

# Article Change Detection Based on Multi-Grained Cascade Forest and Multi-Scale Fusion for SAR Images

Wenping Ma<sup>1</sup>, Hui Yang<sup>1</sup>, Yue Wu<sup>2,\*</sup>, Yunta Xiong<sup>1</sup>, Tao Hu<sup>1</sup>, Licheng Jiao<sup>1</sup> and Biao Hou<sup>1</sup>

- Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation, School of Artificial Intelligence, Xidian University, Xi'an 710071, China; wpma@mail.xidian.edu.cn (W.M.); hyangstu@163.com (H.Y.); tataerchenyingxue@163.com (Y.X.); hsyyhxxk@163.com (T.H.); lchjiao@mail.xidian.edu.cn (L.J.); avcodec@163.com (B.H.)
- <sup>2</sup> School of Computer Science and Technology, Xidian University, Xi'an 710071, China
- Correspondence: ywu@xidian.edu.cn

Received: 5 November 2018; Accepted: 25 December 2018; Published: 12 January 2019



Abstract: In this paper, a novel change detection approach based on multi-grained cascade forest (gcForest) and multi-scale fusion for synthetic aperture radar (SAR) images is proposed. It detects the changed and unchanged areas of the images by using the well-trained gcForest. Most existing change detection methods need to select the appropriate size of the image block. However, the single size image block only provides a part of the local information, and gcForest cannot achieve a good effect on the image representation learning ability. Therefore, the proposed approach chooses different sizes of image blocks as the input of gcForest, which can learn more image characteristics and reduce the influence of the local information of the image on the classification result as well. In addition, in order to improve the detection accuracy of those pixels whose gray value changes abruptly, the proposed approach combines gradient information of the difference image with the probability map obtained from the well-trained gcForest. Therefore, the image edge information can be enhanced and the accuracy of edge detection can be improved by extracting the image gradient information. Experiments on four data sets indicate that the proposed approach outperforms other state-of-the-art algorithms.

Keywords: change detection; gcForest; fusion; synthetic aperture radar; gradient information

# 1. Introduction

Remote sensing image change detection is a process to detect and extract surface changes between images obtained at the same scene but at different times [1–3]. In recent years, the technology of remote sensing image change detection has become widely used in many fields [4–9], such as the aspect on acquisition and update of geographic information data [10,11], the detection and assessment of natural disaster [12], and the military [13,14]. In particular, in the case of natural disasters evaluation [15,16], if the subtle changes in areas where disaster will occur can be detected promptly and the corresponding measure is taken, the loss of life and property caused by natural disasters will be greatly reduced.

With the continuous expansion of image applications, the requirements for image change detection are becoming more accurate. The traditional change detection in remote sensing image includes three steps [17]: image pre-processing, difference image acquisition, and difference image analysis.

The difference image is obtained by images generated at different times, and the size of the different image is the same as the original two images. If the two images are directly subtracted, the speckle noise in SAR images cannot be effectively suppressed because the speckle noise is the



multiplicative noise. The ratio operator can overcome this disadvantage relatively well. However, the local, edge, and class conditional distribution information still needs to be addressed. Therefore, log-ratio (LR) [18] and mean-ratio (MR) [19] have been proposed. According to the process of change detection algorithms, when the difference map is generated, the difference map is further analyzed to generate a change detection result map. The threshold and clustering [20] methods are widely used to analyze the difference image. Threshold methods obtain the optimal threshold and divide the difference image into two categories by using this threshold, such as the Kittler and Illingworth (KI) algorithm [21] and the expectation maximization (EM) algorithm [22]. The clustering algorithms use the similarity between the samples to obtain the change class and the unchanged class. The clustering methods mainly include K-means clustering [23] and fuzzy c-means clustering (FCM) [24]. In [25], the wavelet fusion method that uses the fusion and clustering to obtain the change map is proposed. In addition, many methods including graph-cut, the active contour model, and principal component analysis have been applied to analyze the difference image [26–28]. However, with the increase in the number and the variety of data, these traditional methods cannot achieve higher accuracy. In fact, change detection can be regarded as a classification problem [29] that can use the network model to learn image features and then classify the learned features through a classifier.

In recent years, deep learning has become increasingly popular in different fields. For change detection, deep learning algorithms provide a flexible tool to transform a bitemporal image into a desired feature space and to then capture the key discriminative information and suppress irrelevant variations [30–32]. For example, Zhao et al. [33] believe that a deep belief network (DBN) can be trained to extract features of input data and classify the changed and unchanged pixels in SAR images. In [34], Liu et al. proposed a symmetric convolutional coupling network (SCCN) to detect heterogeneous images. An SCCN is symmetric and each side is composed of the same convolutional layer to extract feature space and suppress noise, and the convolutional neural networks (CNNs) are then trained by using back propagation to learn the concept of change for ternary change detection. These change detection methods based on deep learning are different from traditional methods, which can present a good representation of input data distribution.

The random forest model is simple and easy to handle with large data sets and has relatively fewer parameters to be modified. It has been widely applied in image classification [36,37]. Random forest is composed of many decision trees that act as base estimators and uses a bagging method to train a large number of small decision trees, and finally combines these small decision trees. In addition, they can also be used for unsupervised learning clustering and outlier detection [38]. In [39], completely random trees that have been shown to work well in unsupervised learning and supervised learning independently are employed. In [40], gcForest, which uses a decision tree ensemble approach to do representation learning, is proposed. The forest model shows a very strong learning ability and its performance is quite robust to hyper-parameter settings. Furthermore, an improved random forest node splitting algorithm is proposed in [41] for improving the accuracy of image classification. The random forests show the advantages of the small computational cost, fewer hyper-parameters, and the insensitivity of the hyper-parametric adjustment of the model.

In this paper, we propose a SAR image change detection method based on gcForest. gcForest is selected because it will improve accuracy and reduce the training difficulty. In addition, we make two great improvements. One is multi-scale fusion. Because a single size image block only provides part of the local information, the proposed approach chooses different sizes of image blocks as the input of gcForest, which can learn more image characteristics and reduce the influence of the local information of the image on the classification result. The other is the post-processing. Because it is difficult to detect those pixels whose gray value changes abruptly, the proposed approach combines gradient information of the difference image with the probability map obtained from the well-trained gcForest. The process of the approach we proposed consists of three steps: pre-classification, image fusion, and post-processing: (1) pre-classification is a process that obtains some data with labels; (2) because

different sizes of image blocks contain different information, the proposed method constructs a new model based on gcForest, which selects different sizes of image blocks as input to the network, and the different sizes of image blocks are then trained together; (3) post-processing combines edge distributions with category probability, and classifies the changed and unchanged pixels according to the final probability map.

The rest of this paper is organized as follows. The second section introduces the problem statement and the background of gcForest. The third section is mainly about the proposed change detection framework. The fourth part is the description and analysis of the experimental results. The last section is the conclusion of this article.

## 2. Background

In this section, we describe the motivation of the proposed SAR image change detection method. Meanwhile, the structure of the model is described in detail. Two coregistered intensity [42] and multi-temporal SAR images,  $X_1 = \{x_1(i,j)|1 \le i \le W, 1 \le j \le H\}$  and  $X_2 = \{x_2(i,j)|1 \le i \le W, 1 \le j \le H\}$ , have the same size and were acquired at the same position but at different times. Effective methods are needed to detect the changed areas of the two images within the influence of noise [43] accurately.

## 2.1. Motivation

It is difficult to obtain the final change detection map accurately because of the speckle noise in the SAR image. Therefore, it is important to find a good method that can fully detect the changed area information and suppress the influence of noise. We usually use the information in the image effectively to obtain a changed map with good results. We hope to improve the performance of image change detection algorithms in two ways. Firstly, when we train the model, the size of the image block that we choose will affect the detection result of the image. Secondly, one problem of image change detection is that it is hard to detect the edge part of the image because the edge part of the image exists at the boundary of the image change. The gcForest method we proposed can greatly suppress noise and obtain good detection results, and gcForest [44] is very robust to some parameters. Therefore, the change detection method based on gcForest can learn the characteristics of changed and unchanged areas, and suppress the influence of irrelevant information.

#### 2.2. gcForest

gcForest uses an ensemble approach based on decision trees [40]. It achieves the effect of representation learning by integrating, developing, and connecting them in series before and after the forests formed by trees. Its ability of representation learning can be improved by multi-grained scanning with respect to high-dimensional input data. Compared with the difficult process of adjusting parameters of the deep neural network, gcForest has relatively few parameters. Therefore, the training process is relatively easier and gcForest has less dependence on the parameters. The gcForest has two parts: one is multi-grained scanning and the other is cascade structure. Each hidden layer in the cascade structure is composed of several random forests [45,46], and the features of the first part are cascaded to each level in the second part until the final output of the last level. The final prediction will be obtained by aggregating the class vector at the last level and taking the class with the maximum aggregated value.

#### 2.3. Multi-Grained Scanning

There is a strong spatial relationship between pixels whose positions are close in the image. The window of CNNs can handle this spatial relationship very well [47], and RNN can handle the correlation in time series very well [48]. Similarly, gcForest also uses multi-grained scanning to enhance the cascaded part. The part of multi-grained scanning is shown in Figure 1. Through the block processing on the input data, we will obtain several sub-blocks as follows:

$$sum = (INT(\frac{x-y}{s}+1))^2$$
(1)

where the number of class is n, the input data size is  $x \times x$ , and the window size is selected as  $y \times y$ . The sliding interval is s, INT means rounding down, and sum is the number of sub-blocks. The size of each sub-block is  $y \times y$ . A random forest is used for processing, and the number of the output through the random forest is the same as the numbers of categories in the classification. Each sub-block will receive n-dimensional output through the random forest, and the number of input of random forest corresponds to x - y + 1 blocks. Then, through the processing of random forest and cascading, the final results can be achieved. gcForest has two kinds of random forests, i.e., completely random forests [49] and random forests [50]. The process of building a random forest is roughly as follows:

- 1. Select samples from the original training set randomly and put it back in place, and perform *nt* samplings to generate *nt* training sets.
- 2. For *nt* training sets, we train *nt* decision tree models, respectively.
- 3. For a single decision tree model, assuming that the number of training sample features is *n*, calculate the Gini index and split the best feature.
- 4. Every tree has been split up until all training samples for that node belong to the same class. Pruning is not required during the decision tree splitting.
- 5. The random forest consists of multiple decision trees, and the final classification results are determined by the voting strategy of the results of multiple tree classifiers.



**Figure 1.** Illustration of feature re-representation using sliding window scanning, supposing there are two classes to predict and each forest will output a two-dimensional class vector.

Each completely random forest contains some completely random trees and each random forest also contains some random trees. However, the completely random trees are different from random trees in the candidate feature space. Completely random forests are randomly selected in the complete feature space to split, whereas ordinary random forests are selected in the random feature subspace and select split nodes by Gini coefficients. The Gini index can be obtained by the following:

$$Gini = 1 - \sum_{k=1}^{n} (p_k)^2$$
(2)

where  $p_k$  is the proportion of k class samples in the current sample set, and n is the number of classification. If the value of the Gini is smaller, the purity of the data set is higher. The learning of these two random forests is conducted in a supervised way, and the parameters of random forests are learned by a number of inputs and the corresponding class labels.

# 2.4. Cascade

Each layer of the cascade includes several random forests, and the input of the cascade structure is the output of the multi-grained scanning [26,51]. In order to retain the information of each layer

maximally, the cascading features continue to be embedded in each layer of the network in a cascaded way. The output of random forests in this part of the network is consistent with the number of categories. Therefore, the dimension of the feature obtained is always unchanged after multi-layer processing. In the part of the classifier, several random forests are used to average and maximize their output. The structure about cascade is shown in Figure 2, and the rule is as follows:

$$F = \frac{1}{D} \sum_{d=1}^{D} f_d \tag{3}$$

$$label = \arg\max_{j \le n} F_j \tag{4}$$

where  $f_d$  is the probability of each forest in the last layer, D is the number of forests, and  $F_j$  is the probability of the jth class. Each forest will calculate the percentage of training samples in different classes and then average all the trees in the forest to generate an estimate of the distribution of the classes [52]. The estimated class distribution forms a class vector which is connected to the original feature vector. Each class vector generated by the forest uses the k-fold cross-validation [46]. Each instance will be used as training data for k - 1 times and generate k - 1 class vectors. Those class vectors are then averaged to generate the final class vectors which are the enhanced features of the next level. The performance of the entire cascade will be estimated on the validation set after extending a new level. Not until the results improve will the model stop training.



**Figure 2.** Illustration of the cascade forest structure. The output of each forest is concatenated for re-presentation of the original input.

# 3. Methodology

In this section, the new change detection method based on gcForest and multi-scale image fusion will be put forward. The proposed method consists of three parts: pre-classification, image fusion, and post-processing. First, in the pre-classification part, it introduces the process that receives the initial label. The image fusion part includes the new structure of gcForest that can fuse the different sizes of image blocks. The post-processing is a process that combines the probability map extracted from the proposed model with the gradient information map of the difference image of two original images. Figure 3 shows the process of the proposed method.



Figure 3. Flowchart of the proposed method.

## 3.1. Pre-Classification

The result of pre-classification will affect the final classification result of the proposed model. In the pre-classification, we apply a non-local denoising algorithm (NLM) [53] to reduce as much noise as possible.

For two images at the same place but at different times,  $X_1 = \{x_1(i, j) | 1 \le i \le W, 1 \le j \le H\}$  and  $X_2 = \{x_2(i, j) | 1 \le i \le W, 1 \le j \le H\}$ , a log-ratio operator can be used to obtain a difference image as follows:

$$DI(i,j) = \log \frac{|x_1(i,j)+1|}{|x_2(i,j)+1|}.$$
(5)

Before using the FCM algorithm [54] to classify the difference image, a non-local mean algorithm will reduce the effect of noise in the classification results. The non-local mean algorithm is a method that the estimated value of the current pixel is obtained by averaging the weighted pixels with similar neighborhood structure in the image. Firstly, the difference image takes a point as the center and obtains a window, as shown in Figure 4. The pixels in the window have a similar neighborhood structure. The similarity of the center pixel is calculated with the neighborhood pixels, and the weight is calculated as follows:

$$u(x) = \sum_{y} W(x, y) DI(y)$$
(6)

$$W(x,y) = \frac{1}{Z(x)} \exp(-\frac{\|V(x) - V(y)\|^2}{h^2})$$
(7)

$$\|V(x) - V(y)\|^{2} = \sum_{y} ||DI(x) - DI(y)||^{2}$$
(8)

$$Z(x) = \sum_{y} \exp(-\frac{||V(x) - V(y)||^2}{h^2})$$
(9)

where *x* is the center pixel, *y* is the other pixel in the window, W(x, y) is the weight between *x* and *y*, u(x) is the gray value after using the non-local mean algorithm, DI(x) is the gray value of the pixel in the center, DI(y) is the gray value of the pixel in the neighborhood, Z(x) is the normalization factor, and *h* is the smoothing parameter. The larger *h* is, the smoother the Gaussian function changes. The higher the denoising level is, the more blurred the image will be. The smaller *h* is, the more the image edge details are kept, but too much noise remains. Therefore, *h* should be properly adjusted according to the image.



 $n \times n$  window

Figure 4. Process of the NLM algorithm.

The NLM algorithm makes full use of the redundant information in the image and can retain the details of the image while denoising. After denoising the difference image, the FCM algorithm is used for pre-classification to obtain the initial change detection image. Given that an image block with a size of  $n \times n$  is taken as the center with DI(i, j) in the difference image, the image block is processed into two image blocks, where one image is an image block of size  $n/2 \times n/2$  centered on DI(i, j), and the other image is obtained by downsampling an image block of size  $n \times n$ , so that multi-scale image blocks are obtained. Through this process, the image block of size  $n/2 \times n/2$  can be calculated twice, and the information of the image block will focus on the central part and the impact of the edge part of the image will be reduced. The local features of the image obtained by the method can be described in different scales in a simple form, so that a single-scale input changes to a multi-scale input, which is beneficial to enrich as much local information obtained from the image block as possible.

# 3.2. Image Fusion

In order to avoid the impact of the size selection of the image block on the classification results, the input of multi-scale image blocks will avoid the size factor selection of the image block. At the same time, if the information of multi-scale image blocks is fused, the model can fully learn useful information from local image blocks. This paper chooses different sizes of image blocks together as the input of gcForest, and gcForest can learn different features from different sizes of image blocks. The two different image block feature vectors are then merged to classify. Therefore, gcForest can learn more image feature information by this strategy than that using a single image block input. In this paper, we obtain the  $n1 \times n1$  size and  $n2 \times n2$  size image blocks from pre-processing. As shown in Figure 5, two image blocks with different sizes can obtain two different feature vectors through multi-grained scanning, and two fused vectors are used for classification through the cascade structure. The multi-grained scanning is similar to the sliding operation according to the window size. In order to make the two different sizes of the image block fuse in one layer, the scanning window needs to be set differently. It is proved that the effect of different fused scales is better than a single scale through experiments. This reduces the impact that the size of the image block has on the classification result.



**Figure 5.** Two different size image blocks as the input of the model. These two block features can be fused through the multi-grained part of the model, and the fusion feature is classified by the cascaded part of the model. The output of the multi-grained scanning will cascade with every level of the cascaded part of the model until the best result generates, average every class value, and then select the maximum class value, which is the final prediction class.

#### 3.3. Post-Processing

The image classification result is generally based on the probability that the pixel belongs to each class to determine its label. Generally, when the probability of a certain class is the largest, the pixel can be determined as the class. However, because the pixels between the two types are very different from the surrounding pixels, it is difficult to distinguish them. If the gradient change information of the difference map is combined with the probability map of the pixel, as shown in Figure 6, the accuracy of edge pixel classification can be improved. The gradient of the edge of the image is large and the gradient direction is perpendicular to the edge [55]. When the edge distribution is unknown, the distribution of the edge direction can indicate the outline of the target. Scilicet, the local gradient intensity, and the gradient distribution of each pixel of the difference map can be calculated to detect the edge information of the image. A one-dimensional gradient direction histogram for each small block can be calculated based on the difference map. We can combine the gradient histogram with the pixel probability to obtain the final result, as shown in Figure 7.



Figure 6. Flowchart of the post-processing part.

The gradient magnitude and gradient direction can be calculated by the horizontal and vertical gradients of the difference map.

$$G_1(i,j) = DI(i+1,j) - DI(i-1,j)$$
(10)

$$G_2(i,j) = DI(i,j+1) - DI(i,j-1)$$
(11)

$$G(i,j) = \sqrt{G_1(i,j)^2 + G_2(i,j)^2}$$
(12)

$$M(i,j) = \tan^{-1}(\frac{G_1(i,j)}{G_2(i,j)})$$
(13)

where (i, j) represents the position of the pixel, G(i, j) represents the gradient magnitude of the pixel, and M(i, j) represents the gradient direction of the pixel.



Figure 7. The probability map combined with the gradient histogram of the difference map.

To quantify the gradient direction of a local area which maintains weak sensitivity to the image target edge, we divide the gradient amplitude map and direction map by  $3 \times 3$  window for each pixel point, and divide the gradient direction of the sub-block to eight direction blocks. If the value of the gradient direction of each pixel point in the sub-block belongs to the range in the direction block, the value will be added up in the corresponding direction histogram. Therefore, each pixel in the sub-block of the gradient amplitude image is projected in the histogram according to the gradient direction, and it is mapped into a corresponding angle range block. Using the amplitude as the weight can increase the influence that the direction of each sub-block is normalized. We take  $3 \times 3$  blocks centered on each pixel in the probability map, and map the probability of each pixel in this block to the corresponding angle range. Finally, we obtain the eight-dimensional mean value after averaging the sum of the probability values in each angle range. The process combines the mean of a probability map with amplitude by Equation (13):

$$p = \sum_{m=1}^{8} pm(m) * amp(m)$$
(14)

where pm(m) is the value of the mean probability in the *m*th direction angle, and amp(m) is the amplitude value in the *m*th direction angle. *p* is the final probability after the above processing.

The probability map that contains the edge information can be obtained by combining the eight-dimensional mean value with the eight-dimensional histogram. Therefore, classifying the probability map through the threshold segmentation can obtain the final change detection map.

# 4. Experiments

Four kinds of data sets with different characteristics are used to test the proposed method for confirming the effectiveness of the method. Other methods will be compared with the proposed method according to the evaluation criteria. At the same time, some corresponding parameter analysis will be done for each experimental result.

The Yellow River data set is a section of two SAR images taken by the Radarsat-2 in the Yellow River Estuary area in June 2008 and June 2009 as shown in Figures 8a,b. The original size of these two images is 7666 × 7692 pixels. The noise effect of images collected in 2008 is far greater than the images collected in 2009. Due to the large size of the picture, it is difficult to display detailed information on the small page. We choose four more typical areas (two farmlands, inland waters, and coastline) at different locations. In Figures 8a,b, the area A is the area of the inland water, the area B is the area of coastline, the area C is the area of the Farmland C and the area D is the area of the Farmland D. These places can effectively represent the changed characteristics of the Yellow River. Figure 9 shows the multi-temporal images related to the Yellow River Farmland C in Yellow River, and there are fewer changes compared with Farmland D. In the inland waters, the changed areas are concentrated on the boundaries of the river, as shown in Figure 11. In the coastline areas, the changed areas are concentrated set is a detection of environmental monitoring. The changes areas represent the environment change over a long period.



**Figure 8.** Multitemporal image relating to the Yellow River data set. (a) Image acquired in June 2008. (b) Image acquired in June 2009. Area A: Inland Water; Area B: Costaline; Area C: Farmland C; Area D: Farmland D.



**Figure 9.** Multitemporal images relating to Farmland D of the Yellow River Estuary and the reference image. (**a**) Image acquired in June 2008. (**b**) Image acquired in June 2009. (**c**) The reference image.



**Figure 10.** Multitemporal images relating to Farmland C of the Yellow River Estuary and the reference image. (**a**) Image acquired in June 2008. (**b**) Image acquired in June 2009. (**c**) The reference image.



**Figure 11.** Multitemporal images relating to the inland water of the Yellow River Estuary and the reference image. (a) Image acquired in June 2008. (b) Image acquired in June 2009. (c) The reference image.



**Figure 12.** Multitemporal images relating to the coastline of the Yellow River Estuary and the reference image. (**a**) Image acquired in June 2008. (**b**) Image acquired in June 2009. (**c**) The reference image.

# 4.2. Evaluation Criteria

The evaluation standard value of the change detection result is calculated as follows: (1) FN (the false negative) is the number of the changed pixels that were undetected; (2) FP (the false positive) is the number of the changed pixels that were detected wrongly; (3) OE (the overall error) is the sum of FN and FP.

$$OE = FP + FN. \tag{15}$$

We can calculate the *PCC* (percentage correct classification) to evaluate the result further, as follows:

$$PCC = \frac{TP + TN}{TP + TN + FP + FN}$$
(16)

where *TP* (true positive) represents the number of changed pixels that are correctly detected in both the reference map and the results. *TN* (true negative) represents the number of unchanged pixel that are correctly detected in both the reference map and the results. However, it is difficult to distinguish the detection quality through *PCC*, because when the number of the entire pixels is larger, the *PCC* values obtained by the different methods are similar. Therefore, Kappa is introduced to be a kind of

evaluation criterion. Kappa statistic is a measure of accuracy or agreement based on the difference between the error matrix and chance agreement [56]. Kappa is calculated as follows:

$$Kappa = \frac{PCC - PRE}{1 - PRE} \tag{17}$$

where

$$PRE = \frac{(TP + FP) \cdot NC + (FN + TN) \cdot NU}{N^2}$$
(18)

where *NC* is the actual number of the changed pixels, and NU is the actual number of the unchanged pixels. Kappa involves more detailed information of the classification than *PCC*, but *PCC* only relies on the sum value of *TP* and *TN*.

#### 4.3. Experiment Performance

For each data set, we obtain the pre-classification result firstly. The non-local mean algorithm needs to have the search window size and a smoothing parameter. For each type of data, we adjust the window size and smoothing parameter accordingly. In the experiment, the window size of Farmland D, Farmland C, and the inland water is  $4 \times 4$ , and the smoothing parameters *h* are 0.5; the setting of the window size for the coastline is  $2 \times 2$ , and the smoothing parameter *h* is 0.15.

We obtain the  $3 \times 3$  and  $4 \times 4$  blocks from the two original images of the Farmland D data set, the Farmland C data set, and the inland water data set, and obtain the  $5 \times 5$  and  $6 \times 6$  blocks from the two original images of the coastline data set. We then make them as the data sets used to train the gcForest method we proposed. For the image change detection algorithm, we do not need to manually mark the samples in advance, and the data set is obtained via pre-classification. For the two original images, we select a window centered on each pixel to obtain the test data set. In the image pre-classification part, an initial image change detection label map can be obtained. For the initial label map result, we select a window centered on each pixel and the window size is  $7 \times 7$ . If the label of the neighborhood pixel in the window is the same as that of the center pixel, and the number of the same labels is greater than half of the number of the pixels in the window, we select this central location data from the test data set as the training data set.

In the experiment, there are two types of random forests in the multi-grained scanning part, which are completely random forests and random forests. Each type of random forest includes eight trees, and tree growth occurs until pure leaf is obtained. In the cascading part, each layer contains three completely random forests, each completely random forest includes 10 trees, and tree growth occurs until pure leaf is obtained. We choose the structure of gcForest for the four different types of data in the experiment.

The training data will be used to train a completely random tree forest and a random forest, and the feature vector will be obtained in the multi-grained scanning. The transformed training set data will then be used to train the cascade forest. The transformed feature vectors, augmented with the class vector generated by the previous level of cascade forest, will then be used to train the latter level of the cascade forests. This procedure will be repeated until a convergence of validation performance. As for the test process, the test data set will go through the multi-grained scanning procedure to obtain its corresponding transformed feature representation, and then go through the cascade until the last level. Finally, the probability map that is combined with the edge features of the difference map will be classified, and the change detection results can then be obtained.

# 4.3.1. Results on the Farmland D Data Set

The change detection results will be generated by five methods including the proposed method and four other comparative methods on the Farmland D data set, as shown in Figure 13. Figure 13a shows the the reference images of change detection, and Figures 13b–g are results obtained by FCM, NLMFCM (using the NLM method to denoise the DI and the FCM to obtain the final map), DBN [57],

SCCN [34], wavelet fusion [33], and gcForest (multi-scale input blocks but without adding the gradient information of DI). Figure 13f shows the result of the proposed method. As shown in Figure 13b, the final map generated by FCM is polluted by many white noise spots. This is because the FCM algorithm needs to find the clustering centers of two classes to obtain the result. The error in the clustering center will have an impact on the change detection map, and the clustering center is sensitive to noise. The NLMFCM is used for pre-classification and the NLM algorithm has a good effect in denoising. As shown in Figure 13c, the change detection map obtained by NLMFCM presents fewer white spots than does the FCM, but the details of the results are lost to some degree. In Figure 13d, the DBN algorithm shows an obvious improvement because the final map presents a good result. The DBN algorithm applies deep learning to learn meaningful features but there are too many parameters for setting. Figure 13e shows that using an SCCN algorithm can also have an effect as great as the DBN algorithm does. However, the wavelet fusion in Figure 13f cannot obtain great results, because the great amount of noise cannot be reduced. gcForest is used to test the model without adding the edge feature into the probability, and Figure 13g shows that there is less noise in the result. By contrast, the proposed method that applies the edge feature in the probability map based on gcForest shows an obvious improvement. In particular, training gcForest does not require much time to adjust the parameters. Table 1 presents the values of evaluation criteria. Because the proposed method has some changed pixels that detected wrongly, the FN yield by the proposed method equals to 1090 is not the lowest compared with the six methods, but the FP, OE, PCC, Kappa yield by the proposed method are the best. The results indicate that the proposed method is robust and can reduce the noise.



**Figure 13.** Change detection results of the Farmland D data set: (**a**) Reference. (**b**) FCM. (**c**) NLMFCM. (**d**) DBN. (**e**) SCCN. (**f**) Wavelet fusion. (**g**) gcForest. (**h**) The proposed method.

Method	FN	FP	OE	PCC (%)	κ
FCM	12,237	5176	17,413	76.55	34.32
NLMFCM	2063	2400	4463	93.99	79.51
DBN	660	3487	4147	94.42	79.47
SCCN	1398	2887	4285	94.23	79.65
wavelet fusion	761	4200	4961	93.32	74.95
gcForest	656	1920	2576	96.53	87.84
Proposed	1090	1363	2454	96.70	88.76

Table 1. Value of the evaluation criteria of the Farmland D data set.

Figure 14 shows that most false alarms occur at the edge of the image. In Figure 14b with Figure 14c, the areas of the red circles are the places where the changes in the two figures are obvious. By comparing Figure 14b with Figure 14c, we can find that many false alarms at the edge of the change map have been reduced when we use post-processing. Furthermore, by comparing the three areas of the change detection maps in Figure 15, we see that many false alarms with post-processing in the three areas have been reduced and the effect of edge detection shows some improvements if we use gradient extraction. Therefore, the proposed method can improve the edge change detection, using post-processing can improve the performance of the edge change detection.



**Figure 14.** The false alarms of the change detection map on the Farmland D data set: (**a**) Reference. (**b**) The false alarms of change map that use gcForest. (**c**) The false alarms of the change map that use the proposed method. (The areas of the red circles are the places where the changes in the two figures are obvious.)



Figure 15. Cont.



**Figure 15.** The change detection maps at the three areas on the Farmland D data set: (**a**) Number 1 area in reference. (**b**) Number 1 area with gcForest. (**c**) Number 1 area with the proposed method. (**d**) Number 2 area in reference. (**e**) Number 2 area with gcForest. (**f**) Number 2 area with the proposed method. (**g**) Number 3 area in reference. (**h**) Number 3 area with gcForest. (**i**) Number 3 area with the proposed method.

# 4.3.2. Results on the Farmland C Data Set

For the Farmland C data set, the reference map and the final maps of the proposed method and the comparative methods are shown in Figure 16. FCM obtains the worst performance and there are many white spots in the result. From Table 2, the values of evaluation are the worst in all comparative experiments. The final map obtained by the NLMFCM has many false alarms due to the noise. In Figure 16d, the result obtained by the DBN has a good effect, and the values of *PCC* and *Kappa* are also low, but DBN has a high *FN* because the edge detection is not accurate. The effect of gcForest is better than the SCCN and the DBN, as shown in Figure 16g. The proposed method is shown in Figure 16h. It can be seen that noise spots are few in number. In Table 2, the PCC yielded by the proposed method equals 99.11%, which is the highest among all others. Although the *FN* is close to that obtained by the DBN, the *FP* yielded by the proposed method equals 163, which is lower than the value of 679 obtained by the DBN. Therefore, the proposed method outperforms the other comparison methods.

Method	FN	FP	OE	PCC (%)	κ
FCM	12,126	813	12,939	85.47	34.95
NLMFCM	687	668	1355	98.48	86.36
DBN	697	841	1538	98.27	84.29
SCCN	768	779	1547	98.26	84.38
wavelet fusion	931	1377	2308	97.41	75.76
gcForest	124	685	809	99.09	91.41
Proposed	163	630	793	99.11	91.66

Table 2. Values of the evaluation criteria of the Farmland C data set.



Figure 16. Change detection results of the Farmland C data set: (a) Reference. (b) FCM. (c) NLMFCM.(d) DBN. (e) SCCN. (f) Wavelet fusion. (g) gcForest. (h) The proposed method.

# 4.3.3. Results on the Inland Water Data Set

For the land water data set, the reference map and the final maps of the proposed method and the comparative methods are shown in Figure 17. FCM shows the worst performance in terms of FN and FP. In Figure 17c, the final map obtained by the NLMFCM has less speckle noise than FCM, which indicates that the NLM has a good capability of reducing noise. In Figures 17d,f, the final maps obtained by the DBN and wavelet fusion show a good performance in terms of PCC and Kappa. However, the incorrect detection of a large number of pixels results that the DBN has a high value of FP and the wavelet fusion has a high value of FN. The result of our proposed method is shown in Figure 17h. The noise in the final map obtained by the proposed method is small. The main changed pixels are detected, but they have a high value of FP. The precise detection decreases the FN such that the overall error is lower than the other methods. As shown in Table 3, the proposed method shows the best PCC and Kappa, and the Kappa (81.27%) is higher than that of 80.12% without adding the edge information. False alarms usually occur at the edge of the image. Combining the edge features of the difference map with the probability map is beneficial for reducing the error detection at the edge of the change map.

Method	FN	FP	OE	PCC (%)	к
FCM	24,581	534	25,115	80.56	18.17
NLMFCM	794	927	1721	98.66	78.76
DBN	349	1344	1693	98.67	76.45
SCCN	1060	1235	2295	97.70	61.54
wavelet fusion	1282	794	2076	98.39	76.09
gcForest	580	981	1561	98.79	80.12
Proposed	693	825	1518	98.83	81.27

Table 3. Value of the evaluation criteria of the inland water data set.



**Figure 17.** Change detection results of the inland water data set: (**a**) Reference. (**b**) FCM. (**c**) NLMFCM. (**d**) DBN. (**e**) SCCN. (**f**) Wavelet fusion. (**g**) gcForest. (**h**) The proposed method.

# 4.3.4. Results on the Coastline Data Set

For the coastline data set, the reference map and the final change detection maps of different methods are shown in Figure 18. In the data set, the changed areas are very small. The FCM shows poor performance in terms of change detection. The result generated by the NLMFCM and wavelet fusion have many false alarms and missed alarms. However, the final map obtained by gcForest outperforms the NLMFCM and wavelet fusion, which confirms that gcForest can learn meaningful features and reduce the noise. Figure 18d,e generated by the DBN and the SCCN show that there are few noise spots, and the changed areas are detected precisely. The results obtained by the proposed method are better than that of the DBN and the SCCN. In Table 4, the Kappa yielded by the proposed method is 89.72%, which is higher than 88.76% generated by the DBN. This is because some pixels cannot be detected. However, even if the changed areas are so small that the changed areas are difficult to detected. However, even if the changed area is not large, the proposed method can effectively detect the changed and unchanged areas. As is shown in Table 4, when adding the edge feature into the probability map, *Kappa* and *PCC* are improved. Therefore, the proposed method is effective to some extent; in particular, the effect in the large changed areas, i.e., the farmland and land water, is as good as it is in the small changed areas, i.e., the coastline.

**Table 4.** Value of the evaluation criteria of the coastline data set.

Method	FN	FP	OE	PCC (%)	κ
FCM	29,970	282	30,252	75.99	4.62
NLMFCM	17,337	129	17,466	86.13	10.46
DBN	25	255	280	99.70	88.76
SCCN	59	237	296	99.67	88.12
wavelet fusion	54	26,061	26,115	79.27	7.15
gcForest	48	221	269	99.79	89.23
Proposed	35	220	255	99.80	89.72



**Figure 18.** Change detection results on Area b of the coastline data set: (**a**) Reference. (**b**) FCM. (**c**) NLMFCM. (**d**) DBN. (B) SCCN. (**f**) Wavelet fusion. (**g**) gcForest. (**h**) The proposed method.

# 4.4. Parameter Analysis

#### 4.4.1. Block Size

Selecting a suitable block size is an important step in our proposed method. In the above experiments, we set the block size as  $3 \times 3$  and  $4 \times 4$  or  $5 \times 5$  and  $6 \times 6$  as the input of the proposed model, and fuse the two differently sized blocks in the multi-grained scanning. In this part, we will analyze the effect of different sizes of blocks on the performance. We select different sizes of single blocks for the input of gcForest, and select several different sizes image blocks to be fused in the multi-grained scanning. At the same time, four different data sets are used. The *FN*, *FP*, and *OE* of the single block size are shown in Figure 19, and the results of different sizes of blocks as input are shown in Tables 5–8.

Based on the results of the four data sets, the OE value varies with different single block sizes. For the Farmland D data set, it is best to choose the  $3 \times 3$  block, and for the other data sets, choosing the  $4 \times 4$  blocks is optimal. The fused results show that the effect on two different sizes of fused blocks is different. The reason for this is that fused blocks that are too large or too small may cause the range of the detection to become larger or smaller, which leads to an inaccurate detection range and a higher amount of miss alarms. Moreover, the results of the  $3 \times 3$  and  $4 \times 4$  blocks fused on Farmland D, Farmland C, and the land water are superior because a larger block size may not detect the details. Change detection shows that the larger the block on the three data sets is, the higher the value of *FP* is, which confirms that appropriate sizes of fused blocks are conducive to improved test results. However, this is different from the coastline data set, which is more appropriate for  $5 \times 5$  and  $6 \times 6$  fused blocks. When the size is small, the values of the *FN* and *FP* are high. The changed areas on the coastline data set are small, and it is necessary to select a larger block to obtain more features. Therefore, if the changed areas are large, it is better to choose  $3 \times 3$  and  $4 \times 4$  blocks as the input of the proposed model. If the changed areas are small, it is better to choose  $5 \times 5$  and  $6 \times 6$  blocks as the input of the proposed model. Furthermore, as shown in Tables 5–8, suitably sized fused blocks can gain a balance between FP and FN.



**Figure 19.** The wrong results on several single block size inputs of different data sets. (**a**) Farmland D data set. (**b**) Farmland C data set. (**c**) Inland water data set. (**d**) Coastline data set.

Table 5. The evaluation values of different sizes of fused blocks of the Farmland D data set.

Method	FN	FP	OE	PCC (%)	κ
Win3 + Win4	1090	1363	2454	96.70	88.76
Win3 + Win5	823	1848	2671	96.40	87.49
Win3 + Win6	862	1976	2838	96.17	86.67
Win4 + Win5	722	1983	2705	96.36	87.24
Win4 + Win6	1024	1753	2777	96.26	87.10
Win5 + Win6	1032	1870	2902	96.09	86.48

Table 6. Evaluation values of different sizes of fused blocks of the Farmland C data set.

Method	FN	FP	OE	PCC (%)	к
Win3 + Win4	163	630	793	99.11	96.66
Win3 + Win5	109	782	891	99.00	90.44
Win3 + Win6	115	768	883	99.00	90.55
Win4 + Win5	151	720	871	99.02	90.75
Win4 + Win6	118	768	886	99.01	90.52
Win5 + Win6	123	740	863	99.03	90.79

Table 7. Evaluation values of different sizes of fused blocks of the inland water data set.

Method	FN	FP	OE	PCC (%)	κ
Win3 + Win4	693	825	1518	98.83	81.27
Win3 + Win5	580	1002	1582	98.78	79.81
Win3 + Win6	594	1001	1595	98.77	79.68
Win4 + Win5	583	966	1549	98.80	80.32
Win4 + Win6	609	926	1535	98.81	80.65
Win5 + Win6	650	931	1581	98.78	80.16

Method	FN	FP	OE	PCC (%)	κ
Win3 + Win4	84	268	332	99.74	86.55
Win3 + Win5	56	277	333	99.73	86.41
Win3 + Win6	65	213	279	99.77	88.94
Win4 + Win5	51	216	267	99.79	89.34
Win4 + Win6	46	289	335	99.73	86.21
Win5 + Win6	35	220	255	99.80	89.72

 Table 8. Evaluation values of different sizes of fused blocks of the coastline data set.

The proposed method obtains a good result on the four kinds of data sets. We can conclude three points based on these experiments: (1) the non-local means algorithm helps to reduce the noise in the difference map and obtains a good initial change detection map, which is processed as the data sets of the proposed model; (2) fusing the different sizes of blocks can obtain more information about image features and makes use of the multi-scale features to train gcForest well; (3) combining the edge information obtained from the gradient feature of the probability map can obtain the best change detection map according to the threshold classification.

# 4.4.2. Parameters of Pre-Classification

The non-local mean algorithm needs to have the search window size and a smoothing parameter. For each type of data, we adjust the window size and smoothing parameter accordingly. In this experiment, we change the window size of four types of data and set the smoothing parameters *h* of Farmland D, Farmland C, and the inland water as 0.5, and the smoothing parameters *h* of the coastline as 0.15. The changes of *FN*, *FP*, and *OE* on the four types of data are as shown in Figure 20. The  $4 \times 4$  window size is best for Farmland D, Farmland C, and the inland water, and a  $2 \times 2$  window size is best for the coastline. In Figure 21, we change the smoothing parameter *h* of the four types of data and set the window size of Farmland D, Farmland C, and the inland water as  $4 \times 4$ , the window size of the coastline as  $2 \times 2$ . Figure 21 shows that the smoothing parameters 0.5 are best for Farmland D, Farmland C, and the inland water as 0.5 are best for Farmland D, Farmland C, and the inland water as  $4 \times 4$ , the window size of the coastline as  $2 \times 2$ . Figure 21 shows that the smoothing parameters 0.5 are best for the coastline.



**Figure 20.** The wrong results on the window size in the non-local mean algorithm of four types of data. (a) Farmland D data set. (b) Farmland C data set. (c) Inland water data set. (d) Coastline data set.



**Figure 21.** The wrong results on the smoothing parameter in the non-local mean algorithm of four types of data. (a) Farmland D data set. (b) Farmland C data set. (c) Inland water data set. (d) Coastline data set.

Because Farmland D, Farmland C, and the inland water have relatively large changed areas and have more noise, we will choose a relatively larger search window size. Moreover, the smoothing parameter is a balanced parameter between the denoising ability and the image detail retention ability. If the smoothing parameter is larger, the denoising ability is greater, and if the smoothing parameter is smaller, more details will be saved. Moreover, because Farmland D, Farmland C, and the inland water data have more noise, we need to select a larger smoothing parameter than that of the coastline data. Therefore, setting the window size of Farmland D, Farmland C, and the inland water as  $4 \times 4$ , their smoothing parameters *h* as 0.5, the window size of the coastline as  $2 \times 2$ , and the coastline's smoothing parameter *h* as 0.15 can obtain great results.

# 4.4.3. Parameters of gcForest

gcForest has two parameters that need to be set: the number of trees in the multi-grained scanning and the number of trees in the cascade. In this experiment, we change the number of trees in the multi-grained scanning from 4 to 16 and set the number of trees in the cascade as 10. The results of *FN*, *FP*, and *OE* on the four types of data are as shown in Figure 22. The results do not show many differences. When the number of trees is 8 in the multi-grained scanning, the result are slightly improved. In Figure 23, we change the number of trees in the cascade from 5 to 20 and set the number of trees in the cascade as 8. Figure 23 shows that the values of *OE* will be slightly smaller when the number of trees in the cascade is set to 10.

By the experiments on these four kinds of data, we know that adjusting these parameters in some ranges has no greater effect on the experimental results, and gcForest is robust to parameters because setting the same parameters of gcForest on the four types of data can also obtain good results.



**Figure 22.** The wrong results on the different numbers of trees in the multi-grained scanning of four types of data. (a) Farmland D data set. (b) Farmland C data set. (c) Inland water data set. (d) Coastline data set.



**Figure 23.** The wrong results on the different numbers of trees in the cascade of four types of data. (a) Farmland D data set. (b) Farmland C data set. (c) Inland water data set. (d) Coastline data set.

## 5. Conclusions

This paper presents a novel change detection algorithm based on gcForest and multi-scale image fusion for SAR change detection. The traditional methods are based on the deep learning model that chooses the single block image to train the model, but the proposed method uses the multi-scale input to obtain a better result. In order to strengthen the detection of the pixels whose gray values change abruptly, the gradient information is calculated to combine with the probability map that is produced by the well-trained gcForest. Thus, the proposed method obtains great accuracy and reduces more speckle noise than some change detection methods. Moreover, compared with the deep learning model, the multi-scale gcForest is easy to be trained due to fewer parameters. Experiments on the four kinds of data sets confirm the effectiveness of the proposed method. Compared with several existing methods, the proposed method shows a superior detection performance. Furthermore, although the existing algorithms based on the deep learning model can deal with the noise in the image well, the proposed method considers the multi-scale information and strengthens the characteristics of edge information. In the future, we will pay more attention to the change detection method based on gcForest for the different types of images, which can be optical images and heterogeneous images.

**Author Contributions:** Methodology, H.Y.; validation, H.Y.; improvement, W.M., Y.W. and T.H.; writing—original draft preparation, H.Y.; writing—review and editing, H.Y., Y.W. and Y.X.; project administration, W.M., Y.W., L.J. and B.H.; funding acquisition, W.M., Y.W., L.J. and B.H.

**Funding:** The research was jointly supported by the National Natural Science Foundations of China (No. 61702392, 61671350), and the China Postdoctoral Science Foundation (No. 2018T111022, 2017M623127).

Conflicts of Interest: The authors declare no conflict of interest.

# References

- 1. Kit, O.; Lüdeke, M. Automated detection of slum area change in Hyderabad, India using multitemporal satellite imagery. *Int. Soc. Photogramm. Remote Sens.* **2013**, *83*, 130–137. [CrossRef]
- Radke, R.J.; Andra, S.; Al-Kofahi, O.; Roysam, B. Image change detection algorithms: A systematic survey. *IEEE Trans. Image Process.* 2005, 14, 294–307. [CrossRef] [PubMed]
- 3. Dong, H.; Ma, W.; Wu, Y.; Gong, M.; Jiao, L. Local Descriptor Learning for Change Detection in Synthetic Aperture Radar Images via Convolutional Neural Networks. *IEEE Access* **2018**. [CrossRef]
- 4. Yan, L.; Xia, W.; Zhao, Z.; Wang, Y. A Novel Approach to Unsupervised Change Detection Based on Hybrid Spectral Difference. *Remote Sens.* **2018**, *10*, 841. [CrossRef]
- 5. Liu, W.; Yang, J.; Zhao, J.; Yang, L. A Novel Method of Unsupervised Change Detection Using Multi-Temporal PolSAR Images. *Remote Sens.* **2017**, *9*, 1135. [CrossRef]
- 6. Ma, W.; Wu, Y.; Gong, M.; Xiong, Y.; Yang, H.; Hu, T. Change detection in SAR images based on matrix factorisation and a Bayes classifier. *Int. J. Remote Sens.* **2018**. [CrossRef]
- 7. Ma, W.; Li, X.; Wu, Y.; Jiao, L.; Xing, D. Data fusion and fuzzy clustering on ratio images for change detection in synthetic aperture radar images. *Math Probl. Eng.* **2014**, 2014, 403095. [CrossRef]
- 8. Gong, M.; Zhang, P.; Su, L.; Liu, J. Coupled dictionary learning for change detection from multisource data. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7077–7091. [CrossRef]
- 9. Liu, M.; Zhang, H.; Wang, C.; Wu, F. Change detection of multilook polarimetric SAR images using heterogeneous clutter models. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7483–7494.
- Gokaraju, B.; Turlapaty, A.C.; Doss, D.A.; King, R.L.; Younan, N.H. Change detection analysis of tornado disaster using conditional copulas and Data Fusion for cost-effective disaster management. In Proceedings of the Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA, 13–15 October 2015; pp. 1–8.
- 11. Ke, L.; Lin, Y.; Zeng, Z.; Zhang, L.; Meng, L. Adaptive Change Detection with Significance Test. *IEEE Access* **2018**, *6*, 27442–27450. [CrossRef]
- 12. Ho, S.S.; Wechsler, H. A Martingale Framework for Detecting Changes in Data Streams by Testing Exchangeability. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 2113–2127. [PubMed]

- 13. Sumaiya, M.N.; Kumari, R.S.S. Unsupervised change detection of flood affected areas in SAR images using Rayleigh-based Bayesian thresholding. *Inst. Eng. Technol. Radar Sonar Navig.* **2018**, *12*, 515–522. [CrossRef]
- Azzouzi, S.A.; Vidal-Pantaleoni, A.; Bentounes, H.A. Desertification monitoring in Biskra, Algeria, with Landsat imagery by means of supervised classification and change detection methods. *IEEE Access* 2017, 5, 9065–9072. [CrossRef]
- 15. Lv, Z.; Liu, T.; Wan, Y.; Benediktsson, J.A.; Zhang, X. Post-Processing Approach for Refining Raw Land Cover Change Detection of Very High-Resolution Remote Sensing Images. *Remote Sens.* **2018**, *10*, 472. [CrossRef]
- Wang, K.; Gou, C.; Wang, F.Y. M<sup>4</sup>CD: A Robust Change Detection Method for Intelligent Visual Surveillance. IEEE Access 2018, 6, 15505–15520. [CrossRef]
- 17. Bruzzone, L.; Prieto, D.F. An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images. *IEEE Signal Process. Soc.* **2002**, *11*, 452–466. [CrossRef]
- 18. Bovolo, F.; Bruzzone, L. A detail-preserving scale-driven approach to change detection in multitemporal SAR images. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 2963–2972. [CrossRef]
- Inglada, J.; Mercier, G. A new statistical similarity measure for change detection in multitemporal SAR images and its extension to multiscale change analysis. *IEEE Trans. Geosci. Remote Sens.* 2007, 45, 1432–1445. [CrossRef]
- 20. Zhong, Y.; Ma, A.; Zhang, L. An adaptive memetic fuzzy clustering algorithm with spatial information for remote sensing imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 1235–1248. [CrossRef]
- Bazi, Y.; Bruzzone, L.; Melgani, F. An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images. *IEEE Trans. Geosci. Remote Sens.* 2005, 43, 874–887. [CrossRef]
- 22. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc.* **1977**, *39*, 1–38. [CrossRef]
- 23. Yetgin, Z. Unsupervised change detection of satellite images using local gradual descent. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 1919–1929. [CrossRef]
- 24. Ghosh, A.; Mishra, N.S.; Ghosh, S. Fuzzy clustering algorithms for unsupervised change detection in remote sensing images. *Inf. Sci.* 2011, *181*, 699–715. [CrossRef]
- 25. Gong, M.; Zhou, Z.; Ma, J. Change detection in synthetic aperture radar images based on image fusion and fuzzy clustering. *IEEE Trans. Image Process.* **2012**, *21*, 2141–2151. [CrossRef] [PubMed]
- 26. Zhang, X.; Chen, J.; Meng, H. A novel SAR image change detection based on graph-cut and generalized Gaussian model. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 14–18. [CrossRef]
- 27. Hebel, M.; Arens, M.; Stilla, U. Change detection in urban areas by object-based analysis and on-the-fly comparison of multi-view ALS data. *Int. Soc. Photogramm. Remote Sens.* **2013**, *86*, 52–64. [CrossRef]
- 28. Celik, T. Unsupervised Change Detection in Satellite Images Using Principal Component Analysis and k-Means Clustering. *IEEE Geosci. Remote Sens. Lett.* **2009**, *6*, 772–776. [CrossRef]
- 29. Wu, Y.; Ma, W.; Gong, M.; Li, H.; Jiao, L. Novel fuzzy active contour model with kernel metric for image segmentation. *Appl. Soft Comput.* **2015**, *34*, 301–311. [CrossRef]
- 30. Wu, K.; Du, Q.; Wang, Y.; Yang, Y. Supervised sub-pixel mapping for change detection from remotely sensed images with different resolutions. *Remote Sens.* **2017**, *9*, 284. [CrossRef]
- 31. Zhan, Y.; Fu, K.; Yan, M.; Sun, X.; Wang, H.; Qiu, X. Change Detection Based on Deep Siamese Convolutional Network for Optical Aerial Images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1845–1849. [CrossRef]
- 32. Gao, F.; Dong, J.; Li, B.; Xu, Q. Automatic change detection in synthetic aperture radar images based on PCANet. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1792–1796. [CrossRef]
- 33. Gong, M.; Zhao, J.; Liu, J.; Miao, Q.; Jiao, L. Change detection in synthetic aperture radar images based on deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 125–138. [CrossRef] [PubMed]
- Liu, J.; Gong, M.; Qin, K.; Zhang, P. A deep convolutional coupling network for change detection based on heterogeneous optical and radar images. *IEEE Trans. Neural Netw. Learn. Syst.* 2016, 29, 545–559. [CrossRef] [PubMed]
- Gong, M.; Yang, H.; Zhang, P. Feature learning and change feature classification based on deep learning for ternary change detection in SAR images. *Int. Soc. Photogramm. Remote Sens.* 2017, 129, 212–225. [CrossRef]
- 36. Xu, Z.; Chen, J.; Xia, J.; Du, P.; Zheng, H.; Gan, L. Multisource Earth Observation Data for Land-Cover Classification Using Random Forest. *IEEE Geosci. Remote Sens. Lett.* **2018**, 15, 789–793. [CrossRef]

- 37. Lyu, H.; Lu, H.; Mou, L. Learning a transferable change rule from a recurrent neural network for land cover change detection. *Remote Sens.* **2016**, *8*, 506. [CrossRef]
- 38. Paul, A.; Mukherjee, D.P.; Das, P.; Gangopadhyay, A.; Chintha, A.R.; Kundu, S. Improved Random Forest for Classification. *IEEE Trans. Image Process.* **2018**, *27*, 4012–4024. [CrossRef]
- 39. Mu, X.; Ting, K.M.; Zhou, Z.H. Classification under streaming emerging new classes: A solution using completely-random trees. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 1605–1618. [CrossRef]
- 40. Zhou, Z.H.; Feng, J. Deep forest: Towards an alternative to deep neural networks. arXiv 2017, arXiv:1702.08835.
- Man, W.; Ji, Y.; Zhang, Z. Image classification based on improved random forest algorithm. In Proceedings of the International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 20–22 April 2018; pp. 346–350.
- 42. Wu, Y.; Ma, W.; Gong, M.; Su, L.; Jiao, L. A Novel Point-Matching Algorithm Based on Fast Sample Consensus for Image Registration. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 43–47. [CrossRef]
- 43. Hussain, M.; Chen, D.; Cheng, A.; Wei, H.; Stanley, D. Change detection from remotely sensed images: From pixel-based to object-based approaches. *Int. Soc. Photogramm. Remote Sens.* **2013**, *80*, 91–106. [CrossRef]
- 44. Feng, J.; Zhou, Z.H. AutoEncoder by Forest. arXiv 2017, arXiv:1709.09018.
- 45. Liaw, A.; Wiener, M.; others. Classification and regression by randomForest. R News 2002, 2, 18–22.
- 46. Pal, M. Random forest classifier for remote sensing classification. *Int. J. Remote Sens.* 2005, 26, 217–222. [CrossRef]
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
- 48. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* 2014, arXiv:1406.1078.
- 49. Liu, F.T.; Ting, K.M.; Yu, Y.; Zhou, Z.H. Spectrum of variable-random trees. J. Artif. Intell. Res. 2008, 32, 355–384. [CrossRef]
- Gashler, M.; Giraud-Carrier, C.; Martinez, T. Decision tree ensemble: Small heterogeneous is better than large homogeneous. In Proceedings of the International Conference on Machine Learning and Applications, San Diego, CA, USA, 11–13 December 2008; pp. 900–905.
- 51. Cheng, W.C.; Jhan, D.M. Triaxial accelerometer-based fall detection method using a self-constructing cascade-AdaBoost-SVM classifier. *IEEE J. Biomed. Health Inform.* **2013**, *17*, 411–419. [CrossRef] [PubMed]
- 52. Belgiu, M.; Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. *Int. Soc. Photogramm. Remote Sens.* **2016**, *114*, 24–31. [CrossRef]
- 53. Manjón, J.V.; Carbonell-Caballero, J.; Lull, J.J.; García-Martí, G.; Martí-Bonmatí, L.; Robles, M. MRI denoising using non-local means. *Med. Image Anal.* **2008**, *12*, 514–523. [CrossRef]
- 54. Chuang, K.S.; Tzeng, H.L.; Chen, S.; Wu, J.; Chen, T.J. Fuzzy c-means clustering with spatial information for image segmentation. *Comput. Med. Imaging Graph.* **2006**, *30*, 9–15. [CrossRef]
- Corvee, E.; Bremond, F. Body parts detection for people tracking using trees of histogram of oriented gradient descriptors. In Proceedings of the Advanced Video and Signal Based Surveillance (AVSS), Boston, MA, USA, 29 August–1 September 2010; pp. 469–475.
- Rosin, P.L.; Ioannidis, E. Evaluation of global image thresholding for change detection. *Pattern Recognit. Lett.* 2003, 24, 2345–2356. [CrossRef]
- Zhao, J.; Gong, M.; Liu, J.; Jiao, L. Deep learning to classify difference image for image change detection. In Proceedings of the International Joint Conference on Neural Networks(IJCNN), Beijing, China, 6–11 July 2014; pp. 411–417.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).