Change detection for multispectral images using modified semantic segmentation network

Linzhi Su, Qiaoyun Xie, Fengjun Zhao, and Xin Cao*
Northwest University, School of Information Science and Technology, Xi'an, China

Abstract. Change detection is a significant issue for understanding the changes occurring on the land surface. We propose a change detection approach based on a semantic segmentation network from multispectral (MS) images. Different from the traditional approaches that learn deep features from the change index or establish mapping relations from patches, the proposed approach employs the semantic segmentation network UNet++ for end-to-end change detection. Nevertheless, in UNet++, the deep feature is directly upsampled from the node in the lower level and does not involve much information from the nodes in the other levels. To cope with this problem and further enhance its robustness, the zigzag UNet++ (ZUNet++) is developed. In ZUNet++, the zigzag connection between nodes can be found, so the inputs of the node involve not only the upsampled deep feature but also the downsampled shallow feature, i.e., the network fuses multiple feature information. In addition, as few MS training datasets are available, we designed a strategy in which each MS image is transferred into several pseudo-RGB images; thus the network is trained by available RGB training sets and can be applied to the testing MS datasets. In the experiment, three real testing MS datasets that reflect different types of changes in Xi'an City are used. Experimental results show that, upon determining the appropriate parameter, the proposed ZUNet++ outperforms the other state-of-the-art approaches, demonstrating its feasibility and effectiveness. © *2022 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JRS.16.014518]

Keywords: change detection; multispectral image; semantic segmentation; deep neural networks.

Paper 210355 received Jun. 4, 2021; accepted for publication Feb. 17, 2022; published online Mar. 2, 2022.

1 Introduction

The last few decades have witnessed the rapid development of cities, with old facilities being updated and new constructions being built. Such changes can be detected from various remote sensing images and through change detection techniques.[1–4] In the literature, change detection aims to find the changed area occurring on the land surface in a series of sequential multitemporal images.[5]

Early studies mainly focus on the gray level or patch-based techniques. A three-step framework, which contains image preprocessing, the generation of a change index (CI), and its analysis, has been proposed.[6,7] Image preprocessing includes the image co-registration,[8] geometric correction,[9] and image filtering.[10] In the second step, the CI is usually obtained through a difference operator (which leads to a difference image)[11] or through concatenation (to form a joint feature image).[12] Finally, changes are detected by analyzing the CI. Based on this framework, many basic techniques have been designed. For example, threshold or clustering-based approaches have been proposed in several basic studies.[6,13–17] The change vector analysis (CVA) technique[18] along with its improved versions was then proposed to cope with the problem of particular types of images, such as multispectral (MS) and hyper-spectral (HS) images.[7,12,19,20] These approaches work well when highly accurate results are not necessarily needed and have a relatively low time cost. When it comes to some complicated problems or a demand for more accurate results, however, they are quite limited. Therefore, researchers turned to deep learning

*Address all correspondence Xin Cao, xin_cao@163.com

1931-3195/2022/$28.00 © 2022 SPIE

techniques, which are able to obtain and analyze the deep features from images to detect real changes. Deep learning means adopting deep network architectures to automatically learn hierarchical feature representations using supervised or unsupervised strategies,[21] and it has wide application in many research fields. In the field of change detection, some new approaches, both unsupervised and supervised, were then developed.

For the unsupervised techniques, two main categories are mainly involved. One category is that the deep neural network itself works as an unsupervised method. It usually involves an unsupervised network (i.e., different types of autoencoders and their stacked versions to form deep architectures) and traditional classification algorithms. Lv et al.[22] proposed a simple framework that involves the stacked contractive autoencoder for feature extraction and the K-means clustering algorithm for generating the final map. Geng et al.[23] proposed an unsupervised saliency guided nonnegative- and Fisher-constrained autoencoder (NFCAE). The approach involves extracting a salient region (which probably represents the changed region) from the DI and hierarchical fuzzy C-means clustering.[24] The pseudotraining samples are used to facilitate the NFCAE to obtain reliable detection results. To cope with multiple change detection for single-channel images, Su et al.[25] suggested using stacked denoising AE (DAE) to generate the corresponding multichannel feature imaging, broadening the scope of the application of the CVA technique as the feature change analysis (FCA). A similar method was proposed by Su and Cao in which the novel fuzzy AE was developed to obtain more accurate feature representation.[26] The other unsupervised category involves the training-and-testing process. Different from the supervised approaches, the training inputs are preselected from the given image according to a rough classification, and then all of the image data (as the testing data) is put into the trained network. This process works without necessary supervision. Gao et al.[27] showed a framework that contains deep seminonnegative matrix factorization, which serves as a rough classifier to preselect the training samples, and a singular value decomposition network to obtain reliable high-quality features. Zhang et al.[28] employed the deep belief network to facilitate the FCA, not only capturing the robust features of changed region but also suppressing the irrelevant variations. Similarly, some unsupervised frameworks that involve two different kinds of networks were then put forward. One network is for feature extraction, and the other establishes the function between features from relevant samples. Zhang et al.[29] adopted this idea to cope with multiresolution change detection through the DAE and a simple mapping network. This idea was then improved by Su et al. to cope with the ternary change detection problem.[30] After the feature extraction through stacked DAE, three mapping functions are established for the three classes, and the final changed map is generated through clustering to the corresponding feature mapping index. Gong et al.[31] further improved this framework by updating the feature extraction and mapping networks into the stack sparse AE (SAE) and convolutional neural network (CNN), respectively. Thus the improved framework achieved more robust feature and more flexible mapping functions. Some other excellent networks are also in the related literature. For example, the generative discriminatory network (GAN) learns to produce novel data with the same statistics as the original data, and the generator network finally competes against an adversary.[32] Thus it is useful when there are not many samples for training. Gong et al.[33] developed a GAN-based change detection approach. First, a generator is used to generate fake data; then after a rough segmentation, which discriminates the unlabeled data from the labeled data, the discriminatory network updates the label for each pixel by combining and training the fake, labeled, and unlabeled data together. The generative and discriminatory networks are alternately trained until the objective function converges. Saha et al.[34] further developed the unsupervised CVA technique by choosing a trained CNN, extracting robust features, and facilitating the classification.

On the one hand, these unsupervised techniques are simple in practice and rely little on the training sets as in the supervised approaches. On the other hand, the training samples in some unsupervised techniques are selected or derived from the testing data itself, leading to inaccurate parameters in the networks. In addition, the selection of the training samples by rough classifiers also engenders inaccuracy. To cope with these issues, researchers have built supervised architectures for both training and learning. Wang et al.[35] proposed the general end-to-end 2D CNN (GETNET) for HS image change detection, which is highlighted by their subpixel information fusion method. In GETNET, each pixel is turned into a mixed data cube after linear and nonlinear unmixing, and then the corresponding mixed-affinity matrix is generated, forming the direct inputs of the network. Amin et al.[36] proposed an approach based on CNN; the features are extracted from the CaffeNet model by activating the pretrained CNNs, generating a feature vector for each pixel. Zhan et al.[37] suggested using a supervised deep Siamese CNN to generate two 16-dimensional pixel-wise feature vectors. To generate the final changed map, both approaches employ the Euclidean distance. Mou et al.[38] brought together a CNN and a recurrent neural network (RNN) into a one end-to-end network. In their work, three types of RNN architectures, the fully connected (FC) RNN, the long-short-term memory, and the gated recurrent unit, are used to construct the corresponding subnetwork.

To further detect and recognize the changes on the land surface, recent studies have adopted the semantic segmentation algorithms. Some techniques based on semantic segmentation and recognition networks have been developed. Saha et al.[39] proposed an unsupervised semantic segmentation approach in which the deep feature is extracted and then clustered to obtain the final label for the changed area. In addition to such approaches, the fully convolutional network (FCN)[40] serves as a common and basic tool to build up this architecture. Traditional CNNs usually output a single prediction for each input image, whereas FCNs are able to predict labels for each pixel independently and efficiently with an arbitrary size of input. Hence, the FCNs are especially suited for image semantic analysis. Wiratama et al.[41] showed a Siamese architecture in which two dense-CNN are connected in series for each image and eventually a probability output is generated, hence the name dual-dense convolutional network. Zhang et al.[42] introduced a fully atrous CNN, the encoder of which consists of several fully atrous convolution layers, expanding the receptive field in the convolution process. Daudt et al.[43] made a deep study on semantic change detection and proposed several multitask learning strategies based on integrated deep FCNs. They showed the two simple and two integrated semantic change detection strategies with land cover maps. In addition to the architectures mentioned above, some special versions of FCN, such as SegNet,[44] UNet,[45] and RefineNet,[46] have also become popular. It has been summarized from the literature that UNet can be considered to be one of the standard architectures used for this issue.[47] The general structure of UNet is symmetric (similar to the letter "U"), and it has an encoder that extracts spatial features and a decoder that generates the segmentation map from the feature. UNet is effective for processing medical images,[48] and the related literature also demonstrates its applicability in change detection. Jaturapitpornchai et al.[49] proposed a novel change detection method based on UNet for detecting building construction. The new buildings can be thus detected between two synthetic aperture radar images captured at two different times. A similar architecture was also proposed by Li et al.,[50] who applied the residual UNet to urban building change detection. Hamdi et al.[51] also used UNet to detect the damaged forests, and the network was trained on the database of a forest area in Bavaria, Germany. In addition to these, UNet++,[52] an enhanced version of UNet proposed by Zhou et al., was also applied to the field. UNet++ is also referred to as nested UNet, in which several UNet architectures are nested together and the dense skip connections are established between the nodes at the same depth. Actually, considering UNet++, Alexakis and Armenakis[53] made a further evaluation and comparison of these two on several RGB datasets, and the results by UNet++ were demonstrated to be better than those by UNet. From the corresponding experiments, the authors also suggested using the binary cross-entropy loss with the Dice coefficient function (BCE-Dice loss) to train the network and obtained satisfactory results. Peng et al.[54] studied UNet++ and applied the network to change detection for high-resolution satellite images. The backbone of the network proposed by Zhou et al. was adopted, and using the multiple side-output fusion, the final output was obtained from the four direct outputs. Their qualitative and quantitative results also demonstrate the superiority of UNet++ over the other architectures.

Despite the effectiveness of the aforementioned methods, it is found that supervised semantic change detection still faces some challenges. First, although the skip connections are introduced to enhance the performance, these types of networks still need to be improved by exploring sufficient information from full scales. Second, the training sets are limited because few MS training image pairs are available, whereas natural RGB training images are abundant. To cope with these issues, in this paper, we propose a zigzag UNet++ (ZUNet++) architecture for supervised end-to-end semantic change detection based on MS images. The highlights of the work can be summarized as follows.

(1) Different from the ordinary UNet++ model, in the proposed ZUNet++ architecture, the skip connection of two nodes at the same depth is modified into a zigzag manner, forming an indirect skip connection and thus utilizing more information from the other nodes.
(2) To adapt the available training sets and fully utilize the information provided by the spectral channels, one MS image is transferred into several pseudo-RGB images; thus each channel of the image can be fully tackled, further improving the accuracy.

This paper is organized as follows. Section 2 provides basic background knowledge, and Sec. 3 introduces the proposed ZUNet++ and the entire framework in detail. The datasets and experimental settings and the related evaluation criteria are given in Sec. 4. Experimental results are given in Sec. 5. Finally, we give the concluding remarks in Sec. 6.

2 Background

In this section, an introduction to the background knowledge, including the task description and the available related networks, is given.

2.1 *End-to-End Semantic Change Detection*

Let us consider I_1 and I_2, an image pair consisting of two $A \times B \times S$ MS images taken over the same geographical area at two different times, respectively. A and B are the height and width, and S is the spectral channel number. The aim of change detection is to find a final map I_F that indicates the changes occurring between two times. As summarized in Ref. 43, the problem can be treated as a dense classification problem, aiming to predict a label for each pixel in an input image pair, i.e., achieving semantic segmentation. As introduced in Sec. 1, some approaches involve several steps, in which image features are represented (or extracted) first and then analyzed. In general, this entails an excellent algorithm for extracting and analyzing essential features. In some methods, this process is viewed as two individual aspects in series. For example, Su et al.,[30] Gong et al.,[31] and Gong et al.[33] developed methods that improved the robustness of the features and effectively analyzed them. These approaches, however, involve some complicated networks in both aspects and are mainly applied to unsupervised change detection. Therefore, for supervised change detection, it is necessary to develop an end-to-end network that displays the final result directly from the original inputs.

According to the summaries by Wiratama and Sim,[55] related supervised studies are divided into two categories: the front-end differential network (FDN) and the back-end differential network (BDN). In the FDN, the CI is generated to represent low-level features, and a network is used to analyze it. In the BDN, two Siamese networks are used to extract the respective high-level features of two images, and the final map are generated by computing their corresponding distance. Both categories consist of a rather complicated network and an affiliated simple method. If considered as a whole, these frameworks, whether FDN or BDN, are end-to-end in both training and testing. Several networks have been proposed and adopted to achieve end-to-end semantic change detection, and the FCN as well as some of its modified versions are introduced here.

2.2 *FCN*

FCN is a basic semantic classification tool proposed by Shelhamer et al.[40] A traditional CNN architecture mainly consists of convolutional layers and pooling layers, and the rectified linear unit (ReLU) usually serves as the activation function (AF) and is defined as $\mathrm{ReLU}(t) = \max\{t, 0\}$. At the back-end, the FC layers are applied, mapping the feature data into a vector with a given size. Many networks have been designed for classifying and identifying the inputs by outputting a vector that denotes the memberships (probability) to all possible classes, achieving the image-level classification. Different from traditional CNN architectures, FCN is mainly designed to achieve pixel-level classification for an input image with arbitrary size by showing the classification result of each pixel. The comparison of their typical architectures is shown in Fig. 1.

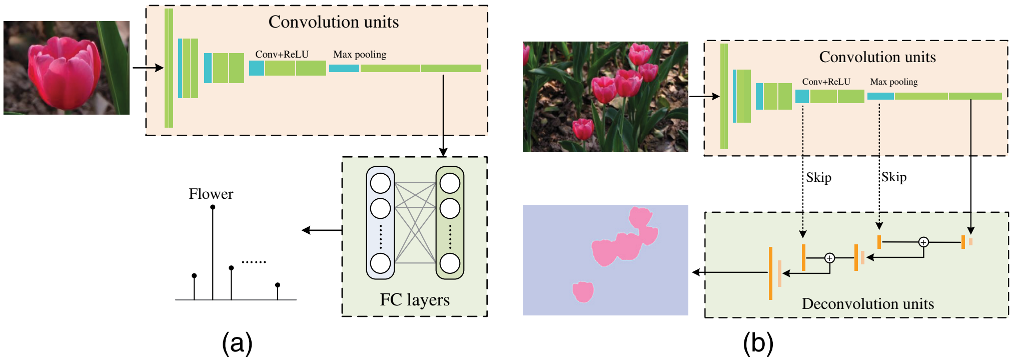


Fig. 1 Basic architectures of (a) CNN and (b) FCN.

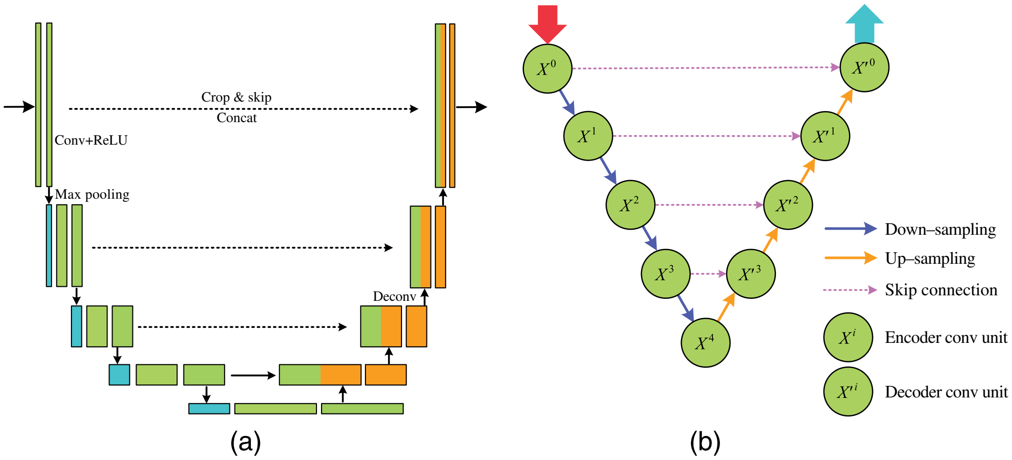


Fig. 2 Basic architectures of UNet: (a) one typical detailed UNet architecture and (b) backbone of UNet.

2.3 *UNet*

UNet, a special form of FCN, is a symmetric encoder–decoder structure. The encoder part consists of several convolutional units, and the decoder part is symmetric with respect to the encoder in the structure. A typical architecture of UNet is shown in Fig. 2(a) with its abstract backbone shown in Fig. 2(b).

There are two main distinctions between a normal FCN and UNet. First, the upsampling process (decoder) in FCN merely involves deconvolution, whereas the decoder part of UNet also involves several convolution layers to deepen the network, forming a symmetric architecture. Second, the skip connections are a summation in FCN and a concatenation in UNet. These are also considered to be the main advantages of UNet over FCN.

Despite these advantages, some problems can still be found. First, UNet is not very flexible because there is only one end-to-end pathway. Second, the skip connection is quite simple, and the node at the upsampling path only receives the skip connection from that the same level. Therefore, based on UNet, the UNet++ architecture is developed, and in this paper we propose the ZUNet++ architecture to further improve the performance of the network.

3 Methodology

In this section, we first introduce the UNet++ architecture followed by our proposed ZUNet++. Then we show the way to apply ZUNet++ to MS image change detection. Finally, the establishment of the loss function is discussed.

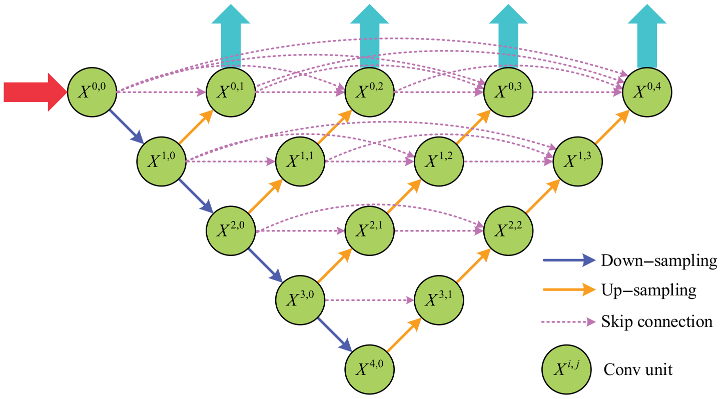


Fig. 3 Basic architecture of UNet++.

3.1 *Nested UNet: UNet++*

UNet++, as introduced in Sec. 1, is the nested version of UNet. A typical UNet++ architecture consists of convolution units, downsampling and upsampling modules, and, above all, skip connections. Figure 3 shows the main architecture used in Ref. 54.

In Fig. 3, $X^{i,j}$ is the convolution unit, where i denotes the number of the downsampling level and j denotes the number of the convolutional layer along the skip connection path. $X^{0,0}$ is the initial node with the original image input. The relationship illustrated in Fig. 3 can also be demonstrated mathematically in the following equation:

$$x^{i,j} = \begin{cases} \mathcal{F}(\mathcal{D}(x^{i-1,j})), & j = 0 \\ \mathcal{F}(\mathcal{C}(x^{i,0}, \dots, x^{i,j-1}, \mathcal{U}(x^{i+1,j-1}))), & j \geq 1 \end{cases}, \tag{1}$$

where $x^{i,j}$ is the output of $X^{i,j}$ and $\mathcal{F}(\cdot)$ denotes the convolution operation followed by an AF (e.g., ReLU). $\mathcal{C}(\cdot)$, $\mathcal{D}(\cdot)$, and $\mathcal{U}(\cdot)$ denote the concatenation, the downsampling, and the upsampling operations, respectively. Actually, when $j = 0$, $X^{i,j}$ is at the downsampling path; thus it is directly related to its direct lower-level node. When $j > 0$, $X^{i,j}$ is at several upsampling paths and is not only related to its direct higher-level node but also is skip-connected by the previous nodes at the same level, similar to DenseNet.[56] This rearranged skip connection gives its remarkable characteristics and is the main difference between UNet and UNet++.[54] Let us take node $X^{0,4}$ in Fig. 3 as an example and make a comparison of the skip connection to the corresponding node in Fig. 2(b), i.e., X'^{0}. The skip connection applied to X'^{0} is only from X^{0} in the UNet architecture, whereas $X^{0,4}$ receives skip connections from $X^{0,0}$, $X^{0,1}$, $X^{0,2}$, and $X^{0,3}$ in the UNet++ architecture. In addition, UNet++ contains several output nodes ($X^{0,1}$, $X^{0,2}$, $X^{0,3}$, and $X^{0,4}$ in Fig. 3), and these outputs can be either fused to generate the final output or viewed as an individuals, enhancing the flexibility compared with UNet.

In general, UNet++ performs better than UNet because of its excellent skip connection strategy. Nevertheless, these skip connections are between two nodes at the same level without sufficiently utilizing the information provided by the other levels. Therefore, ZUNet++ is proposed to cope with this issue and to achieve a more robust end-to-end semantic change detection.

3.2 *Backbone of ZUNet++*

In the UNet++ architecture, when $j > 0$, the node $X^{i,j}$ is at an upsampling pathway and only receives skip connections from $X^{i,0}, \dots, X^{i,j-1}$. These nodes share the same value of i, i.e., they are at the same level. To fully utilize the information provided by the adjacent nodes at both the same level and different levels, we propose the ZUNet++ architecture shown in Fig. 4.

A comparison between ZUNet++ and UNet++ shows that the skip connections between $X^{0,j}$ and $X^{0,j+1}$ stay the same for the two networks. Nevertheless, there is no skip connection between $X^{i,j}$ and $X^{i,j+1}$ ($i > 0$), which are two adjacent nodes at the same level. Instead, it is modified into

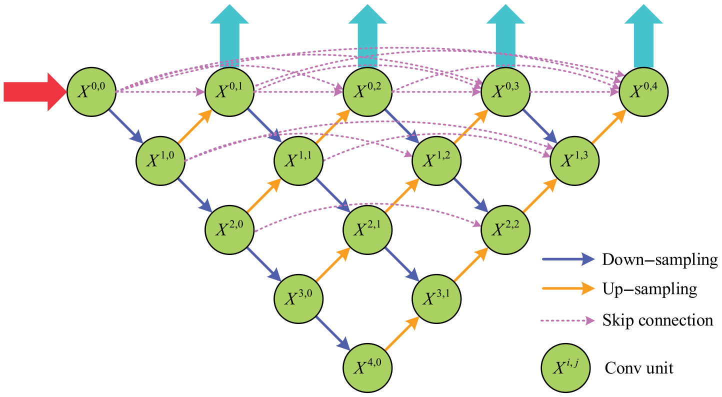


Fig. 4 Proposed ZUNet++ architecture.

a zigzag propagation as $X^{i,j} \to X^{i-1,j+1} \to X^{i,j+1}$. This process involves up and downsampling operations, which means that $X^{i,j+1}$ also receives information from the upper level. In addition, despite the modification of such a connection, the skip connection still exists for $X^{i,j}$ and $X^{i,j+k}$ $(k > 1)$. In this way, $x^{i,j}$ in ZUNet++ can be expressed as

$$x^{i,j} = \begin{cases} \mathcal{F}(\mathcal{C}(x^{i,0},\dots,x^{i,j-1},\mathcal{U}(x^{i+1,j-1}))), & i=0,\ j>0 \\ \mathcal{F}(\mathcal{D}(x^{i-1,j})), & i>0,\ j=0 \\ \mathcal{F}(\mathcal{C}(\mathcal{U}(x^{i+1,j-1}),\mathcal{D}(x^{i-1,j}))), & i>0,\ j=1 \\ \mathcal{F}(\mathcal{C}(x^{i,0},\dots,x^{i,j-2},\mathcal{U}(x^{i+1,j-1}),\mathcal{D}(x^{i-1,j}))), & i>0,\ j>1 \end{cases}. \tag{2}$$

Let us, again, take node $X^{0,4}$ for example and make a comparison with UNet++. In UNet++ shown in Fig. 3, only one basic path from $X^{0,0}$ to $X^{0,4}$ can be found: $X^{0,0} \to X^{1,0} \to X^{2,0} \to X^{3,0} \to X^{4,0} \to X^{3,1} \to X^{2,2} \to X^{1,3} \to X^{0,4}$, i.e., the data are restricted within this path; thus it is quite difficult to learn more robust features. In ZUNet++, however, there are many paths available to $X^{0,4}$. Take three feasible paths P_1, P_2, and P_3 for examples:

$$P_1\colon\ X^{0,0}\overset{\mathcal{D}}{\to}X^{1,0}\overset{\mathcal{D}}{\to}X^{2,0}\overset{\mathcal{D}}{\to}X^{3,0}\overset{\mathcal{D}}{\to}X^{4,0}\overset{\mathcal{U}}{\to}X^{3,1}\overset{\mathcal{U}}{\to}X^{2,2}\overset{\mathcal{U}}{\to}X^{1,3}\overset{\mathcal{U}}{\to}X^{0,4},$$

$$P_2\colon\ X^{0,0}\overset{\mathcal{D}}{\to}X^{1,0}\overset{\mathcal{U}}{\to}X^{0,1}\overset{\mathcal{D}}{\to}X^{1,1}\overset{\mathcal{U}}{\to}X^{0,2}\overset{\mathcal{D}}{\to}X^{1,2}\overset{\mathcal{U}}{\to}X^{0,3}\overset{\mathcal{D}}{\to}X^{1,3}\overset{\mathcal{U}}{\to}X^{0,4},$$

$$P_3\colon\ X^{0,0}\overset{\mathcal{U}}{\to}X^{1,0}\overset{\mathcal{D}}{\to}X^{0,1}\overset{\mathcal{D}}{\to}X^{1,1}\overset{\mathcal{U}}{\to}X^{2,1}\overset{\mathcal{D}}{\to}X^{3,1}\overset{\mathcal{U}}{\to}X^{2,2}\overset{\mathcal{U}}{\to}X^{1,3}\overset{\mathcal{D}}{\to}X^{0,4},$$

where $\overset{\mathcal{D}}{\to}$ and $\overset{\mathcal{U}}{\to}$ denote the downsampling and upsampling propagations, respectively. P_1 is exactly the same path as that in UNet++, and it contains sequential downsamplings followed by sequential upsamplings. P_2 is arranged in a zigzag manner with alternate downsampling and upsampling. P_3 is the mixture of the two cases. Many more similar paths can be found, and these paths actually involve many more nodes than those in a single P_1. Thus deep robust features are further extracted and utilized in both training and testing processes, facilitating the fusion of multiscale features from the nodes at different levels.

3.3 *Basic Convolution Unit in ZUNet++*

Here an introduction to the inner structure of the basic unit $X^{i,j}$ in ZUNet++ is made. Based on Ref. 54, the basic convolution unit of ZUNet++ is designed as shown in Fig. 5.

Here “Conv2D” means the 2D convolution layer, and here we use 3×3 filters with both padding and stride set as 1. “BN” and “AF” are short for batch normalization and activation function, respectively. Here a discussion is made on the BN layer and AF.

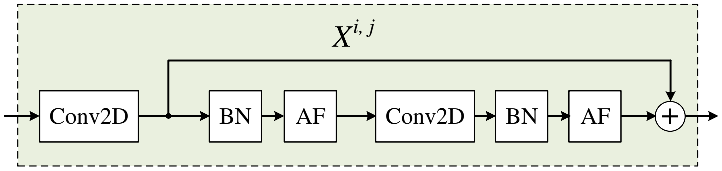


Fig. 5 Illustration of the basic convolution unit.

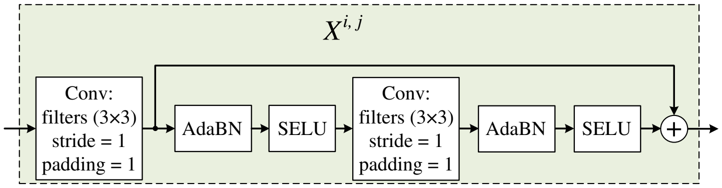


Fig. 6 Detailed illustration of the basic convolution unit used in the ZUNet++ architecture.

BN is first proposed by Ioffe and Szegedy.[57] Li et al.[58] made a further study on BN and then proposed the adaptive BN (AdaBN) for practical domain adaptation to reduce the influence by domain shift. Remote sensing images involve the difference of distribution between various datasets, and Saha et al.[59] demonstrated the effectiveness of AdaBN when dealing with the change detection problem. Hence, we employ the AdaBN to make the network more adaptive here.

As for the AF, it is usual to use the ReLU function. Nevertheless, there are some other types of AFs, and the scaled exponential linear unit (SELU) is one of these. SELU is defined as

$$\mathrm{SELU}(t) = \begin{cases} \rho t, & t \geq 0 \\ \rho\alpha(\mathrm{e}^t - 1), & t < 0 \end{cases}, \tag{3}$$

where $\rho \approx 1.051$ and $\alpha \approx 1.673$, as given in Ref. 60.

According to the theoretical analysis and practical experiments in Ref. 61, SELU has two quite conspicuous advantages over ReLU. One is its speed of convergence regardless of the choice of hyperparameter when most optimization algorithms are adopted, and the other is its ability to use a big and deep network and gain a better test accuracy.[61] Hence, in the basic unit of ZUNet++, the SELU is used as the AF.

Based on the discussion above, the final detailed form of $X^{i,j}$ in ZUNet++ is shown in Fig. 6.

3.4 *Framework of MS Image Change Detection Using ZUNet++*

Each testing MS image that we use consist of four spectral channels: red (R), green (G), blue (B), and near-infrared (NIR). However, few training datasets have exactly the same spectral property as the testing ones, i.e., it is difficult to find abundant RGB-NIR image pairs that can serve as the training data here. At the same time, we have yet to find a group of RGB data with 10,000 image pairs obtained by Google Earth (DigitalGlobe). To tackle this disparity problem, we utilize these RGB data as the training sets and do not directly input the four-spectral MS images in the testing process. Instead, one MS image I is first decomposed to generate four pseudo-RGB images $I^{(1)}$, $I^{(2)}$, $I^{(3)}$, and $I^{(4)}$ by combining every three spectral channels. Upon the decomposition of an RGB-NIR image into four three-channel images, we also notice that one of them is a real RGB image and the other three are pseudo-RGB images. Each pseudo-one involves an NIR channel along with two from the RGB channels. Therefore, this operation not only allows the testing data to adapt the trained network but also fully utilizes the spectral information. This spectral recombination strategy is depicted in Fig. 7 intuitively.

Change detection involves two images, and therefore four groups of pseudo-RGB image pairs are generated. Let $I_1^{(k)}$ and $I_2^{(k)}$ be two corresponding pseudo-RGB images ($k = 1,2,3,4$), and they are concatenated to generate $I^{(k)}$, which serves as the direct input of

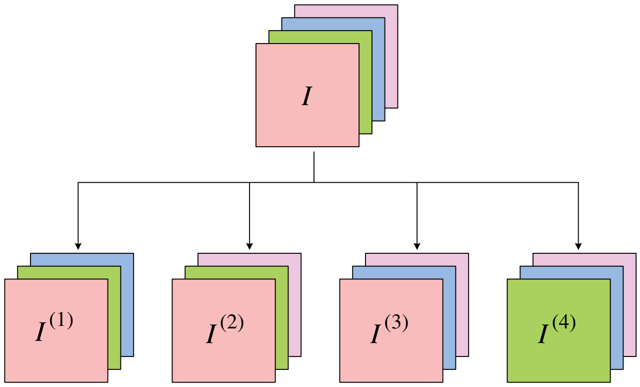


Fig. 7 Decomposition of one four-spectral MS image into four pseudo-RGB images.

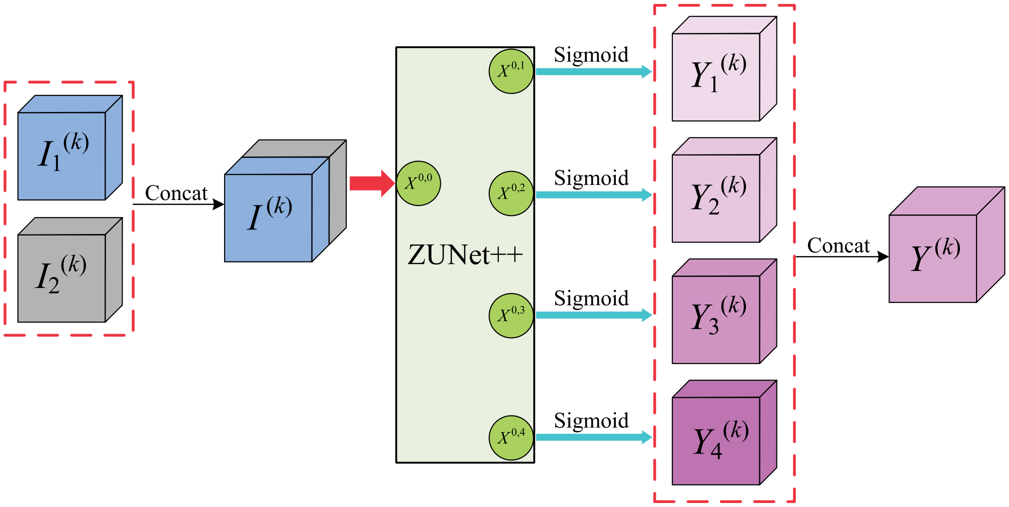


Fig. 8 Illustration of the generation of the fusion feature map $Y^{(k)}$.

ZUNet++. The network has four individual output nodes: $X^{0,1}$, $X^{0,2}$, $X^{0,3}$, and $X^{0,4}$, and a sigmoid layer is followed to obtain the corresponding output results: $Y_1^{(k)}$, $Y_2^{(k)}$, $Y_3^{(k)}$, and $Y_4^{(k)}$, which is similar to the operation in Ref. 54. Then they are concatenated to generate the fusion feature map $Y^{(k)}$:

$$Y^{(i)} = \mathcal{C}(Y_1^{(i)}, Y_2^{(i)}, Y_3^{(i)}, Y_4^{(i)}). \tag{4}$$

This process is illustrated in Fig. 8.

Upon generating four fusion feature maps $Y^{(1)}$, $Y^{(2)}$, $Y^{(3)}$, and $Y^{(4)}$ that correspond to four channel combinations shown in Fig. 7, we concatenate them to further generate the integrated feature map Y, which involves the entire spectral information and reflects the deep feature:

$$Y = \mathcal{C}(Y^{(1)}, Y^{(2)}, Y^{(3)}, Y^{(4)}). \tag{5}$$

Finally, the fuzzy C-means clustering algorithm is utilized to generate the final change detection map I_F. This is illustrated in Fig. 9.

3.5 *Loss Function*

In Figs. 4 and 8, four output nodes from ZUNet++ are determined: $X^{0,1}$, $X^{0,2}$, $X^{0,3}$, and $X^{0,4}$. Based on Ref. 52, the overall loss function L is defined as

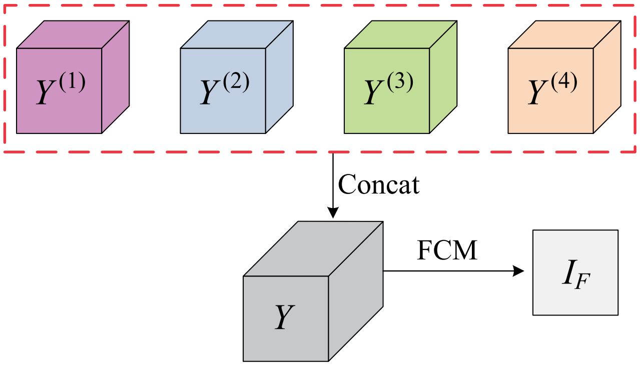


Fig. 9 Illustration of the generation of the final change detection map.

$$L = \frac{1}{4}\sum_{k=1}^{4} L^k, \tag{6}$$

where L^k represents the loss function corresponding to the output node $X^{0,k}$ and L^k is defined as

$$L^k = L^k_{\text{bce}} + \lambda L^k_{\text{dice}}, \tag{7}$$

where L^k_{bce} and L^k_{dice} denote the binary cross entropy and the dice coefficient loss, respectively. Obviously, L^k is the weighted summation of the two terms with the weight λ, the value of which is tested in the experiments. Here we omit the superscript k if no confusion is engendered.

L_{bce} is a commonly used in machine learning. Let us suppose that y is the real label array of samples and $\hat{y}$ is the corresponding estimated value. Here we have $y_m \in \{0,1\}$, $m = 1,2,\dots,N$, where N is the sample number. Thus L_{bce} is defined as

$$L_{\text{bce}} = -\frac{1}{2N}\sum_{m=1}^{N}\{y_m \log[s(\hat{y}_m)] + (1-y_m)\log[1-s(\hat{y}_m)]\}, \tag{8}$$

where $s(\cdot)$ is the sigmoid function.

The dice coefficient loss is also used in semantic segmentation and is defined as

$$L_{\text{dice}} = 1 - \frac{2|y \cap \hat{y}|}{|y| + |\hat{y}|}, \tag{9}$$

where

$$\begin{cases} |y| = \sum_{m=1}^{N} y_m \\ |\hat{y}| = \sum_{m=1}^{N} \hat{y}_m \\ |y \cap \hat{y}| = \sum_{m=1}^{N} \hat{y}_m y_m \end{cases}. \tag{10}$$

In practice, Eq. (9) is usually modified as Eq. (11).

$$L_{\text{dice}} = 1 - \frac{2|y \cap \hat{y}| + \delta}{|y| + |\hat{y}| + \delta}, \tag{11}$$

where δ is a small positive number. This operation gives the case in which the denominator equals 0, which avoids overfitting to some extent.

4 Testing Datasets and Experimental Settings

This section shows the testing datasets and experimental settings as well as the introduction to some evaluation metrics.

4.1 *Testing Datasets*

In the experiments, three four-channel MS datasets that reflect the changes in Xi'an City by the WorldView-2 satellite are used to test the effectiveness. Each of these three involves one image taken in August 2013 and one taken in August 2015 as well as a reference changed map that is obtained through on-the-spot investigation. Note that, although the datasets are from the same satellite, two different sensors were used to generate the two images in every dataset, which makes it a little difficult to detect real changes.

The first dataset is the factory dataset, which shows the construction of an automobile factory at Huyi District near the Qingling Mountains (Fig. 10). This place was a farmland before 2012, and the factory began and finished its construction in 2013 and 2015, respectively. The size of either image is $244 \times 257 \times 4$.

The second dataset is the Jinghe dataset and reflects the changes in the south side of the Jinghe River with a size of $174 \times 161 \times 4$ (Fig. 11). These land changes include the emergence of a roundabout and several foundations built later for modern buildings.

The third dataset is the park dataset and shows the construction of a new park in the Xixian New Area with a size of $156 \times 145 \times 4$ (Fig. 12). The government began its construction in 2014 and was still under construction in 2015. An obvious change can be seen from the two images.

4.2 *Experimental Settings*

First, we make a test on the parameter λ, which is the weight in Eq. (6). Here λ is set to 0, 0.25, 0.5, 0.75, and 1, and the optimal value can be selected according to the performance of the results.

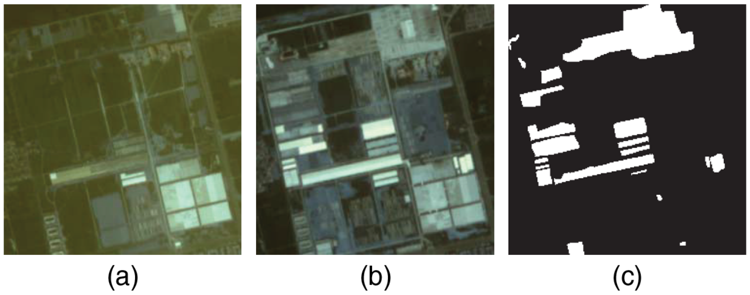


Fig. 10 Factory dataset: (a) image taken in 2013, (b) image taken in 2015, and (c) reference ground truth map.

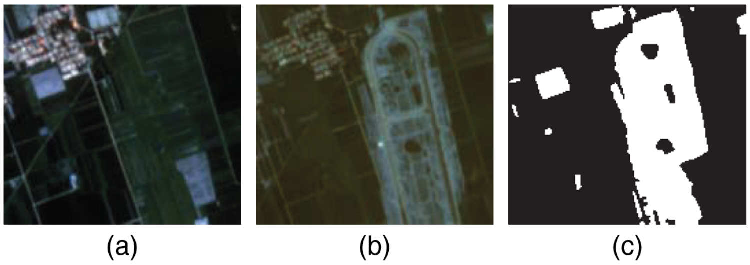


Fig. 11 Jinghe dataset: (a) image taken in 2013, (b) image taken in 2015, and (c) reference ground truth map.

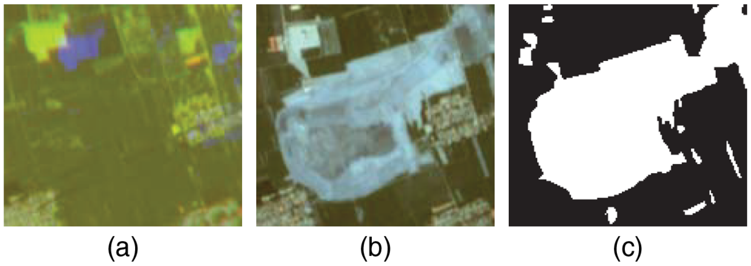


Fig. 12 Park dataset: (a) image taken in 2013, (b) image taken in 2015, and (c) reference ground truth map.

Upon determining the optimal value of λ, the experimental results from several comparison algorithms are given to show the excellent performance of ZUNet++. These comparison results are FCN, SegNet, UNet, and UNet++. Note that a true-and-false comparison is made for UNet++ and ZUNet++. It is used to demonstrate the effectiveness of concatenating the outputs from the four output nodes, which are $X^{0,1}$, $X^{0,2}$, $X^{0,3}$, and $X^{0,4}$ in Fig. 2. A "true" result is generated from the concatenation of the four output nodes, whereas a "false" one is only from $X^{0,4}$.

4.3 *Evaluation Metrics*

As a binary classification problem, a 2×2 classification matrix $\mathbf{P}$ is adopted to describe the performance as illustrated in the following equation:

$$\mathbf{P} = \frac{1}{A \times B} \begin{bmatrix} \mathrm{TN} & \mathrm{FP} \\ \mathrm{FN} & \mathrm{TP} \end{bmatrix}, \tag{12}$$

where TP, FP, TN, and FN denote the number of true positives, false positives, true negatives, and false negatives, respectively. Three evaluation metrics, the precision value (Pr), the recall value (Re), and the $F1$-score ($F1$), are employed to describe the performance of detecting changes. Pr and Re are calculated as shown in the following equation:

$$\begin{cases} \mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \\ \mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \end{cases}. \tag{13}$$

And $F1$ is defined as their harmonic mean value:

$$F1 = \frac{2\ \mathrm{Pr} \cdot \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}}. \tag{14}$$

These three metrics emphasize the performance of the detection of changes, and generally higher values correspond to a better detection result. To illustrate the overall performance of classification, the percentage correct classification (PCC) and the kappa coefficient (KC) are used. PCC is calculated as

$$\mathrm{PCC} = \frac{\mathrm{TP} + \mathrm{TN}}{A \times B} = \mathrm{tr}(\mathbf{P}). \tag{15}$$

KC is calculated as

$$\mathrm{KC} = \frac{\mathrm{tr}(\mathbf{P}) - \mathbf{q}^{\mathrm{T}}\mathbf{P}^2\mathbf{q}}{1 - \mathbf{q}^{\mathrm{T}}\mathbf{P}^2\mathbf{q}}, \tag{16}$$

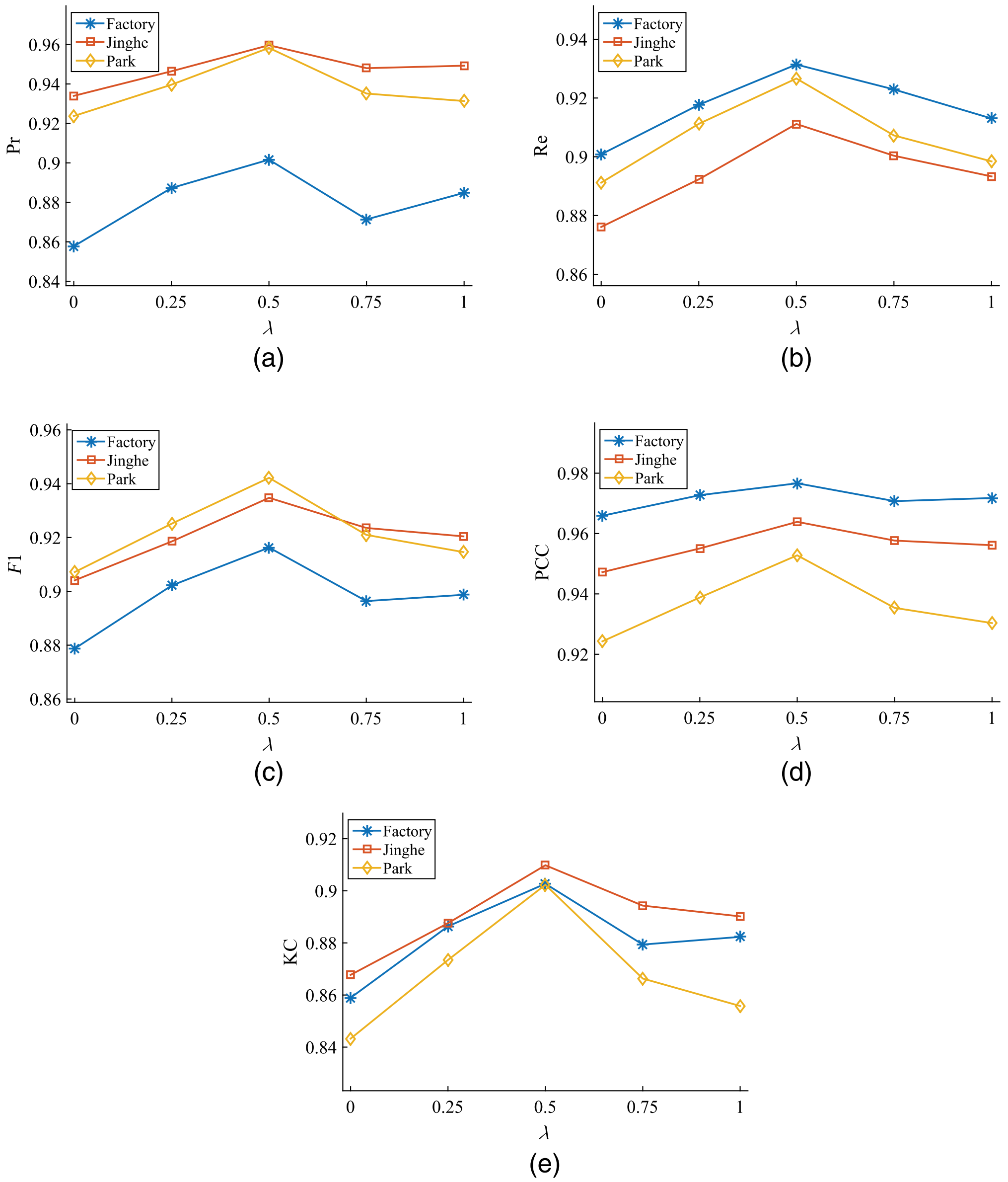
where $\mathbf{q} = [1,1]^{\mathrm{T}}$. The derivation of Eq. (16) can be found in Refs. 30 and 62.

5 Experimental Results

5.1 *Parameter Testing*

This experiment determines the optimal value of the parameter λ, which varies from 0 to 1 at intervals of 0.25. Figure 13 illustrates the effect of λ on the accuracy of ZUNet++.

It can be seen that, when $\lambda = 0.5$, quantitative evaluation results achieve their maximum. Other values of λ may lead to results that are not so good or stable. In particular, when $\lambda = 0$ (the case in which the loss function only involves L_{bce}), all five criterion values indicate low values, and this suggests the indispensability of L_{dice}. A comparison between the cases with $\lambda = 0.25$ and $\lambda = 0.5$ indicates that L_{dice} should not account for such a small percentage. On the other hand, a further increase in the value of λ (0.75 and 1 here) also leads to downward trends of the criterion values. Therefore, λ can be set to 0.5 to balance the two parts better in the following experiments.



(a)

(b)

(c)

(d)

(e)

Fig. 13 Effect of λ on (a) Pr, (b) Re, (c) $F1$, (d) PCC, and (e) KC.

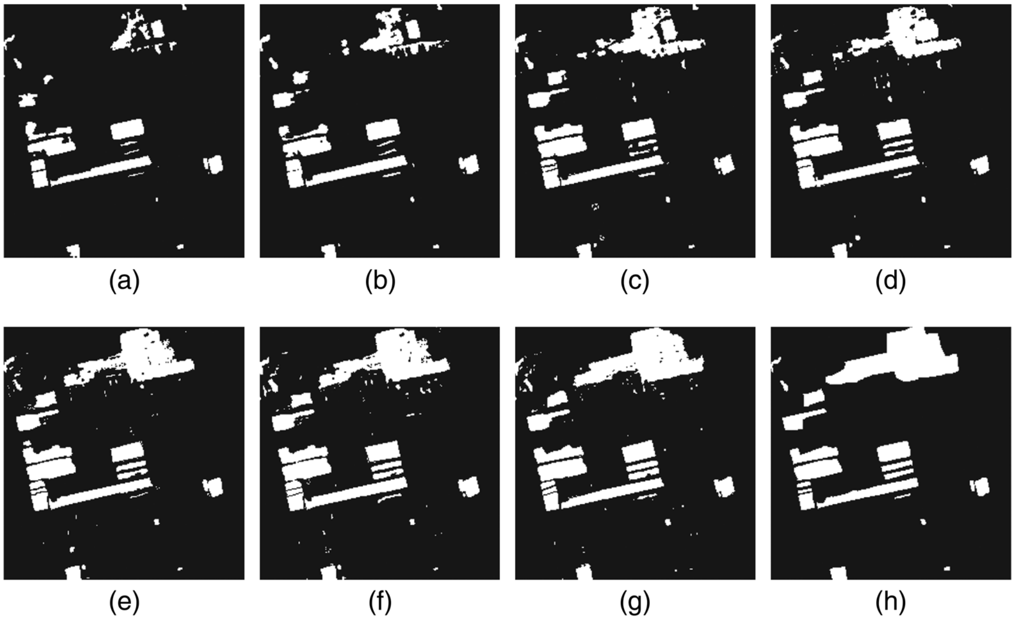


Fig. 14 Results from the factory dataset by (a) FCN, (b) SegNet, (c) UNet, (d) false UNet++, (e) true UNet++, (f) false ZUNet++, and (g) true ZUNet++. (h) the reference map.

5.2 *Results from the Factory Dataset*

The final maps from the factory dataset by different approaches are shown in Fig. 14, and the corresponding quantitative results are listed in Table 1.

From Fig. 14, all of the approaches are capable of detecting some obvious changes. Therefore, we focus more on detecting some subtle changes. In general, two nested networks, UNet++ and ZUNet++, show better detection results than the other networks in terms of the changes in the north, which is subtle and not so obvious. In the true-and-false comparison experiment, as anticipated, the true results perform better than the corresponding false results, demonstrating the effectiveness of concatenating the four outputs in the network. The advantages of ZUNet++ over UNet++ can be seen by making a comparison of the metric values in terms of their true experiments in Table 1. The values of $F1$, PCC, and KC from ZUNet++ reach 0.9162, 0.9766, and 0.9027, respectively, higher than those from UNet++, indicating the excellent performance of the zigzag propagation.

Table 1 Quantitative evaluation results of different approaches from the factory dataset.

	Pr	Re	$F1$	PCC	KC
FCN	0.9682	0.4787	0.6406	0.9263	0.6048
SegNet	0.9284	0.5987	0.7279	0.9386	0.6951
UNet	0.9159	0.7461	0.8223	0.9558	0.7974
UNet++_false	0.9116	0.7960	0.8499	0.9614	0.8279
UNet++_true	0.9003	0.8648	0.8822	0.9683	0.8639
ZUNet++_false	0.8690	0.8890	0.8789	0.9664	0.8593
ZUNet++_true	0.9016	0.9314	0.9162	0.9766	0.9027

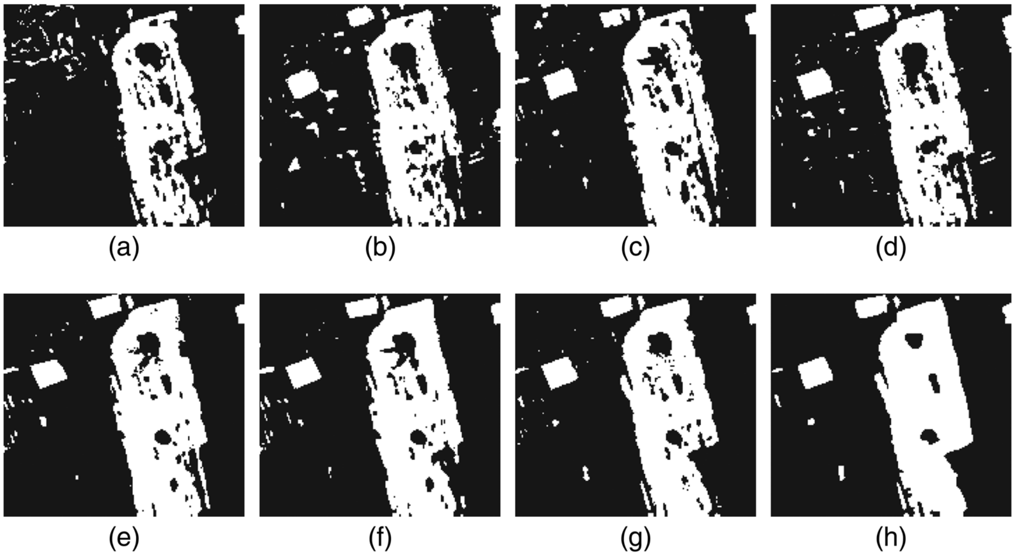


Fig. 15 Results from the Jinghe dataset by (a) FCN, (b) SegNet, (c) UNet, (d) false UNet++, (e) true UNet++, (f) false ZUNet++, and (g) true ZUNet++. (h) the reference map.

Table 2 Quantitative evaluation results of different approaches from the Jinghe dataset.

	Pr	Re	$F1$	PCC	KC
FCN	0.9058	0.7157	0.7997	0.8981	0.7326
SegNet	0.8812	0.8017	0.8396	0.9130	0.7800
UNet	0.8986	0.8448	0.8709	0.9288	0.8218
UNet++_false	0.9155	0.8415	0.8770	0.9329	0.8310
UNet++_true	0.9069	0.9099	0.9084	0.9478	0.8720
ZUNet++_false	0.9142	0.9015	0.9078	0.9480	0.8716
ZUNet++_true	0.9596	0.9112	0.9348	0.9639	0.9098

5.3 *Results from the Jinghe Dataset*

Figure 15 shows the final maps from the Jinghe dataset by different approaches, and Table 2 lists the quantitative results.

It can be seen that the obvious change in the Jinghe dataset is the constructions of the roundabout and a few building foundations around it, which is detected by all of the approaches. For the detailed detection, however, the two nested networks perform much better. This is manifested in two aspects, which are their excellent capability of edge detection and noise suppression in the heterogenous and homogenous areas, respectively. In addition, the true-and-false comparison also demonstrated the essentiality and superiority of the final concatenation operation. In addition, a comparison of the quantitative results in Table 2 shows the superiority of ZUNet++ to Unet++. Actually, such a detection contrast can also be seen visually by Figs. 15(e) and 15(g).

5.4 *Results from the Park Dataset*

The final maps from the park dataset by different approaches are shown in Fig. 16 with their corresponding quantitative results listed in Table 3.

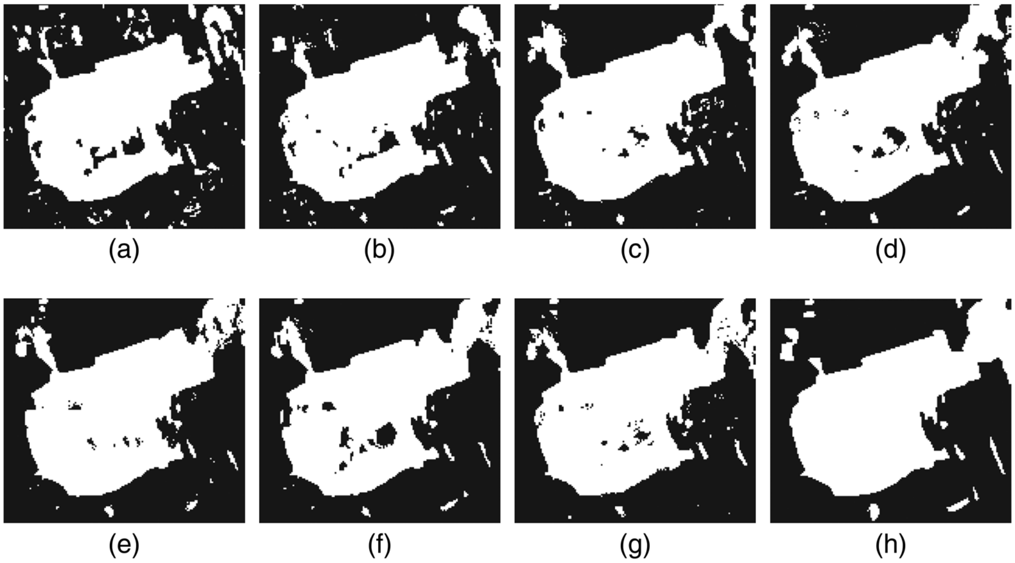


Fig. 16 Results from the park dataset by (a) FCN, (b) SegNet, (c) UNet, (d) false UNet++, (e) true UNet++, (f) false ZUNet++, and (g) true ZUNet++. (h) the reference map.

Table 3 Quantitative evaluation results of different approaches from the park dataset.

	Pr	Re	$F1$	PCC	KC
FCN	0.9142	0.8535	0.8828	0.9059	0.8044
SegNet	0.9455	0.8721	0.9073	0.9260	0.8459
UNet	0.9146	0.8948	0.9046	0.9216	0.8381
UNet++_false	0.9315	0.9024	0.9167	0.9319	0.8592
UNet++_true	0.9427	0.9210	0.9318	0.9440	0.8843
ZUNet++_false	0.9488	0.8921	0.9196	0.9352	0.8655
ZUNet++_true	0.9583	0.9266	0.9422	0.9528	0.9023

The park dataset is characterized by its large changed area, and the results from the true experiments [Figs. 16(e) and 16(g)] are closer to the reference map than the others. In addition, from Table 3, the values of $F1$, PCC, and KC by the proposed ZUNet++ network in its true experiment are 0.9422, 0.9528, and 0.9023, respectively, which are higher than those by the true UNet++ network. The results suggest that ZUNet++ also applies to such extensive changes with higher accuracy than other approaches.

6 Concluding Remarks

Change detection has facilitated urban study to a large extent by analyzing multitemporal images. In this paper, we propose the ZUNet++ framework to cope with the semantic change detection problem for MS images. Serving as a modified version of UNet++, ZUNet++ involves nodes that retain more information from the other levels, which makes the network more robust and flexible. In addition, according to the characteristics of the training and testing data, the four channels in one MS image are recombined into four pseudo-RGB images, making it possible to apply the trained network to the testing data. Experimental results from the real satellite data also demonstrates the effectiveness of the proposed ZUNet++.

In general, the contributions of the work can be summarized as follows. First, it further develops the nested network for end-to-end semantic change detection and is able to utilize more information between levels, which is robust and flexible in the training and testing process. Second, the framework is also designed to tackle the problem of characteristics disparity between training and testing data by adopting the spectral recombination strategy. These two contributions are considered to broaden the applicability of the available frameworks. Despite these advantages, some hyper-parameters in the framework are still worth studying. In addition, it is considered that the technique can also be extended to the problem of multiple change detection. Therefore, in the future, we will put more emphasis on its further study.

References

1. M. K. Ridd and J. Liu, "A comparison of four algorithms for change detection in an urban environment," *Remote Sens. Environ.* **63**(2), 95–100 (1998).
2. J. Xu et al., "Urban change detection with polarimetric advanced land observing satellite phased array type L-band synthetic aperture radar data: a case study of Tai'an, China," *J. Appl. Remote Sens.* **7**(1), 073481 (2013).
3. H. Hammer et al., "Comparison of multiple methods for detecting changes in urban areas in TerraSAR-X data," *Proc. SPIE* **10005**, 100050Y (2016).
4. M. Boldt et al., "Practical approach for synthetic aperture radar change analysis in urban environments," *J. Appl. Remote Sens.* **13**(3), 034528 (2019).
5. A. Singh, "Review article digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.* **10**(6), 989–1003 (1989).
6. Y. Bazi, L. Bruzzone, and F. Melgani, "An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images," *IEEE Trans. Geosci. Remote Sens.* **43**(4), 874–887 (2005).
7. F. Bovolo, S. Marchesi, and L. Bruzzone, "A framework for automatic and unsupervised detection of multiple changes in multitemporal images," *IEEE Trans. Geosci. Remote Sens.* **50**(6), 2196–2212 (2012).
8. M. Gong et al., "A novel coarse-to-fine scheme for automatic image registration based on SIFT and mutual information," *IEEE Trans. Geosci. Remote Sens.* **52**(7), 4328–4338 (2014).
9. P. Wang et al., "Geometric correction method to correct the influence of attitude jitter on remote sensing imagery using compressive sampling," *J. Appl. Remote Sens.* **9**(1), 095077 (2015).
10. W. B. Abdallah and R. Abdelfattah, "Two-dimensional wavelet algorithm for interferometric synthetic aperture radar phase filtering enhancement," *J. Appl. Remote Sens.* **9**(1), 096061 (2015).
11. M. Gong, Y. Cao, and Q. Wu, "A neighborhood-based ratio approach for change detection in SAR images," *IEEE Geosci. Remote Sens. Lett.* **9**(2), 307–311 (2012).
12. S. Liu et al., "Unsupervised multitemporal spectral unmixing for detecting multiple changes in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.* **54**(5), 2733–2748 (2016).
13. G. Moser and S. B. Serpico, "Generalized minimum-error thresholding for unsupervised change detection from SAR amplitude imagery," *IEEE Trans. Geosci. Remote Sens.* **44**(10), 2972–2982 (2006).
14. L. Su et al., "Unsupervised change detection in SAR images based on locally fitting model and semi-EM algorithm," *Int. J. Remote Sens.* **35**(2), 621–650 (2014).
15. M. Gong, Z. Zhou, and J. Ma, "Change detection in synthetic aperture radar images based on image fusion and fuzzy clustering," *IEEE Trans. Image Process.* **21**(4), 2141–2151 (2012).
16. M. Gong et al., "Fuzzy clustering with a modified MRF energy function for change detection in synthetic aperture radar images," *IEEE Trans. Fuzzy Syst.* **22**(1), 98–109 (2014).
17. A. M. Lal and S. M. Anouncia, "Adapted sparse fusion with constrained clustering for semisupervised change detection in remotely sensed images," *J. Appl. Remote Sens.* **11**(1), 016013 (2017).

18. F. Bovolo and L. Bruzzone, “A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain,” *IEEE Trans. Geosci. Remote Sens.* **45**(1), 218–236 (2007).
19. S. Liu et al., “Hierarchical unsupervised change detection in multitemporal hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.* **53**(1), 244–260 (2015).
20. S. Liu et al., “Sequential spectral change vector analysis for iteratively discovering and detecting multiple changes in hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.* **53**(8), 4363–4378 (2015).
21. Y. Bengio, “Learning deep architectures for AI,” *Found. Trends Mach. Learn.* **2**(1), 1–127 (2009).
22. N. Lv et al., “Deep learning and superpixel feature extraction based on contractive autoencoder for change detection in SAR images,” *IEEE Trans. Ind. Inf.* **14**, 5530–5538 (2018).
23. J. Geng et al., “Saliency-guided deep neural networks for SAR image change detection,” *IEEE Trans. Geosci. Remote Sens.* **57**, 7365–7377 (2019).
24. A. B. Geva, “Hierarchical unsupervised fuzzy clustering,” *IEEE Trans. Fuzzy Syst.* **7**(6), 723–733 (1999).
25. L. Su et al., “Detecting multiple changes from multi-temporal images by using stacked denosing autoencoder based change vector analysis,” in *Int. Joint Conf. Neural Networks* (2016).
26. L. Su and X. Cao, “Fuzzy autoencoder for multiple change detection in remote sensing images,” *J. Appl. Remote Sens.* **12**(3), 035014 (2018).
27. F. Gao et al., “Change detection in SAR images based on deep semi-NMF and SVD networks,” *Remote Sens.* **9**, 435 (2017).
28. H. Zhang et al., “Feature-level change detection using deep representation and feature change analysis for multispectral imagery,” *IEEE Geosci. Remote Sens. Lett.* **13**, 1666–1670 (2016).
29. P. Zhang et al., “Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images,” *ISPRS J. Photogramm. Remote Sens.* **116**, 24–41 (2016).
30. L. Su et al., “Deep learning and mapping based ternary change detection for information unbalanced images,” *Pattern Recognit.* **66**, 213–228 (2017).
31. M. Gong, H. Yang, and P. Zhang, “Feature learning and change feature classification based on deep learning for ternary change detection in SAR images,” *ISPRS J. Photogramm. Remote Sens.* **129**, 212–225 (2017).
32. I. Goodfellow et al., “Generative adversarial nets,” in *Adv. Neural Inf. Process. Syst.*, pp. 2672–2680 (2014).
33. M. Gong et al., “A generative discriminatory classified network for change detection in multispectral imagery,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **12**, 321–333 (2019).
34. S. Saha, F. Bovolo, and L. Bruzzone, “Unsupervised deep change vector analysis for multiple-change detection in VHR images,” *IEEE Trans. Geosci. Remote Sens.* **57**(6), 3677–3693 (2019).
35. Q. Wang et al., “GETNET: a general end-to-end 2-D CNN framework for hyperspectral image change detection,” *IEEE Trans. Geosci. Remote Sens.* **57**, 3–13 (2018).
36. A. M. E. Amin, Q. Liu, and Y. Wang, “Convolutional neural network features based change detection in satellite images,” *Proc. SPIE* **10011**, 100110W (2016).
37. Y. Zhan et al., “Change detection based on deep Siamese convolutional network for optical aerial images,” *IEEE Geosci. Remote Sens. Lett.* **14**(10), 1845–1849 (2017).
38. L. Mou, L. Bruzzone, and X. X. Zhu, “Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery,” *IEEE Trans. Geosci. Remote Sens.* 1–12 (2018).
39. S. Saha et al., “Unsupervised deep joint segmentation of multitemporal high-resolution images,” *IEEE Trans. Geosci. Remote Sens.* **58**(12), 8780–8792 (2020).
40. E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 640–651 (2017).

41. W. Wiratama et al., “Dual-dense convolution network for change detection of high-resolution panchromatic imagery,” *Appl. Sci.* **8**, 1785 (2018).
42. C. Zhang et al., “Detecting large-scale urban land cover changes from very high resolution remote sensing images using CNN-based classification,” *ISPRS Int. J. Geo-Inf.* **8**, 189 (2019).
43. R. C. Daudt et al., “Multitask learning for large-scale semantic change detection,” *Comput. Vision Image Understanding* **187**, 102783 (2019).
44. V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: a deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(12), 2481–2495 (2017).
45. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: convolutional networks for biomedical image segmentation,” *Lect. Notes Comput. Sci.* **9351**, 234–241 (2015).
46. G. Lin et al., “RefineNet: multi-path refinement networks for high-resolution semantic segmentation,” in *IEEE Conf. Comput. Vision and Pattern Recognit.* (2017).
47. L. Khelifi and M. Mignotte, “Deep learning for change detection in remote sensing images: comprehensive review and meta-analysis,” *IEEE Access* **8**, 126385–126400 (2020).
48. X. Li et al., “H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from ct volumes,” *IEEE Trans. Med. Imaging* **37**(12), 2663–2674 (2018).
49. R. Jaturapitpornchai et al., “Newly built construction detection in SAR images,” *Remote Sens.* **11**, 1444 (2019).
50. L. Li et al., “Urban building change detection in SAR images using combined differential image and residual U-Net network,” *Remote Sens.* **11**, 1091 (2019).
51. Z. Hamdi, M. Brandmeier, and C. Straub, “Forest damage assessment using deep learning on high resolution remote sensing data,” *Remote Sens.* **11**, 1976 (2019).
52. Z. Zhou et al., “UNet++: a nested U-Net architecture for medical image segmentation,” *Lect. Notes Comput. Sci.* **11045**, 3–11 (2018).
53. E. Alexakis and C. Armenakis, “Evaluation of UNet and UNet++ architectures in high resolution image change detection applications,” *ISPRS Int. Arch. Photogramm. Remote Sens. and Spatial Inf. Sci.* **XLIII-B3-2020**, 1507–1514 (2020).
54. D. F. Peng, Y. J. Zhang, and H. Y. Guan, “End-to-end change detection for high resolution satellite images using improved UNet++,” *Remote Sens.* **11**, 1382 (2019).
55. W. Wiratama and D. Sim, “Fusion network for change detection of high-resolution panchromatic imagery,” *Appl. Sci.* **9**, 1441 (2019).
56. G. Huang et al., “Densely connected convolutional networks,” in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)* (2017).
57. S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *Proc. 32nd Int. Conf. Mach. Learn.*, JMLR.org (2015).
58. Y. Li et al., “Adaptive batch normalization for practical domain adaptation,” *Pattern Recognit.* **80**, 109–117 (2018).
59. S. Saha, F. Bovolo, and L. Bruzzone, “Destroyed-buildings detection from VHR SAR images using deep features,” *Proc. SPIE* **10789**, 107890Z (2018).
60. G. Klambauer et al., “Self-normalizing neural networks,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, pp. 972–981, Curran Associates Inc. (2017).
61. K. Pratama and D.-K. Kang, “The effect of hyperparameter choice on ReLU and SELU activation function,” *Int. J. Adv. Smart Converg.* **6**(4), 73–79 (2017).
62. M. G. Gong, H. L. Yang, and P. Z. Zhang, “Feature learning and change feature classification based on deep learning for ternary change detection in SAR images,” *ISPRS J. Photogramm. Remote Sens.* **129**, 212–225 (2017).

Linzhi Su received his BS degree in artificial intelligence from Xidian University in 2011 and his PhD in electronic circuit and systems from Xidian University in 2016. He is an assistant professor at Northwest University. His current research interests include image processing and machine learning.

Qiaoyun Xie received her BS degree in software engineering from Shanxi University in 2018. She is a postgraduate student. Her current research interests include remote sensing and image processing.

Fengjun Zhao received his BS degree in detection guidance and control engineering from Xidian University in 2010 and his PhD in signal and information processing from Xidian University in 2015. He is an associate professor at Northwest University. His current research interests include deep neural networks and image processing.

Xin Cao received his BS degree in detection guidance and control engineering from Xidian University in 2011 and his PhD in electronic circuit and systems from Xidian University in 2016. He is an associate professor at Northwest University. His current research interests include artificial intelligence systems, machine learning, and image processing.