ISPRS Journal of Photogrammetry and Remote Sensing 166 (2020) 183–200

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images

Chenxiao Zhang[a], Peng Yue[b,c,d,*], Deodato Tapete[e], Liangcun Jiang[b], Boyi Shangguan[b], Li Huang[b], Guangchao Liu[b]

[a] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, Wuhan University, 129 Luoyu Road, Wuhan, Hubei 430079, China
[b] School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Road, Wuhan, Hubei 430079, China
[c] Hubei Province Engineering Center for Intelligent Geoprocessing (HPECIG), Wuhan University, 129 Luoyu Road, Wuhan, Hubei 430079, China
[d] Collaborative Innovation Center of Geospatial Technology, 129 Luoyu Road, Wuhan, Hubei 430079, China
[e] Italian Space Agency (ASI), Via del Politecnico snc, 00133, Rome, Italy

ARTICLE INFO

Keywords:
Change detection
Deep supervision network
Image fusion
High resolution remote sensing image
Image difference discrimination

ABSTRACT

Change detection in high resolution remote sensing images is crucial to the understanding of land surface changes. As traditional change detection methods are not suitable for the task considering the challenges brought by the fine image details and complex texture features conveyed in high resolution images, a number of deep learning-based change detection methods have been proposed to improve the change detection performance. Although the state-of-the-art deep feature based methods outperform all the other deep learning-based change detection methods, networks in the existing deep feature based methods are mostly modified from architectures that are originally proposed for single-image semantic segmentation. Transferring these networks for change detection task still poses some key issues. In this paper, we propose a deeply supervised image fusion network (IFN) for change detection in high resolution bi-temporal remote sensing images. Specifically, highly representative deep features of bi-temporal images are firstly extracted through a fully convolutional two-stream architecture. Then, the extracted deep features are fed into a deeply supervised difference discrimination network (DDN) for change detection. To improve boundary completeness and internal compactness of objects in the output change maps, multi-level deep features of raw images are fused with image difference features by means of attention modules for change map reconstruction. DDN is further enhanced by directly introducing change map losses to intermediate layers in the network, and the whole network is trained in an end-to-end manner. IFN is applied to a publicly available dataset, as well as a challenging dataset consisting of multi-source bi-temporal images from Google Earth covering different cities in China. Both visual interpretation and quantitative assessment confirm that IFN outperforms four benchmark methods derived from the literature, by returning changed areas with complete boundaries and high internal compactness compared to the state-of-the-art methods.

1. Introduction

Change detection aims to identify differences in multi-temporal images of the same area. Monitoring differences in bi-temporal remotely sensed images is crucial to the understanding of land surface changes. Using remote sensing images for change detection has been widely applied for various applications, such as disaster damage assessment, land cover mapping, and urban expansion investigation (Jin et al., 2013; Mundia and Aniya, 2005; Wang and Xu, 2010). With the development of high resolution optical sensors (e.g., WorldView-3, GeoEys-1, QuickBird, and Gaofen-2), the increasing availability of high resolution remote sensing images has widened the range of potential applications of change detection in high resolution bi-temporal images.

Studies on change detection have been carried out for decades in the remote sensing community. Traditional change detection methods can be broadly categorized into three classes: 1) image arithmetical-based, 2) image transformation-based, and 3) post classification methods. Image arithmetical-based methods directly compare pixel values from multi-temporal images to produce image difference maps upon which thresholds are applied to classify pixels into changed class or

* Corresponding author at: School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Road, Wuhan, Hubei 430079, China.
E-mail address: pyue@whu.edu.cn (P. Yue).

https://doi.org/10.1016/j.isprsjprs.2020.06.003
Received 3 December 2019; Received in revised form 20 May 2020; Accepted 3 June 2020

unchanged class. Arithmetical operations, such as image differencing (Singh, 1986), image regression (Jackson, 1983), and image rationing (Todd, 1977), are typically used for image comparison. The key of image arithmetical-based methods is the decision of where to place the threshold boundaries to separate changed pixels from unchanged pixels (Singh, 1989). Recently, some other machine learning-based methods such as random forest regression, support vector machine, and kernel regression, have been proposed for remote sensing image change detection (Zerrouki et al., 2018; Luppino et al., 2019; Padron-Hidalgo et al., 2019). For example, Luppino et al. (2019) firstly utilize affinity matrices to create pseudo training data from co-located patches. Then, four different machine learning algorithms are tested over the pseudo training data to transform the domain of one image to the other domain to realize heterogeneous change detection. Image transformation-based methods transform image spectral combinations into a specific feature space to discriminate changed pixels. Principal Component Analysis (PCA) is one of the most widely used algorithms in the image transformation-based methods for dimensionality reduction (Kuncheva and Faithfull, 2014). Saha et al. (2019) propose to use cycle-consistent Generative Adversarial Network (CycleGAN) to *trans*-code images from different sensors into the same domain in an unsupervised way, and further realize change detection through deep feature change vector analysis. Since pixel-based analysis neglect spatial contextual information, object-based change detection methods are proposed. The main idea of object-based methods is to extract features from segmented image-objects and identify changes in the state of objects (Chen et al., 2012). In the work of Celik (2009), PCA is applied on image difference maps to extract representative features from objects. In post classification methods, bi-temporal images are independently classified and labeled. Changed areas are extracted through a direct comparison of the classification results (Wu et al., 2017). Arithmetical-based and transformation-based methods are the typical unsupervised methods. To improve the performance of change detection, many scholars treat change detection as a problem of explicitly finding land-cover transitions in a supervised way. The most popular supervised method is post classification. Post classification methods bypass the difficulties in change detection from raw images at different times. However, these methods are highly sensitive to the classification results (Deng et al., 2008). Arithmetical-based and transformation-based methods are highly dependent on the empirically designed algorithms for discriminative feature extraction, which fail to achieve satisfying results on high resolution images. Moreover, errors generated by the derivation of difference images in pixel-based methods, the uncertainty of segmented objects in object-based methods, and the misclassification errors of bi-temporal images in post classification methods are inevitably propagated through different stages of change detection and in the end affect negatively the results. The fine image details and complex texture features conveyed in high resolution images introduce new challenges for the change detection task. This has led to the rising of deep learning-based change detection methods. In pixel-based and object-based methods, deep features of pixels or objects are firstly extracted through deep learning techniques such as deep belief network, stacked denoising autoencoder and convolutional neural network (CNN) (El Amin et al., 2017; Lei et al., 2019b; Zhang et al., 2016). Afterward, difference images or change vectors are generated by deep feature comparison. The final change maps are produced by clustering or threshold-based classification methods. Benefiting from the strong high level feature extraction ability of deep neural networks, these methods achieve superior performances than traditional methods. However, the error propagation problem still exists in pixel-based and object-based methods. , Deep feature based methods implemented with fully convolutional network (FCN) are proposed (Alcantarilla et al., 2018; Bromley et al., 1994; Caye Daudt et al., 2018; Daudt et al., 2018; Peng et al., 2019; Shelhamer et al., 2017). Deep feature based methods transform bi-temporal images into high-level spaces and take deep features as analysis unit. These methods integrate feature extraction and difference discrimination operation within the networks to directly produce the final change maps in an end-to-end manner. It should be noted that most of these networks are modified from networks that are proposed for single-image semantic segmentation. Transferring these networks for change detection in bi-temporal images often comes across with some crucial problems including the lack of informative deep features of individual raw images in early-fusion methods, the low representative raw image features in late-fusion methods and the heterogeneous feature fusion problems.

In this paper, we present a deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. Firstly, highly representative deep features of bi-temporal images are extracted in parallel through a fully convolutional two-stream architecture. Then, the extracted features are sequentially fed into the difference discrimination network for change detection. Attention modules are exploited in the difference discrimination network for effective fusion of raw image deep features and image difference features to help change map reconstruction. Moreover, to further improve the network performance, deep supervision is proposed by directly introducing change detection losses in intermediate layers in the difference discrimination network.

The rest of the paper is organized as follows: Section 2 reviews the current deep learning-based change detection methods. Problem statements and the proposed solutions are also presented in this section. Section 3 presents the proposed methodology. Experiments and discussions are given in Section 4. Section 5 concludes the paper.

2. Related work and problem statement

After a broad review in the literature, in Section 2.1, we classify the deep learning-based change detection methods into three categories based on the analysis unit: 1) pixel-based methods, 2) object-based methods, and 3) deep feature based methods. Afterward, problems of existing deep learning-based methods are stated. In order to frame our work within the state-of-the-art, we focus our research on deep feature based methods. Therefore, in Section 2.2, problems of existing deep feature based methods are specifically discussed, and a brief synopsis of the proposed solution is presented.

2.1. Deep learning-based change detection methods

Pixel-based methods follow a similar procedure with traditional image transformation-based methods. Images are firstly transformed into deep feature spaces by deep neural networks and the deep features are pixel-wise compared to distinguish changed pixels. For example, Zhang et al. (2016) utilized a deep belief network to transform bi-temporal images into a high level feature space. Then, the bi-temporal change features are mapped into a 2-D polar domain to characterize the change information. Saha et al. (2019) used a CNN that was pre-trained for semantic segmentation to extract deep features from raw images. Deep features were compared to generate change maps through change vector analysis. In Hou et al. (2017) and Peng and Guan (2019), pre-trained VGG-16 architectures are applied over multi-temporal images for feature extraction. In summary, in pixel-based methods, firstly, deep learning is applied to transform raw images into high level feature spaces. Then, difference maps are generated by means of deep feature selection and pixel-wise comparison. Finally, threshold-based methods, clustering-based methods, and change vector analysis are applied on difference maps to discriminate changed pixels. Object-based methods take segmented objects as analysis units for change detection. Lei et al. (2019b) proposed to apply stacked denoising autoencoders on segmented superpixels to realize superpixel-based change detection. Lv et al. (2018) used a stacked contractive autoencoder to extract temporal change features from superpixels, then simple linear clustering was used to produce change maps. In El Amin et al. (2017), deep features extracted by a pre-trained CNN from multi-scale regions were

concatenated for deep feature expansion in order to overcome the limited features of the single superpixel. The object-based change detection methods followed the "image segmentation - object feature extraction - feature difference analysis - clustering or threshold classification" procedure. In summary, for both pixel-based and object-based change detection methods, deep learning techniques, such as deep belief network, stacked denoising autoencoders, CNNs, are utilized to extract pixel-level or object-level deep features from which difference maps or difference feature vectors are derived. In this way, original images are transformed into a high-level feature space the key information of which is retained, while noises are eliminated. Among the above-mentioned deep learning techniques, CNNs are the most widely used deep feature extractor (Lv et al., 2018). Although methods applying deep learnings for representative feature extraction outperform image arithmetical-based and transformation-based methods, error propagation effects still exist. In addition, generating change maps from image difference maps still encounters the drawbacks of, for example, the empirical threshold determination, that exists in traditional change detection methods.

Most of the pixel and object based methods do not use supervision from labeled data. To overcome the drawbacks of pixel-based and object-based methods, and furthermore, to learn informative guidance from the available labeled images, an end-to-end deep learning architecture, fully convolutional neural network (FCN) has been introduced for deep feature based change detection in a supervised manner. FCN was firstly proposed for image segmentation task by Long et al. (2015). By replacing the fully connected layers with upsampling layers, the downsampled feature can be restored to the original size of input images to realize end-to-end classification. The methods can be classified into two classes according to how they manage bi-temporal images: early-fusion methods and late-fusion methods. In early-fusion methods, images taken at different times are stacked as one input image. For example, in Alcantarilla et al. (2018), two bi-temporal street view images with three channels were firstly concatenated as one image with six channels. Then, the six-channel image was fed into an FCN to realize street view image change detection. Similarly, Peng et al. (2019) combined bi-temporal remote sensing images as one input, which was then fed into a modified UNet++ architecture for remote sensing image change detection. Rather than combining bi-temporal images as one input in early-fusion methods, late-fusion methods take each of the bi-temporal images as an input (Bromley et al., 1994). Firstly, features of bi-temporal images are separately extracted by two independent pipelines in the network. Then, the extracted features are further combined and compared in the following network layers to generate change maps. Based on this conception, Siamese networks have been proposed for change detection. Siamese networks consist of two subnetworks with the same layer settings and parameter values, each subnetwork receives one image as input. Caye Daudt et al. (2018) proposed to use Siamese networks for change detection and compared them with early fusion methods. The comparison results proved the efficiency of the Siamese network. Based on the work by Caye Daudt et al. (2018), Guo et al. (2018) improved the performance by modifying the fully convolutional Siamese network with the addition of contrastive loss in the network.

2.2. Problem statement and proposed solutions

Deep feature based methods integrate raw image feature extraction and image difference discrimination within the FCNs in an end-to-end manner. Parameters in the networks are automatically updated by back-propagations. After a set of training epochs, parameters in the networks are fine-tuned to have the ability of change detection. Although the state-of-the-art deep feature based methods have achieved superior performances than any other methods, there remain some limitations in existing deep feature based architectures:

1) To migrate semantic segmentation networks for change detection task, early-fusion methods simply concatenate bi-temporal images as one image to meet the requirement of the single input. These segmentation networks share a key similarity: using skip connections to combine deep features with low-level features (Zhou et al., 2018). Since bi-temporal images are fused as one input before being fed into the networks, early layers in early fusion networks fail to provide informative deep features of individual raw images to help image reconstruction, which consequently results in change maps with broken object boundaries and poor object internal compactness.
2) To extract deep features of bi-temporal images, late-fusion methods apply networks receiving two inputs to process each of the bi-temporal images separately. Feature extraction and difference discrimination are chained through a stack of convolutional and pooling layers within an integrated network in which lower layers are responsible for feature extraction and deeper layers are responsible for difference discrimination. As the back-propagation is performed from the last difference discrimination layer to the first feature extraction layer in the integrated framework, the presence of vanishing gradients would prevent gradients to flow backward and the lower layers from learning useful features (Glorot and Bengio, 2010; Mao et al., 2018; Lei et al., 2019a; Liu et al., 2020). Consequently, the poorly-trained feature extraction layers (i.e., lower layers) would produce bi-temporal image features in low representativeness and thus after the difference discrimination process change maps would be affected by poor image qualities.
3) Concatenating deep raw image features and image difference features poses a fusion problem. Let us denote features extracted from bi-temporal images $T1$ and $T2$ as f_{T1} and f_{T2}, respectively. Image difference features are represented as d_{T1-T2}. The two independently extracted feature sets f_{T1} and f_{T2} consist of high-level features from images $T1$ and $T2$, respectively. They are effective for raw image reconstruction while they lack image difference information. d_{T1-T2} is computed through a set of convolutional and pooling operations based on f_{T1} and f_{T2}, it contains informative features that represent image differences between $T1$ and $T2$. $d_{T1-T2}$is responsible for difference discrimination while it lacks individual raw image features. The problem of how to effectively fuse features in different domains (i.e., f_{T1}, f_{T2} and d_{T1-T2}) for change detection needs to be fixed.

The proposed image fusion network addresses the above-mentioned problems in effective manners. Feature extraction is conducted by an independently trained fully convolutional two-stream architecture to acquire highly representative deep bi-temporal image features. To include raw bi-temporal image features for better change map reconstruction, deep bi-temporal image features and image difference features are layer-wisely concatenated in the difference discrimination network. The concatenated features are fused by means of attention modules to effectively overcome the heterogeneity problem. Additionally, to further improve the difference distinguishing ability of the network, deep supervision is proposed to effectively train intermediate layers in the difference discrimination network.

3. Methodology

The proposed network architecture is presented in Section 3.1. In Section 3.2 and 3.3, we present the key network components, i.e. the attention modules and deep supervisions, respectively. Section 3.4 provides details of model training, including data augmentation process, training process, and loss function.

3.1. Network architecture

The proposed image fusion network (IFN) takes a pair of bi-temporal images (i.e., the pre-change image T1 and the post-change image

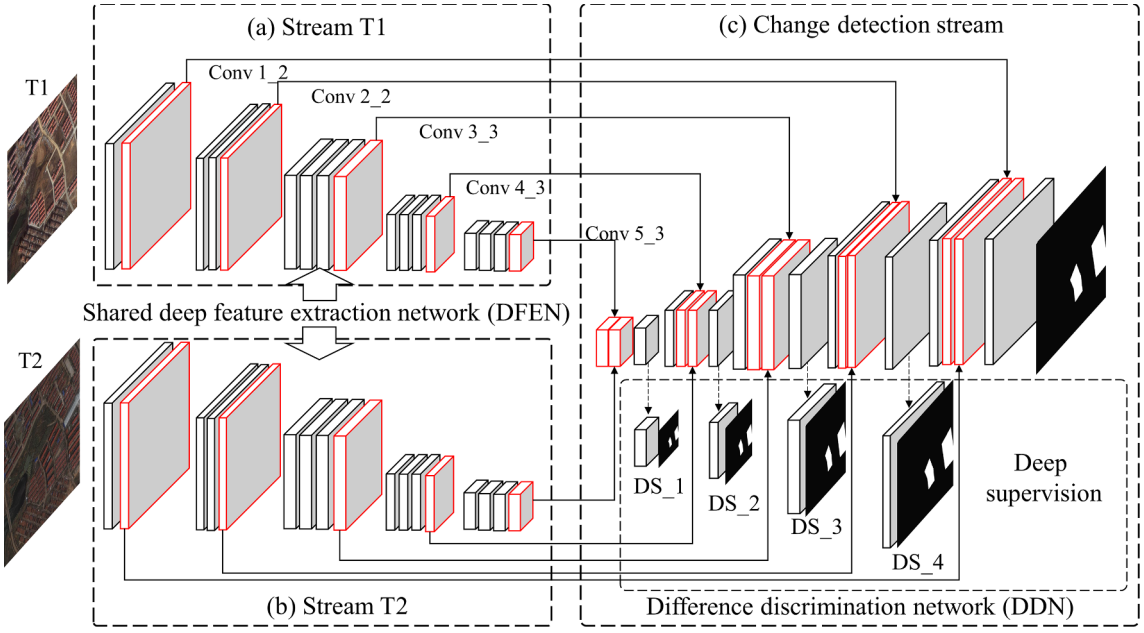

Fig. 1. The proposed image fusion network (IFN) architecture.

T2) as separate inputs into two parallel streams (i.e. Stream T1 and T2, respectively; Fig. 1a-b) This allows the original features of each individual bi-temporal image to be preserved as much as possible. A deep feature extraction network (DFEN) with shared structure and parameters is then applied to both streams for raw image feature extraction. Afterward, the two streams converge to a change detection stream (Fig. 1c), which exploits a difference discrimination network (DDN) to produce change maps. DS_1, DS_2, DS_3, and DS_4 are four deep supervision modules embedded in DDN.

The three streams are structured as follows:

(a) Stream T1 - T1 image is the input and image feature extraction is performed by means of DFEN. For network construction, we take layers before pool5 of VGG16 (Simonyan and Zisserman, 2014) as the backbone of DFEN.
(b) Stream T2 - T2 image is the input. For effective difference discrimination, bi-temporal images should be transformed into the same feature space that is comparable. Therefore, structure and parameters of DFEN in Stream T1 is shared with Stream T2. The extracted T2 image features are combined with T1 image features in the same scales to provide both high-level and low-level raw image features for the change detection stream.
(c) Change detection stream - Once deep features of the two bi-temporal images are extracted through stream T1 and T2, change detection is performed by DDN in the change detection stream.

After progressive abstraction by means of a stacked convolutional and pooling layers, the deepest layers in Stream T1 and T2 acquire large receptive field and compact global information. Hence Conv5_3 acts as the initial input of the DDN to produce a preliminary global change map with compact size. Early layers in DFEN (i.e., Conv4_3, Conv3_3, Conv2_2, and Conv1_2) which contain lower level local structure information of bi-temporal images are skip-connected to DDN layers with the same scales in order to complement individual bi-temporal image features. It is worth mentioning that the existing late-fusion methods (Caye Daudt et al., 2018; Guo et al., 2018) also adopt the two-stream architecture to handle bi-temporal images. However, as pointed out in Section 2.2, the presence of a vanishing gradient problem in an integrated structure makes the network training slow and ineffective (Lee et al., 2015). Based on the observation that a discriminative classifier trained on highly discriminative features gains better performance than a discriminative classifier trained on less discriminative features, in this paper we use a pre-trained DFEN for raw image feature extraction to enhance feature representativeness of each individual bi-temporal images. It should be noted that pre-trained CNNs have also been widely utilized in pixel-based and object-based methods (Saha et al., 2019; Hou et al., 2017; Peng and Guan, 2019). However, these methods employ traditional algorithms such as threshold-based, cluster-based and change vector analysis over extracted features for unsupervised change detection. Therefore, in this case, we intentionally explore the availability of using a pre-trained network in an adaptively learning framework. To the best of our knowledge, this is the first attempt of using the pre-trained network for change detection in an end-to-end manner within the scope of supervised change detection.

The structure of DDN is shown in Fig. 2. DDN starts with the deepest raw image features from stream T1 and T2. Three convolutional layers are sequentially applied over the combined deep image features (i.e., T1_Conv5_3 and T2_Conv5_3) to generate preliminary global image difference feature maps with compact sizes. Then a spatial attention module is applied to extract spatial attention maps (i.e., SAM_1) for feature map refinement across the spatial dimension. To bring back the resolution to that of the raw images, the refined image difference feature maps (i.e., IDF_1) are upsampled to the enlarged feature maps Up_IDF_1. Features in different layers contain different levels of information abstracted from raw images. Deep layer features contain more global information yet less local information, while early layer features contain richer local object details but poor global information. Because skip connections bridge high level features with low level features for better segmentation results in image segmentation networks (Ronneberger et al., 2015; Zhou et al., 2018), we incorporate lower level features (i.e., T1_Conv4_3 and T2_Conv4_3) of individual raw images with the upsampled image difference features Up_IDF_1 in order to recover fine-grained details and better masks of changed objects. However, T1_Conv4_3 and T2_Conv4_3 represent deep features (f_{T1}, f_{T2}) of bi-temporal images, while Up_IDF_1 are image difference features that are denoted as d_{T1-T2}. A direct combination of the heterogeneous features inevitably increases the training difficulty. To fuse raw image deep features with image difference features in an efficient way, we use a channel attention module to emphasize target-relevant channels while suppressing target-irrelevant channels. Afterward, a

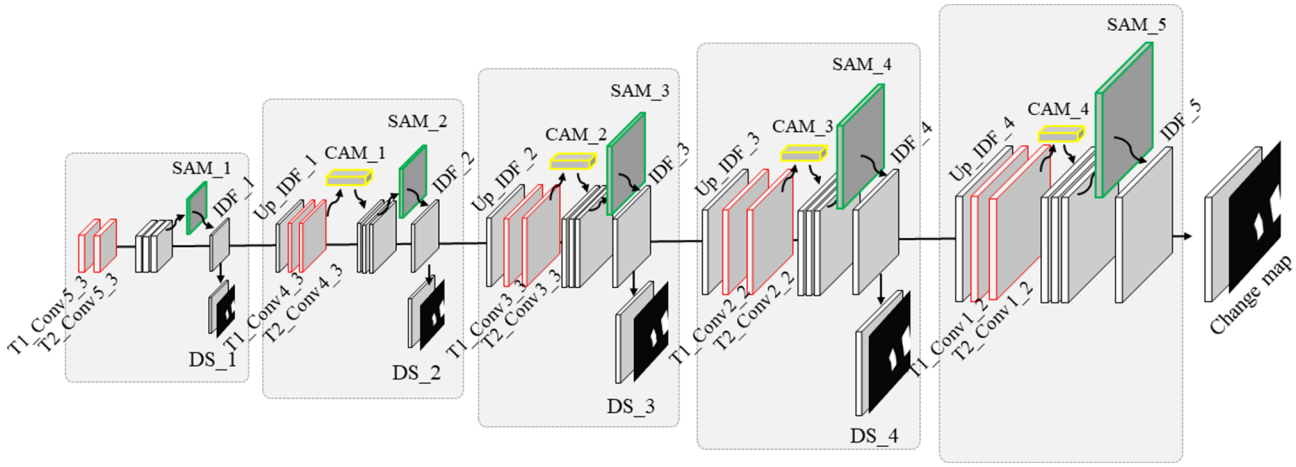

Fig. 2. Structure of the difference discrimination network (DDN).

spatial attention module is applied to enlarge distances between changed pixels and unchanged pixels. The motivation and description of attention modules are provided in Section 3.1.1.

In this way, fused with raw image deep features, image difference features are hierarchically extracted to produce final change maps with fine image details. Additionally, the four deep supervision modules (i.e., DS_1, DS_2, DS_3, DS_4) are introduced after each refined image difference features. The deep supervision is introduced in the intermediate layers in the difference discrimination network to guarantee the network to be well trained and enhance network performance. Details on the deep supervision are presented in Section 3.1.2.

3.1.1. Attention module for feature fusion

CNNs are effective to transform raw images from RGB color spaces $f_i^d (d = 1, 2, 3)$ into high-level feature spaces $f_d (d = 1, \cdots, D)$, where D denotes dimension of features and d is the d-th feature. But not all high-level features are helpful for image difference discrimination (Saha et al., 2019). Such irrelevant features contribute to increase the training difficulty of the network (Woo et al., 2018). Additionally, combination of features in different domains (i.e. raw image features (f_{T1}, f_{T2}) with image difference features (d_{T1-T2}), brings a heterogeneity problem. Therefore, we introduce attention modules to effectively fuse features in different domains.

Channel attention module: Firstly, the combined features are channel-wise refined through channel attention maps. The importance of each of the channels are encoded in channel attention maps the weights of which are automatically recalibrated in the network. By multiplying features $f_d (d = 1, \cdots, D)$ with the corresponding attention map weights $w_d (d = 1, \cdots, D)$, channels that are relevant to change detection are emphasized while the irrelevant channels are suppressed. In this way, channel attention module focuses on 'which channel' to learn from the combined heterogeneous features. For example, assuming that the object of the same area was a car in the pre-change image T1 and was replaced by a house in the post-change image T2, the boundary and context information of car and house are stored in f_{T1} and f_{T2}, respectively. Ideally, in the resulting change map, the algorithm should return a changed area with a complete boundary of the house. By utilizing channel attention maps, f_{T2} that contains house information is emphasized to produce a better change map, together with the image difference features. At the same time, f_{T1} that contains car information is suppressed, since it is helpless for the generation of the change map. Channel attention module is shown in Fig. 3.

Computation of the Channel attention map (M_c^F) is detailed as follows:

$$M_c^F = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \quad (1)$$

where F denotes input feature maps (see Fig. 3). Firstly, spatial information of input feature maps is squeezed through an average pooling (*AvgPool*) operation and a max pooling (*MaxPool*) operation along spatial axis. Assuming that there are C feature maps with the size of $H \times W$, after the pooling operations, feature maps will be squeezed into two vectors with the size of $C \times 1 \times 1$. Then, each vector is forwarded to a shared multi-layer perception (*MLP*). Outputs through the shared *MLP* is merged using element-wise summation. Finally, a sigmoid function denoted as σ is attached to assign the attention weights of each channel.

Spatial attention module: Then, the channel-wise refined features are further refined by means of the spatial attention module across the spatial dimension. Similarly, the importance of each pixel location in feature maps is encoded in spatial attention maps. By iteratively receiving feedbacks from ground truth maps during the network training process, the spatial attention module is trained to gain the ability of adaptively recalibrating the weights of each pixel location and finally outputs spatial attention maps wherein locations of changed and unchanged pixels are assigned with higher and lower importance, respectively. Specifically, a spatial attention map $w_p (p = 1, \cdots, N)$ is firstly computed by the spatial attention module, where N denotes the total number of pixels of each feature map, and p is the p-th pixel location in the map. Then for each high-level feature map f in the feature map collection $f_d (d = 1, \cdots, D)$, f is element-wisely multiplied with w_p to realize the spatial-wise refinement. The refined feature maps can be denoted as $f_d \otimes w_p$. After the spatial-wise refinement, changed pixels are emphasized by being multiplied with higher weights while unchanged pixels are suppressed by being multiplied with lower weights. In this way, the network can rapidly approach the changed regions. In the above mentioned case (i.e. object changed from a car in pre-change image to a house in post-change image), the spatial attention module computes a spatial attention map wherein pixels located within the house boundary (i.e., changed pixels) have higher weights than pixels falling outside (i.e., unchanged pixels). Thus, the combined features are improved across the spatial dimension. The structure of the spatial attention module is shown in Fig. 4.

The Spatial attention map (M_s^F) is computed as follows:

$$M_s^F = \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \quad (2)$$

where $f^{7\times7}$ denotes a convolution operation with the filter size of 7×7 and [;] denotes the concatenate operation (see Fig. 4). Firstly, input feature maps F are average pooled and max pooled by *AvgPool* and *MaxPool* layers along the channel axis, respectively. Then, *AvgPool* and *MaxPool* features are concatenated and convolved by a convolutional layer $f^{7\times7}$. σ is used in the end to produce the final M_s^F.

In summary, the channel attention module focuses on "which

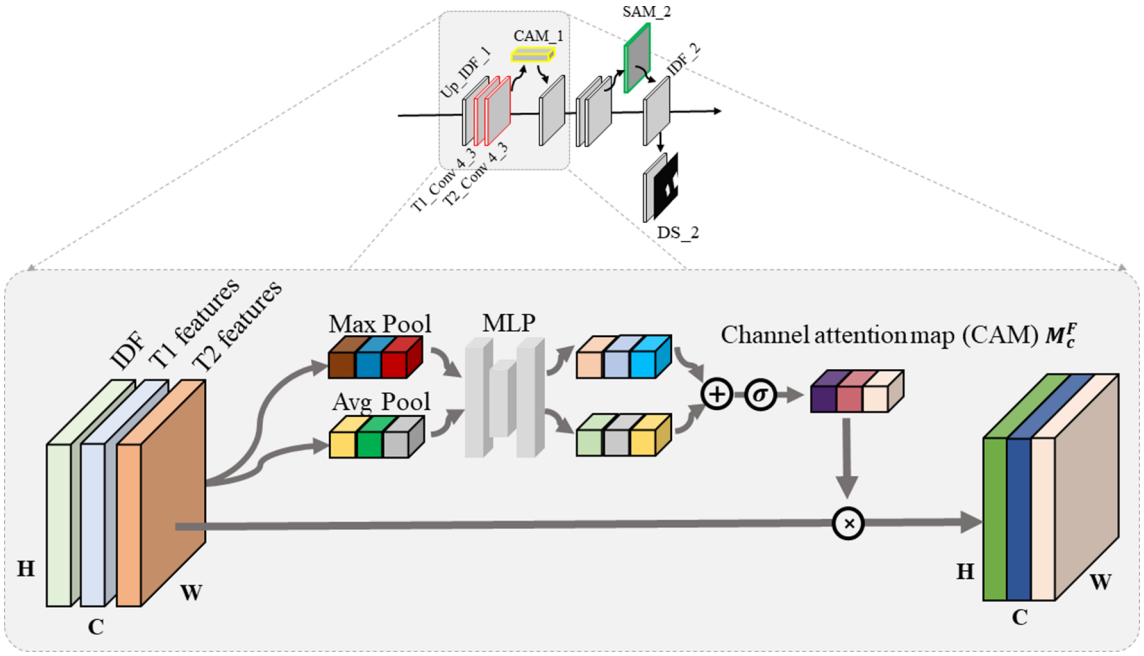

Fig. 3. Channel attention module for channel-wise refinement of combined heterogeneous features.

channel" to learn from the combined heterogeneous features. The spatial attention module focuses on "which area" to learn from the combined feature maps. On the one hand, the channel attention map selects the important features and discard the unnecessary ones from both raw image feature set and image difference feature set, respectively. As a result, the redundancy in both sets can be mitigated. On the other hand, the channel attention map explicitly determines the importance of each selected feature and outputs an informative feature combination from the blending heterogeneous features. This attention-based feature fusion can automatically explore the importance of individual raw image features and image difference features to effectively solve the heterogeneous problem across the channel dimension and further recalibrate the importance of each pixel location across the spatial dimension to rapidly approach interested areas.

After the computation of M_c^F and M_s^F, input features F are refined as follows:

$$F_c^r = F \otimes M_c^F \tag{3}$$

$$F_s^r = F \otimes M_s^F \tag{4}$$

where $\otimes$ denotes element-wise multiplication; F_c^r and F_s^r denote refined feature maps by channel and spatial modules, respectively. In the proposed architecture, we perform spatial attention on the first combined deep features (i.e., T1_ Conv5_3 and T2_ Conv5_3) as follows:

$$F_s^r = Conv([\text{T1\_Conv5\_3}; \text{T2\_Conv5\_3}]) \otimes M_s^F \tag{5}$$

where *Conv* represents the convolution operation. After the spatial

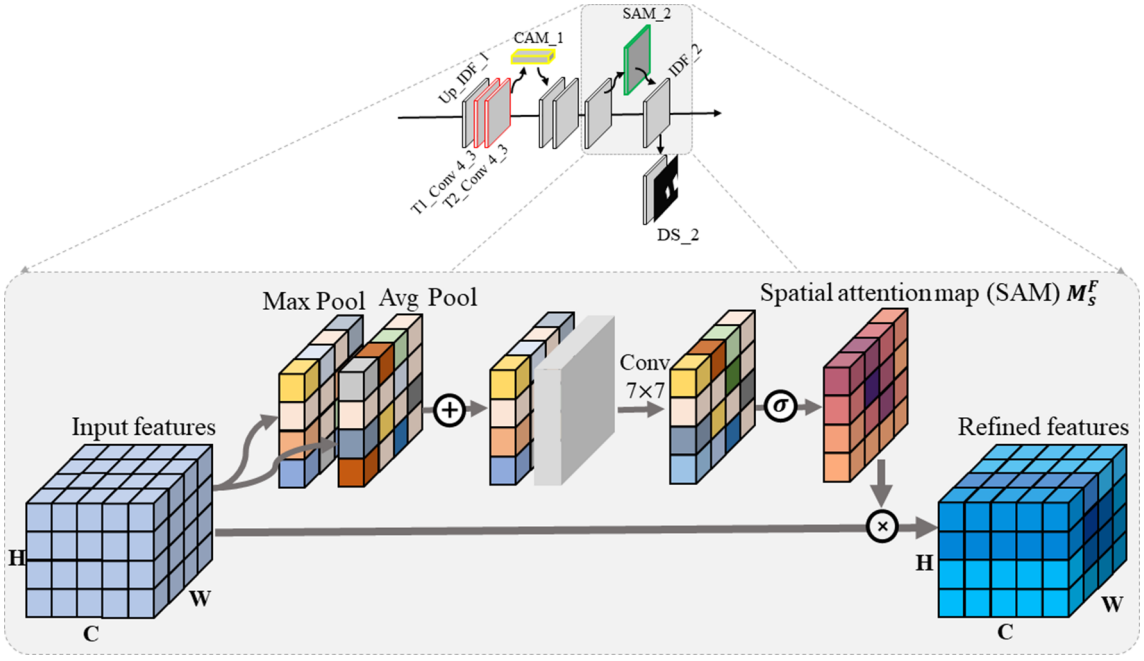

Fig. 4. Spatial attention module for spatial-wise refinement of combined heterogeneous features.

attention module, distances of feature values on changed and unchanged areas are enlarged to facilitate difference discrimination. For the following combined features in difference discrimination network, features describing raw images and t features describing image differences are fused. To emphasize relevant feature maps and suppress irrelevant feature maps, we firstly perform a channel attention module on the combined feature sets. Then, the refined feature maps are fed into a set of convolutional layers. Finally, the spatial attention module is further performed. The computation is as follows:

$$F_c^r = Conv((\text{[T1\_Conv4\_3; T2\_Conv4\_3; } Up\_IDF\_1\text{]}) \otimes M_c^F) \otimes M_s^F \tag{6}$$

where M_c^F represents the channel attention map of the combined deep features (i.e., T1_ Conv4_3, T2_ Conv4_3 and Up_IDF_1). Then the channel-wise refined features will go through a set of convolutional layers $Conv$. A spatial attention map M_s^F is then computed and element-wisely multiplied with the convolved channel-wise refined features to get the final spatial-wise refined deep features F_c^r.

3.1.2. Deep supervision for network performance enhancement

The key feature of artificial neural network (ANN) is using the back-propagation algorithm for weights update to improve the network. Given an ANN and an error function, back-propagation calculates the gradient of the error function with respect to each neuron weight, with the gradient of the last layer weights being calculated first and the gradient of the first layer weights being calculated last. Considering a network with the loss function E that defines the total error between the desired output and the calculated output, w_{kj} denotes the weight between neuron j in the last output layer and neuron k in the previous layer. Then, the output o_j for neuron j is defined as:

$$o_j = \varphi\left(\sum_{k=1}^{K} w_{kj} o_k\right) \tag{7}$$

where φ represents the activation function; o_k denotes outputs from neurons in the previous layer; K denotes the number of neurons in the previous layer. Errors of w_{ij} can be derived by calculating the partial derivative using the chain rule:

$$\partial E/(\partial w_{ij}) = (\partial E/\partial o_j) \times (\partial o_j/\partial w_{ij}) = (\partial E/\partial o_j) \times (\partial o_j/\partial \varphi) \times o_i \tag{8}$$

Eq. (8) represents the calculation of the neuron errors in the last layer. For neurons in the intermediate layers, the factor $\partial E/\partial o_j$ is defined as the weighted sum of the neurons in the next layer:

$$\partial E/\partial o_j = \sum_{l=1}^{N} ((\partial E \times \partial o_l) \times (\partial o_l/\partial \varphi) \times w_{jl}) \tag{9}$$

where N represents the number of neurons in the next layer; o_l represents the output of neuron l in the next layer; w_{jl} denotes the weight between neuron j and neuron l. Therefore, the error with respect to w_{ij} can be calculated in a recursive expression as follows:

$$\partial E/\partial w_{ij} = (\partial o_j/\partial \varphi) \times o_i \times \sum_{l=1}^{N} ((\partial E/\partial o_l) \times (\partial o_l/\partial \varphi) \times w_{jl}) \tag{10}$$

Then, according to the learning rate α, the gradient descent strategy updates the weight w_{ij} as follows:

$$w_{ij} = w_{ij} - \alpha \times (\partial E/\partial w_{ij}) \tag{11}$$

As we can see in Eq. (10), if the activation function φ uses the sigmoid algorithm, the derivative $\partial o_j/\partial \varphi$ is smaller than 1. As the error back-propagates from the last layer to intermediate layers, multiplication of multiple $\partial o_j/\partial \varphi$ will produce much smaller values, creating an even tinier gradient for weights in shallow layers, which is known as the vanishing gradients. Even though alternative activation functions such as ReLU (Glorot et al., 2011) and PReLU (He et al., 2015) have been proposed attempting to overcome the vanishing gradient problem, the problem still exists. Because weights w_{jl} in Eq. (10) are generally initialized to be smaller than 1.0, multiplication of w_{jl} also produces much smaller values. In most of the cases, the vanishing gradient problem will still occur. The randomly initialized neurons in the first few layers are the slowest to train (Glorot and Bengio, 2010) and lead to the poorly trained early layers. Consequently, the poorly-trained early layer will have a negative effect on the following layers in the network. Typically, the longer the distance that error back-propagates, the smaller the gradient that the early layer gets. Some scholars have also demonstrated that the vanishing gradient problem occurred in the long chained networks during the back-propagation process would affect the change detection performance (Mao et al., 2018; Lei et al., 2019a; Liu et al., 2020). Therefore, instead of using an integrated network that performs back-propagation from difference discrimination layers to feature extraction layers, in this paper we break the entire change detection task into two stages to shorten the error propagation distance, which has been described in Section 3.1. Furthermore, to alleviate the vanishing gradient problem and enhance the performance of difference discrimination network, we introduce deep supervision to efficiently train intermediate layers in difference discrimination network. Rather than solely relying on the gradually back-propagated gradients from the output layer, intermediate layers are additionally supervised by change maps in various spatial resolutions. By receiving direct feedbacks from change maps, intermediate layers produce features with higher discriminative for change area discrimination. Finally, the difference discrimination network is expected to much more rapidly approach changed regions. Specifically, for each spatial-wise refined image difference feature sets in difference discrimination network, we associate each of them with a downsampled change map in the same scale. Deep supervision is employed as follows:

$$O^i = \sigma(f^{1\times1}(IDF_i)) \tag{12}$$

where O^i denotes the ith side output DS_i; $f^{1\times1}$ denotes a convolutional layer with kernel size of 1×1; σ denotes the Sigmoid activation layer; IDF_i represents the refined image difference features computed through attention modules. During the training process, the loss of each deep supervision is computed independently and is directly back-propagated to intermediate layers. In this way, intermediate layers in the network are effectively trained and weights of intermediate layers can be finely updated, thus alleviating the presence of vanishing gradient. As a result, the following layers will face an easy change detection task, when previous layers have been finely trained for preliminary change detection. By introducing multiple deep supervisions in the network, the performance of difference discrimination network is improved.

3.2. Model training

3.2.1. Data augmentation

To increase the diversity of training data, we use a set of image preprocesses for data augmentation: 1) image rotation: bi-temporal image pairs are rotated 45°, 90°, 135°, 180°, and 270°; 2) image flipping: bi-temporal image pairs are horizontally flipped, 3); image noise adding: we randomly add 200 salt and pepper noises on T1 images; 4) image blurring: a Gaussian blur filter is applied to T1 images to produce blurred T1 images; 5) image smoothing: a smooth filter is applied to T1 images to produce smoothed T1 images.

Note that for augmentation methods 3) image noise adding, 4) image blurring, and 5) image smoothing, only T1 images are augmented, while the corresponding T2 images remain the same. This is because we intentionally enlarge the image quality gaps between T1 and T2 images, in order to produce a challenging dataset for model robustness testing.

3.2.2. Training process

The deep feature extraction network (DFEN) and the difference discrimination network (DDN) in the proposed method are independently trained during the training process. Since there is a lack of

available models that were pre-trained on a wide range of remote sensing data, for DFEN training, layers before pool5 in VGG16 network are pre-trained on the 'ImageNet' dataset (Jia Deng et al., 2009). For DDN, raw bi-temporal images are firstly fed into the pre-trained DFEN to produce deep features. Then, the extracted deep features are fed into the DDN to distinguish changed areas. To effectively train neurons in DDN, we introduce deep supervision by assigning ground truth maps in different sizes to each side branch. Ground truth maps are generated by linear interpolation algorithm. Sizes of derived ground truth maps are defined as follows:

$$(gt^l_{DS_1}, gt^w_{DS_1}) = (gt^L/16, gt^W/16) \tag{13}$$

$$(gt^l_{DS_2}, gt^w_{DS_2}) = (gt^L/8, gt^W/8) \tag{14}$$

$$(gt^l_{DS_3}, gt^w_{DS_3}) = (gt^L/4, gt^W/4) \tag{15}$$

$$(gt^l_{DS_4}, gt^w_{DS_4}) = (gt^L/2, gt^W/2) \tag{16}$$

where gt^L and gt^W denote length and width of original ground truth maps, respectively; $(gt^l_{DS_1}, gt^w_{DS_1})$, $(gt^l_{DS_2}\ gt^w_{DS_2})$, $(gt^l_{DS_3}\ gt^w_{DS_3})$, and $(gt^l_{DS_4}\ gt^w_{DS_4})$ denote length and width of ground truth maps in DS_1, DS_2, DS_3, DS_4, respectively.

3.2.3. Loss function

Binary cross-entropy, usually applied for binary classification problems, is defined as:

$$L_{bce} = -t_i \log(y_i) - (1 - t_i)\log(1 - y_i) \tag{17}$$

where t_i represents the ground truth value of pixel i; $t_i = 1$ if the ground truth pixel belongs to the changed class. Otherwise, $t_i = 0$. y_i represents the predicted probability of pixel i belonging to the changed class. $1 - y_i$ represents the probability of pixel i belonging to the unchanged class. It can be observed in Eq. (17) that a very small value of y_i on changed class leads to a very large L_{bce}. For example, if y_i gets zero, $-\log(y_i)$ will be infinitely great, an infinitely great loss will corrupt the network. In our method, we use sigmoid binary cross-entropy as part of our loss function. Sigmoid binary cross-entropy is defined as:

$$L_{sig\_bce} = -t_i \log(\sigma(y_i)) - (1 - t_i)\log(\sigma(1 - y_i)) \tag{18}$$

where σ denotes the sigmoid function. To weaken the effect of unbalanced categories, we combine sigmoid binary cross-entropy loss with dice coefficient loss. Dice coefficient loss is defined as:

$$L_{dice} = 1 - (2y_i t_i)/(y_i + t_i)) \tag{19}$$

where y_i represents the predicted probability of pixel i belonging to changed class, t_i represents ground truth value of pixel i. Finally, total loss of the proposed network is defined as:

$$L = L_{sig\_bce} + L_{dice} \tag{20}$$

4. Experiments and discussions

4.1. Datasets

Two datasets are utilized in the experiments for comprehensive benchmark comparison. The first dataset is released by Lebedev et al. (2018). Tested on this dataset, a modified Unet++ architecture achieved the best performance (Peng et al., 2019). The second dataset is a challenging dataset that is manually collected from Google Earth. Different from the literature that train and test model with images covering the same area, we train and test the model with the second dataset covering different areas in order to evaluate the model generalization ability. On one hand, images taken over different cities are used as the training dataset, which increases the difficulty of discriminating changed areas and filtering image differences caused by noises; on the other hand, images taken over another city are used for model testing, aiming to challenge the generalization ability of models. Specifically, for model training, bi-temporal images of Beijing, Chengdu, Shenzhen, Chongqing, and Wuhan are clipped into 394 sub-image pairs with sizes of 512×512. After data augmentation, a collection of 3940 bi-temporal image pairs is acquired. We randomly select 90% of the dataset for model training and the rest 10% for model validation. Xian image pair is clipped into 48 image pairs for model testing.

4.2. Parameter setting

All the convolutional kernels in convolutional layers are set to 3×3. After each concatenation layer, number of filters in the following convolutional layer is set to half of the number of channels in combined features. For example, a deep feature extraction network produces T1_Conv1_2 with 512 channels. Concatenating T1_Conv1_2 and T2_Conv1_2 will get a set of feature maps with 1024 channels. Then the number of filters in the following convolutional layer is set to 512. Loss function described in Section 3.4.3 is used in the network. In total, there are totally five losses in the network, with the weight of each loss being set to 1. Initial learning rate is set to 0.0001 and drops by 10% when the loss stops decreasing for 5 epochs. Model training is finished when f1 score on the validation dataset does not improve for 20 epochs.

4.3. Benchmark methods

To evaluate the effectiveness of the proposed method, we perform change detection using the following four benchmark methods and compare their performances on the two datasets:

(1) Unet++_MSOF

Unet++_MSOF (Peng et al., 2019) is proposed based on the architecture of Unet++ (Zhou et al., 2018). Unet++_MSOF is an early fusion model that takes bi-temporal images as one input. By fusing multiple side outputs of Unet++, the method outperforms all the other state-of-the-art change detection methods on the first dataset.

(2) FC-Sima-conc

Fully convolutional Siamese-Concatenation (FC-Siam-conc) is proposed by Caye Daudt et al. (2018) for satellite image change detection. The method firstly applies a Siamese encoding stream to extract deep features from bi-temporal images, and then concatenate the extracted deep features by the decoding stream for change detection. All the parameters in encoding and decoding streams are updated at each training epoch.

(3) FC-Sima-diff

Fully convolutional Siamese-Difference (FC-Siam-diff) (Caye Daudt et al., 2018) has a similar network architecture with FC-Siam-conc. The difference lies in that FC-Siam-diff concatenates the absolute value of deep feature differences in the decoding stream, instead of concatenating both deep features from the encoding stream.

(4) FCN-PP

Fully convolutional network with pyramid pooling (FCN-PP) (Lei et al., 2019a) is proposed for landslide detection in remote sensing images. The method applies a U-shape architecture. Pyramid pooling is utilized in the network to enlarge the receptive filed in order to overcome the limitations of traditional global pooling.

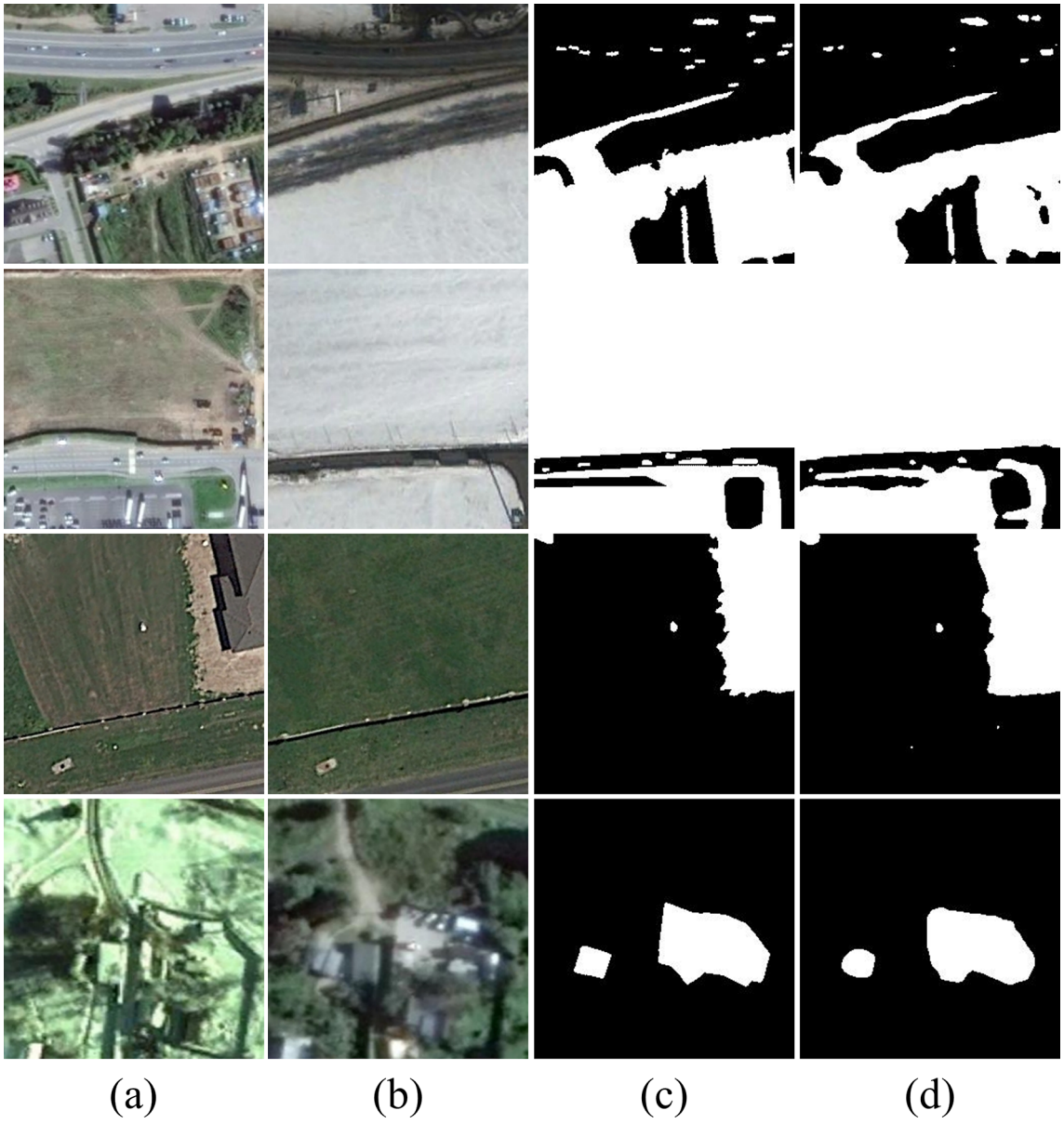

Fig. 5. Large area change detection results on the first dataset. (a) T1 images. (b) T2 images. (c) Ground truth maps. (d) Change maps produced by IFN.

4.4. Experimental results

4.4.1. Benchmark comparison

Both datasets are processed with the proposed change detection method. In the case of the first dataset, the results are compared with the performances of the selected four benchmark methods reported in the literature (Peng et al., 2019). Whereas, the second dataset is also processed with the four benchmark methods, and the respective results are compared. Change detection results of the proposed method and benchmark methods are compared. Comparison is based on visual interpretation and quantitative assessment. For quantitative assessment, the following four indices are: Precision (P), recall (R), F1 score (F1), and overall accuracy (OA) are used as evaluation metrics.

Starting from the change detection results achieved on the first dataset (Fig. 5 and Fig. 6).

When large area change areas are detected, IFN successfully returns the changed areas with complete boundaries and high internal compactness. A demonstrative example is shown in Fig. 12 (d). In the case of small area changes in Fig. 6, almost all the small changed objects are distinguished in the change maps. Although the bi-temporal images T1 and T2 in Fig. 6 contain some noises, IFN successfully filters such noise and the change maps show pixels that have actually changed. For example, in the second bi-temporal image pair in Fig. 6, the trees canopies in image T1 are vigorous, while leaves of the trees have fallen in image T2 due to seasonal changes. IFN recognizes these image noises, and classify these pixels as unchanged areas in the change maps. For both large area and small area change detection tasks, visual interpretation of IFN change maps confirms that the performance is excellent and results match with ground truth maps.

This is further corroborated based on the quantitative assessment (Table 1). IFN achieves the best performance with the highest OA (97.71%), P (94.96%), and F1 (0.9030). Unet++_MSOF achieves the highest R with 87.11%, which is slightly better than that in IFN (R 86.08%). One possible reason is that there is some image noise that has affected some of the changed/unchanged objects between the bi-temporal images. Noise embedded in changed/unchanged objects would produce image difference features wherein changed/unchanged pixels have higher values. Therefore, more changed pixels are detected by Unet++_MSOF. Meanwhile, due to the lack of guidance from individual raw image features, Unet++_MSOF tends to misclassify more

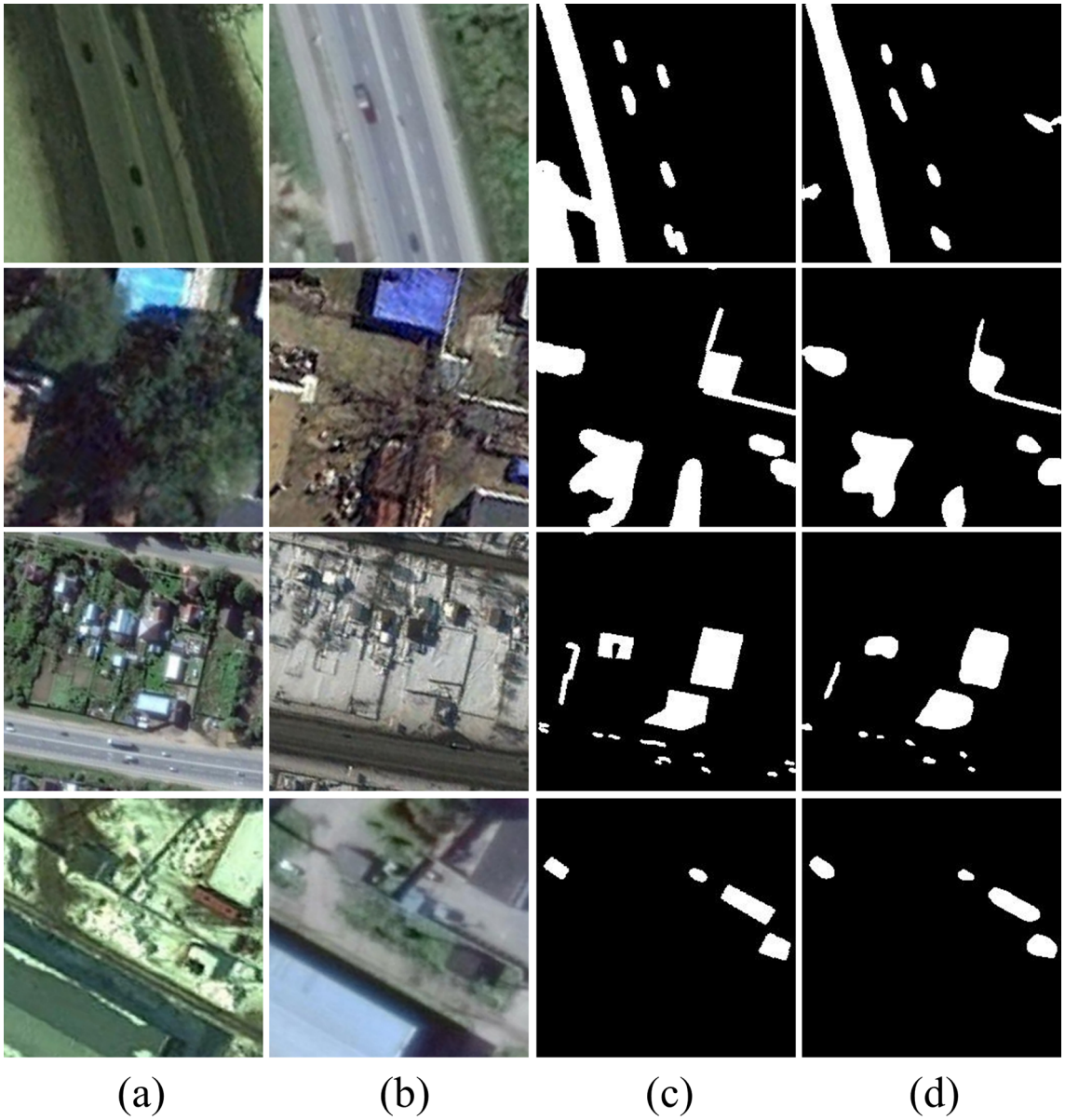

Fig. 6. Small area change detection results on the first dataset. (a) T1 images. (b) T2 images. (c) Ground truth maps. (d) Change maps produced by IFN.

Table 1
Quantitative results of IFN and the four benchmark methods.

Method	P (%)	R (%)	F1	OA (%)
IFN	94.96	86.08	0.9030	97.71
Unet++_MSOF	89.54	87.11	0.8756	96.73
FCN-PP	82.64	80.60	0.8047	95.36
FC-Siam-conc	84.41	82.50	0.8250	95.72
FC-Siam-diff	85.78	83.64	0.8373	95.75

unchanged pixels as changed pixels, resulting in a lower OA (89.54%). Compared with FCN-PP, FC-Siam-conc, and FC-Siam-diff, IFN achieves significant improvements on both P and F1 scores. Compared with Unet++_MSOF, IFN achieves 5.42% increase on P, and 0.027 increase on F1.

To further evaluate the performance of IFN, comparison is made between the results achieved with IFN and the four benchmark methods on the challenging second dataset (Figs. 7-11).

In all the examples, IFN achieves the best change detection results. The detected changed areas are returned with complete boundaries and high internal compactness. Specifically, for large area change detection in Figs. 7-9, IFN extensively outperforms all the other benchmark methods. Unet++_MSOF fails to detect some of changed areas (Fig. 7e and Fig. 9e) while it misclassifies some of the unchanged areas that are instead returned as changed areas (Fig. 8e). The other three benchmark methods (i.e., FCN-PP, FC-Siam-conc, FC-Siam-diff) tend to misclassify more unchanged pixels as changed pixels, producing change maps with lower object compactness and inaccurate object boundaries. For small objects change detection (Figs. 9-11), although IFN fails to detect some of small changed objects (e.g. Fig. 9d), it still achieves the best performance (Fig. 10d and Fig. 11d) compared with the other four benchmark methods which produce coarse change maps with severe pepper and salt noises.

From a quantitative point of view (Table. 2), IFN achieves the best performance with the highest P (67.11%), R (67.54%), F1 (0.6733), and OA (88.86%). FC-Siam-conc and FC-Siam-diff gain the worst performances with the lowest P (41.83%, 51.51%), F1 (0.4917, 0.5769), and OA (79.05%, 83.66%) scores. This is because the Siamese network integrates the training of raw image feature extraction with difference discrimination, resulting in deep raw image features with less

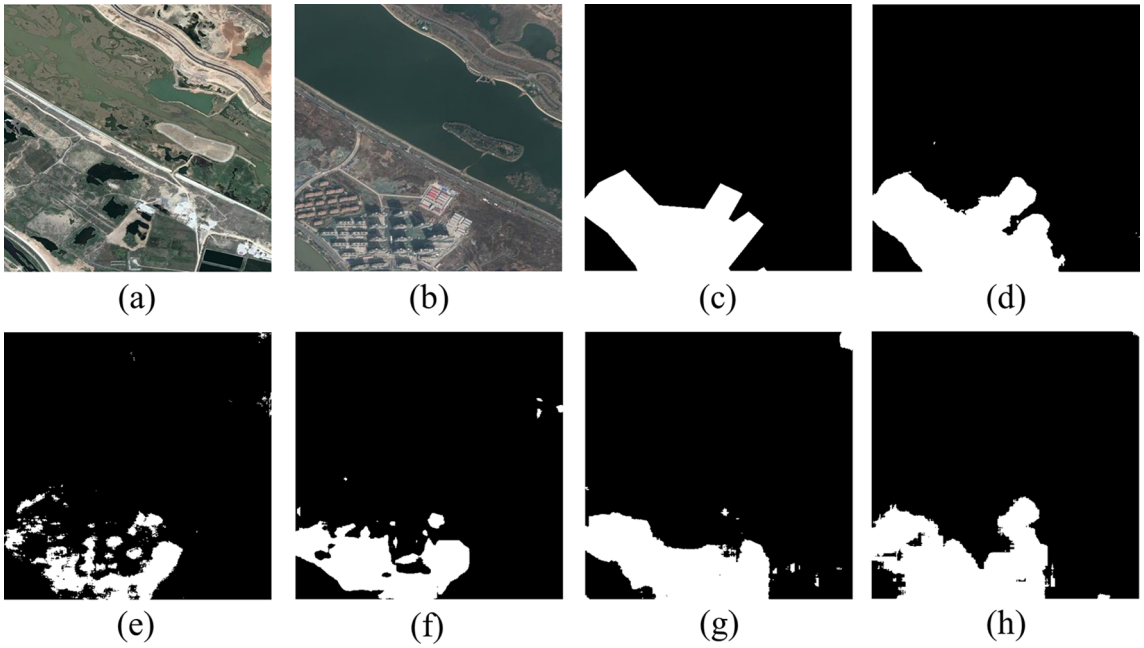

Fig. 7. The first example results on large area change detection task. (a) T1 image. (b) T2 image. (c) Ground truth map. Change maps produced by means of (d) IFN, (e) Unet++_MSOF, (f) FCN-PP, (g) FC-Siam-conc, (h) FC-Siam-diff.

representativeness. The low representative deep features of raw images, in turn, hamper difference discrimination. Unet++_MSOF and FCN-PP achieve better performances than FC-Siam-conc and FC-Siam-diff. However, since bi-temporal images are concatenated as one input of the network, image difference discrimination starts at the very beginning of the network. Though intermediate feature maps are concatenated through skip-connections, deep features of raw images can hardly be provided to help image reconstruction. As a consequence, change maps show broken object boundaries and poor object internal compactness.

4.4.2. Validation of deep supervision

The proposed IFN architecture introduces deep supervision after each feature fusion operation. Therefore, four deep supervision branches are used in IFN. In order to evaluate the effect of deep supervision, we build four networks with different deep supervision settings based on the architecture of IFN. The four networks are constructed as: 1) IFN-DS_1 keeps DS_1, and discards DS_2, DS_3, DS_4, 2) IFN-DS_12 keeps DS_1 and DS_2, and discards DS_3 and DS_4, 3) IFN-DS_123 only discards DS_4, and keeps DS_1, DS_2, and DS_3, 4) IFN-DS_0 discards all the deep supervision branches in IFN. Layer parameters (e.g., convolutional kernel) and training hyper-parameters (e.g., loss function) of the four networks are set the same. The four networks are trained on the second dataset to validate the effect of deep supervision. Learning curves of these networks are presented in Fig. 12.

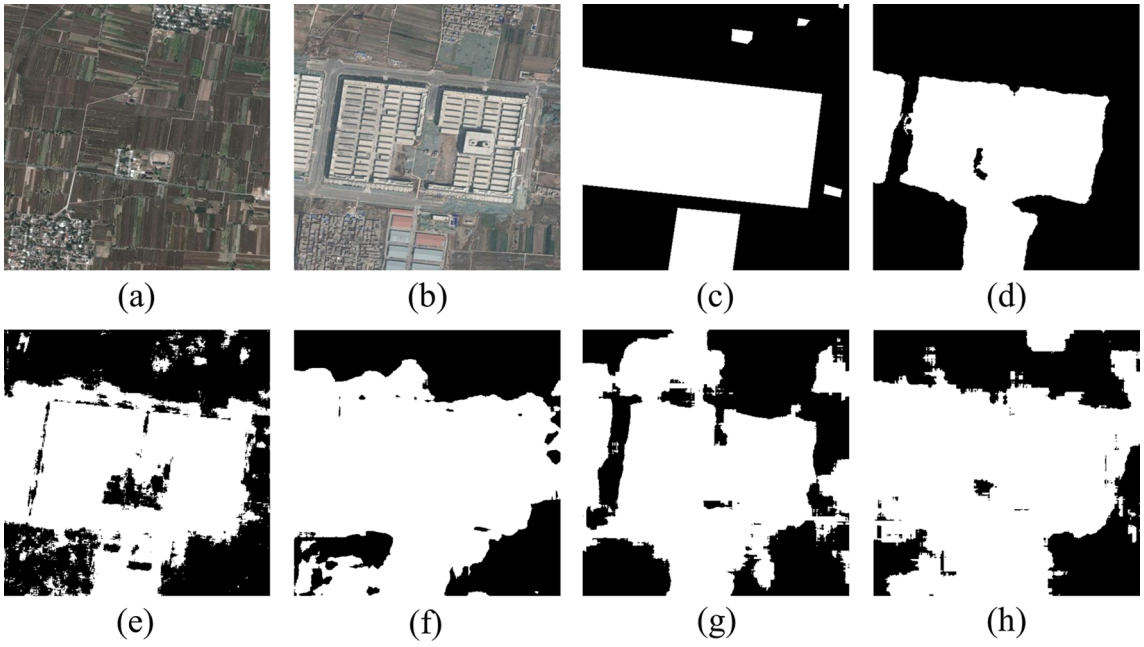

Fig. 8. The second example results on large area change detection task. (a) T1 image. (b) T2 image. (c) Ground truth map. Change maps produced by means of (d) IFN, (e) Unet++_MSOF, (f) FCN-PP, (g) FC-Siam-conc, (h) FC-Siam-diff.

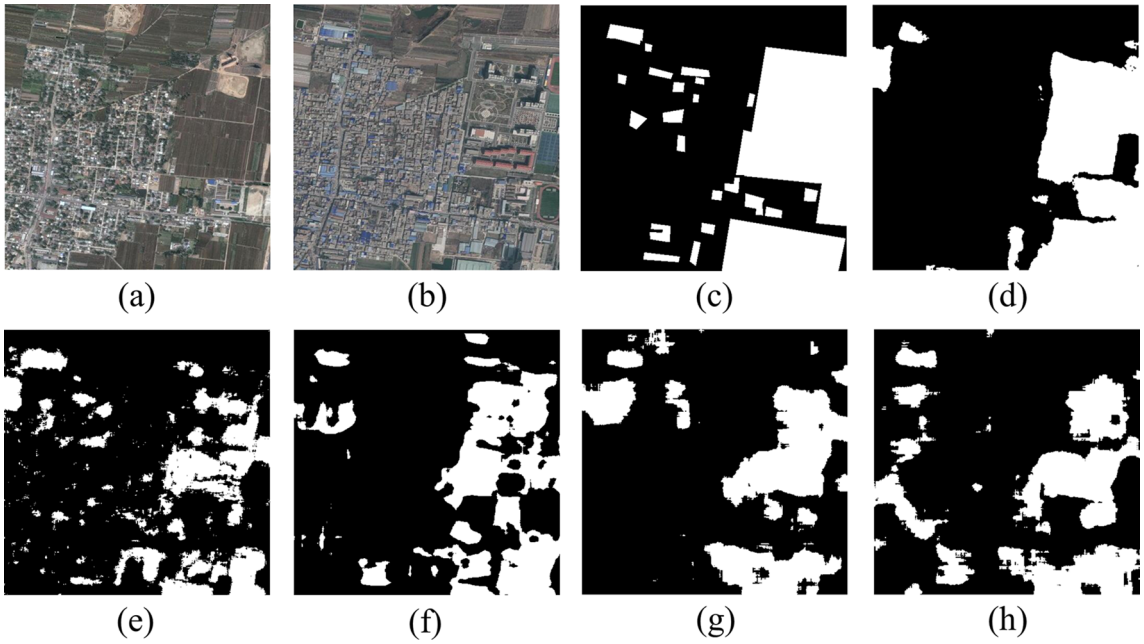

Fig. 9. The third example results on large area change detection task. (a) T1 image. (b) T2 image. (c) Ground truth map. Change maps produced by means of (d) IFN, (e) Unet++_MSOF, (f) FCN-PP, (g) FC-Siam-conc, (h) FC-Siam-diff.

Learning curves computed from training dataset demonstrate how well the model is trained. Learning curves computed from validation dataset demonstrate the goodness of model generalization ability. As shown in Fig. 12c, the training loss curve of IFN drops at the fastest rate compared to the other four networks. IFN is the best trained model. Additionally, in Fig. 12f, the gradually descending validation loss curve of IFN achieves the smallest value, which demonstrates the outstanding generalization ability of IFN. Comparatively, IFN-DS_0 has the slowest model converge speed as shown in Fig. 12a-c. Moreover, IFN-DS_0 gains the poorest generalization ability with the smallest accuracy and F1 score achieved in Fig. 12d and Fig. 12e. The other 3 networks, i.e., IFN-DS_1, IFN-DS_12, and IFN-DS_123 perform better than IFN-DS_0.

Fig. 13 and Fig. 14 allow a visual comparison of the change detection performance achieved with the four networks and IFN on selected examples.

IFN-DS_0 produces the poorest change maps (Fig. 13b, Fig. 14b) with shattered boundaries and lower object compactness. Moving from these change maps to those produced by IFN (Fig. 13f, Fig. 14f), it is clear that object boundaries become more complete and object internal compactness is higher, as more deep supervision branches are involved into the network.

As shown in Table 3, we see that IFN achieves the highest F1 (0.6733) and R (67.54%) scores. Lowest F1 and R scores are achieved with IFN-DS_0 (0.5620, 49.09%). IFN-DS_1, IFN-DS_12, IFN-DS_123

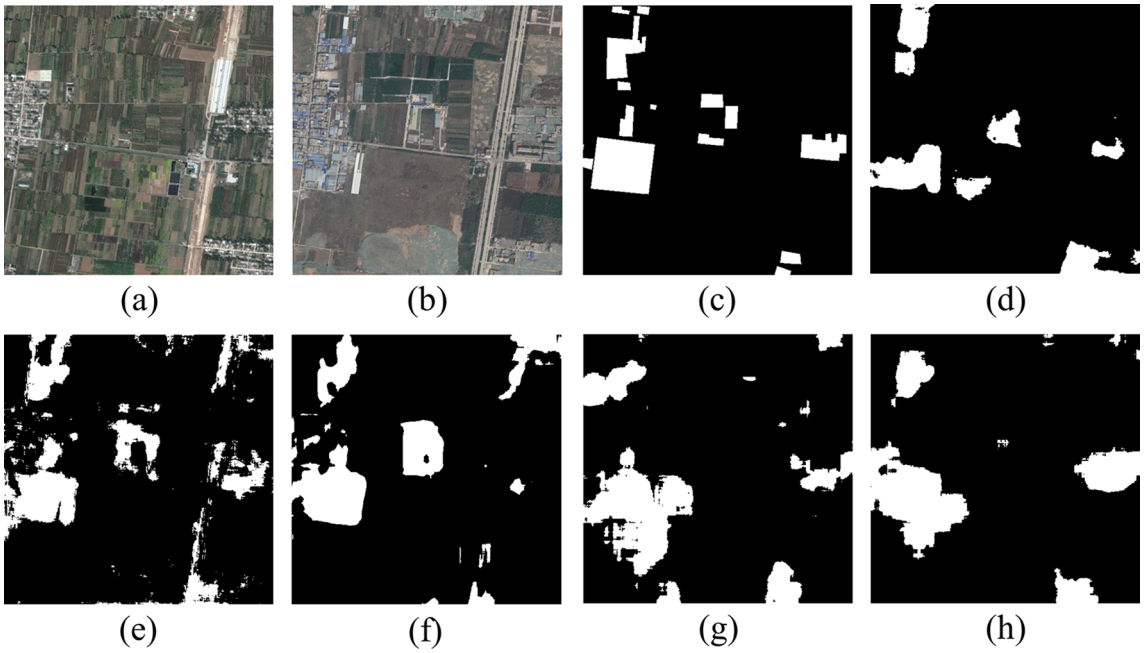

Fig. 10. The first example results on small area change detection task. (a) T1 image. (b) T2 image. (c) Ground truth map. Change maps produced by means of (d) IFN, (e) Unet++_MSOF, (f) FCN-PP, (g) FC-Siam-conc, (h) FC-Siam-diff.

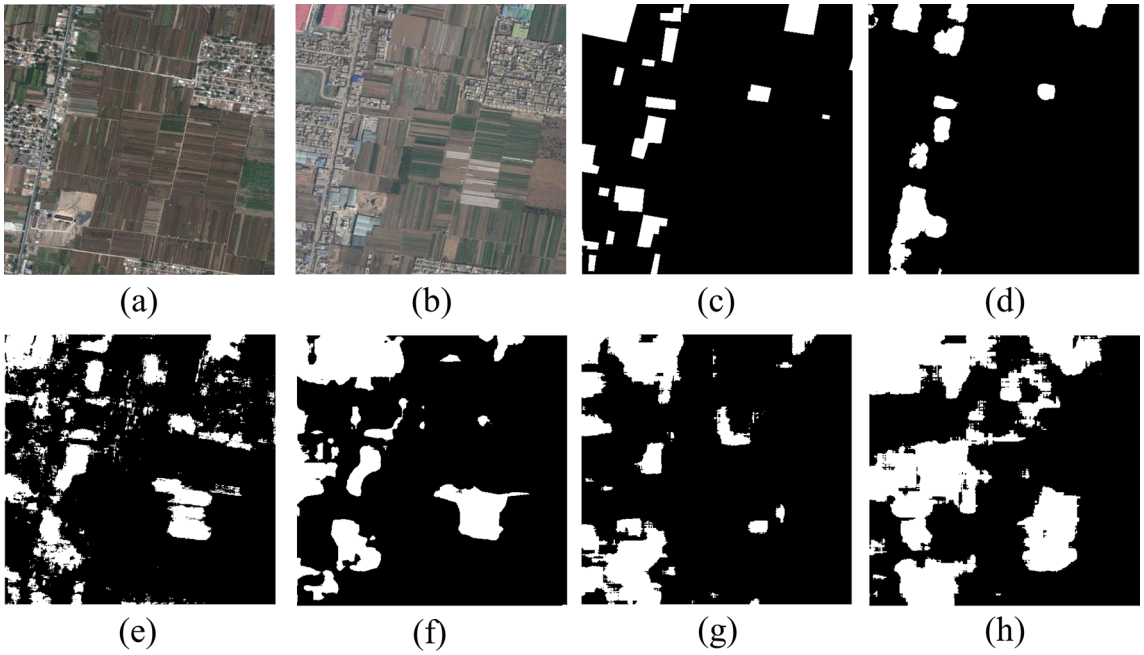

Fig. 11. The second example results on small area change detection task. (a) T1 image. (b) T2 image. (c) Ground truth map. Change maps produced by means of (d) IFN, (e) Unet++_MSOF, (f) FCN-PP, (g) FC-Siam-conc, (h) FC-Siam-diff.

gain intermediate performances between IFN-DS_0 and IFN. IFN-DS_12 gets the highest OA 88.91% which is comparable with IFN (OA 88.86%), but its F1 score is much lower. In summary, IFN achieves the best change detection performance.

5. Conclusions

In this paper, we explicitly explore the mechanisms and point out the key limitations of the state-of-the-art deep learning-based change

Table 2
Quantitative results of IFN and the four benchmark methods.

Method	P (%)	R (%)	F1	OA (%)
IFN	67.11	67.54	0.6733	88.86
Unet++_MSOF	59.83	65.91	0.6273	86.68
FCN-PP	56.40	67.03	0.6126	85.59
FC-Siam-conc	41.83	59.63	0.4917	79.05
FC-Siam-diff	51.51	65.54	0.5769	83.66

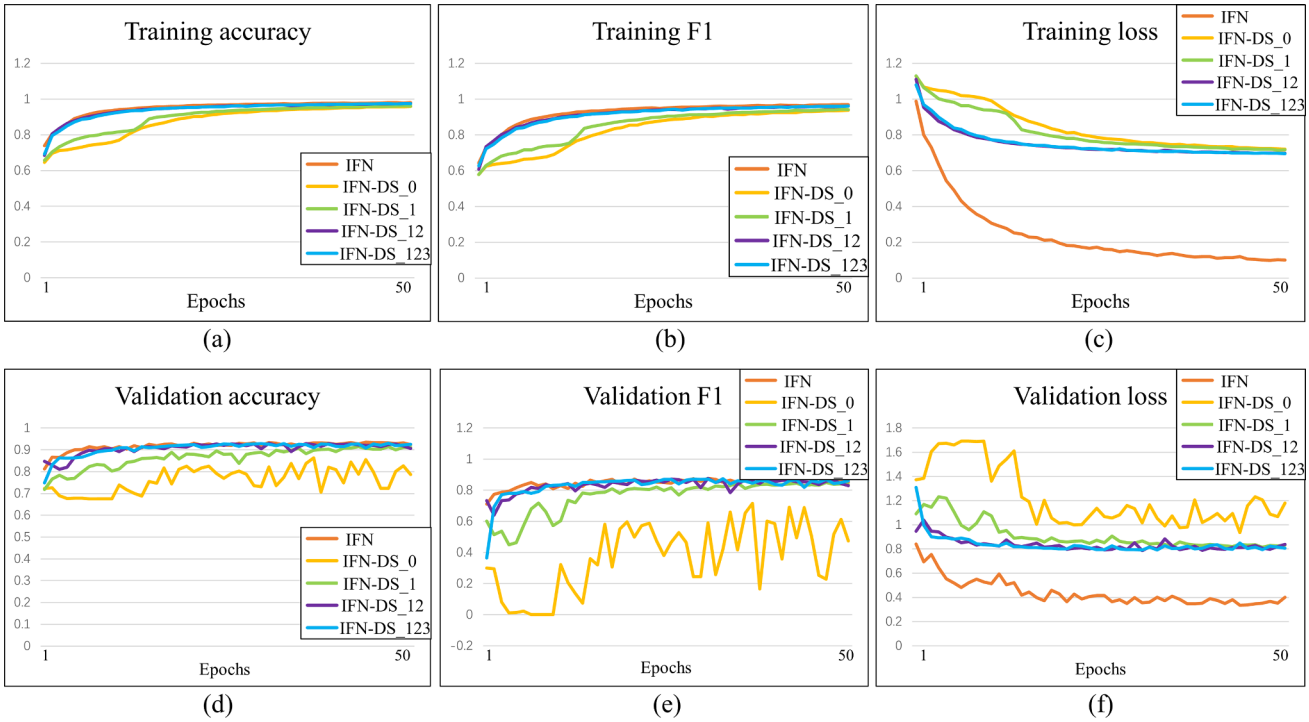

Fig. 12. Comparison of training and validation learning curves on the second dataset. (a) Training accuracy. (b) Training F1. (c) Training loss. (d) Validation accuracy. (e) Validation F1. (f) Validation loss. X-axis is the epoch, whereas y-axis is score.

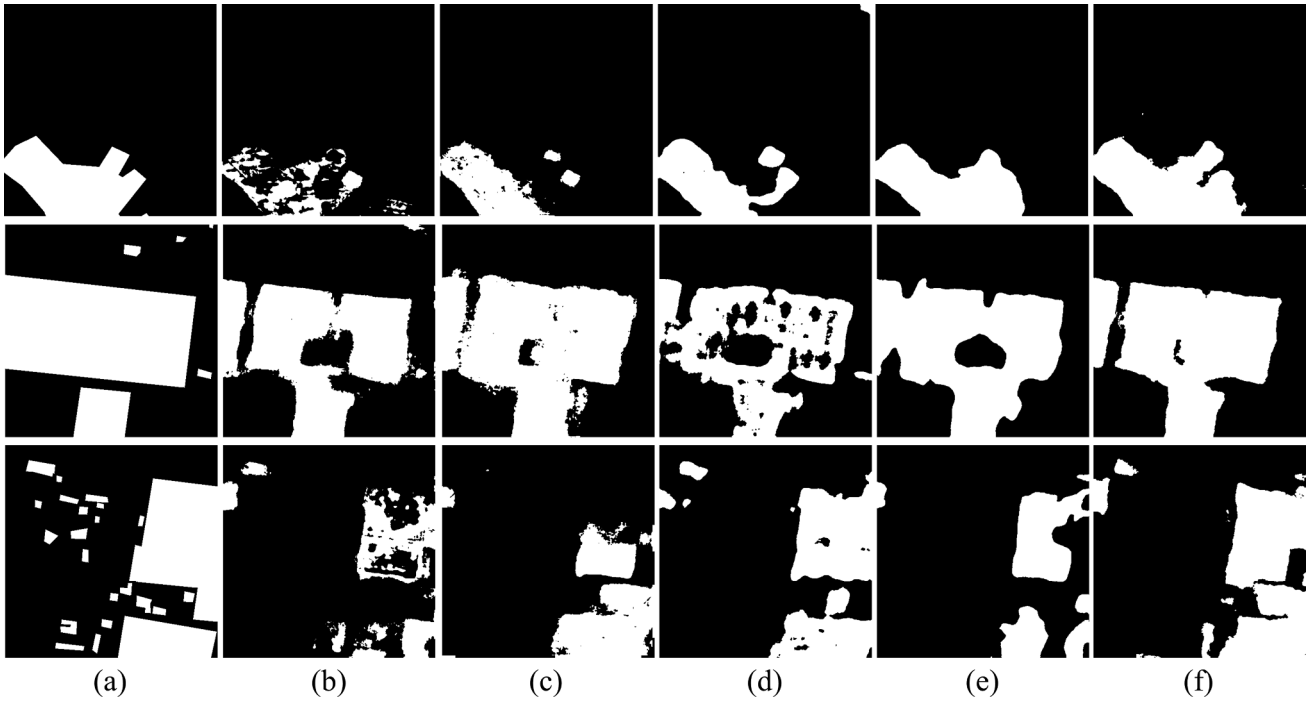

Fig. 13. Large area change detection results. (a) Ground truth map. Change maps produced by means of (b) IFN-DS_0, (c) IFN-DS_1, (d) IFN-DS_12, (e) IFN-DS_123, (f) IFN.

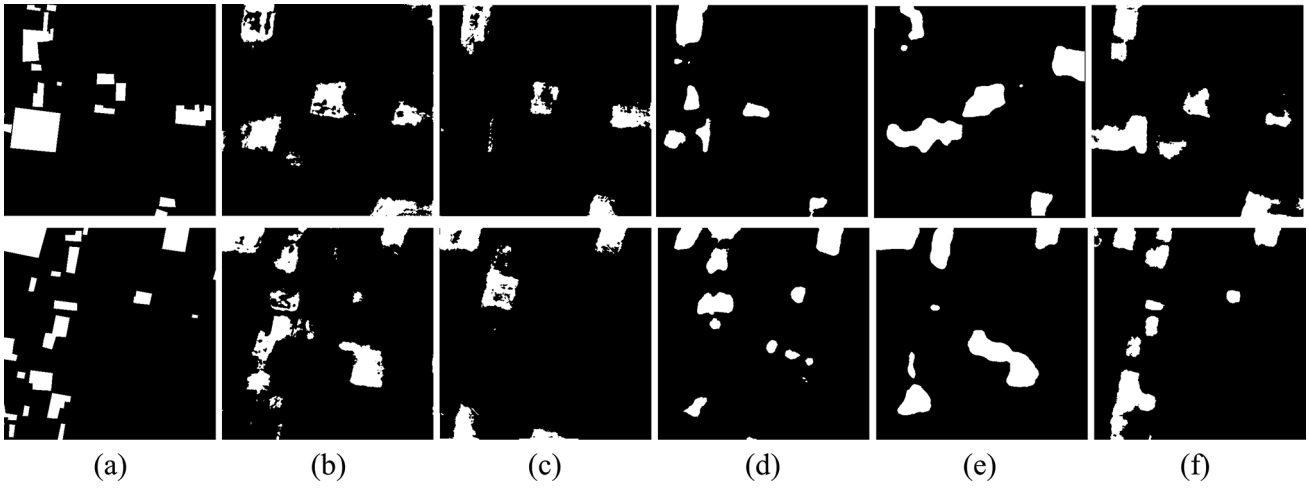

Fig. 14. Small area change detection results. (a) Ground truth map. Change maps produced by means of (b) IFN-DS_0, (c) IFN-DS_1, (d) IFN-DS_12, (e) IFN-DS_123, (f) IFN.

Table 3
Quantitative results of IFN and four modified networks.

Method	P (%)	R (%)	F1	OA (%)
IFN	67.11	67.54	0.6733	88.86
IFN-DS_123	64.73	58.71	0.6157	87.55
IFN-DS_12	72.39	56.18	0.6326	88.91
IFN-DS_1	60.02	64.83	0.6233	86.69
IFN-DS_0	65.73	49.09	0.5620	87.00

detection methods, including the early-fusion and late-fusion architectures. Rather than simply propose a modified network based on the existed architectures, we analyze the reason behind the problem and propose a deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. Feature extraction of bi-temporal images is conducted by an independently trained fully convolutional two-stream architecture to enhance feature representativeness. Raw image features are fused with image difference features in difference discrimination network to complement multi-level information of changed objects for better change map reconstruction. To overcome the heterogeneity problem of feature fusion, attention modules are applied over the combined features to adaptively emphasize important features while suppressing irrelevant features across the channel and spatial dimensions. Besides, to improve the performance of difference discrimination network, we propose deep supervision by introducing direct feedbacks from downsampled change maps into intermediate layers in the network.

The proposed method was evaluated on two datasets. The first dataset is a publicly available dataset with very high image resolutions. Benchmark comparison on the first dataset demonstrates a satisfying performance of the proposed method. The second dataset consists of multi-source bi-temporal images from Google Earth covering different

cities in China. On one hand, images taken over different cities are used as the training dataset, which increases the difficulty of discriminating changed areas in different scenarios; on the other hand, images taken over another city are used for model testing, aiming to test the generalization ability of the proposed method. Tested on the second dataset, IFN outperforms all the benchmark methods. Both experiments demonstrate the effectiveness and robustness of IFN. Future works will investigate the availability of the proposed method for change detection in heterogeneous bi-temporal images, for example, change detection task on bi-temporal Synthetic Aperture Radar and optical images.

Acknowledgements

We appreciate the reviewers and editors for their constructive comments that helped improve the quality of the paper. The work was supported by Major State Research Development Program of China (No. 2017YFB0504103), National Natural Science Foundation of China (No. 41722109), Hubei Provincial Natural Science Foundation of China (No. 2018CFA053), and Wuhan Yellow Crane Talents (Science) Program (2016).

Declaration of Interest Statement

Each of the authors confirms that no part of this manuscript has been previously published, nor is any part is currently under consideration by any other journal. Additionally, each of the authors has approved the contents of this paper and have agreed to the submission policies of ISPRS Journal of Photogrammetry and Remote Sensing.

Appendix A. – The first dataset

The dataset contains 7 pairs of season-varying images with 47,252,700× pixels for manual ground truth creation and 4 season-varying images with minimal changes and resolution of 19,001,000× pixels for adding additional objects manually. Spatial resolution of the images varies from 3 to 100 cm/px. Images are clipped into sub-images with sizes of 256,256× to generate the final dataset. The final dataset contains 16,000 images pairs of which 10,000 image pairs for model training, 3000 for model validation and model testing, respectively. Note that data augmentation is not applied on the first dataset considering the large data volume of the first dataset. The corresponding ground truth maps are also provided in the dataset. The experimental setups in the manuscript including the training dataset, validation dataset, test dataset, image sizes, and image resolutions are exactly the same as in Lebedev et al. (2018). Fig. A.1 shows example images of the first dataset.

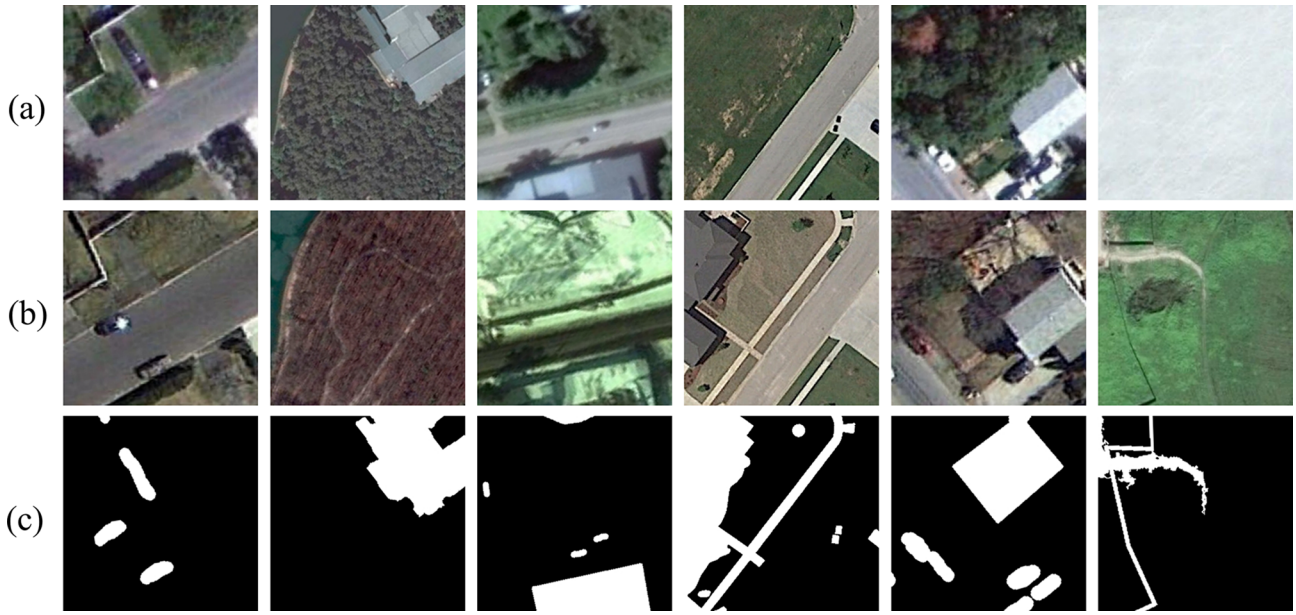

Fig. A1. Example images of the first dataset. (a) T1 images. (b) T2 images. (c) Ground truth maps. Changed areas are in white color, unchanged areas are in black color.

Appendix B. – The second dataset

The dataset consists of 6 large co-registered bi-temporal image pairs that are collected from six major cities in China. As shown in Figs. B.1-B.6, on one hand, these season-varying images provided by Google Earth are collected by a variety of sensors, resulting in a dataset with varied image brightness and contrast; on the other hand, the multi-source image pairs collected from six cities contain land objects in great differences, both of which make the second dataset more challenging for change detection. Note that in Fig. B.2 (b), Fig. B.3(a) and Fig. B.3(b), due to the limited size of single optical image, images taken at similar times are stitched together to generate T1 and T2 images. For the ground truth map, we manually compare the bi-temporal images and mark changed areas by observing appearance and disappearance of land cover objects, while ignoring the changes caused by season changing and image brightness. Specifically, we firstly pick T1 images as the “master” and T2 images as the “slave”. Then we observe the appearance and disappearance of land cover objects such as roads, buildings, croplands, water bodies on the T2 image. If the land cover type has been significantly changed on the T2 image, then we mark it as changed area. It should be noted that T2 images are only compared to T1 images if they both cover the same city. For spectral and texture changes caused by season changing in T2 images, we mark it as unchanged area. For example, in T2 image of Chengdu city, there is fewer grass growing on bare lands compared with T1 image. This might be caused by drought. We consider such kind of image change as a seasonal change and mark those areas as unchanged areas on the ground truth map. The dataset is described in Table B.1.

Fig. B1. Beijing images. (a) T1 image. (b) T2 image. (c) Ground truth map.

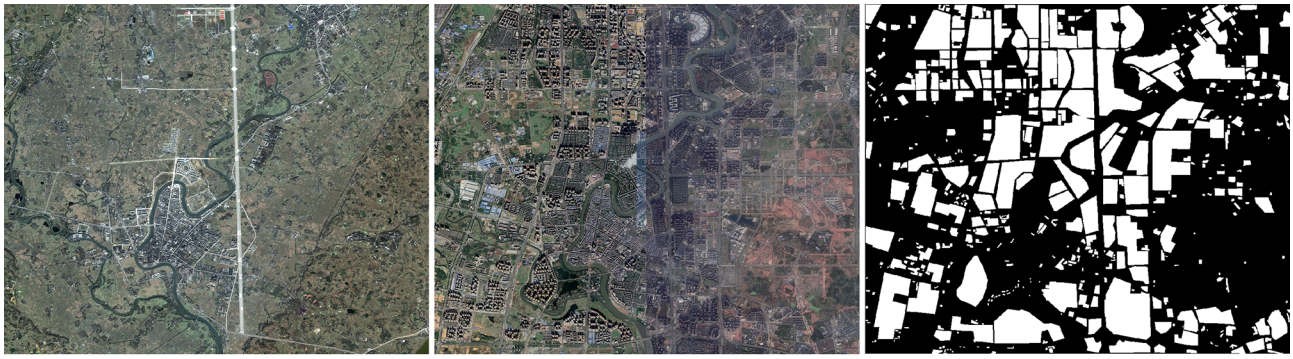

Fig. B2. Chengdu images. (a) T1 image. (b) T2 image. (c) Ground truth map.

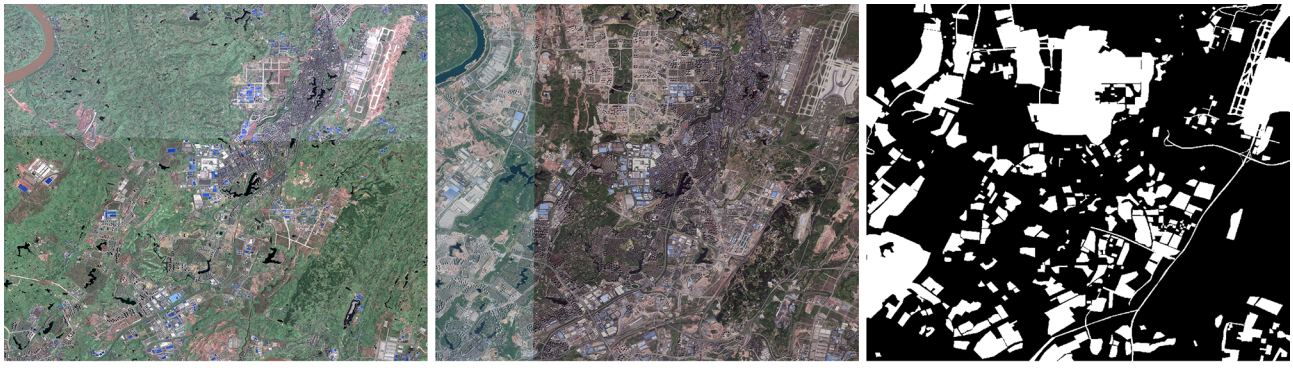

Fig. B3. Chongqing images. (a) T1 image. (b) T2 image. (c) Ground truth map.

Fig. B4. Shenzhen images. (a) T1 image. (b) T2 image. (c) Ground truth map.

(a) (b) (c)

Fig. B5. Xian images. (a) T1 image. (b) T2 image. (c) Ground truth map.

(a) (b) (c)

Fig. B6. Wuhan images. (a) T1 image. (b) T2 image. (c) Ground truth map.

Table B1
Description of the second dataset. Images of Beijing, Chengdu, Shenzhen, Chongqing, and Wuhan are used for model training. Xian images are used for model testing. Spatial resolution of all the images is 2 m.

City	Time T1	Time T2	Image size
Beijing	2013	2018	6236×4652
Chengdu	2001	2018	4412×3636
Shenzhen	2002	2017	2010×1464
Wuhan	2009	2017	6963×4555
Chongqing	2009	2019	6542×5492
Xian	2003	2018	4392×3140

References

Alcantarilla, P.F., Stent, S., Ros, G., Arroyo, R., Gherardi, R., 2018. Street-view change detection with deconvolutional networks. Auton. Robots. 42, 1301–1322. https://doi.org/10.1007/s10514-018-9734-5.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R., 1994. Signature verification using a“ siamese” time delay neural network. InAdvances in neural information processing systems. 737–744. https://doi.org/10.1142/s0218001493000339.

Caye Daudt, R., Le Saux, B., Boulch, A., 2018. Fully convolutional siamese networks for change detection. In: in: Proceedings - International Conference on Image Processing, ICIP, pp. 4063–4067. https://doi.org/10.1109/ICIP.2018.8451652.

Celik, T., 2009. Unsupervised change detection in satellite images using principal component analysis and κ-means clustering. IEEE Geosci. Remote Sens. Lett. 6, 772–776. https://doi.org/10.1109/LGRS.2009.2025059.

Chen, G., Hay, G.J., Carvalho, L.M.T., Wulder, M.A., 2012. Object-based change detection. Int. J. Remote Sens. 33, 4434–4457. https://doi.org/10.1080/01431161.2011.648285.

Daudt, R.C., Saux, B. Le, Boulch, A., Gousseau, Y., 2018. High Resolution Semantic Change Detection. arXiv preprint arXiv:1810.08452.

Deng, J.S., Wang, K., Deng, Y.H., Qi, G.J., 2008. PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data. Int. J. Remote Sens. 16, 4823–4838. https://doi.org/10.1080/01431160801950162.

El Amin, A.M., Liu, Q., Wang, Y., 2017. Zoom out CNNs features for optical remote sensing change detection, in: 2017 2nd International Conference on Image, Vision and Computing, ICIVC 2017. 2, 812-817. https://doi.org/10.1109/ICIVC.2017.7984667.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics. 9, 249–256.

Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier neural networks. Journal of Machine Learning Research. 315–323.

Guo, E., Fu, X., Zhu, J., Deng, M., Liu, Y., Zhu, Q., Li, H., 2018. Learning to Measure Change: Fully Convolutional Siamese Metric Networks for Scene Change Detection. arXiv preprint arXiv:1810.09111.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. The IEEE International Conference on Computer Vision. 1026–1034.

Hou, B., Wang, Y., Liu, Q., 2017. Change Detection Based on Deep Features and Low Rank. IEEE Geosci. Remote Sens. Lett. 14, 2418–2422. https://doi.org/10.1109/LGRS.2017.2766840.

Jackson, R.D., 1983. Spectral indices in N-Space. Remote Sens. Environ. 13, 409–421. https://doi.org/10.1016/0034-4257(83)90010-X.

Jia Deng, Wei Dong, Socher, R., Li-Jia Li, Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database. pp. 248-255. https://doi.org/10.1109/cvprw.2009.5206848.

Jin, S., Yang, L., Danielson, P., Homer, C., Fry, J., Xian, G., 2013. A comprehensive change detection method for updating the National Land Cover Database to circa 2011. Remote Sens. Environ. 132, 159–175. https://doi.org/10.1016/j.rse.2013.01.012.

Kuncheva, L.I., Faithfull, W.J., 2014. PCA feature extraction for change detection in multidimensional unlabeled data. IEEE Trans. Neural Networks Learn. Syst. 25, 69–80. https://doi.org/10.1109/TNNLS.2013.2248094.

Lebedev, M.A., Vizilter, Y.V., Vygolov, O.V., Knyaz, V.A., Rubis, A.Y., 2018. Change detection in remote sensing images using conditional adversarial networks. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII–2, 565–571. https://doi.org/10.5194/isprs-archives-XLII-2-565-2018.

Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z., 2015. Deeply-supervised nets. Artificial intelligence and statistics. 562–570.

Lei, T., Zhang, Y., Lv, Z., Li, S., Liu, S., Nandi, A.K., 2019a. Landslide Inventory Mapping From Bitemporal Images Using Deep Convolutional Neural Networks. IEEE Geosci. Remote Sens. Lett. 16, 982–986. https://doi.org/10.1109/LGRS.2018.2889307.

Lei, Y., Liu, X., Shi, J., Lei, C., Wang, J., 2019b. Multiscale Superpixel Segmentation with Deep Features for Change Detection. IEEE. Access. 36600–36616. https://doi.org/10.1109/ACCESS.2019.2902613.

Liu, J., Gong, M., Qin, A.K., Tan, K.C., 2020. Bipartite Differential Neural Network for Unsupervised Image Change Detection. Neural Networks Learn. Syst, IEEE Trans 10.1109/TNNLS.2019.2910571.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully Convolutional Networks for Semantic Segmentation. The IEEE Conference on Computer Vision and Pattern Recognition. 3431–3440.

Luppino, L.T., Bianchi, F.M., Moser, G., Anfinsen, S.N., 2019. Unsupervised image regression for heterogeneous change detection. IEEE Trans. Geosci. Remote Sens. https://doi.org/10.1109/TGRS.2019.2930348.

Lv, N., Chen, C., Qiu, T., Sangaiah, A.K., 2018. Deep Learning and Superpixel Feature Extraction Based on Contractive Autoencoder for Change Detection in SAR Images. IEEE Trans. Ind. Informatics. 14, 5530–5538. https://doi.org/10.1109/TII.2018.2873492.

Mao, T., Liu, W., Zhao, Y., Huang, J., 2018. Change Detection in Semantic Level for SAR Images, in: 2018 3rd IEEE International Conference on Image, Vision and Computing, ICIVC 2018. https://doi.org/10.1109/ICIVC.2018.8492796.

Mundia, C.N., Aniya, M., 2005. Analysis of land use/cover changes and urban expansion of Nairobi city using remote sensing and GIS. Int. J. Remote Sens. 26, 2831–2849. https://doi.org/10.1080/01431160500117865.

Padron-Hidalgo, J.A., Laparra, V., Longbotham, N., Camps-Valls, G., 2019. Kernel Anomalous Change Detection for Remote Sensing Imagery. IEEE Trans. Geosci. Remote Sens. 57, 7743–7755. https://doi.org/10.1109/TGRS.2019.2916212.

Peng, D., Guan, H., 2019. Unsupervised change detection method based on saliency analysis and convolutional neural network. J. Appl. Remote Sens. 13, 024512. https://doi.org/10.1117/1.jrs.13.024512.

Peng, D., Zhang, Y., Guan, H., 2019. End-to-end change detection for high resolution satellite images using improved UNet++. Remote Sens. 11, 1382. https://doi.org/10.3390/rs11111382.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-319-24574-4_28.

Saha, S., Bovolo, F., Bruzzone, L., 2019. Unsupervised deep change vector analysis for multiple-change detection in VHR Images. IEEE Trans. Geosci. Remote Sens. 57, 3677-3693. https://doi.org/10.1109/TGRS.2018.2886643Saha, S., Bovolo, F., Bruzzone, L., 2019. Unsupervised Multiple-Change Detection in VHR Multisensor Images Via Deep-Learning Based Adaptation, in: IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, 5033–5036. https://doi.org/10.1109/igarss.2019.8900173.

Shelhamer, E., Long, J., Darrell, T., 2017. Fully Convolutional Networks for Semantic Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 640–651. https://doi.org/10.1109/TPAMI.2016.2572683.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Singh, A., 1989. Review Articlel: Digital change detection techniques using remotely-sensed data. Int. J. Remote Sens. 10, 989–1003. https://doi.org/10.1080/01431168908903939.

Singh, A., 1986. Change detection in the tropical forest environment of northeastern India using Landsat. Remote Sens. Trop. L. Manag, 237–254.

Todd, W.J., 1977. Urban and regional land use change detected by using Landsat data. J. Res. US Geol. Surv. 5, 529–534.

Wang, F., Xu, Y.J., 2010. Comparison of remote sensing change detection techniques for assessing hurricane damage to forests. Environ. Monit. Assess. 162, 311–326. https://doi.org/10.1007/s10661-009-0798-8.

Woo, S., Park, J., Lee, J.Y., Kweon, I.S., 2018. CBAM: Convolutional block attention module. Lecture Notes in Computer Science. 3–19. https://doi.org/10.1007/978-3-030-01234-2_1.

Wu, C., Du, B., Cui, X., Zhang, L., 2017. A post-classification change detection method based on iterative slow feature analysis and Bayesian soft fusion. Remote Sens. Environ. 199, 241–255. https://doi.org/10.1016/j.rse.2017.07.009.

Zerrouki, N., Harrou, F., Sun, Y., 2018. Statistical Monitoring of Changes to Land Cover. IEEE Geosci. Remote Sens. Lett. 15, 927–931. https://doi.org/10.1109/LGRS.2018.2817522.

Zhang, H., Gong, M., Zhang, P., Su, L., Shi, J., 2016. Feature-Level Change Detection Using Deep Representation and Feature Change Analysis for Multispectral Imagery. IEEE Geosci. Remote Sens. Lett. 13, 1666–1670. https://doi.org/10.1109/LGRS.2016.2601930.

Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J., 2018. Unet++: A nested u-net architecture for medical image segmentation. Lecture Notes in Computer Science. 3–11.