement.