

# Robust unsupervised small area change detection from SAR imagery using deep learning

Xinzheng Zhang<sup>a,b,\*</sup>, Hang Su<sup>a</sup>, Ce Zhang<sup>c,d,\*</sup>, Xiaowei Gu<sup>e</sup>, Xiaoheng Tan<sup>a,b</sup>, Peter M. Atkinson<sup>c,f,g</sup>

<sup>a</sup> School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, 400044, China

<sup>b</sup> Chongqing Key Laboratory of Space Information Network and Intelligent Information Fusion, Chongqing, 400044, China

<sup>c</sup> Lancaster Environment Centre, Lancaster University, Lancaster, LA1 4YQ, United Kingdom

<sup>d</sup> UK Centre for Ecology & Hydrology, Library Avenue, Lancaster, LA1 4AP, United Kingdom

<sup>e</sup> Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, United Kingdom

<sup>f</sup> Geography and Environmental Science, University of Southampton, Highfield, Southampton SO17 1BJ, UK

<sup>g</sup> Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, 11A Datun Road, Beijing 100101, China

## Abstract

Small area change detection using synthetic aperture radar (SAR) imagery is a highly challenging task, due to speckle noise and imbalance between classes (changed and unchanged). In this paper, a robust unsupervised approach is proposed for small area change detection using deep learning techniques. First, a multi-scale superpixel reconstruction method is developed to generate a difference image (DI), which can suppress the speckle noise effectively and enhance edges by exploiting local, spatially homogeneous information. Second, a two-stage centre-constrained fuzzy *c*-means clustering algorithm is proposed to divide the pixels of the DI into changed, unchanged and intermediate classes with a parallel clustering strategy. Image patches belonging to the first two classes are then constructed as pseudo-label training samples, and image patches of the intermediate class are treated as testing samples. Finally, a convolutional wavelet neural network (CWNN) is designed and trained to classify testing samples into changed or unchanged classes, coupled with a deep convolutional generative adversarial network (DCGAN) to increase the number of changed class within the pseudo-label training samples. Numerical experiments on four real SAR datasets demonstrate the validity and robustness of the proposed approach, achieving up to 99.61% accuracy for small area change detection.

## Keywords:

Change detection; Synthetic aperture radar; Difference image; Fuzzy *c*-means algorithm; Deep learning.

## 1. Introduction

Remotely sensed change detection focuses on identifying land-cover changes by analysing the images observed over the same scene at different times (Huang et al., 2011; Li et al., 2020; Cao et al., 2020). In particular, remotely sensed synthetic aperture radar (SAR) imagery is adopted widely for change detection, owing to its ability to penetrate cloud cover, and its insensitivity to atmospheric and lighting conditions (Wang et al., 2016; Gong et al., 2017; Li et al., 2019). Change detection using SAR imagery has been applied in a variety of real-world settings, such as for earthquake damage assessment (Brunner et al., 2010), forest mapping (Pantze et al., 2014) and flood monitoring (Kim et al., 2020). Over the past decades, tremendous effort has been made to develop automatic change detection methods using multi-temporal SAR images. Amongst them, machine learning is currently considered as the most promising and evolving set of approaches (Gong et al., 2017). In general, machine learning-based change detection can be divided into supervised and unsupervised approaches (Li et al., 2019; Geng et al., 2019). The major issue in relation to the supervised approach is the lack of ground reference data, and it often involves manual labelling processes that are labour-intensive and time-consuming (Saha et al., 2020). Thus, unsupervised approaches are developed and employed widely in this field (Li et al., 2015; Jia et al., 2016). The major components of unsupervised approaches include: 1) image pre-processing, 2) difference image (DI) generation, and 3) analysis of the DI and the classification of pixels into changed and unchanged classes (Li et al., 2020; Wang et al., 2020).

The image pre-processing step is used to co-register the SAR images (Lei et al., 2019) and filter speckle noise (Gong et al., 2012; Gong et al., 2016; Li et al., 2018). In SAR images, speckle noise severely degrades the quality of images, thus affecting the performance of subsequent tasks. Numerous methods have been developed to remove speckle noise from SAR images, such as Frost filtering (Sun et al., 2020), non-local means approach (Deledalle et al., 2015), and deep learning method (Cozzolino et al., 2020).

DI generation is aim to provide guidance for post-processing. Subtraction (S) and logarithmic ratio (LR) operators are two basic techniques to obtain pixel-by-pixel DI. Some approaches exploited spatial information in a local window for DI generation, such as the mean ratio (MR) and the neighbourhood-based ratio (NR) (Inglada et al., 2007; Gong et al., 2012), where speckle noise is reduced and the discrimination capability of DI is enhanced.

58 The DI analysis is a critical step, by which change detection is converted into a binary classification task  
59 (e.g. Thresholding or  $k$ -means clustering). Typically, fuzzy  $c$ -means (FCM) is a widely adopted clustering  
60 approach in change detection using SAR imagery (Gong et al., 2012; Gao et al., 2019; Li et al., 2019).  
61 Several FCM variants have been proposed to reduce the speckle noise while retaining spatial details (Krinidis  
62 et al., 2010; Gong et al., 2012; Tian et al., 2018). Recent researches suggest that the DI should be divided  
63 into three categories: high-probability changed, high-probability unchanged, and an intermediate class (Li et  
64 al., 2019; Kalaiselvi et al., 2020). The intermediate class represents pixels that are difficult to discriminate  
65 by a specific clustering algorithm, which are generally differentiated by a deep learning classifier such as a  
66 convolutional neural network (CNN) or a convolutional-wavelet neural network (CWNN) (Li et al. 2019;  
67 Gao et al., 2019).

68 One common issue in unsupervised deep learning approaches is to deal with small area changes from  
69 multi-temporal SAR images. Such a problem exists in multiple steps as follows:

70 **DI generation:** most DI generation methods adopt rectangular windows to characterise local spatial  
71 information. Such strategy will smooth out small area changes or fine details along the changed and  
72 unchanged region, where changed pixels are difficult to identify (Wang et al., 2020).

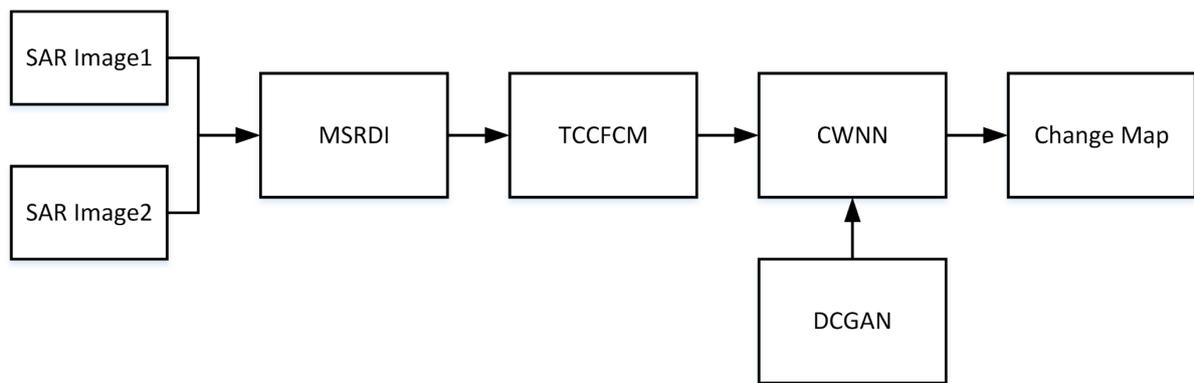
73 **The clustering-based DI analysis:** many FCM-based methods fail to produce accurate results, due  
74 mainly to the use of overall optimisation objective, which forces the cluster prototype of the minority  
75 (changed) class to migrate to the majority (unchanged) class, particularly for the imbalanced data distribution.

76 **Training a deep learning classifier:** The number of changed pixels is far less than unchanged pixels in  
77 small area change detection, where the pseudo-label training samples of the changed class are insufficient to  
78 support the training of deep learning classifier, leading to poor classification accuracy.

79 To address these issues, we developed a novel set of methods as Robust Unsupervised Small Area  
80 Change Detection (RUSACD) approach. First, a novel DI generation method is proposed based on multi-  
81 scale superpixel reconstruction (MSRDI), by exploiting the local spatial information within a superpixel (i.e.,  
82 an image object) instead of a regular window. Multi-scale information is extracted in superpixel  
83 segmentation and DI reconstruction, followed by fusion to achieve the final MSRDI. Second, a two-stage  
84 centre-constrained FCM algorithm (TCCFCM) is developed to group imbalanced data. The TCCFCM is  
85 designed to identify the preliminary and reliable clustering centres of the changed and unchanged classes.  
86 These cluster centres are used as constraints to build the objective function and to prevent incorrect migration

87 of cluster prototypes. Finally, a deep convolutional generative adversarial network (DCGAN) was embedded  
88 in this framework for data augmentation, which addressed the issue of insufficient training samples and class  
89 imbalance.

90 The major contributions include:1) a novel MSRDI generating method was developed, which can  
91 effectively enhance the quality and separability of the DI. 2) TCCFCM was proposed for analysing MSRDI,  
92 coupled with a parallel clustering strategy to divide the pixels of MSRDI into three classes: changed,  
93 unchanged and intermediate classes. 3) a DCGAN was applied to increase the number of training samples  
94 and the visual features of the changed class and, thus, achieve a balance of training samples amongst classes.  
95 This was followed by a CWNN to differentiate the intermediate class into changed and unchanged classes.  
96 The new method was tested comprehensively on six real SAR image datasets. The overall pipeline of the  
97 RUSACD is shown in Fig. 1, demonstrating how each component is linked together in a modular design.  
98



99  
100 Fig.1 Simple flow diagram illustrating the processing pipeline.

## 101 2. Related work

### 102 2.1 DI generation

103 In the step of DI generation, several methods have been developed to exploit neighbourhood spatial  
104 information in a local window, such as MR and NR. In (Zhuang et al., 2018), the authors proposed a spatial-  
105 temporal adaptive neighbourhood-based ratio to select optimal window size for solving the shortcoming of  
106 fixed-size rectangular window. Further, an adaptive generalized likelihood ratio test (AGLRT) was  
107 developed to weaken the geometric degradation of the DI caused by integrating heterogeneous pixels  
108 (Zhuang et al., 2020). Recently, Wang et al. (2020) developed a method of generating DI using irregular  
109 spatial information, in which the clear boundaries and textures were preserved by matching hypergraphs.

## 2.2 DI analysis based on clustering

One classical algorithm for DI analysis is  $k$ -means clustering. In (Celik, 2009), principal component analysis (PCA) feature extraction combined with  $k$ -means clustering was applied to assign each pixel of the DI into the changed or unchanged class. Another typical algorithm is the FCM, which is more prevalent than the former in SAR-image change detection (Gong et al., 2012; Gao et al., 2019; Li et al., 2019). The FCM clustered the Gabor feature vectors of the DI in SAR-based change detection (Li et al., 2015). To control the sensitivity to noise, a fuzzy local information c-means (FLICM) algorithm was developed by incorporating local spatial information and grey-level characteristics (Krinidis et al., 2010). Gong et al. (2012) reformulated the FLICM (RFLICM) using coefficient of variation to replace Euclidean distance, and employed it for change detection in SAR images. Tian et al. (2018) proposed an edge-weighted FCM by introducing a piecewise smooth prior to balance the trade-off between noise reduction and edge preservation. Li et al. (2019) developed a spatial FCM (SFCM) algorithm with a spatial function added into the fuzzy membership for noise suppression.

## 2.3 Application of deep learning to change detection

Recently, deep learning has been introduced to discriminate pixels of the intermediate class for SAR-based change detection, thanks to its ability to learn high-level semantic and contextual features automatically. There are various deep neural networks available for change detection, such as PCANet, nonnegative- and Fisher-constrained autoencoder (NFCAE), CNN and CWNN (Gao et al., 2016; Geng et al., 2019; Li et al., 2019; Gao et al., 2019). CWNN is able to suppress speckle noise and suitable for processing SAR data by incorporating a wavelet-constrained pooling layer compared with a standard CNN (Duan et al., 2017). To train the deep learning model, a large amount of training samples is needed to ensure its classification performance. When training data are insufficient or imbalanced among all classes, data augmentation is usually adopted to handle the issue. Common augmentation techniques include affine transformations such as rotation, translation and scaling, as well as linear combination strategy (Shorten et al., 2019; Gao et al., 2019). However, these simple strategies are limited the generalisation capability of deep learning models (Frid-Adar et al., 2018). Generative adversarial networks (GANs) are powerful deep networks for training sample generation (Goodfellow et al., 2014; Shorten et al., 2019). The GAN model consists of two networks (generator and discriminator) that are trained in an adversarial fashion, where the

generator creates fake images and the discriminator discriminates between real and fake images. Radford et al. (2015) further developed a DCGAN, where both generator and discriminator were composed of deep CNNs with fully convolution layers. The adversarial pair in DCGAN can learn a hierarchy of representations both from image parts and entire images, making it superior to simple linear generation methods for data augmentation (Frid-Adar et al., 2018).

### 3. The proposed approach

Implementation of the proposed RUSACD approach is shown in Fig. 2. First, two SAR images are filtered to denoise the image and enhance spatial features in a pre-processing stage, establishing a primary DI through the LR operator. Superpixel-based reconstruction is performed on the primary DI in multi-scale spaces, and the results fused into a MSRDI. Second, Gabor wavelet feature extraction is applied to the MSRDI, with Gabor feature vectors acquired at the pixel level. Parallel TCCFCM clustering is conducted to cluster these feature vectors as pseudo-labels (the changed, unchanged and intermediate classes). Third, the image patches together with pseudo-labels are obtained from two SAR images. The DCGAN is employed to enrich the pseudo-label image patches of the changed class. Finally, a CWNN is trained using these image patches with pseudo-labels, and is used to allocate the intermediate pixels to the changed or unchanged class. Thus, all pixels are marked with a unique label as changed or unchanged, obtaining the final change map. For clarity, pixels allocated to the changed and unchanged categories by the parallel TCCFCM are defined as simple pixels, since they can be discriminated easily. In contrast, intermediate pixels are treated as hard pixels due to the difficulty in discrimination by clustering alone.

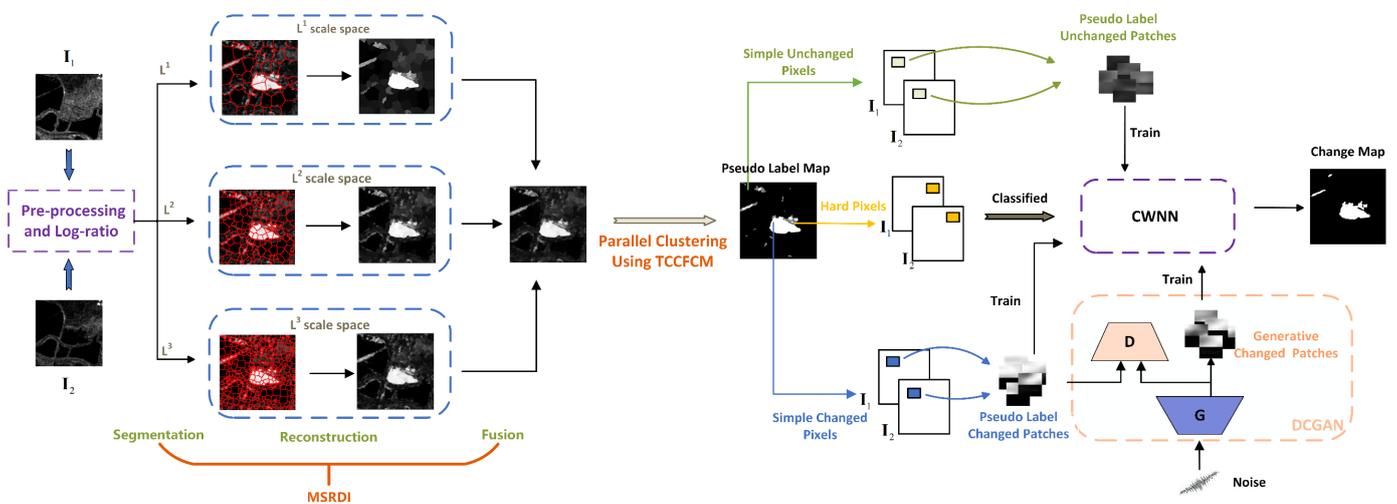


Fig. 2. Flowchart illustrating the proposed RUSACD methodology. Pseudo labels are drawn by three colours, black indicates unchanged pixels, white indicates changed pixels, and gray indicates intermediate (hard) pixels in Pseudo Label Map.

### 3.1 Pre-processing

A weighted average filter is designed for pre-processing, which can suppress noise and enhance spatial domain features effectively. This filter  $\mathbf{W}^\eta$  is defined as a  $\eta \times \eta$  matrix, where  $\eta$  is an odd number. Each element value of the matrix is determined by the distance between the current position and the centre of the matrix, as shown in Eq. (1). The centre of the matrix is defined as  $w_{(\eta+1)/2, (\eta+1)/2} = 2 / \eta^2$ . The proposed filter maintains edges while suppressing speckle noise compared to other filters such as the mean filter, and retains the continuity of the image.

$$w_{ij} = \frac{1}{\eta^2 \sqrt{\left(\frac{\eta+1}{2} - i\right)^2 + \left(\frac{\eta+1}{2} - j\right)^2}} \quad i, j = 1, 2, \dots, \eta \quad (1)$$

Given multi-temporal SAR images  $\mathbf{I}_1$  and  $\mathbf{I}_2$ , where  $\mathbf{I}_1, \mathbf{I}_2 \in \mathbf{R}^{N_r \times N_c}$ ,  $N_r$  and  $N_c$  refer to the row and column of the images respectively. The convolution of each SAR image using  $\mathbf{W}^\eta$  is taken to acquire two images  $\mathbf{I}_1^w(\eta) = \mathbf{I}_1 * \mathbf{W}^\eta$  and  $\mathbf{I}_2^w(\eta) = \mathbf{I}_2 * \mathbf{W}^\eta$ , where  $*$  denotes the 2-D convolution operation.

### 3.2 Generating DI based on multi-scale superpixel reconstruction

Simple Linear Iterative Clustering (SLIC) (Achanta et al., 2012) is used as the basis for superpixel segmentation. A multiscale superpixel reconstruction (MSRDI) is designed here for DI generation. The formal steps for MSRDI include:

**Step 1:** A log-ratio operator is used to generate the primary DI, and the log-ratio image  $\mathbf{I}_{LR}$  is calculated as  $\mathbf{I}_{LR} = \left| \log(\mathbf{I}_2^w / \mathbf{I}_1^w) \right|$ .

**Step 2:** The filter template  $\mathbf{W}^\eta$  is used to smooth the speckle in the log-ratio (LR) image  $\mathbf{I}_{LR}$ . The processed log-ratio image is described as  $\mathbf{I}_{SLR}(\eta) = \mathbf{I}_{LR} * \mathbf{W}^\eta$ .

**Step 3:** SLIC is used to split  $\mathbf{I}_{SLR}$  into superpixels, each of which includes a number of pixels. Those superpixels are denoted as a set  $\{\mathbf{S}_l\}_{l=1}^{l=L}$ , where  $l$  and  $L$  represent the index and number of superpixels respectively. Each pixel in  $\mathbf{I}_{SLR}$  is located in a corresponding superpixel  $\mathbf{S}_l$ . The median is an indicator to evaluate the level of a superpixel value, which is resistant to speckle noise. The mean contains homogeneous information of the corresponding superpixel. The mean value of each superpixel  $o_l$  is obtained by calculating the average value of all pixels inside, which are collected into a set  $\{o_l\}_{l=1}^{l=L}$ . Similarly, the median of each superpixel  $p_l$  can be calculated and collected into a set  $\{p_l\}_{l=1}^{l=L}$ . The superpixel-based DI is reconstructed pixel-by-pixel as

$$I_{SRDI}(i, j) = \alpha_1 I_{i,j} + \alpha_2 p_l + \alpha_3 o_l \quad (2)$$

where  $I_{i,j}$  is the pixel value at position  $(i, j)$  of  $\mathbf{I}_{LR}$ , corresponding to the superpixel  $S_l$ . The above superpixel segmentation and reconstruction are carried out at multiple scales to exploit multi-scale spatial context information.  $T$  superpixel-based DIs are reconstructed in each scale.

**Step 4:** These  $T$  superpixel-based reconstruction DIs are fused to generate the MSRDI.

$$\mathbf{I}_{MSRDI} = \frac{1}{T} \sum_{t=1}^T \mathbf{I}_{SRDI}^t \quad (3)$$

The generation process of MSRDI is shown in Fig. 2. Different scale spaces are corresponding to different  $L$  for superpixel segmentation.

In addition, a one-dimensional signal is designed to demonstrate how MSRDI can restrain speckle noise and enhance edges (Fig. 3). Compared to other DI generation methods, the MSRDI (Figure 3(d)) has the following advantages: (1) the gap between pixels within a class (changed or unchanged) tends to be small, (2) a significant distinction is present between the two classes, and (3) the boundaries are enhanced and the details are well preserved.

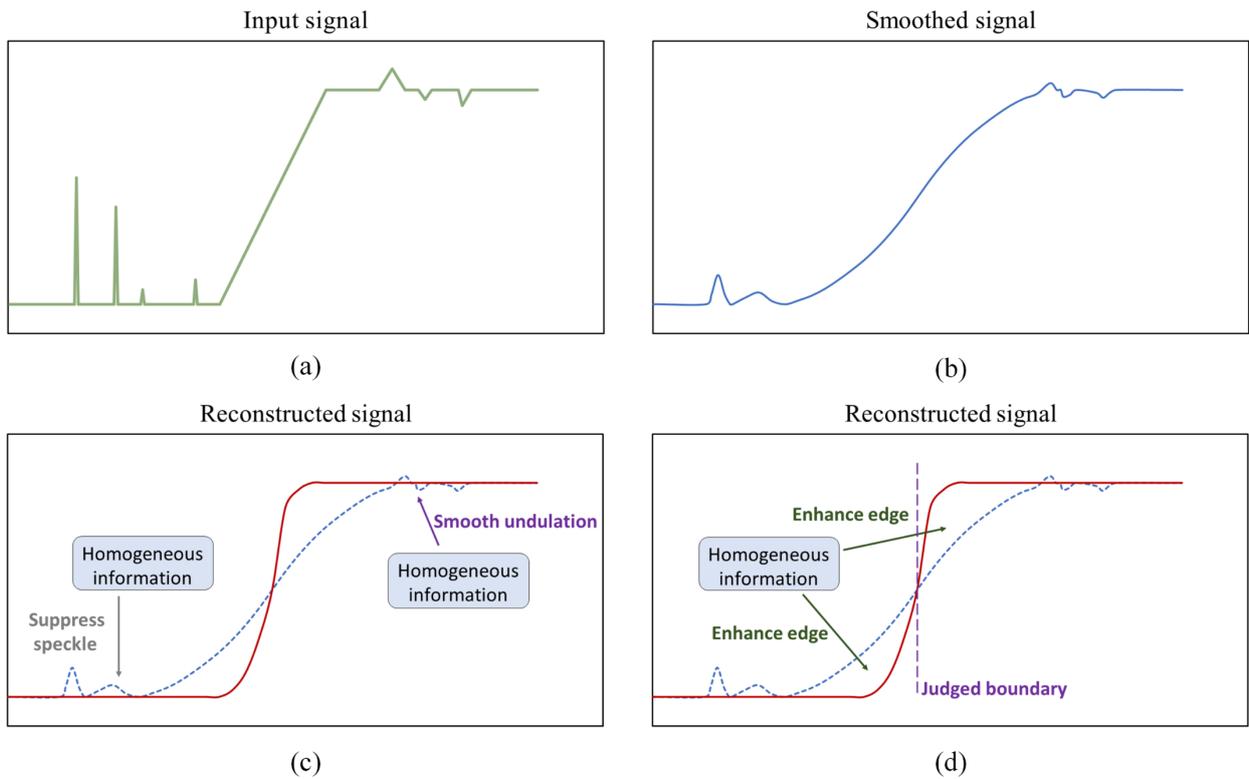


Fig. 3. Diagram using a one-dimensional signal to illustrate how MSRDI can minimise speckle noise and enhance edges. (a) The input signal with speckle and distortion. (b) The smoothed signal using filter  $\mathbf{W}^n$ . (c) The reconstructed signal, in which noise is suppressed and distortion is mitigated. (d) The reconstructed signal, where the edges are enhanced.

### 3.3 Parallel TCCFCM clustering

A two-stage centre-constrained FCM algorithm (TCCFCM) is designed for imbalanced data clustering to solve the small area change detection problem (Fig. 4). The first stage of TCCFCM is to find the preliminary and reliable clustering centres. The most important feature information provided by the DI is the pixel values: high-intensity pixels represent the changed class and low-intensity pixels refer to the unchanged class. Therefore, we select the top  $N_p$  samples with prominent high-intensity  $\{x_p^h\}_{p=1}^{N_p}$  and top  $N_p$  samples with low  $\{x_p^l\}_{p=1}^{N_p}$  values. These samples are then clustered by FCM to obtain cluster centres  $v_c^{pre}$ . Those cluster centres are used as constraints in the second stage to prevent incorrect transfer of cluster prototypes (Fig. 4 (c) and (d)).

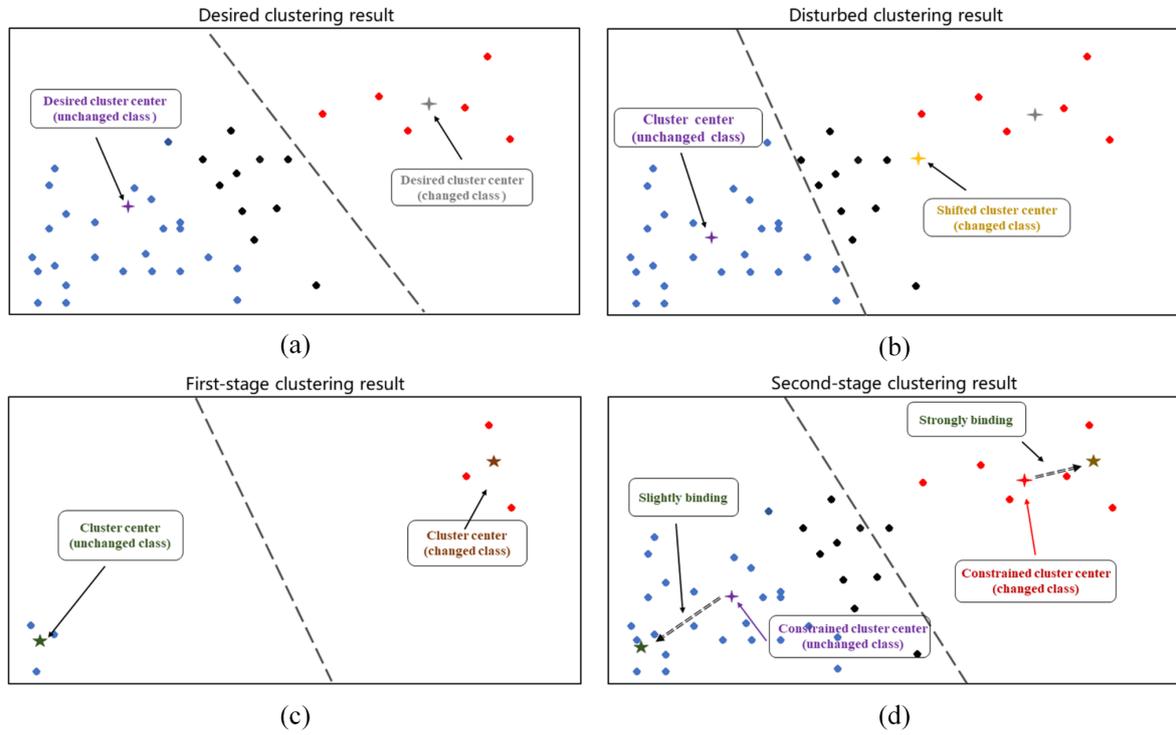


Fig. 4. Illustration of the principle of TCCFCM clustering. The red dots, blue dots and black dots represent pixels or samples of the changed class, low-intensity pixels or samples of the unchanged class, and medium-intensity pixels or samples of the unchanged class respectively. In (c), we obtain reliable clustering centres in the first stage. The reliable centres then constrain cluster prototypes to increase accuracy in the second stage, as shown in (d).

The second stage of TCCFCM performs clustering on all samples  $\{x_n\}_{n=1}^{N_r N_c}$ . The objective function is defined as:

$$J = \sum_{c=1}^2 \sum_{n=1}^{N_r N_c} u_{cn}^m \left\| (1 - \beta_c) x_n + \beta_c v_c^{pre} - v_c \right\|^2 \quad (4)$$

where  $\beta_c$  is control parameter. We impose a strong constraint by setting a large value for the minority control parameter  $\beta_1 (c = 1)$ , and a small value for the majority control parameter  $\beta_2 (c = 2)$  to impose a

223 weak constraint. Hence, the two parameters are simplified as  $\beta_1 = \beta$  and  $\beta_2 = 0.7 \cdot \beta$ , which means that  
 224  $\beta_1$  and  $\beta_2$  are controlled uniformly by  $\beta$ . The element  $u_{cn}$  of the membership partition matrix  $\mathbf{U}$  and  
 225 the cluster centre is derived as:

$$226 \quad u_{cn} = \frac{1}{\sum_{j=1}^2 \left( \frac{\|(1-\beta_c)x_n + \beta_c v_c^{pre} - v_c\|^2}{\|(1-\beta_j)x_n + \beta_j v_j^{pre} - v_j\|^2} \right)^{1/(m-1)}} \quad (5)$$

$$227 \quad v_c = (1-\beta_c) \frac{\sum_{n=1}^N u_{cn}^m x_n}{\sum_{n=1}^N u_{cn}^m} + \beta_c v_c^{pre} \quad (6)$$

228 where the initial membership partition matrix ( $\mathbf{U}^1$ ) is set randomly at the first stage, and the initial  
 229 membership partition matrix ( $\mathbf{U}^2$ ) at the second stage is derived by the clustering centre  $v_c^{pre}$ .

230 A parallel clustering strategy based on TCCFCM is designed to discriminate simple pixels of both  
 231 changed and unchanged classes, and hard pixels of the intermediate class. At the beginning of clustering, the  
 232 sigmoid function is employed to perform two nonlinear mappings with different parameters on the MSRDI.  
 233 One mapping prefers the significant changes, and the other prefers the weak changes. Then, a Gabor wavelet  
 234 transform is used for feature extraction on the two mapped MSRDI (Li et al., 2015; Gao et al, 2016),  
 235 implemented by convolving  $\mathbf{I}_{MSRDI}$  with a set of Gabor kernels in eight directions on  $\gamma$  scales. The  
 236 maximum response is considered as a feature at this scale. The Gabor feature vector is expressed as  
 237  $\mathbf{z} = [z_0, z_1, \dots, z_{\gamma-1}]$ . The details of the parallel TCCFCM clustering are as follows:

238 **Step 1:** Normalize and centralize the MSRDI  $\mathbf{I}_{MSRDI}$ .

239 **Step 2:** Apply two sigmoid mappings to  $\mathbf{I}_{MSRDI}$  with two different parameter sets at each pixel to achieve  
 240  $\mathbf{I}_1^M = \text{sigmoid}(\mathbf{I}_{MSRDI}; \mu_1)$  and  $\mathbf{I}_2^M = \text{sigmoid}(\mathbf{I}_{MSRDI}; \mu_2)$ , where  $\mu_1$  and  $\mu_2$  represent two parameters. The  
 241 sigmoid function is given by Eq. (7).

$$242 \quad \text{sigmoid}(x; \mu) = \frac{1}{1 + e^{-(x+\mu)}} \quad (7)$$

243 **Step 3:** Apply Gabor feature extraction to  $\mathbf{I}_1^M$  and  $\mathbf{I}_2^M$ , and two Gabor feature vector sets are obtained,  
 244 comprising  $\mathbf{Z}^1 = [\mathbf{z}_1^1, \mathbf{z}_2^1, \dots, \mathbf{z}_{N_r N_c}^1]$  and  $\mathbf{Z}^2 = [\mathbf{z}_1^2, \mathbf{z}_2^2, \dots, \mathbf{z}_{N_r N_c}^2]$ , where  $\mathbf{z}_i^j \in \mathbf{R}^\gamma$  ( $i = 1, 2, \dots, N_r N_c; j = 1, 2$ )  
 245 represents a Gabor feature vector.

246 **Step 4:** TCCFCM is utilised to perform two-class clustering on  $\mathbf{Z}^1$  and  $\mathbf{Z}^2$  to obtain two label sets  
 247  $\mathbf{Y}^1 = [y_1^1, y_2^1, \dots, y_i^1, \dots, y_{N_r N_c}^1]$  and  $\mathbf{Y}^2 = [y_1^2, y_2^2, \dots, y_i^2, \dots, y_{N_r N_c}^2]$ , respectively, where  $y_i^j$  represents a label

248 corresponding to a Gabor feature vector. The value of  $y_i^j$  is either 0 or 1. The simple average operation  
 249 is used to encode the two label sets, obtaining the final label set  $\mathbf{Y}=[y_1, y_2, \dots, y_s, \dots, y_{N_r N_c}]$ , where  
 250  $y_i = (y_i^1 + y_i^2) / 2$ .

251 **Step 5:** If  $y_i = 1$ , the corresponding pixel is assigned to the changed class  $\omega_c$ . If  $y_i = 0$ , the  
 252 corresponding pixel is assigned to the unchanged class  $\omega_{uc}$ . The others represent hard pixels assigned to the  
 253 hard class  $\omega_h$ .

### 254 3.4 Deep learning CWNN and DCGAN

255 To increase the accuracy of change detection for a small area, a deep CWNN is constructed to classify  
 256 hard pixels. This is augmented by a DCGAN to increase the sample size for the rare changed class. A patch  
 257 of size  $\lambda \times 2\lambda$  is selected, centred on a pixel of the changed class pseudo-label in the SAR image  $\mathbf{I}_1$ .  
 258 Another patch is acquired in the same position from the SAR image  $\mathbf{I}_2$  in the same way. Both patches are  
 259 concatenated into a new patch with  $2\lambda \times 2\lambda$  size as a pseudo-label training sample of the changed class.  
 260 All pseudo-label training samples are produced according to the above method, denoted as  $\{\mathbf{P}_{\tau_1}^c\}_{\tau_1=1}^{N_1}$  and  
 261  $\{\mathbf{P}_{\tau_2}^{uc}\}_{\tau_2=1}^{N_2}$ , where  $\mathbf{P}_{\tau_1}^c, \mathbf{P}_{\tau_2}^{uc} \in \mathbf{R}^{2\lambda \times 2\lambda}$ , representing the changed and unchanged patches, respectively. Hard  
 262 patches  $\{\mathbf{P}_q^h\}_{q=1}^{N_h}$  relative to hard pixels are obtained in the same way. Thereafter, a DCGAN is used to  
 263 enrich the pseudo-label training samples of the changed class. **The input vectors  $\boldsymbol{\phi}$  are initialized using**  
 264 **random Gaussian noise**, and are fed into the generator  $G(\cdot)$  to generate false images, and the discriminator  
 265  $D(\cdot)$  is built to distinguish the fake images from false and real images. Here, a convolutional generative  
 266 model  $G(\cdot)$  is built, with an objective max-min function as:

$$267 \quad \begin{aligned} V(G, D) &= L^f + L^r \\ &= \mathbf{E}[\log(1 - D(G(\boldsymbol{\phi})))] + \mathbf{E}[\log(D(\mathbf{P}_{\tau_1}^c))] \end{aligned} \quad (8)$$

$$268 \quad G^* = \arg \min_G \max_D V(G, D) \quad (9)$$

269 **The structure of the designed deep learning model is illustrated in Fig. 5.** The trained generator  $G^*$  is  
 270 used to expand the data to  $\{\mathbf{P}_{\tau}^c\}_{\tau=1}^{\tau=N_T}$ , in which  $N_T - N_1$  image patches are generated as fake image patches.  
 271  $2N_T$  samples are selected to form a training dataset  $\{\mathbf{P}_{\tau}^c, \mathbf{P}_{\tau}^{uc}\}_{\tau=1}^{\tau=N_T}$ , where  $\mathbf{P}_{\tau}^c, \mathbf{P}_{\tau}^{uc} \in \mathbf{R}^{2\lambda \times 2\lambda}$ . The  
 272 corresponding label  $\{L_{\tau}^c, L_{\tau}^{uc}\}_{\tau=1}^{\tau=N_T}$  is defined by the parallel TCCFCM clustering result. All the pseudo-label  
 273 training samples and labels are fed into the CWNN to classify the hard patches, where hard samples  
 274  $\{\mathbf{P}_q^h\}_{q=1}^{N_h}$  are classified into changed and unchanged categories. The final change map can be obtained by  
 275 combining the labels of hard pixels and simple pixels.

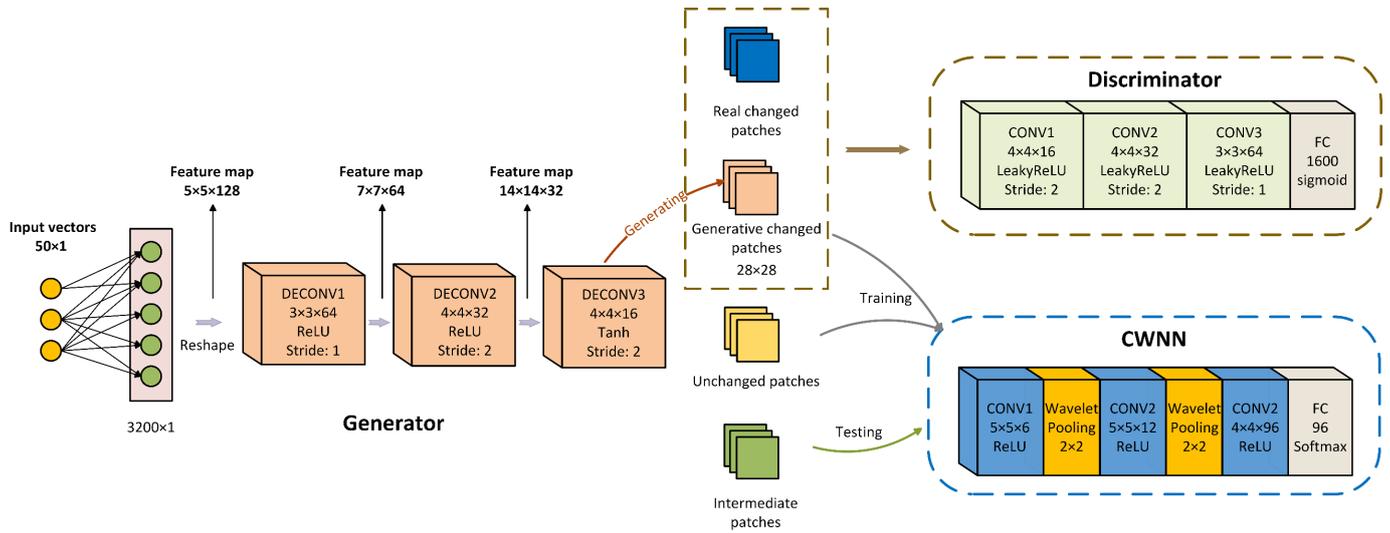
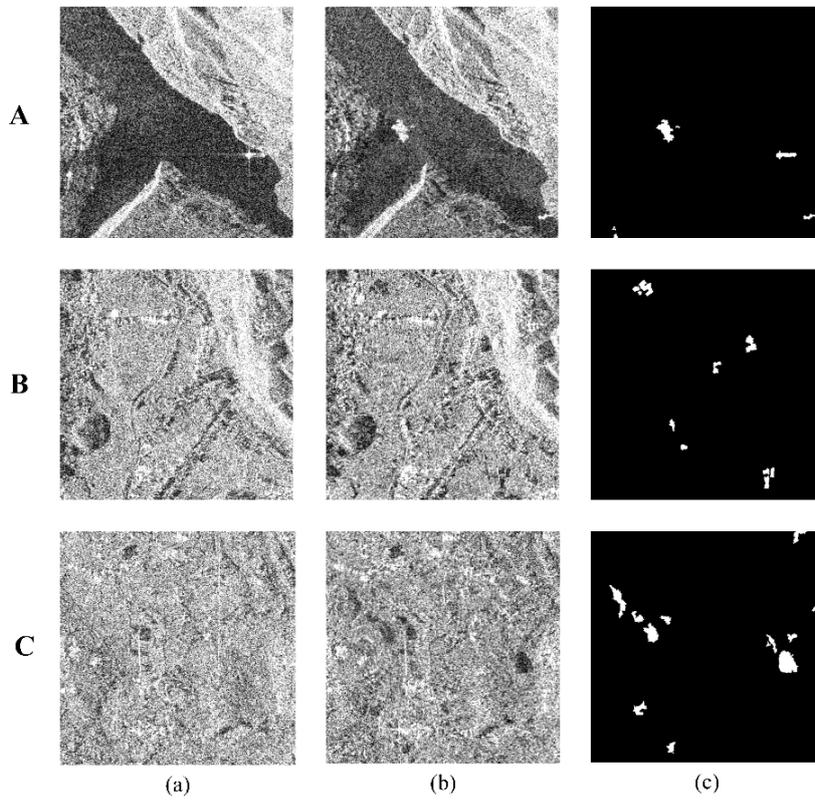


Fig. 5. Network structure of DCGAN and CWNN. Padding is employed in DECONV2, DECONV3, CONV1 and CONV2 in the generator and discriminator, where CONV and DECONV represent the convolutional layer and deconvolutional layer, respectively. FC is the fully connected layer.

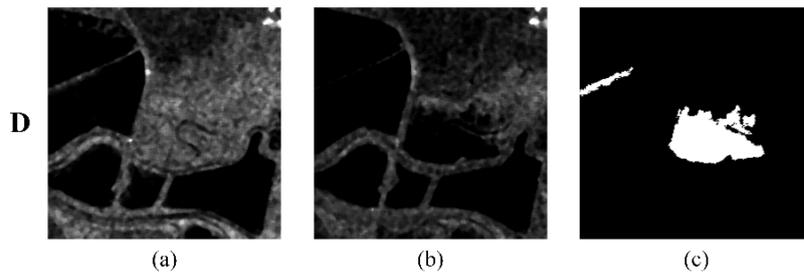
## 4. Numerical experiments

### 4.1 Experimental datasets

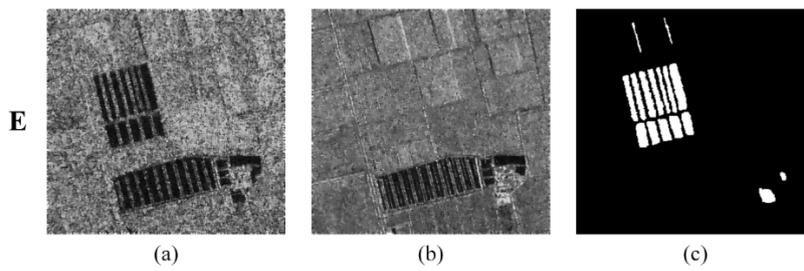
Six real multi-temporal SAR datasets were used to evaluate the performance of the proposed approach. Three of these six datasets were acquired over the province of Guizhou in China by the COSMO-SkyMed SAR sensor on June 2016 and April 2017. As shown in Fig. 6, the first image pair (dataset A) with ground reference map consists of mountains and a river, the second image pair (dataset B) with ground reference map includes hills, plains and buildings, and the third image pair (dataset C) with ground reference map is mostly plains. Image pair of dataset D was acquired over the city of San Francisco, USA by the ERS-2 SAR sensor in August 2003 and May 2004 (Fig. 7). Image pair of dataset E (relating to Farmland C of Yellow River Estuary) was acquired by the Radarsat-2 satellite in June 2008 and June 2009 (Fig. 8), and Image pair of dataset F (relating to Ottawa flood) was acquired by Radarsat-1 satellite in May 1997 and August 1997 (Fig. 9). Speckle noise is shown in datasets A, B and C, which is very challenging for change detection. From the ground reference maps in Fig. 6, it is clear that the proportion of changed pixels is small compared with unchanged pixels. The details are recorded in Table 1. In contrast, datasets D, E and F are relatively balanced with lower speckle noise than the former three datasets, which are used as benchmarks to evaluate the robustness and effectiveness of our approach. In this research,  $N_c$  and  $N_{uc}$  refer to the number of changed and unchanged pixels in the ground reference map, respectively.



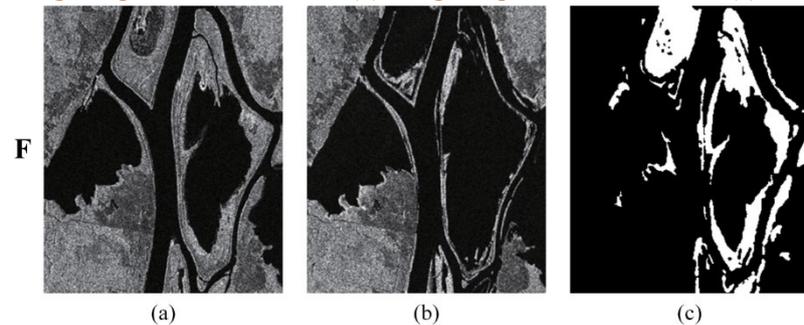
297  
298 Fig. 6. Dataset A, B and C. (a) Image acquired in June 2016, (b) Image acquired in April 2017. (c) Ground reference map



299  
300 Fig. 7. Dataset D. (a) Image acquired in August 2003. (b) Image acquired in May 2004. (c) Ground reference map.



301  
302 Fig. 8. Dataset E. (a) Image acquired in June 2008. (b) Image acquired in June 2009. (c) Ground reference map.



303  
304 Fig. 9. Dataset F. (a) Image acquired in July 1997. (b) Image acquired in August 1997. (c) Ground reference map.

**Table 1.** Summary of the experimental datasets.

Datasets	Size	$N_c$	$N_{uc}$	$N_c : N_{uc}$
A	400×400	1066	158934	1:149
B	400×400	1492	158508	1:106
C	400×400	3467	156533	1:45
D	256×256	4685	60851	1:13
E	293×308	5378	84866	1:16
F	350×290	16049	85428	1:5

## 306 4.2 Evaluation criteria

307 Five quantitative indicators were used to test the performance of the change detection methods: false  
 308 alarm (FA) rate, missed detection (MD) rate, percentage correct classification (PCC), Kappa coefficient  
 309 (KC), and  $F_1$  score. The changed pixels are designated as positive and the unchanged pixels are assigned as  
 310 negative, then, the confusion matrix is constructed. The false positive ( $FP$ ) denotes the number of  
 311 unchanged pixels detected as the changed class, and the false negative ( $FN$ ) refers the number of changed  
 312 pixels misclassified as unchanged. The true negative ( $TN$ ) and true positive ( $TP$ ) indicates the number of  
 313 changed and unchanged pixel correctly detected respectively.

314 1) **FA**: The false alarm rate is given by  $P_{FA} = \frac{FP}{FP + TP} \times 100\%$ .

315 2) **MD**: The missed detection rate is calculated as  $P_{MD} = \frac{FN}{FN + TP} \times 100\%$ .

316 3) **PCC**: Accuracy of pixel-based classification can be expressed as

$$317 \text{PCC} = \frac{TP + TN}{TP + FP + TN + FN} \quad (10)$$

318 4) **KC**: Kappa coefficient is used for consistency checks, defined as

$$319 \text{KC} = \frac{\text{PCC} - \text{PRE}}{1 - \text{PRE}} \quad (11)$$

$$\text{PRE} = \frac{(TP + FN) \times (TP + FP) + (TN + FP) \times (TN + FN)}{(TP + FN + TN + FP)^2}$$

320 5) **F1**:  $F_1$  score is an essential indicator of classification performance, which is defined as

$$321 F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (12)$$

$$\text{precision} = \frac{TP}{TP + FP} \quad \text{recall} = \frac{TP}{TP + FN}$$

### 4.3 Experimental setup and parameter setting

All experiments were implemented on a PC with a 3.3-GHz four-core CPU and 24-GB memory. The GAN training and sample augmentation were implemented with NVIDIA GeForce RTX 2080s GPU with 8-GB memory and PyTorch1.7.0. The CWNN training and testing were implemented using CPU and MATLAB2020a.

The hyper-parameters for each module were set separately. The weighted average filter kernel  $\mathbf{W}^n$  size was set to  $\eta = 3$ . In MSRDI, SLIC was conducted on four scales ( $T=4$ ), and  $L^1 = 100$ ,  $L^2 = 500$ ,  $L^3 = 1000$ ,  $L^4 = 2000$  were used to obtain spatial information at the different scales on dataset A-D.  $L^1 = 4000$ ,  $L^2 = 8000$ ,  $L^3 = 16000$ ,  $L^4 = 32000$  were used for datasets E and F. Two sigmoid mapping parameters  $\mu_1 = -0.2$  and  $\mu_2 = 0.3$  were set for unchanged and changed classes, respectively.  $\mu_2$  was larger than 0.2, which ensured the reliable samples fed into the DCGAN. The Gabor feature extraction dimension was  $\gamma = 6$ , and the constraint parameter of TCCFCM was set as  $\beta = 0.5$ . Finally, the size of the image patch was chosen as  $\lambda = 14$ .

The training of the DCGAN was as follows: 1) the Adam optimiser with a learning rate of 0.0003 was used to optimise the generator, and a learning rate of 0.0006 to optimise the discriminator. 2) Each iteration gave the real image a random value in the range 0.8 to 1 as a score, and the score of the fake image was provided randomly between 0 and 0.2. The number of epochs was 10000, and 640 image patches were selected and divided into 10 batches for training. A total of 4000 pseudo-label samples were used to train the CWNN with a learning rate of 0.0001, and epochs set as 50.

### 4.4 Model complexity analysis

In the MSRDI module, the complexity of the reconstruction method is linear in the number of pixels, with  $O(N)$  complexity similar to SLIC. The proposed TCCFCM introduces cluster prototype constraints without increasing the computational complexity compared with standard FCM. The TCCFCM has  $O(NI\rho c^2)$  complex, where  $N$  represents the number of pixels,  $I$  denotes the number of iterations and  $\rho$  and  $c$  refer to the dimensions of the pixel sample and number of clusters, respectively. The complexity of the deep learning models is listed in Table 2. In addition, 640 image patches of size  $28 \times 28$  pixels were used for CWNN training, and employed for testing the forward propagation time. The same strategy was applied to record the running time of DCGAN.

**Table 2.** Description of the deep learning models illustrating their complexity. K and M represent thousands and millions, respectively, FLOPs refer to floating point operations and MB is Megabytes.

Networks	Parameters (K)	Model size (MB)	FLOPs (M)	Forward time (s)	Training time (s / epoch)
CWNN	20.88	0.163	0.247	1.54	2.56
Generator	278	2.224	16.8	0.19	0.52
Discriminator	30.192	0.242	0.922	0.012	

## 5. Results and discussion

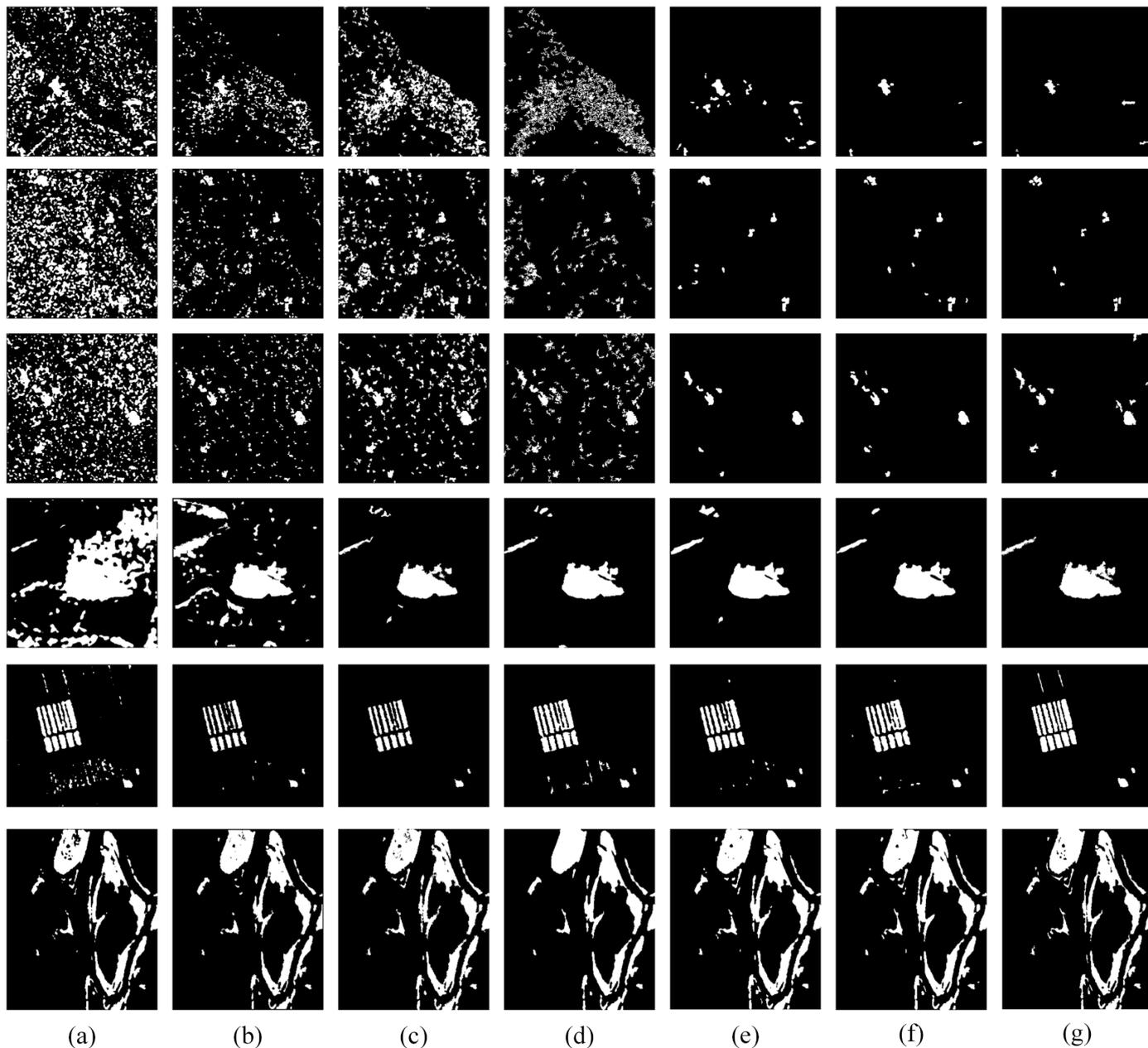
### 5.1 Results and comparison

To demonstrate the effectiveness of the proposed RUSACD, four benchmark methods were compared, including: PCA k-means (PCAK) (Celik, 2009), neighbourhood ratio and extreme learning machine (NRELM) (Gao et al., 2016), Gabor feature extraction and FCM with PCANet (GFPCANet) (Gao et al., 2016), and FCM with CWNN (FCWNN) (Gao et al., 2019). We also applied TCCFCM to cluster the MSRDI into two categories (changed and unchanged) as a simple, fast and effective variant change detection method MTCCFCM. The experimental results are shown in Fig. 8, while the quantitative accuracy metrics are given in Table 3. In addition, other methods used for comparison on datasets D, E and F (Table 4, 5, 6) include saliency-guided detection with  $k$ -means clustering (SGK) (Zheng et al., 2017), stacked autoencoder and FCM with CNN (SAEFCNN) (Gong et al., 2017), saliency-guided deep neural network (SGDNN) (Geng et al., 2019), adaptive generalised likelihood ratio test (AGLRT) (Zhuang et al., 2020), PCANet with Saliency detection (SDPCANet) (Li et al., 2019), spatial FCM and CNN (SFCNN) (Li et al., 2019), FCM and Deep Belief Network (FDBN) (Gong et al., 2016) and hypergraph-based change detection framework (HCDF) (Wang et al., 2020), the numerical results of these methods are acquired from published articles.

From Table 3, RUSACD could maintain a change detection accuracy of more than 99%, which is the best among all the methods. The  $F_1$  and KC of RUSACD are greater than 70%, which is significantly superior to the other methods. Both PCAK and NRELM produced a low accuracy due to the high sensitivity of  $k$ -means clustering and extreme learning machine (ELM) to noise. Datasets A, B and C contained massive speckle noise, which led to a very large number of false alarms with limited accuracy. The change detection accuracy for GFPCANet and FCWNN is low since the lack of guiding information in DI and clustering labels. For dataset D (with low noise), most approaches showed excellent detection capabilities, except for PCAK. For the proposed RUSACD, the PCC reached the highest accuracy of 99.24%, and both  $F_1$  and KC exceeded 94%. The MTCCFCM achieved competitive results compared with methods based on deep

376 learning. It can be concluded that RUSACD is very effective for small area change detection from multi-  
377 temporal SAR images with strong speckle noise.

378 In addition, RUSACD also exhibits outstanding detection performance on benchmark datasets as shown  
379 in Table 5 and Table 6. For dataset E, RUSACD achieves best performance on PCC,  $F_1$  and KC among all  
380 methods. Although the performance of RUSACD is not best on dataset F, it achieves competitive  
381 performance compared with other state-of-the-art methods. The MSRDI is designed for SAR images with  
382 strong speckle noise, and it restrains some tiny discrete changed pixels as noise on dataset F.



384 Fig. 10. Comparison of the final change maps on all datasets. (a) PCAK. (b) NRELM. (c) GFPCANet. (d) FCWNN.  
385 (e) MTCCFCM. (f) RUSACD. (g) Ground reference map.

**Table 3.** Accuracy metrics for the different change detection methods. Best results are shown in bold.

Criterion	FP	FN	P <sub>FA</sub> (%)	P <sub>MD</sub> (%)	PCC (%)	KC (%)	F <sub>1</sub> (%)	
Dataset A	PCAK	27822	<b>1</b>	96.45	<b>0.1</b>	82.61	5.69	6.86
	NRELM	11312	42	92.00	4.10	92.90	13.74	14.76
	GFPCANet	20935	9	95.37	0.88	86.91	7.71	8.84
	FCWNN	18108	383	96.58	37.37	88.44	5.34	6.49
	MTCCFCM	2118	107	69.76	10.44	98.61	44.68	45.21
	<b>RUSACD</b>	<b>390</b>	241	<b>33.22</b>	23.51	<b>99.61</b>	<b>71.11</b>	<b>71.31</b>
Dataset B	PCAK	43095	<b>47</b>	96.76	<b>3.15</b>	73.04	4.56	6.28
	NRELM	8066	208	86.27	13.94	94.83	22.44	23.69
	GFPCANet	14553	102	91.28	6.84	90.84	14.49	15.94
	FCWNN	7822	602	73.19	17.36	94.74	38.47	40.48
	MTCCFCM	677	330	36.81	22.12	99.37	69.45	69.77
	<b>RUSACD</b>	<b>596</b>	315	<b>33.62</b>	21.11	<b>99.43</b>	<b>71.81</b>	<b>72.10</b>
Dataset C	PCAK	31048	<b>241</b>	90.59	<b>6.95</b>	80.44	13.70	17.10
	NRELM	6527	729	70.45	21.03	95.47	41.15	43.01
	GFPCANet	11583	331	78.69	9.55	92.55	32.11	34.49
	FCWNN	7194	417	87.00	27.95	95.24	20.77	22.03
	MTCCFCM	<b>159</b>	1459	<b>7.37</b>	42.08	98.99	70.79	71.28
	<b>RUSACD</b>	196	1406	8.68	40.55	<b>99.00</b>	<b>71.53</b>	<b>72.01</b>

**Table 4.** Accuracy metrics for the different change detection methods on dataset D. Best results are shown in bold.

Criterion	FP	FN	P <sub>FA</sub> (%)	P <sub>MD</sub> (%)	PCC (%)	KC (%)	F <sub>1</sub> (%)
AGLRT	<b>231</b>	552	<b>5.27</b>	11.78	98.80	91.10	91.80
SGK	561	489	11.79	10.44	98.40	88.02	88.88
SAEFCNN	231	604	5.36	12.89	98.73	90.04	90.72
SGDNN	321	509	7.14	10.86	98.73	90.28	90.96
PCAK	13784	182	75.38	3.88	78.69	31.40	39.20
NRELM	4466	<b>51</b>	49.08	<b>1.09</b>	93.11	63.81	67.23
GFPCANet	321	337	6.88	7.19	99.00	92.42	92.97
FCWNN	545	150	10.73	3.20	98.94	92.31	92.88
MTCCFCM	621	148	12.04	3.16	98.83	91.55	92.19
<b>RUSACD</b>	279	217	5.88	4.63	<b>99.24</b>	<b>94.33</b>	<b>94.74</b>

**Table 5.** Accuracy metrics for the different change detection methods on dataset E. Best results are shown in bold.

Criterion	FP	FN	P <sub>FA</sub> (%)	P <sub>MD</sub> (%)	PCC (%)	KC (%)	F <sub>1</sub> (%)
FDBN	659	<b>641</b>	12.21	<b>11.92</b>	98.56	87.17	87.93
SFCNN	380	903	7.83	16.79	98.58	86.71	87.46
PCAK	1269	652	21.17	12.12	97.87	81.98	83.11
NRELM	76	1958	2.17	36.41	97.75	75.95	77.08
GFPCANet	<b>24</b>	1535	<b>0.62</b>	28.54	98.27	82.25	83.14
FCWNN	589	705	11.19	13.11	98.57	87.08	87.84
MTCCFCM	250	1013	5.42	18.84	98.60	86.62	87.36
<b>RUSACD</b>	360	837	7.35	15.56	<b>98.67</b>	<b>87.65</b>	<b>88.35</b>

**Table 6.** Accuracy metrics for the different change detection methods on dataset F. Best results are shown in bold.

Criterion	FP	FN	P <sub>FA</sub> (%)	P <sub>MD</sub> (%)	PCC (%)	KC (%)	F <sub>1</sub> (%)
AGLRT	882	678	5.74	4.22	98.46	94.13	94.89
SDPCANet	820	944	5.15	5.88	98.26	93.45	94.48
FDBN	<b>569</b>	620	<b>3.56</b>	3.86	98.83	95.59	96.29
SFCNN	<b>569</b>	782	3.59	4.87	98.67	94.97	95.76
HCDF	615	<b>516</b>	3.81	<b>3.22</b>	<b>98.89</b>	<b>95.83</b>	<b>96.49</b>
PCAK	755	1718	5.00	10.70	97.56	90.62	92.06
NRELM	577	1690	3.86	10.53	97.77	91.37	92.68
GFPCANet	1047	821	6.43	5.12	98.16	93.13	94.22
FCWNN	956	933	5.95	5.81	98.14	93.01	94.12
MTCCFCM	1527	539	8.96	3.36	97.96	92.54	93.76
RUSACD	1335	560	7.94	3.49	98.13	93.12	94.24

## 5.2 Performance analysis of MSRDI

Five DI generation methods were compared in the experiment: LR, SLR (smoothed LR by the proposed weighted average filter), NR, CDI (Zheng et al., 2014) and MSRDI. The results for Datasets C and D are shown in Fig. 11. Compared with the other DIs, the images in MSRDI are smooth, the edges are strengthened and the outliers are suppressed since MSRDI exploits the local spatial information in superpixels at multiple scales. MSRDI has a strong ability to suppress noise and retain clear edges, particularly for small area change detection (the first row of Fig. 11).

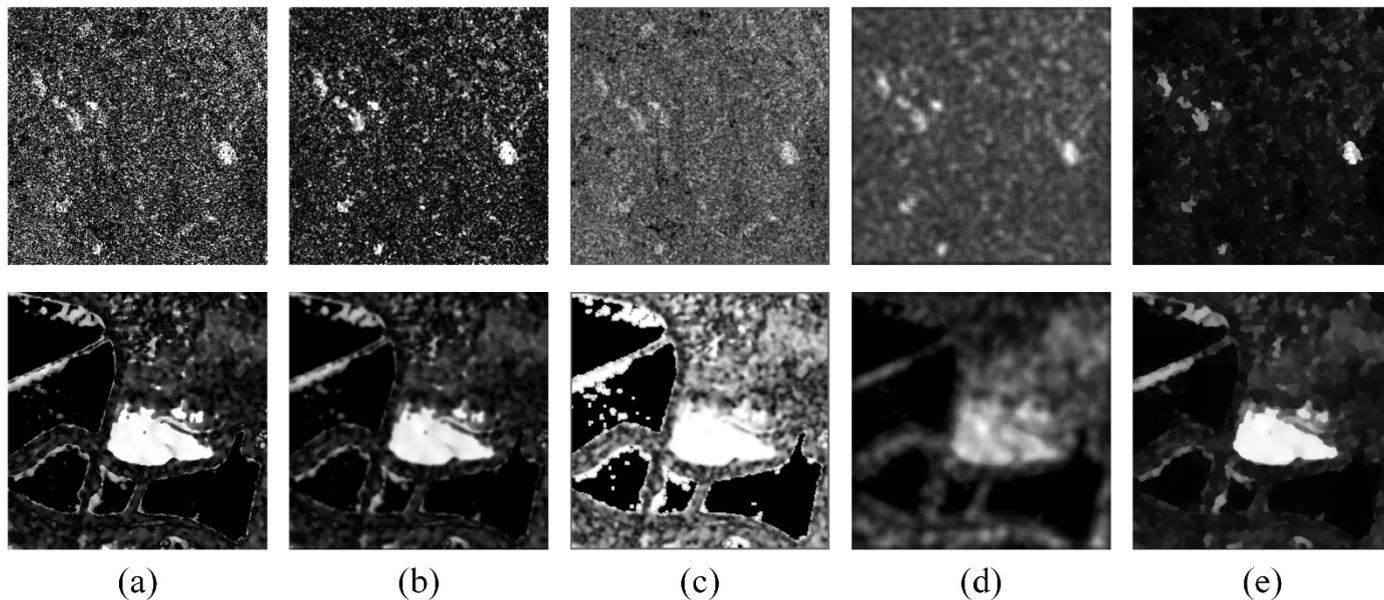


Fig. 11. DIs of the five methods on datasets C and D. (a) LR. (b) SLR. (c) NR. (d) CDI. (e) MSRDI.

The performance of MSRDI was further tested using the Otsu thresholding method to segment all DIs, as shown in Fig. 12 and Table 7. MSRDI is superior to the other methods, with the lowest false alarm rate and the highest accuracy. The NR-based DI and detection results are unsatisfactory for both datasets. For

dataset C with strong speckle noise, the mean and median operators in MSRDI were beneficial for suppressing speckle noise. For Dataset D, MSRDI was able to enhance the edges and strengthen the separability of the DI, while smoothing the image.

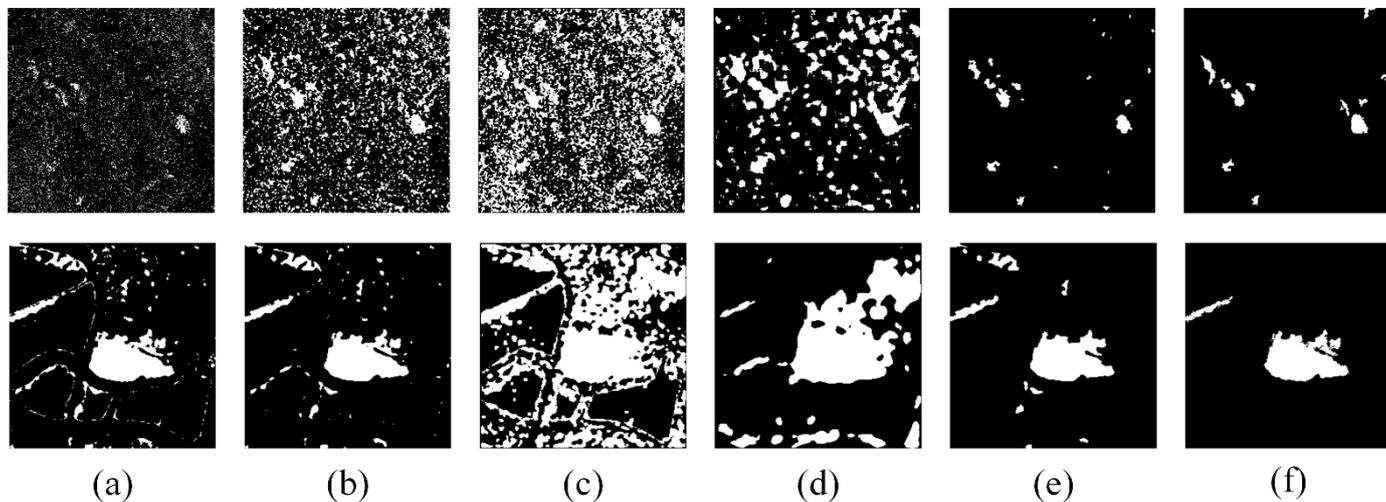


Fig. 12. Change maps of the five DI methods using Otsu thresholding on dataset C and D. (a) LR. (b) SLR. (c) NR. (d) CDI. (e) MSRDI. (f) Ground reference map.

**Table 7.** Change detection accuracy metrics for five DIs acquired by Otsu thresholding. Best results are shown in bold.

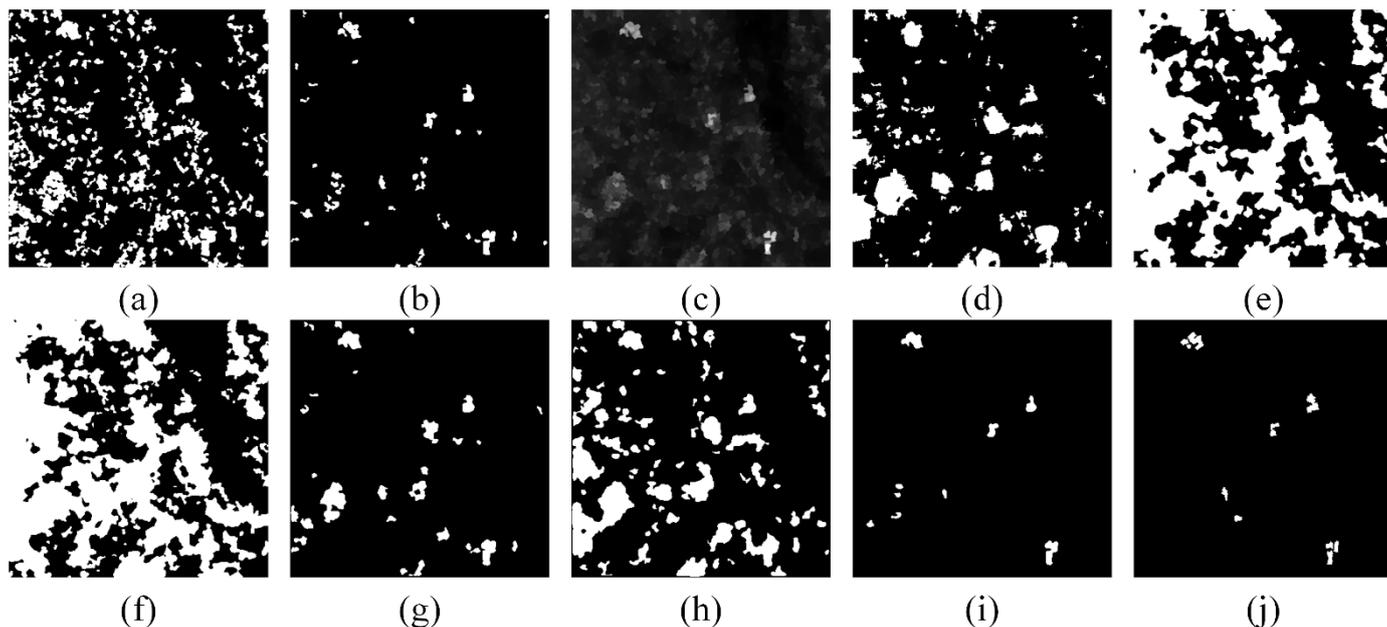
Criterion	Dataset C					Dataset D				
	FP	FN	PCC (%)	KC (%)	F <sub>1</sub> (%)	FP	FN	PCC (%)	KC (%)	F <sub>1</sub> (%)
LR	12083	2000	91.20	14.28	17.24	2917	146	95.33	72.34	74.77
SLR	30911	309	80.49	13.42	16.83	1952	130	96.82	79.71	81.40
NR	54044	236	66.08	6.83	10.64	23295	<b>4</b>	64.45	18.71	28.66
CDI	24977	<b>153</b>	84.29	17.69	20.87	11947	152	81.54	35.67	42.83
<b>MSRDI</b>	<b>785</b>	1110	<b>98.82</b>	<b>70.72</b>	<b>71.33</b>	<b>1237</b>	174	<b>97.85</b>	<b>85.32</b>	<b>86.48</b>

### 5.3 Performance analysis of TCCFCM

Six segmentation methods were used to segment the MSRDI of dataset B to achieve change maps, as shown in Fig. 13 and Table 8. Here, GTCCFCM represents segmentation based on Gabor feature extraction together with TCCFCM. GKM represents the combination of Gabor feature extraction and the k-means clustering algorithm. GFCM refers to Gabor feature extraction with FCM clustering. **In addition, two methods were used as benchmarks, including classic FCM and TCCFCM algorithms with Gabor wavelet features extracted from LR images.**

The change map of GTCCFCM produced reliable detection results, with the highest accuracy of 99.37%, demonstrating robustness for small area change detection. Other methods such as FCM produced a significant number of false alarms. The k-means algorithm shows some benefits by minimising the incorrect transfer of cluster centres, although it still contains several medium-value unchanged pixels within the cluster

426 prototype of changed class, leading to increased false alarms. Fig. 13(b) and the results of LGTCCFCM  
 427 demonstrate the effectiveness of the proposed algorithm, without overly relying on the high-quality MSRDI.



428  
 429 Fig. 13. Change maps of the image segmentation methods on dataset B. (a) LGFCM. (b) LGTCCFCM. (c) MSRDI. (d)  
 430 Otsu threshold. (e) FLICM. (f) RFLICM. (g) GKM. (h) GFCM. (i) GTCCFCM. (j) Ground reference map.

431 **Table 8.** Change detection accuracy metrics acquired on dataset B. Best results are shown in bold.

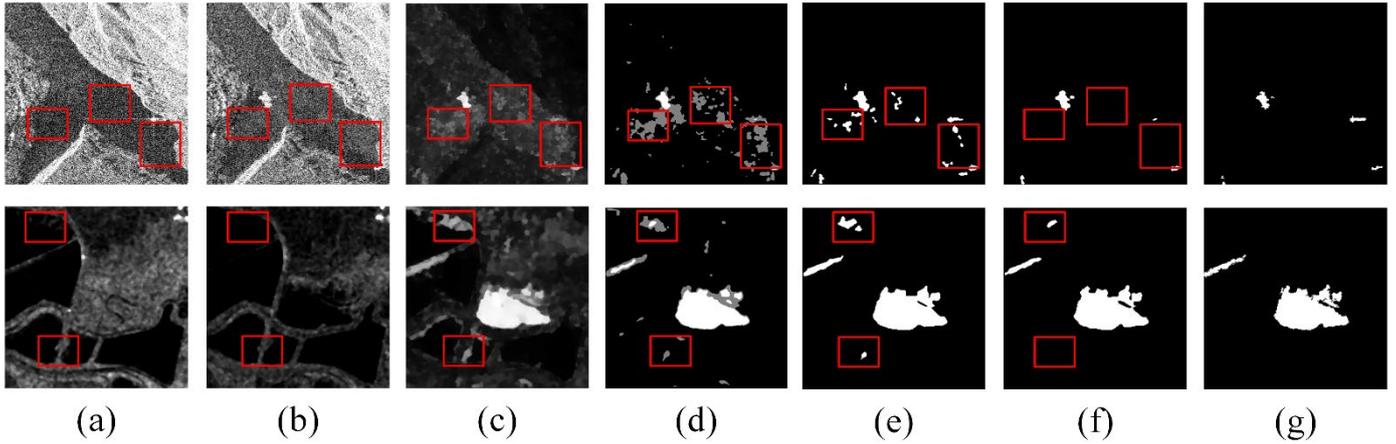
Criterion	FP	FN	P <sub>FA</sub> (%)	P <sub>MD</sub> (%)	PCC (%)	KC (%)	F <sub>1</sub> (%)
LGFCM	26718	44	94.86	2.95	83.27	8.14	9.76
LGTCCFCM	2614	222	67.30	14.88	98.23	46.53	47.25
Otsu	17522	126	92.77	8.45	88.97	11.88	13.41
FLICM	70990	3	97.95	0.2	55.63	2.24	4.03
RFLICM	78150	25	98.16	1.68	51.14	1.82	3.62
GKM	5272	128	79.45	8.58	96.63	32.54	33.56
GFCM	24242	32	94.32	2.14	84.83	9.14	10.74
GTCCFCM	677	330	36.81	22.12	99.37	69.45	69.77

432  
 433 **Table 9.** Simple pixel classification accuracy by the proposed parallel TCCFCM clustering strategy. The PCC<sub>c</sub> and PCC<sub>uc</sub>  
 434 refer to the accuracy of the changed class and the unchanged class respectively.

Dataset	PCC <sub>c</sub> (%)	PCC <sub>uc</sub> (%)
A	89.35	99.98
B	83.71	99.82
C	99.76	99.25
D	97.91	99.97

435 Table 9 shows the accuracy of simple pixel classification using the parallel clustering strategy embedded  
 436 in TCCFCM. The accuracy (PCC) is greater than 99%, showing that the parallel TCCFCM clustering  
 437 strategy is able to provide credible pseudo-label training samples for deep learning models.

## 5.4 Performance analysis of deep learning model



439

440

441

442

443

Fig. 14. Analysis of the difficulties in accurately detecting small area changes in dataset A and D. (a) and (b) are the original SAR images. (c) MSRDI. (d) Label map of parallel TCCFCM clustering; white pixels indicate simple pixels of the changed class, black pixels represent simple pixels of the unchanged class, grey pixels indicate hard pixels. (e) Change map by MTCCFCM. (f) Change map by RUSACD. (g) Ground reference map.

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

Two experiments were conducted on datasets A and D using deep learning (CWNN with DCGAN). In Fig. 14, the red windows indicate that many hard pixels are difficult to distinguish in MSRDI, which are classified accurately by the deep learning models. For Dataset D, there are some slight distortions between the bi-temporal SAR images, and they do not represent real land cover changes. RUSACD can distinguish well between the distortions and real changed pixels with the help of deep learning. In dataset A, the proposed RUSACD significantly increased the KC from 44.68% to 71.11%, and  $F_1$  from 45.21% to 71.31% compared with MTCCFCM. This is further validated by the three other datasets, showing that CWNN with DCGAN can precisely classify hard pixels, thus, increasing change detection accuracy. By contrast, many false alarms are shown when classifying all pixels based on MSRDI and TCCFCM without the deep learning models.

The second experiment demonstrated the sample augmentation for small area change detection, where several data augmentation methods were used for comparative analysis, including simple linear generation (SLG) (Gao et al., 2019), ADASYN (He et al., 2008), and our DCGAN method. As shown in Fig. 15, DCGAN was the most effective augmentation method with the highest accuracy in  $PCC^H$  and  $F_1$  score. The CWNN with sample augmentation can achieve greater accuracy than the benchmark on datasets B, C and D. Although the benchmark experiment results maintain a high accuracy on dataset A, it depends on the distribution of hard pixels, which are mostly unchanged pixels in dataset A. The trained CWNN without sample augmentation failed to differentiate hard pixels, and classified them all as unchanged pixels.

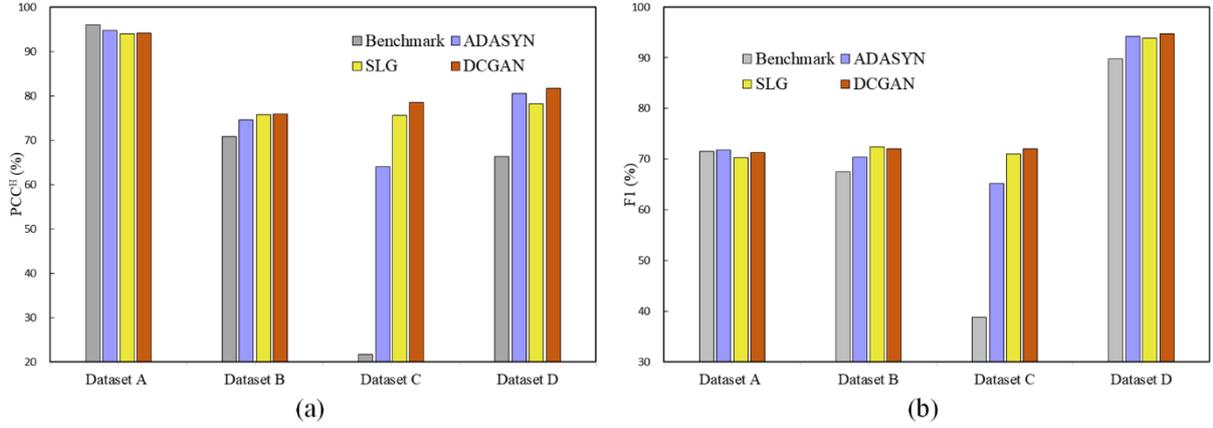


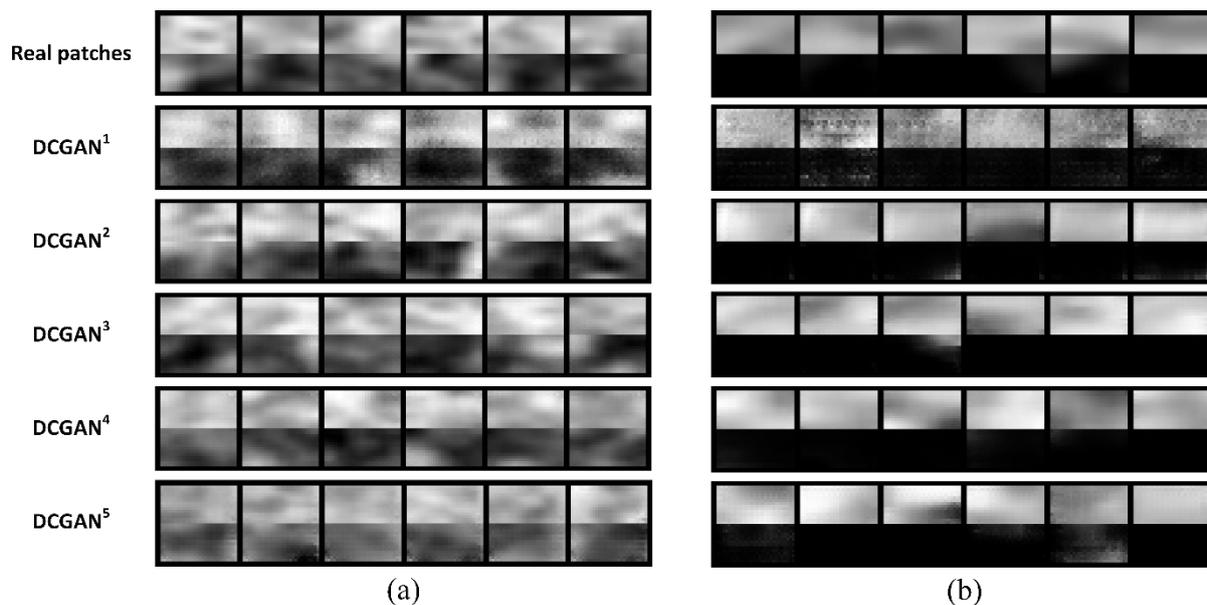
Fig. 15. Comparison of different sample augmentation methods. (a) The classification accuracy of hard pixels is the criterion. (b) The F1 score of the final change map is defined as the criterion.

## 5.5 Ablation study

We designed two ablation experiments to evaluate the robustness of hard pixel classification as a function of different network structures. The first experiment used generators with different numbers of convolution layers and convolution kernels to create the changed image patches. The network architectures of all DCGAN versions are listed in Table 10. Thereafter, the generated changed image patches and real pseudo-label image patches were used jointly to train the CWNN. The network structure of CWNN was fixed in the first experiment, with two datasets (B and D) adopted. The accuracy of hard pixel classification is listed in Table 11. In addition, the image patches generated by DCGAN with different network structures are shown in Fig. 14.  $C_i$  represents the convolutional layer, where  $i$  indexes the layer.

From Table 11, DCGAN<sup>3</sup> shows advantages over the benchmark methods, achieving a PCC<sup>H</sup> of 80.82% on dataset D. The patches generated by DCGAN<sup>1</sup> with one-layer and a few convolution kernels were not beneficial to training a strong CWNN, since the simple network of DCGAN<sup>1</sup> was unable to learn the visual features of the change classes effectively (Fig.16). In contrast, DCGAN<sup>4</sup> and DCGAN<sup>5</sup> employed more layers and convolutional kernels, and resulted in learned features similar to the training data. For dataset B, the CWNN combined with DCGAN<sup>4</sup> achieved the best performance, with a PCC<sup>H</sup> of 71.89%, slightly higher than the 71.73% of DCGAN<sup>3</sup>. This is because dataset B presents a more complex background texture, which requires rich convolution kernels to learn detailed texture information. Taking all factors into account, DCGAN<sup>3</sup> was selected in the proposed RUSACD. The experimental results also demonstrated that a reasonable structure of DCGAN was useful to learn deep features of the changed class, where the generated

483 samples include rich feature representations, thereby enhancing the discrimination capability of the trained  
 484 CWNN.



485  
 486 Fig.16. The changed image patches generated by DCGAN. (a) dataset B. (b) dataset D.

487 **Table 10.** The architecture of different versions of DCGANs. **G** and **D** refer to generator and discriminator respectively

DCGAN <sup>1</sup>		DCGAN <sup>2</sup>		DCGAN <sup>3</sup>		DCGAN <sup>4</sup>		DCGAN <sup>5</sup>	
<b>G</b>	<b>D</b>								
C <sub>1</sub> :8×8×64	C <sub>1</sub> :8×8×32	C <sub>1</sub> :6×6×64	C <sub>1</sub> :6×6×32	C <sub>1</sub> :4×4×64	C <sub>1</sub> :4×4×16	C <sub>1</sub> :4×4×64	C <sub>1</sub> :4×4×8	C <sub>1</sub> :4×4×64	C <sub>1</sub> :4×4×4
		C <sub>2</sub> :4×4×32	C <sub>2</sub> :4×4×64	C <sub>2</sub> :4×4×32	C <sub>2</sub> :4×4×32	C <sub>2</sub> :4×4×32	C <sub>2</sub> :4×4×16	C <sub>2</sub> :3×3×32	C <sub>2</sub> :3×3×8
				C <sub>3</sub> :3×3×16	C <sub>3</sub> :3×3×64	C <sub>3</sub> :4×4×16	C <sub>3</sub> :4×4×32	C <sub>3</sub> :3×3×16	C <sub>3</sub> :3×3×16
						C <sub>4</sub> :3×3×8	C <sub>4</sub> :3×3×64	C <sub>4</sub> :3×3×8	C <sub>4</sub> :3×3×32
								C <sub>5</sub> :3×3×4	C <sub>5</sub> :3×3×64

488  
 489 **Table 11.** The PCC (%) of hard pixel classification using DCGANs with different architectures for data augmentation.

490 The best results are shown in bold.

	DCGAN <sup>1</sup>	DCGAN <sup>2</sup>	DCGAN <sup>3</sup>	DCGAN <sup>4</sup>	DCGAN <sup>5</sup>
Dataset B	70.51	70.64	71.73	<b>71.89</b>	71.84
Dataset D	74.95	74.25	<b>80.82</b>	77.74	74.22

491 The second experiment used CWNNs with different numbers of layers and convolution kernels to  
 492 classify hard pixels, and the results are shown in Table 12, with the fixed DCGAN<sup>3</sup> being used. The CWNN<sup>3</sup>  
 493 acquired the highest accuracy (average PCC<sup>H</sup> of 78.56%). The accuracy of shallow and deep CWNNs were  
 494 limited according to the PCC<sup>H</sup>. CWNNs with small numbers of layers or convolution kernels tend to under-  
 495 fit because they are oversimplified models with insufficient learning capacity. On the contrary, CWNNs with  
 496 a large number of convolution kernels or layers tended to overfit through training, resulting in a decrease in

classification accuracy. Based on the above experimental results, CWNN<sup>3</sup> is demonstrated to be the most suitable network architecture with a powerful discriminative capability.

**Table 12.** The PCC (%) of hard pixel classification using CWNN with different architectures. The best results are shown in bold.

	CWNN <sup>1</sup>	CWNN <sup>2</sup>	CWNN <sup>3</sup>	CWNN <sup>4</sup>	CWNN <sup>5</sup>	CWNN <sup>6</sup>
Architecture	C <sub>1</sub> : 5×5×6 Pooling: 2 FC: 864	C <sub>1</sub> : 5×5×6 Pooling: 2 C <sub>2</sub> : 5×5×12 Pooling: 2 FC :256	C <sub>1</sub> : 5×5×6 Pooling: 2 C <sub>2</sub> : 5×5×12 Pooling: 2 C <sub>3</sub> : 4×4×96 FC: 96	C <sub>1</sub> : 5×5×6 Pooling: 2 C <sub>2</sub> : 3×3×12 Pooling: 2 C <sub>3</sub> : 2×2×48 Pooling: 2 C <sub>4</sub> : 2×2×96 FC: 96	C <sub>1</sub> : 5×5×2 Pooling: 2 C <sub>2</sub> : 5×5×4 Pooling: 2 C <sub>3</sub> : 4×4×32 FC: 32	C <sub>1</sub> : 5×5×18 Pooling: 2 C <sub>2</sub> : 5×5×36 Pooling: 2 C <sub>3</sub> : 4×4×288 FC: 288
Dataset A	86.33	96.63	92.98	94.88	<b>97.96</b>	95.12
Dataset B	49.54	67.39	<b>71.73</b>	54.09	65.64	62.01
Dataset C	68.37	64.50	68.69	65.52	64.51	<b>70.12</b>
Dataset D	71.56	77.90	<b>80.82</b>	70.52	70.16	68.60
Average	68.95	76.61	<b>78.56</b>	71.25	74.57	73.96

## 5.6 Parameter analysis

To test the sensitivity of the results to parameter choice, we investigated the effect of the constraint parameter  $\beta$  in TCCFCM on change detection accuracy. We fixed other parameters, and assigned a set of values to  $\beta$  as shown in Fig. 15. RUSACD was not sensitive to the parameter  $\beta$  for all datasets, and the change detection accuracy was the highest (Fig. 15(b)). The MTCCFCM exhibited stable and satisfactory performance for datasets B, C and D (Fig. 15(a)). The accuracy for dataset A declined when  $\beta$  was less than 0.4. These results indicate that RUSACD is robust and accurate compared with MTCCFCM for small area change detection.

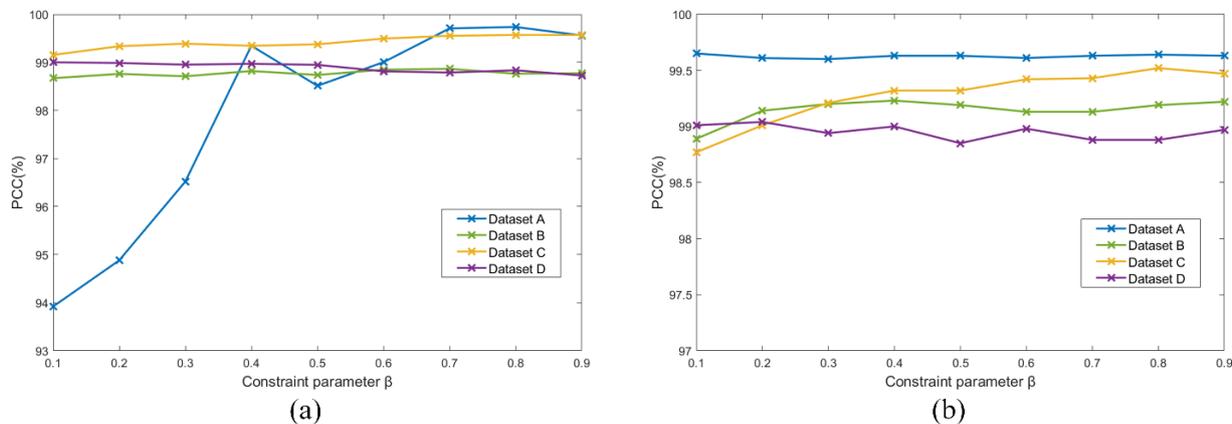


Fig. 17. The relationship between change detection accuracy and parameter  $\beta$ . (a) Results of MTCCFCM. (b) Results of RUSACD.

## 5.7 Discussion

In multi-temporal SAR image change detection, the speckle noise and image distortion produced in the image acquisition process often cause false changes, leading to a decrease in change detection accuracy. For small area change detection, the number of false changed pixels may be similar to the number of real changed pixels, leading to the problems addressed in this paper. In this research, we proposed RUSACD, as a step-wise modular framework, and demonstrated that it was the most accurate method across a series of state-of-the-art benchmarks for small area change detection.

In the DI generation module, the developed MSRDI aimed to increase the gap between unchanged and changed areas, as well as suppressing speckle noise. It was also able to reduce the interference caused by false-changed pixels.

In the clustering module in the DI analysis, existing FCM-based algorithms usually produce incorrect shifts of cluster prototypes when the number of changed pixels is far less than for unchanged pixels. The TCCFCM was proposed to maintain the appropriate direction of the clustering algorithm through the optimisation process by adopting preliminary centres as constraint terms in the objective function. Furthermore, instead of clustering into two categories, TCCFCM used a parallel clustering strategy, dividing the MSRDI into three classes, which was beneficial for discriminating accurately between false changed pixels and real changed pixels.

In the intermediate (hard) pixel classification module, the CWNN was employed for small area change detection. However, as is typical for small area change detection, there were too few training samples of the changed class to train the CWNN. To avoid an imbalance between the changed and unchanged training samples a DCGAN was applied to enrich the training samples of the minority class. Further, compared to other data augmentation methods, DCGAN explored the visual feature space through the learning of image patches, and captured effectively the greatest potential discriminative information for the changed class. The augmented data cover the feature space of the changed class, which supports the training of CWNN.

Rigorous evaluation was undertaken of each module in RUSACD by comparing against several benchmark approaches. The experimental results demonstrate the effectiveness and robustness of each module in RUSACD. The proposed method has the benefit that each module can be extended readily and, indeed, the entire framework can be developed as an end-to-end deep network. In future research, the method

540 should be evaluated for a wider range of datasets and applications such as change detection using optical  
541 Sentinel-2 images.

## 542 **6. Conclusion**

543 In this paper, a modular RUSACD framework composed of MSRDI, TCCFCM, CWNN and DCGAN  
544 was proposed for small area change detection from multi-temporal SAR images. Experiments on four real  
545 SAR datasets demonstrated the effectiveness and robustness of each module of RUSACD for small area  
546 change detection. The MSRDI in the framework enhances the edges of changed areas and minimises speckle  
547 noise and background pixels, increasing the separability between changed and unchanged pixels. The  
548 TCCFCM avoids incorrect transfer of clustering prototypes caused by class imbalanced samples, and  
549 together with a parallel clustering strategy, the pseudo-label training samples are selected automatically in  
550 an unsupervised approach. CWNN obtains discriminative features from the two original SAR images and  
551 the MSRDI, while DCGAN further expands the pseudo-label training samples of the changed class, solving  
552 the problem of insufficient training samples. **In addition, the adversarial learning process of DCGAN can  
553 capture effectively the potential visual features of the changed class, which supports the training of a strong  
554 classifier.** We conclude that RUSACD offers many advantages for small area change detection and produced  
555 consistently the greatest accuracy of classification amongst the tested benchmarks.

## 556 **Acknowledgement**

557 This work was supported by the National Science Foundation of China (61301224) and the Chongqing  
558 Basic and Frontier Research Project (cstc2017jcyjA1378). **The authors are grateful to the anonymous  
559 reviewers for their constructive comments which increased greatly the quality of this manuscript.**

## 560 **Declaration of Competing Interest**

561 The authors declare that they have no conflicts of interest to disclose.

## 562 **Reference**

- 563 Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S., 2012. SLIC superpixels compared to state-of-the-art  
564 superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(11), 2274–2282.  
565 <https://doi.org/10.1109/TPAMI.2012.120>.
- 566 Brunner, D., Lemoine, G., Bruzzone, L., 2010. Earthquake damage assessment of buildings using VHR optical and SAR  
567 imagery. *IEEE Trans. Geosci. Remote Sens.* 48(5), 2403–2420. <https://doi.org/10.1109/tgrs.2009.2038274>.

568 Cao, S., Du, M., Zhao, W., Hu, Y., Mo, Y., Chen, S., Cai, Y., Peng, P., Zhang, C., 2020. Multi-level monitoring of three-  
569 dimensional building changes for megacities: Trajectory, morphology, and landscape. *ISPRS J. Photogramm. Remote*  
570 *Sens.* 167, 54–70. <https://doi.org/10.1016/j.isprsjprs.2020.06.020>.

571 Celik, T., 2009. Unsupervised change detection in satellite images using principal component analysis and k-means  
572 clustering. *IEEE Geosci. Remote Sens. Lett.* 6(4), 772-776. <https://doi.org/10.1109/LGRS.2009.2025059>.

573 Cozzolino, D., Verdoliva, L., Scarpa, G., Poggi, G., 2020. Nonlocal CNN SAR image despeckling. *Remote Sens.* 17(6),  
574 1097-1101. <https://doi.org/10.3390/rs12061006>.

575 Deledalle, C-A., Denis, L., Tupin, F., Reigber, A., Jager, M., 2015. NL-SAR: A unified nonlocal framework for resolution-  
576 preserving (Pol) (In) SAR denoising. *IEEE Trans. Geosci. Remote Sens.* 53(4), 2021–2038.  
577 <https://doi.org/10.1109/TGRS.2014.2352555>.

578 Duan, Y., Liu, F., Jiao, L., Zhao, P., Zhang, L., 2017. SAR Image segmentation based on convolutional-wavelet neural  
579 network and markov random field. *Pattern Recognit.* 64, 255-267. <https://doi.org/10.1016/j.patcog.2016.11.015>.

580 Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H., 2018. GAN-based synthetic medical  
581 image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing.* 321, 321-331.  
582 <https://doi.org/10.1016/j.neucom.2018.09.013>.

583 Gao, F., Dong, J., Li, B., Xu, Q., 2016. Automatic change detection in synthetic aperture radar images based on PCANet.  
584 *IEEE Geosci. Remote Sens. Lett.* 13(12), 1792–1796. <https://doi.org/10.1109/LGRS.2016.2611001>.

585 Gao, F., Dong, J., Li, B., Xu, Q., Xie, C., 2016. Change detection from synthetic aperture radar images based on  
586 neighborhood-based ratio and extreme learning machine. *J. Appl. Remote Sens.* 10(4), 046019. <https://doi.org/10.1117/1.jrs.10.046019>.

587

588 Gao, F., Wang, X., Gao, Y., Dong, J., Wang, S., 2019. Sea ice change detection in SAR images based on convolutional-  
589 wavelet neural networks. *IEEE Geosci. Remote Sens. Lett.* 16(8), 1240–1244. <https://doi.org/10.1109/LGRS.2019.2895656>.

590

591 Geng, J., Ma, X., Zhou, X., Wang, H., 2019. Saliency-guided deep neural networks for SAR image change detection. *IEEE*  
592 *Trans. Geosci. Remote Sens.* 57(10), 7365-7377. <https://doi.org/10.1109/TGRS.2019.2913095>.

593 Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014.  
594 Generative adversarial nets. *Proc. NIPS.* 2672-2680. <https://arxiv.org/pdf/1406.2661>.

595 Gong, M., Cao, Y., Wu, Q., 2012. A neighborhood-based ratio approach for change detection in SAR images. *IEEE Geosci.*  
596 *Remote Sens. Lett.* 9(2), 307-311. <https://doi.org/10.1109/LGRS.2011.2167211>.

597 Gong, M., Yang, H., Zhang, P., 2017. Feature learning and change feature classification based on deep learning for ternary  
598 change detection in SAR images. *ISPRS J. Photogramm. Remote Sens.* 129, 212–225.  
599 <https://doi.org/10.1016/j.isprsjprs.2017.05.001>.

600 Gong, M., Zhao, J., Liu, J., Miao, Q., Jiao, L., 2016. Change detection in synthetic aperture radar images based on deep  
601 neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 27(1), 125–138. <https://doi.org/10.1109/tnnls.2015.2435783>.

602 Gong, M., Zhou, Z., Ma, J., 2012. Change detection in synthetic aperture radar images based on image fusion and fuzzy  
603 clustering. *IEEE Trans. Image Process.* 21(4), 2141-2151. <https://doi.org/10.1109/TIP.2011.2170702>.

604 He, H., Bai, Y., Garcia, E. A., Li, S., 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. Proc.  
605 Int'l Joint Conf. Neural Netw. 1322-1328. <https://doi.org/10.1109/ijcnn.2008.4633969>.

606 Huang, S., Cai, X., Chen, S., Liu, D., 2011. Change detection method based on fractal model and wavelet transform for  
607 multitemporal SAR images. Int. J. Appl. Earth Obs. Geoinf. 13(6), 863–872. <https://doi.org/10.1016/j.jag.2011.05.018>.

608 Inglada, J., Mercier, G., 2007. A new statistical similarity measure for change detection in multitemporal SAR images and  
609 its extension to multiscale change analysis. IEEE Trans. Geosci. Remote Sens. 45(5), 1432-1445.  
610 <https://doi.org/10.1109/TGRS.2007.893568>.

611 Jia, L., Li, M., Zhang, P., Wu, Y., Zhu, H., 2016. SAR image change detection based on multiple kernel k-means clustering  
612 with local-neighborhood information. IEEE Geosci. Remote Sens. Lett. 13(6), 856–860.  
613 <https://doi.org/10.1109/lgrs.2016.2550666>.

614 Kalaiselvi, S., Gomathi, V., 2020.  $\alpha$ -cut induced fuzzy deep neural network for change detection of SAR images. Appl. Soft  
615 Comput. 95, 106510. <https://doi.org/10.1016/j.asoc.2020.106510>.

616 Kim, Y., Lee, M. J., 2020. Rapid change detection of flood affected area after collapse of the Laos Xe-Plain Xe-Namnoy  
617 dam using Sentinel GRD data. Remote Sens. 12(12), 1978. <https://doi.org/10.3390/rs12121978>.

618 Krinidis, S., Chatzis, V., 2010. A robust fuzzy local information c-means clustering algorithm. IEEE Trans. Image Process.  
619 19(5), 1328-1337. <https://doi.org/10.1109/TIP.2010.2040763>.

620 Lei, Y., Liu, X., Shi, J., Lei, C., Wang, J., 2019. Multiscale superpixel segmentation with deep features for change detection.  
621 IEEE Access. 7, 36600-36616. <https://doi.org/10.1109/ACCESS.2019.2902613>.

622 Li, H., Gong, M., Wang, C., Miao, Q., 2018. Self-paced stacked denoising autoencoders based on differential evolution for  
623 change detection. Appl. Soft Comput. 71, 698–714. <https://doi.org/10.1016/j.asoc.2018.07.021>.

624 Li, H. C., Celik, T., Longbotham, N., Emery, W. J., 2015. Gabor feature based unsupervised change detection of  
625 multitemporal SAR images based on two-level clustering. IEEE Geosci. Remote Sens. Lett. 12(12), 2458-2462.  
626 <https://doi.org/10.1109/LGRS.2015.2484220>.

627 Li, H. C., Yang, G., Yang, W., Du, Q., Emery, W. J., 2020. Deep nonsmooth nonnegative matrix factorization network  
628 factorization network with semi-supervised learning for SAR image change detection. ISPRS J. Photogramm. Remote  
629 Sens. 160, 167–179. <https://doi.org/10.1016/j.isprsjprs.2019.12.002>.

630 Li, L., Wang, C., Zhang, H., Zhang, B., Wu, F. 2019. Urban building change detection in SAR images using combined  
631 differential image and residual U-Net network. Remote Sens. 11(9). 1091. <https://doi.org/10.3390/rs11091091>.

632 Li, M., Li, M., Zhang, P., Wu, Y., Song, W., An, L., 2019. SAR image change detection using PCANet guided by saliency  
633 detection. IEEE Geosci. Remote Sens. Lett. 16(3), 402–406. <https://doi.org/10.1109/LGRS.2018.2876616>.

634 Li, Y., Peng, C., Chen, Y., Jiao, L., Zhou, L., Shang, R., 2019. A deep learning method for change detection in synthetic  
635 aperture radar images. IEEE Trans. Geosci. Remote Sens. 57(8), 5751–5763. <https://doi.org/10.1109/TGRS.2019.2901945>.

636

637 Pantze, A., Santoro, M., Fransson, J. E. S. 2014. Change detection of boreal forest using bi-temporal ALOS PALSAR  
638 backscatter data. Remote Sens. Environ. 155, 120–128. <https://doi.org/10.1016/j.rse.2013.08.050>.

639 Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative  
640 adversarial networks. arXiv preprint arXiv:1511.06434. <https://arxiv.org/pdf/1511.06434.pdf>.

- 641 Saha, S., Bovolo, F., Bruzzone, L., 2020. Building change detection in VHR SAR images via unsupervised deep tanscoding.  
642 IEEE Trans. Geosci. Remote Sens. <https://doi.org/10.1109/TGRS.2020.3000296>.
- 643 Shorten, C., Khoshgoftaar, T. M., 2019. A survey on image data augmentation for deep learning. J. Big Data, 6(1).  
644 <https://doi.org/10.1186/s40537-019-0197-0>.
- 645 Sun, Z., Zhang, Z., Chen, Y., Liu, S., Song, Y., 2020. Frost filtering algorithm of SAR images with adaptive windowing  
646 and adaptive tuning factor. IEEE Geosci. Remote Sens. Lett. 17(6), 1097–1101.  
647 <https://doi.org/10.1109/LGRS.2019.2939208>.
- 648 Tian, D., Gong, M., 2018. A novel edge-weight based fuzzy clustering method for change detection in SAR images. Inf. Sci.  
649 467, 415–430. <https://doi.org/10.1016/j.ins.2018.08.015>.
- 650 Wang, J., Yang, X., Yang, X., Jia, L., Fang, S., 2020. Unsupervised change detection between SAR images based on  
651 hypergraphs. ISPRS J. Photogramm. Remote Sens. 164, 61–72. <https://doi.org/10.1016/j.isprsjprs.2020.04.007>.
- 652 Wang, S., Jiao, L., Yang, S., 2016. SAR Images Change detection based on spatial coding and nonlocal similarity pooling.  
653 IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 9(8), 3452–3466. <https://doi.org/10.1109/jstars.2016.2547638>.
- 654 Zheng, Y., Jiao, L., Liu, H., Zhang, X., Hou, B., Wang, S., 2017. Unsupervised saliency-guided SAR image change detection.  
655 Pattern Recognit, 61, 309–326. <https://doi.org/10.1016/j.patcog.2016.07.040>.
- 656 Zheng, Z., Zhang, X., Hou, B., Liu, G., 2014. Using combined difference image and k-means clustering for SAR image  
657 change detection. IEEE Geosci. Remote Sens. Lett. 11(3), 691–695. <https://doi.org/10.1109/LGRS.2013.2275738>.
- 658 Zhuang, H., Fan, H., Deng, K., Yao, G., 2018. A spatial-temporal adaptive neighborhood-based ratio approach for change  
659 detection in SAR images. Remote Sens. 10(8), 1295. <https://doi.org/10.3390/rs10081295>.
- 660 Zhuang, H., Tan, Z., Deng, K., Yao, G., 2020. Adaptive generalized likelihood ratio test for change detection in SAR image.  
661 IEEE Geosci. Remote Sens. Lett. 17(3), 416–420. <https://doi.org/10.1109/LGRS.2019.2922198>.