# AGCDetNet: An Attention-Guided Network for Building Change Detection in High-Resolution Remote Sensing Images

Kaiqiang Song and Jie Jiang

***Abstract*—While deep learning-based methods have gained considerable improvements in remote sensing (RS) image change detection (CD), scale variations and pseudochanges hinder most supervised methods' performance. The CD networks derived from other fields can be fronted with false alarms and miss detections in high-resolution RS images due to the weak feature representation ability. In this article, an attention-guided end-to-end change detection network (AGCDetNet) is proposed based on the fully convolutional network and attention mechanism. AGCDetNet learns to enhance the feature representation of change information and achieve accuracy improvements using spatial attention and channel attention. A spatial attention module (SPAM) promotes the discrimination between the changed objects and the background by adding the learned spatial attention to the deep features. Channelwise attention-guided interference filtering unit (CIFU)/atrous spatial pyramid pooling (CG-ASPP) module enhances the representation of multilevel features and multiscale context, respectively. Extensive experiments have been conducted on several public datasets for performance evaluation, including LEVIR-CD, WHU, Season-Varying, WV2, and ZY3. Experiment results demonstrate that AGCDetNet outperforms several state-of-the-art methods of accuracy and robustness. Specifically, AGCDetNet achieves the best F1-score on two datasets, i.e., LEVIR-CD (0.9076) and Season-Varying (0.9654).**

***Index Terms*—Attention mechanism, end-to-end change detection (CD), fully convolutional network (FCN), high resolution, remote sensing (RS) images.**

## I. Introduction

CHANGE detection (CD) aims to identify and locate changes of the objects of interest from multitemporal remote sensing (RS) images acquired over the same geographical region at different times [1]. As a particular CD task, building CD [2], [3] focuses on identifying the changed buildings from RS images. CD methods usually generate a binary change map, in which pixels are classified into the foreground changed objects or the background. Buildings are the most representative human-made structures among numerous geospatial objects. Building CD has a wide range of applications, such as urbanization monitoring, illegal building identification, construction land supervision, and damage assessment [4]–[7]. Automatic building CD plays a vital role in the field of RS image analysis. It also prevents the time-consuming and labor-intensive manual interpretation work with the increasing number of RS images.

With the rapid development of RS techniques during the last few decades, numerous high-resolution and very high resolution aerial or satellite images can be acquired, e.g., the most representative optical images [8], [9]. The detailed structures and change information of geospatial objects can be observed and analyzed. Moreover, changed building instances with accurate boundaries can be identified and located by a promising CD method. However, building CD can be confronted with many challenges, such as scale variations, class imbalance, and pseudochanges in the high-resolution RS imagery [6], [10]. Buildings vary in scale and appearance in different regions or countries. The multiscale problem makes it difficult to locate and recognize the changed objects of various scales. Building CD in urban areas can be more challenging than that in rural areas due to other geospatial objects' interference. For example, vehicles and containers can be easily incorrectly detected as buildings even through manual interpretation [11]. Besides, the masks of crowded building groups are easy to stick together. Class imbalance is a common problem, which means the number of changed pixels is far less than that of the unchanged pixels. Models built on an imbalanced dataset tend to predict pixels as unchanged pixels even if they changed [10]. Most importantly, CD can be fronted with false alarms and missed detections due to the pseudochanges caused by geometric registration errors and spectral differences between bitemporal images. In general, bitemporal RS images have been homogenized by geometric and radiometric correction. However, the roofs cannot be completely geometrically aligned due to relief displacement, as shown in Fig. 1(a) and (b). The misaligned boundaries easily result in false alarms such as isolated noise in the background region. Illumination variations lead to spectral changes of the roof and ground between two images acquired at different times, as shown in Fig. 1(c) and (d). If not treated carefully, roofs with different colors can be misclassified as changed buildings when they are actually unchanged.

Manuscript received December 30, 2020; revised March 17, 2021; accepted May 2, 2021. Date of publication May 4, 2021; date of current version May 26, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61725501, in part by the National Key Research and Development Program of China under Grant 2019YFA0706002, and in part by the High Performance Computing Resources at Beihang University *(Corresponding author: Jie Jiang.)*

The authors are with the School of Instrumentation and Optoelectronic Engineering, Key Laboratory of Precision Opto-Mechatronics Technology, Ministry of Education, Beijing Advanced Innovation Center for Big Data Based Precision Medicine, Beihang University, Beijing 100191, China (e-mail: kaiqiangsong@buaa.edu.cn; jiangjie@buaa.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2021.3077545

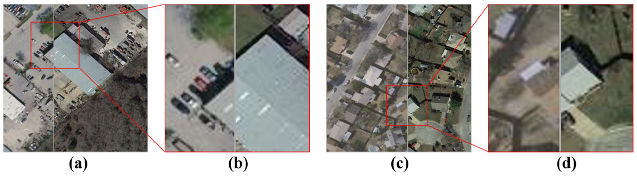


Fig. 1. Illustrations of (a), (b) geometric registration errors and (c), (d) spectral changes. We stitched the bitemporal images together and used the red boxes to highlight the pseudochanges. Zoom-in for an improved view. (b) and (d) are the local enlarged views of (a) and (c), respectively.

According to whether requiring training samples, CD methods can be categorized as either supervised or unsupervised. The unsupervised methods are appealing because they have no requirement of building suitable training sets. Most of the unsupervised methods are developed based on the difference images (DI), which indicate the magnitude of change [12]–[14]. DI is usually generated by image differencing, image ratio, or change vector analysis [13]. Most of the unsupervised methods obtain the change map by classifying DI into changed or unchanged through thresholding-based [12] or clustering-based approaches [15]. Meanwhile, some transformation methods are used to transform the raw feature vector into a new feature space, such as the principal components analysis (PCA) [16] and the iterative reweighted multivariate alteration detection (IR-MAD) [17] algorithms. For instance, Celik [15] proposed the PCA-kMeans method for pixel-based CD. It applies PCA on the nonoverlapping blocks of DI to extract feature vectors and utilizes the k-means clustering algorithm to determine whether a corresponding pixel has changed. Though attractive in applications without requiring ground truth data, DI-based methods generate many false alarms and missed detections in the detection results. With the increasing spatial resolution of RS images, these pixel-based methods exhibit worse performance and unstable against the pseudochanges. Therefore, the object-based methods are developed in high spatial resolution RS image CD [5], [18]. Compared with pixel-based methods sensitive to spectral variability, object-based methods are more effective in exploring spatial information and improving accuracy by employing the image object as the basic processing unit. However, the detection errors highly depend on the results obtained by different segmentation strategies. To alleviate the effect of the segmentation error, Lv *et al.* [19] combined the conditional random field method with the object-based technique to explore the spectral–spatial information. Alternatively, based on heuristics prior knowledge from the ground truth, supervised approaches combined with some machine learning algorithms have gained popularity and achieved promising performance [20], [21]. The postclassification and multidate classification can generate the change mask and the information of change directions simultaneously. However, the handcrafted features hinder their performance due to the limited representation of high-level semantic information in the traditional CD methods.

With the booming breakthrough of deep learning techniques, CD methods have gradually evolved from traditional [18]–[23] to deep learning-based methods [24]. Specifically, supervised CD methods based on deep neural networks (DNNs) achieve promising performance through available labeled samples. The CD networks built on convolutional neural networks (CNNs) and fully convolutional network (FCN) [25] have gained popularity. Recently, U-shaped encoder–decoder architectures, derived from some semantic segmentation networks [26], [27], have drawn considerable attention [10], [28]–[35]. The high-quality change maps generated in the decoder rely on the feature representations learned by the encoder. However, given the inherent shortage of the backpropagation (BP) algorithm, the vanishing gradient problem becomes the main concern when training a deep network. The weak discrimination of deep features becomes one of the main concerns [35]. U-shaped methods attempt to improve accuracy by concatenating feature maps from different levels through multiple skip connections [28]–[30], [34], [35]. It facilitates the gradual recovery of spatial details through consecutive upsampling in the decoder. However, they ignore that different channels of features gain response to different semantics [36], of which some negatively affect the procedure of difference discrimination. The useful features cannot be adequately mined since the context of different levels was treated equally. Despite the powerful representation ability of CNNs, these methods can be insufficient for building CD in high-resolution RS images. The problem mainly lies in the following aspects: One is the insufficient feature representations of change information from the lack of modeling of foreground changed objects; the other is the lack of interference filtering when passing context information through skip connections.

The main contributions are summarized as follows.

1) This article presents an attention-guided end-to-end network (AGCDetNet) for building CD in high-resolution RS images. AGCDetNet learns to enhance the feature representations using spatial attention and channel attention.
2) The proposed spatial attention module (SPAM) learns a spatial attention map by incorporating prior knowledge into a scaled dot-product model to estimate each pixel location's change probability in deep features. SPAM promotes discrimination between the change objects and the background by adding the learned attention map on the deep features.
3) Two channelwise attention-guided modules, i.e., channelwise attention-guided interference filtering unit (CIFU) and CG-ASPP are proposed to enhance multilevel features and multiscale context representations. Both adopt multiple multilayer perceptrons (MLP) to generate channel attention weights by sharing one hidden layer to explore semantic similarity among different features and applying private output layers to focus on their difference.

## II. Related Work

A brief review of deep learning-based CD methods is concluded in this section. Most state-of-the-art CD methods have been exploited on the basis of DNNs, such as CNNs [10], [11], [28], [29], [32]–[38]; generative adversarial networks (GANs) [34], [39]–[42]; and recurrent neural networks (RNNs) [31], [43], [44].

Transferred deep learning-based CD methods have been proposed [10], [45]–[48] due to the powerful representation ability of CNNs. Some pretrained CNNs for image classification are used as feature extractors to extract deep features from raw bitemporal images. For instance, Zhang and Shi [10] adopted the VGG16 [49] pretrained on scene-level RS images to obtain features at different depths and constructed a feature difference CNN (FDCNN) for supervised CD. However, these methods can be sensitive to the data shift among different domains [48] and suffer from underfitting. Alternatively, Ji *et al.* [11] proposed a postclassification method, which has shown advantages in building CD. The network consists of a Mask R-CNN [50]-based building extraction network and a U-shaped CD network. The pair of classification building maps generated by Mask R-CNN are fed into the CD network to obtain change maps. However, the postclassification method cannot be trained in an end-to-end manner. Meanwhile, the detection accuracy heavily relies on the performance of the building extraction network.

Instead, some supervised CD methods attempt to apply CNNs trained in an end-to-end manner through the available labeled samples. Specifically, most existing CD networks adopt the U-shaped encoder–decoder architecture, where the encoder consists of an early fusion [28], [29], [37] or late fusion [10], [30]–[35] framework for feature extraction. The former takes the concatenated bitemporal images as an input, whereas the latter extracts features from the two images in parallel. In accordance with whether weights are shared, late-fusion networks can be divided into the Siamese and pseudo-Siamese structures [51]. For example, Hou *et al.* [34] proposed a Siamese variant of U-Net [26] for building CD and called it W-Net. WNet conducts comparisons in the feature domain and obtains difference features in a learning manner. Zhang *et al.* [35] proposed a deeply supervised image fusion network (IFN) based on a pseudo-Siamese structure for deep feature extraction and difference discrimination. Besides, the early-fusion framework has drawn attention due to its ease of use for directly learning latent difference features at the beginning stage of the network. Daudt *et al.* [28] extended the fully convolutional early fusion (FC-EF) [30] model with some residual block plugins to facilitate the training of the deeper network. Then, the proposed FC-EF-Res outperforms the fully convolutional Siamese-concatenation model (FC-Siam-diff) [30]. Peng *et al.* [29] adopted an improved UNet++ with dense skip connections for context reusing and took the fusion strategy of multiple side outputs to achieve high accuracy. However, most existing methods ignore pseudochanges and redundant contexts. Given the lack of modeling of foreground changed objects, these methods have weak discrimination of the pseudochanges and result in false alarms.

Some attempts consider CD as an image translation problem and adopt GANs as CD networks [34], [39]–[42]. GAN-based methods have less requirement of labeled samples. They provide a new feasible solution to improve the generalization ability. However, GANs are more difficult and time-consuming to train compared with CNNs. Alternatively, RNNs are well known to be good at processing sequential data. Some attempts have introduced RNNs as a natural candidate for not only extracting spatial–spectral features but also mining the temporal dependencies among bitemporal images [31], [43], [44]. By combining CNNs and RNNs, they attempt to extract spatial–spectral–temporal features. However, these methods take small image patches as inputs for feature extraction, e.g., 5×5 pixels. Due to the limitation of input size, insufficient exploration of spatial context makes them unstable against pseudochanges.

Alternatively, the attention mechanism [52] has been widely studied and embedded into deep CNNs in many computer vision tasks [53]–[62]. To the best of our knowledge, works that utilize the attention mechanism for CD based on RS images are not sufficient [6], [35], [37], [38], [63], [64]. Specifically, the self-attention mechanism [53] is effective in modeling long-range dependencies and generating discriminative features. Chen and Shi [6] proposed STANet, which integrates the self-attention module into a pyramid structure to model the spatial–temporal relationships between any pair of pixels at different times and positions. Zhang *et al.* [35] used the spatial and channelwise attention modules (CBAM) proposed in [56] in the difference discrimination network for the effective fusion of raw image features and image difference features. Peng *et al.* [37] proposed a simplified UNet++ called DDCNN based on the dense upsampling attention units. Chen *et al.* [38] extended the scene segmentation network DANet [58] to DASNet with a weighted double-margin contrastive loss for addressing the class imbalance problem in CD.

## III. Proposed Method

This article presents an attention-guided network called AGCDetNet for building CD in high-resolution RS images. AGCDetNet adopts an FCN-based encoder–decoder architecture and implements a fully automatic end-to-end CD. The main concerns are enhancing the feature representation and promoting the model's discrimination ability by incorporating the attention mechanism. Instead of considering all spatial locations equally, spatial attention makes our model focus on where the changed objects lie. Similarly, channelwise attention emphasizes the task-relevant channels and suppresses the task-irrelevant feature channels by assigning weights to features from the channel dimension. Without loss of generality, $I^{(1)}$ and $I^{(2)}$ denote the first- and second-temporal images acquired over the same geographical target region at different times. $I^{(1)}$ and $I^{(2)}$ have been homogenized through necessary preprocessing, such as geometric coregistration and radiometric correction. And *GT* indicates the reference change map/ground truth. This work focuses on identifying the changed buildings and generating a binary change map *CM*, in which pixels are divided into either foreground changed objects or background. In this manner, the foreground changed objects refer to the changed buildings, including newly built, demolished, and reconstructed buildings. The background indicates the unchanged buildings, as well as other geospatial objects.

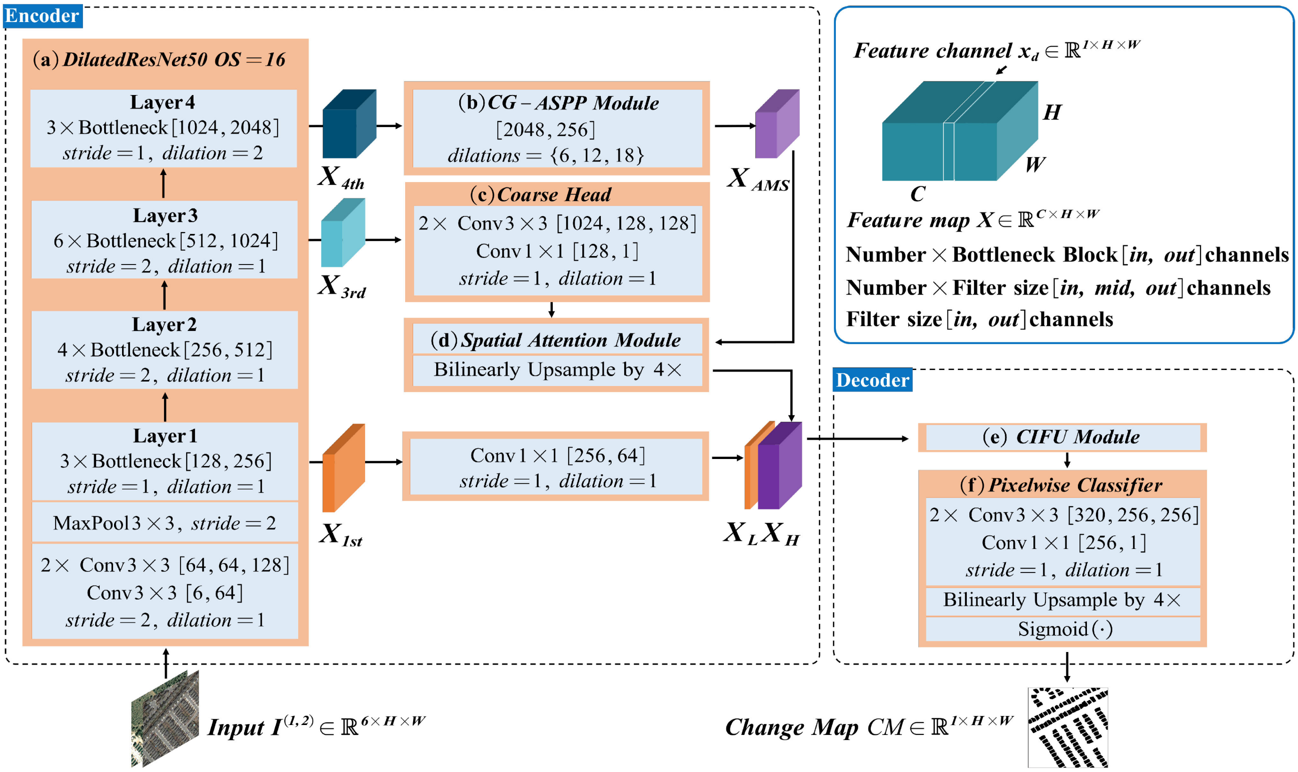

Encoder
(a) DilatedResNet50 OS = 16
Layer4
3 × Bottleneck [1024, 2048]
stride = 1, dilation = 2
Layer3
6 × Bottleneck [512, 1024]
stride = 2, dilation = 1
Layer2
4 × Bottleneck [256, 512]
stride = 2, dilation = 1
Layer1
3 × Bottleneck [128, 256]
stride = 1, dilation = 1
MaxPool 3 × 3, stride = 2
2× Conv 3 × 3 [64, 64, 128]
Conv 3 × 3 [6, 64]
stride = 2, dilation = 1
X_4th
X_3rd
X_1st
(b) CG – ASPP Module
[2048, 256]
dilations = {6, 12, 18}
(c) Coarse Head
2× Conv 3 × 3 [1024, 128, 128]
Conv 1 × 1 [128, 1]
stride = 1, dilation = 1
(d) Spatial Attention Module
Bilinearly Upsample by 4×
Conv 1 × 1 [256, 64]
stride = 1, dilation = 1
X_AMS
X_L X_H
Feature channel x_d ∈ ℝ^{1×H×W}
H
W
C
Feature map X ∈ ℝ^{C×H×W}
Number × Bottleneck Block [in, out] channels
Number × Filter size [in, mid, out] channels
Filter size [in, out] channels
Decoder
(e) CIFU Module
(f) Pixelwise Classifier
2× Conv 3 × 3 [320, 256, 256]
Conv 1 × 1 [256, 1]
stride = 1, dilation = 1
Bilinearly Upsample by 4×
Sigmoid (·)
Input I^{(1,2)} ∈ ℝ^{6×H×W}
Change Map CM ∈ ℝ^{1×H×W}

Fig. 2. Network architecture and attention modules of the proposed AGCDetNet. Zoom-in for an improved view. (a) DilatedResNet50 backbone with OS=16. (b) CG-ASPP module. (c) Coarse head for predicting a coarse change map. (d) SPAM. (e) CIFU. (f) Pixelwise classifier for predicting the change map. Best viewed in color. The ReLU nonlinear activation function and batch normalization (BN) at the tail of convolution layers are not shown for brevity.

### *A. Network Architecture*

The proposed AGCDetNet adopts an FCN-based encoder–decoder architecture and implements a fully automatic end-to-end CD, as shown in Fig. 2. AGCDetNet takes an early-fusion framework that consists of a single-stream encoder and a lightweight decoder. The former extracts feature from input, e.g., taking as an input a pair of concatenated images with 6 bands. The latter classifies the extracted features into two classes and then generates a binary change map through binarization. During training, network parameters are iteratively updated by minimizing the loss between the forward output and the reference using the BP algorithm.

*1) Encoder:* The encoder of AGCDetNet consists of four core components, i.e., 1) a *DilatedResNet50* backbone networ; 2) a *CG-ASPP* module; 3) a coarse head; and 4) a *SPAM*. This section mainly introduces the backbone network; the other components will be described later.

*DilateResNet50 Backbone Network*: As shown in Fig. 2(a), the backbone network of AGCDetNet is composed of $\mathbf{Layer}1$, $\mathbf{Layer}2$, $\mathbf{Layer}3$, and $\mathbf{Layer}4$. $\mathbf{Layer}1$ downsamples the input by four times and transforms $I^{(1,2)} \in \mathbb{R}^{6\times \boldsymbol{H}\times \boldsymbol{W}}$ into a 3-D tensor $\boldsymbol{X}_{1\boldsymbol{st}} \in \mathbb{R}^{256\times\frac{\boldsymbol{H}}{4}\times\frac{\boldsymbol{W}}{4}}$. Similarly, $\boldsymbol{X}_{3\boldsymbol{rd}} \in \mathbb{R}^{1024\times\frac{\boldsymbol{H}}{8}\times\frac{\boldsymbol{W}}{8}}$ and $\boldsymbol{X}_{4\boldsymbol{th}} \in \mathbb{R}^{2048\times\frac{\boldsymbol{H}}{16}\times\frac{\boldsymbol{W}}{16}}$ are feature maps extracted from $\mathbf{Layer}3$ and $\mathbf{Layer}4$, respectively.

*a) Introducing residual network [65]:* Typically, a deep CNN designed for image classification, after removing the last global pooling layer and fully connected layer, is used as a backbone network for feature extraction. The backbone network of AGCDetNet is derived from ResNet-50 [65] since residual network has the advantage of alleviating the degradation problem during training. In this work, the first 7 × 7 convolution of ResNet-50 is replaced with three consecutive 3 × 3 convolutions to reduce network parameters while keeping the receptive field size. Finally, $\mathbf{Layer}1$consists of three stacked 3 × 3 convolutions followed by a MaxPool layer and three stacked bottleneck residual blocks. $\mathbf{Layer}2$, $\mathbf{Layer}3$, and $\mathbf{Layer}4$ consist of 4, 6, and 3 stacked botttlenect residual blocks, respectively. The backbone network reduces the size of input through downsampling and convolution operations and extracts feature maps with varying degrees of semantics. Specifically, low-level features encode rich spatial details but lack semantic information. High-level features are accurate in semantic representation but coarse in spatial resolution.

*b) Introducing atrous/dilated convolutions* [66]*:* Previous works suggest that both high-level semantic features and detailed information are important in CD. Applying atrous convolutions [66] in deep layers is an effective way to enhance the receptive field and maintain the spatial resolution of high-level features. The kernel size of atrous convolution can be formulated as $\{(d-1)\times(k-1)+k\}$, where $k$ indicates the kernel size of standard convolution and $d$ indicates the dilated rate. Atrous convolutions can be flexibly configured to extract high-resolution feature maps with strong semantic by setting different dilation rates [67]. Therefore, we apply atrous convolutions at the last several layers of the backbone network, termed ***DilatedResNet50***. Without loss of generality, $\boldsymbol{OS}(output\_stride)$ indicates the input spatial resolution ratio to the output feature resolution. For instance, a regular ResNet-based backbone network extracts features through four layers with five consecutive downsampling processes, where $\boldsymbol{OS}$ is set to 32. If atrous convolutions with a dilation rate of 2 are only configured in the $\mathbf{Layer}4$, $\boldsymbol{OS}$ will be set to 16. If configured in the last two layers ($\mathbf{Layer}3$ and $\mathbf{Layer}4$) with a dilation rate of 2 and 4, $\boldsymbol{OS}$ will be set to 8.

*2) Decoder:* The decoder only consists of a CIFU module followed by a pixelwise classifier, as shown in Fig. 2(e) and (f). The former attempts to enhance the representation of multilevel features using channelwise attention. A $1 \times 1$ convolution is applied to transform low-level feature $\boldsymbol{X}_{\boldsymbol{1st}} \in \mathbb{R}^{256\times\frac{\boldsymbol{H}}{4}\times\frac{\boldsymbol{W}}{4}}$ into $\boldsymbol{X}_{\boldsymbol{L}} \in \mathbb{R}^{64\times\frac{\boldsymbol{H}}{4}\times\frac{\boldsymbol{W}}{4}}$, and high-level feature $\boldsymbol{X}_{\boldsymbol{H}} \in \mathbb{R}^{256\times\frac{\boldsymbol{H}}{4}\times\frac{\boldsymbol{W}}{4}}$ is obtained through applying a fourfold bilinear upsampling on the features extracted by SPAM. Then, low-level and high-level features are fed into CIFU for refinement, which will be described later. The pixelwise classifier consists of two consecutive $3 \times 3$ convolutions followed by a $1 \times 1$ convolution and Sigmoid layer. The classifier classifies the extracted features and recovers the change probability activation map to the input size through a fourfold bilinear upsampling, i.e., $\boldsymbol{CM}_{\boldsymbol{prob}} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$. Finally, binary change map $\boldsymbol{CM} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ is generated by applying the fixed threshold segmentation on $\boldsymbol{CM}_{\boldsymbol{prob}} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ for binarization. It can be formulated as shown in the following equation:

$$\boldsymbol{CM}_{\boldsymbol{i,j}} = \begin{cases} 1, \text{ if } \boldsymbol{CM}_{\boldsymbol{prob}_{i,j}} > T \\ 0, \text{ otherwise} \end{cases}. \tag{1}$$

The subscript $i, j (1 \leqslant i \leqslant H, 1 \leqslant j \leqslant W)$ indicate the indexes of the height and width, respectively. *T* indicates a fixed binarization threshold to determine whether a pixel has changed, where *T* is empirically set to 0.5. A pixel is classified as changed if and only if the change probability is larger than *T*; otherwise, it is classified as background.

### *B. Spatial Attention Module*

The main concern of SPAM is to learn a spatial attention map to promote discrimination between the change objects and the background in deep features. Given the deep features $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$, SPAM encodes the change probability of each pixel in a spatial attention map $\boldsymbol{SA} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ by incorporating prior knowledge into a scaled dot-product model. Specifically, a coarse change map predicted by a coarse head serves as the prior knowledge. As shown in Fig. 3, SPAM mainly consists of three steps as follows: 1) perform a prior knowledge-guided feature squeeze, 2) estimate the spatial attention map, and 3) assign spatial attention weights to feature maps for refinement.

*1) Prior-Knowledge-Guided Feature Squeeze:* SPAM takes coarse features $\boldsymbol{X}_{\boldsymbol{3rd}} \in \mathbb{R}^{4\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$ and strong semantic deep features $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$ extracted by CG-ASPP as input, and outputs augmented features $\hat{\boldsymbol{X}} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$. First, $\boldsymbol{X}_{\boldsymbol{3rd}} \in \mathbb{R}^{4\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$ is fed into the coarse head to predict a coarse change map $\boldsymbol{L}_{\boldsymbol{cm}} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ under the supervision of the reference during training. $\boldsymbol{L}_{\boldsymbol{cm}} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ is reshaped to $\boldsymbol{L}_{\boldsymbol{cm}} \in \mathbb{R}^{1\times\boldsymbol{N}}, \boldsymbol{N} = \boldsymbol{H}\cdot\boldsymbol{W}$ followed by a softmax normalization to obtain the prior knowledge that indicates where changed objects lie in the coarse features. Besides, $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$ is reshaped to $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{N}}$.

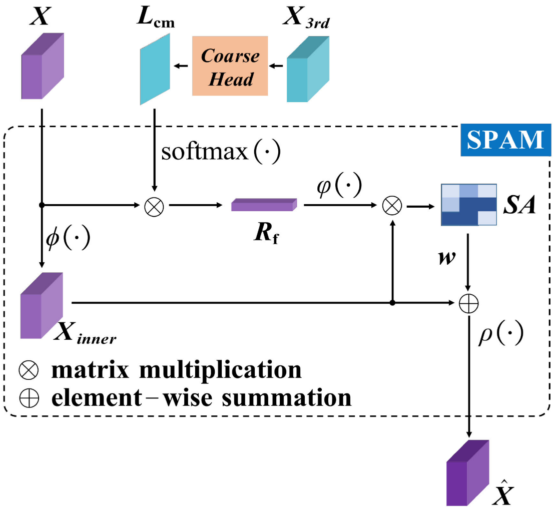


Fig. 3. SPAM module. Zoom-in for an improved view. Best viewed in color.

Next, a prior knowledge-guided feature squeeze across the spatial dimension is applied to aggregate the global context of $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{N}}$. Specifically, a dot-product similarity function is used to calculate a sparse vector $\boldsymbol{R}_{\mathbf{f}} \in \mathbb{R}^{\boldsymbol{C}\times 1}$ through multiplying $\boldsymbol{L}_{\boldsymbol{cm}} \in \mathbb{R}^{1\times\boldsymbol{N}}$ with each feature channel of $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{N}}$. It is implemented by matrix multiplication for high computational efficiency and can be formulated as follows:

$$\boldsymbol{R}_{\mathbf{f}} = \boldsymbol{X}(\textbf{softmax}\,(\boldsymbol{L}_{\boldsymbol{cm}}))^{\boldsymbol{T}}. \tag{2}$$

In this way, the key information of deep features is compressed into the sparse vector $\boldsymbol{R}_{\mathbf{f}} \in \mathbb{R}^{\boldsymbol{C}\times 1}$, where each element of $\boldsymbol{R}_{\mathbf{f}} \in \mathbb{R}^{\boldsymbol{C}\times 1}$ indicates the response of each feature channel of $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{N}}$ to the changed objects. A high activation response indicates that the corresponding feature channel has a high contribution to the changed objects.

*2) Estimate the Spatial Attention Map:* Based on the fact that different feature channels respond to different semantics, the most important feature channels can be selected using the sparse vector $\boldsymbol{R}_{\mathbf{f}} \in \mathbb{R}^{\boldsymbol{C}\times 1}$ to construct the spatial attention map. Given a *C*-dimensional feature vector at the *i*th location, i.e., $x_i = (x_1, \ldots, x_c) \in \mathbb{R}^{1\times\mathrm{C}} (i = 1, \ldots, N)$, the dot-product obtained by multiplying $x_i \in \mathbb{R}^{1\times\mathrm{C}}$ with $\boldsymbol{R}_{\mathbf{f}} \in \mathbb{R}^{\boldsymbol{C}\times 1}$ determines the change probability of the *i*th pixel in deep features. Therefore, the dot-product indicates the degree how much the model should pay attention. In this way, the spatial attention map $\boldsymbol{SA} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ can be estimated through calculating all the dot-products across the spatial dimension. It is implemented by a scaled dot-product model using matrix multiplication as shown in the following equation:

$$\boldsymbol{SA} = \frac{\hat{\boldsymbol{X}}^{\boldsymbol{T}}_{\boldsymbol{inner}}\hat{\boldsymbol{R}}_{\mathbf{f}}}{\sqrt{\boldsymbol{C}}} = \frac{\phi(\boldsymbol{X})^{\boldsymbol{T}}\varphi\,(\boldsymbol{R}_{\mathbf{f}})}{\sqrt{\boldsymbol{C}}}. \tag{3}$$

Specifically, nonlinearity transformation functions $\varphi(\cdot)$ and $\phi(\cdot)$ are applied to transform $\boldsymbol{R_f} \in \mathbb{R}^{\boldsymbol{C}\times 1}$ and $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$ to $\hat{\boldsymbol{R}}_{\boldsymbol{f}} \in \mathbb{R}^{\boldsymbol{C}\times 1}$ and $\boldsymbol{X_{inner}} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$, respectively. $\boldsymbol{X_{inner}} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$ is reshaped to $\hat{\boldsymbol{X}}_{\boldsymbol{inner}} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{N}}$. $\varphi(\cdot)$ and $\phi(\cdot)$ are implemented by two $3\times 3$ convolutions followed by BN and ReLU layers. The dot-product result is normalized by $\sqrt{C}$ to alleviate the effect of the number of feature channels on the spatial attention map during training. Meanwhile, to keep the same spatial dimension with the inner features $\boldsymbol{X_{inner}} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$, $\boldsymbol{SA} \in \mathbb{R}^{\boldsymbol{N}\times 1}$ is reshaped to $\boldsymbol{SA} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$. Thus, $\boldsymbol{SA} \in \mathbb{R}^{\boldsymbol{N}\times 1}$ encodes the change probability of each pixel in deep features. The above procedure of estimating the spatial attention map is the so-called modeling of foreground changed objects.

*3) Assign Spatial Attention Weights to Feature Maps:* The last step is to assign attention weights to each feature channel of original features for refinement and generate augmented features $\hat{\boldsymbol{X}} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$ with powerful discrimination. It is implemented by adding $\boldsymbol{SA} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ to each feature channel $\boldsymbol{X_d} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ through an elementwise summation across the spatial dimension and followed by a nonlinear transformation $\rho(\cdot)$, as shown in the following equation:

$$\hat{\boldsymbol{X}}_{\boldsymbol{d}} = \boldsymbol{\rho}\left(\boldsymbol{X_d} + \boldsymbol{w}\cdot\boldsymbol{SA}\right), (\boldsymbol{d} = 1, \ldots, \boldsymbol{C}) \tag{4}$$

where $\hat{\boldsymbol{X}}_{\boldsymbol{d}} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ indicates the $d$th feature channel of $\hat{\boldsymbol{X}} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$, and $w$ is a learnable scale factor to adjust the activation of $\boldsymbol{SA} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$. $w$ is initialized as zero and learn the weight during training. $\rho(\cdot)$ is implemented by a $1\times 1$ convolution followed by BN and ReLU layers. Besides, $\boldsymbol{SA} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ is also supervised by the reference during training.

The pixels that have a large change probability in deep features will be assigned with high activation and vice versa. SPAM highlights the pixel locations where the changed buildings are located and promotes the discrimination between the changed objects and the background. It helps to reduce false alarms caused by the distractions of pseudochanges and other geospatial objects.

### C. Channelwise Attention-Guided Modules

*1) Channelwise Attention-Guided Interference Filtering Unit:* Previous works suggests that low-level high-resolution features are crucial for refining high-level semantic features and achieving high-quality change maps [6], [29], [35]. A CIFU is proposed to enhance the representation of multilevel features. CIFU is derived from the channelwise attention module that encodes the importance of different feature channels in a channel attention weight vector. Feature channels can be recalibrated according to the learned attention weight.

As shown in Fig. 4, CIFU applies two MLPs to explore the channelwise interdependence among multilevel features and learn private channel attention weight vectors. CIFU mainly consists of three steps: 1) perform a joint feature squeeze; 2) explore channel attention weight vectors; and 3) perform channelwise interference filtering.

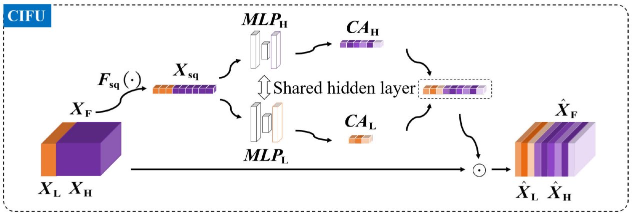


Fig. 4. CIFU module. Zoom-in for an improved view. Best viewed in color.

*a) Performing a joint feature squeeze:* First, the high-level feature $\boldsymbol{X_H} \in \mathbb{R}^{\boldsymbol{C_H}\times\boldsymbol{H}\times\boldsymbol{W}}$ and low-level feature $\boldsymbol{X_L} \in \mathbb{R}^{\boldsymbol{C_L}\times\boldsymbol{H}\times\boldsymbol{W}}$ are concatenated along the channel axis to obtain a joint feature $\boldsymbol{X_F} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$, where $\boldsymbol{C} = \boldsymbol{C_H} + \boldsymbol{C_L}$. Next, a shared feature squeeze layer $F_{sq}(\cdot)$ is applied to squeeze the joint feature $\boldsymbol{X_F} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$ into a channelwise descriptor $\boldsymbol{X_{sq}} \in \mathbb{R}^{1\times 1\times\boldsymbol{C}}$. In this way, $\boldsymbol{X_{sq}} \in \mathbb{R}^{1\times 1\times\boldsymbol{C}}$ aggregates the global context of $\boldsymbol{X_F} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$ across the spatial dimension. The feature squeezing is implemented by a global average pooling layer for its simplicity and computational efficiency. It can be formulated as follows:

$$\boldsymbol{X_{sq}} = \boldsymbol{F_{sq}}\left(\boldsymbol{X_F}\right) = \frac{1}{\boldsymbol{HW}}\sum_{\boldsymbol{i}=1}^{\boldsymbol{H}}\sum_{\boldsymbol{j}=1}^{\boldsymbol{W}}\boldsymbol{X_F}\left(i, j\right). \tag{5}$$

*b) Exploring channel attention weight vectors:* CIFU applies two MLPs, i.e., $\boldsymbol{MLP_L}(\cdot)$ and $\boldsymbol{MLP_H}(\cdot)$, to explore the interdependence among the feature channels of $\boldsymbol{X_{sq}} \in \mathbb{R}^{1\times 1\times\boldsymbol{C}}$ and learn channel attention weight vectors for $\boldsymbol{X_L} \in \mathbb{R}^{\boldsymbol{C_L}\times\boldsymbol{H}\times\boldsymbol{W}}$ and $\boldsymbol{X_H} \in \mathbb{R}^{\boldsymbol{C_H}\times\boldsymbol{H}\times\boldsymbol{W}}$. In particular, $\boldsymbol{MLP_L}(\cdot)$ and $\boldsymbol{MLP_H}(\cdot)$ share one hidden layer to capture semantic similarity among different features and apply private output layers to focus on their difference. Specifically, the shared hidden layer is implemented by a fully connected layer followed by a ReLU activation function $\vartheta(\cdot)$. The output layers are implemented by a fully connected layer followed by a Sigmoid activation function $\sigma(\cdot)$. $\boldsymbol{MLP_L}(\cdot)$ and $\boldsymbol{MLP_H}(\cdot)$ simultaneously take $\boldsymbol{X_{sq}} \in \mathbb{R}^{1\times 1\times\boldsymbol{C}}$ as an input and output private channelwise attention $\boldsymbol{CA_H} \in \mathbb{R}^{1\times 1\times\boldsymbol{C_H}}$ and $\boldsymbol{CA_L} \in \mathbb{R}^{1\times 1\times\boldsymbol{C_L}}$, respectively. In short, it can be formulated as follows:

$$\begin{cases} \boldsymbol{CA_L} = \boldsymbol{MLP_L}\left(\boldsymbol{X_{sq}}\right) = \sigma\left(\vartheta\left(\boldsymbol{X_{sq}\Omega}\right)\boldsymbol{\Omega_L}\right) \\ \boldsymbol{CA_H} = \boldsymbol{MLP_H}\left(\boldsymbol{X_{sq}}\right) = \sigma\left(\vartheta\left(\boldsymbol{X_{sq}\Omega}\right)\boldsymbol{\Omega_H}\right) \end{cases} \tag{6}$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{\boldsymbol{C}\times\frac{\boldsymbol{C}}{r}}$ indicates the shared hidden layer weights; and $\boldsymbol{\Omega_L} \in \mathbb{R}^{\boldsymbol{C}\times\frac{\boldsymbol{C_L}}{r}}$ and $\boldsymbol{\Omega_H} \in \mathbb{R}^{\boldsymbol{C}\times\frac{\boldsymbol{C_H}}{r}}$ indicate the output layer weights of $\boldsymbol{MLP_L}(\cdot)$ and $\boldsymbol{MLP_H}(\cdot)$, respectively. To reduce the parameter overhead, the size of hidden output activation is set to $\vartheta(\boldsymbol{X_{sq}\Omega}) \in \mathbb{R}^{1\times 1\times\frac{\boldsymbol{C}}{r}}$, where $r$ indicates the reduction ratio and is set to 16 as default due to the large number of the channel dimension.

*c) Performing channelwise interference filtering:* Channelwise interference filtering for different features are implemented by elementwise multiplication using the above channelwise attention weight vectors $\boldsymbol{CA_H} \in \mathbb{R}^{1\times 1\times\boldsymbol{C_H}}$ and $\boldsymbol{CA_L} \in \mathbb{R}^{1\times 1\times\boldsymbol{C_L}}$. It can be formulated as follows:

$$\begin{cases} \hat{\boldsymbol{X}}_{\boldsymbol{L}} = \boldsymbol{CA_L} \odot \boldsymbol{X_L} \\ \hat{\boldsymbol{X}}_{\boldsymbol{H}} = \boldsymbol{CA_H} \odot \boldsymbol{X_H} \end{cases} \tag{7}$$

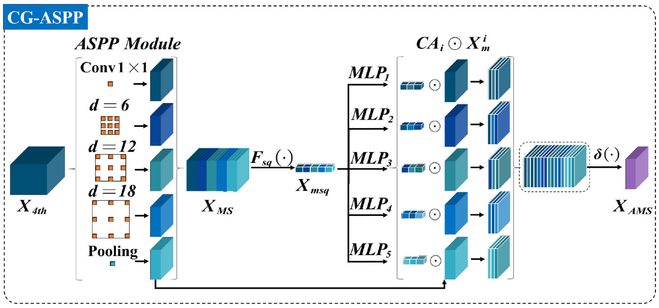


Fig. 5. CG-ASPP module. Zoom-in for an improved view. Best viewed in color.

where $\odot$ denotes the elementwise multiplication operation between a channel attention weight and the corresponding feature channel. $\hat{\boldsymbol{X}}_{\boldsymbol{H}} \in \mathbb{R}^{\boldsymbol{C_H} \times \boldsymbol{H} \times \boldsymbol{W}}$ and $\hat{\boldsymbol{X}}_{\boldsymbol{L}} \in \mathbb{R}^{\boldsymbol{C_L} \times \boldsymbol{H} \times \boldsymbol{W}}$ represent the refined features after channelwise interference filtering. Then, $\hat{\boldsymbol{X}}_{\boldsymbol{H}} \in \mathbb{R}^{\boldsymbol{C_H} \times \boldsymbol{H} \times \boldsymbol{W}}$ and $\hat{\boldsymbol{X}}_{\boldsymbol{L}} \in \mathbb{R}^{\boldsymbol{C_L} \times \boldsymbol{H} \times \boldsymbol{W}}$ are concatenated along the channel axis to obtain the refined difference features $\hat{\boldsymbol{X}}_{\boldsymbol{F}} \in \mathbb{R}^{\boldsymbol{C} \times \boldsymbol{H} \times \boldsymbol{W}}$. During the separate channelwise interference filtering, it enhances the representation of the changed-relevant feature channels and suppresses the irrelevant channels that have a large response to the contours and textures of the background. With the help of CIFU, the network can accurately locate and identify the foreground changed objects and improve boundary completeness and internal compactness.

*2) CG-ASPP Module:* The atrous spatial pyramid pooling (ASPP) module [68] can effectively capture the context of objects of various scales. On this basis, a channelwise attention-guided CG-ASPP module is built on the top of the backbone to capture multiscale context and improve the semantic consistency among multiscale features. The idea of the CG-ASPP module is derived from the CIFU. The main difference is that CG-ASPP explores the interdependence among multiscale features to improve their semantic consistency in the encoder, whereas the CIFU focuses on the interference filtering of the context information flows from encoder to decoder. Besides, CG-ASPP module extends channelwise attention to enhance the representation of multiscale context derived from the original ASPP module. CG-ASPP also consists of three steps: 1) perform a joint feature squeeze; 2) explore separate channelwise attention; and 3) assign attention for multiscale context refinement.

As shown in Fig. 5, ASPP module [68] captures multiscale context from the high-level semantic features. ASPP module consists of five parallel branches. Each branch is composed of a 1 × 1 convolution, three parallel 3 × 3 atrous convolutions with different dilation rates $d = \{6,\ 12,\ 18\}$, and an image-level feature obtained by global average pooling followed by a 1 × 1 convolution. Each branch is followed by a BN and ReLU layer to facilitate convergence. Consequently, five scale features $X_m^i \in \mathbb{R}^{C \times H \times W}, (i = 1, \ldots, 5)$ are simultaneously obtained.

Similarly, the concatenated multiscale features $\boldsymbol{X_{MS}} \in \mathbb{R}^{\boldsymbol{5C} \times \boldsymbol{H} \times \boldsymbol{W}}$ are squeezed into a channelwise descriptor $\boldsymbol{X_{msq}} \in \mathbb{R}^{1 \times 1 \times 5\boldsymbol{C}}$ by a global average pooling layer $F_{sq}(\cdot)$ across the spatial dimension. Furthermore, five multilayer perceptrons $MLP_i(\cdot), (i = 1, \ldots, 5)$, are constructed to mine the interdependence among the channels of $\boldsymbol{X_{msq}} \in \mathbb{R}^{1 \times 1 \times 5\boldsymbol{C}}$ and explore the latent semantic relation among different scale features. Sequentially, CG-ASPP generates five different channelwise attention vectors $\boldsymbol{CA_i} \in \mathbb{R}^{1 \times 1 \times \boldsymbol{C}}$ in a learning manner. $MLP_i(\cdot)$ forwards the descriptor $\boldsymbol{X_{msq}} \in \mathbb{R}^{1 \times 1 \times 5\boldsymbol{C}}$ into the shared hidden layer and then outputs the attention via their private output layers, respectively. In short, it is formulated as follows:

$$\begin{aligned} \boldsymbol{CA_i} &= \boldsymbol{MLP_i}\left(\boldsymbol{X_{msq}}\right) \\ &= \boldsymbol{\sigma}\left(\vartheta\left(\boldsymbol{X_{msq}} \boldsymbol{\Omega_m}\right) \boldsymbol{\Omega_i}\right), (i = 1, \ldots, 5) \end{aligned} \tag{8}$$

where $\boldsymbol{\Omega_m} \in \mathbb{R}^{\boldsymbol{C} \times \frac{\boldsymbol{5C}}{r}}$ indicates the shared hidden layer weights and $\boldsymbol{\Omega_i} \in \mathbb{R}^{\frac{\boldsymbol{5C}}{r} \times \boldsymbol{C}}$ denotes the output layer weights of $MLP_i(\cdot)$. $r$ is the reduction ratio of hidden layer parameters, and it is set to 16 as default to reduce the parameter overhead. $\sigma(\cdot)$ and $\vartheta(\cdot)$ represent the Sigmoid and ReLU activation functions.

Finally, the learned channelwise attention is assigned to the original scale features for improving their semantic consistency. The set of refined multiscale features is denoted by $\{CA_i \odot X_m^i \mid i = 1, \ldots, 5\}$. Besides, a 1 × 1 convolution layer $\delta(\cdot)$ is used to integrate the refined multiscale features and transform them into an augmented joint feature $X_{AMS} \in \mathbb{R}^{C \times H \times W}$.

### *D. Loss Function Definition*

Loss function plays a key role in improving the performance and facilitating the convergence of networks during training. In general, the cross-entropy loss is utilized to measure the similarity between two probability distributions. The binary cross-entropy loss function (*BCE* loss) is an intuitive candidate since binary CD classifies the pixels into changed or unchanged pixels. The loss function can be formulated as follows:

$$\boldsymbol{L_{BCE}} = -\frac{1}{\boldsymbol{N}} \sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \left(\boldsymbol{y_n} \mathbf{log}\left(\hat{\boldsymbol{y}}_{\boldsymbol{n}}\right) + \left(1 - \boldsymbol{y_n}\right) \mathbf{log}\left(1 - \hat{\boldsymbol{y}}_{\boldsymbol{n}}\right)\right) \tag{9}$$

where $N$ is the number of samples, $y_n \in \{0, 1\}$ indicates an unchanged or changed pixel of the ground truth, and $\hat{y}_n \in [0, 1]$ denotes the prediction of the network.

The *BCE* loss is suitable in the case of class balance. However, the number of background pixels is always far more than that of foreground changed pixels. The network trained on unbalanced training samples with the *BCE* loss may tend to classify most pixels to be the background pixels and result in a high miss alarm rate [10]. To alleviate the problem, a soft Jaccard term is introduced [69] and the loss function can be formulated as follows:

$$\begin{aligned} \boldsymbol{L_{BCD}} = &-\boldsymbol{\lambda}_1 \frac{1}{\boldsymbol{N}} \sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \left(\boldsymbol{y_n} \mathbf{log}\left(\hat{\boldsymbol{y}}_{\boldsymbol{n}}\right) + \left(1 - \boldsymbol{y_n}\right) \mathbf{log}\left(1 - \hat{\boldsymbol{y}}_{\boldsymbol{n}}\right)\right) \\ &+ \left(1 - \boldsymbol{\lambda}_1\right) \frac{1}{\boldsymbol{N}} \mathbf{log}\left(\sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \frac{\boldsymbol{y_n} \cdot \hat{\boldsymbol{y}}_{\boldsymbol{n}}}{\boldsymbol{y_n} + \hat{\boldsymbol{y}}_{\boldsymbol{n}} - \boldsymbol{y_n} \cdot \hat{\boldsymbol{y}}_{\boldsymbol{n}}}\right) \end{aligned} \tag{10}$$

where the second term indicates the soft *Jaccard*, and $\lambda_1 \in [0, 1]$ is the weight factor to balance *BCE* and *Jaccard* loss.

TABLE I
EXPERIMENT DATASETS

| *Datasets* | *Data Collection* | *Spatial Resolution* | *Number of Changed/Unchanged Pixels* | *Size of Samples* | *Number of Samples* | | |
|---|---|---|---|---|---|---|---|
| | | | | | *training set* | *validation set* | *test set* |
| LEVIR-CD | Google Earth | 0.5 m/pixel | 30,913,975/637,028,937 | 512 × 512 | 4,016 | 1,024 | 512 |
| WHU | Aerial Image | 0.2 m/pixel | 21,352,815/477,759,663 | 512 × 512 | 3,528 | – | 1,024 |
| Season-Varying | Google Earth | 3-100 cm/pixel | 134,068,750/914,376,178 | 256 × 256 | 10,000 | 3,000 | 3,000 |
| WV2 Site 1 | Worldview-2 | 2 m/pixel | 270,438/1,777,323 | 256 × 256 | 2,888 | – | – |
| WV2 Site 2 | Worldview-2 | 2 m/pixel | 376,610/1,671,151 | 1431 × 1431 | – | – | 1 |
| ZY3 | Ziyuan 3 | 5.8 m/pixel | 27,195/228,827 | 559 × 458 | – | – | 1 |

To facilitate the performance of the SPAM, $L_{BCE}$ and $L_{BCD}$ losses are used to supervise the learning procedure of $\boldsymbol{SA} \in \mathbb{R}^{1\times \boldsymbol{H}\times \boldsymbol{W}}$ and $\boldsymbol{L_{cm}} \in \mathbb{R}^{1\times \boldsymbol{H}\times \boldsymbol{W}}$ under the supervision of the reference during training, respectively. Consequently, the complete CD loss can be formulated as follows:

$$\boldsymbol{L} = \boldsymbol{L_m} + \boldsymbol{\lambda_c L_c} + \boldsymbol{\lambda_s L_s} \tag{11}$$

where $L_m$ , $L_c$, and $L_s$ represent the master branch loss, coarse change map loss, and spatial attention map loss, respectively. The first two terms adopt the *BCD* loss, and the last term adopts pure *BCE* loss. $\lambda_c$ and $\lambda_s$ are the balance weights.

## IV. EXPERIMENTS

### A. Datasets

As shown in Table I, four public representative datasets were used for model training and evaluation, including LEVIR-CD [6], WHU [8], Season-Varying [39], and WV2 and ZY3 [10]. We applied the criteria as recommended by the creators to split the datasets.

*1) LEVIR-CD Dataset[1]:* The dataset consists of 637 pairs of coregistered VHR Google Earth images and the reference change masks. Original images have a size of 1024×1024 pixels with a spatial resolution of 0.5 m/pixel. These images, with a period of 5–14 years, were collected from 20 different regions that sit in several cities in Texas of the US. The dataset mainly focuses on building-related changes, i.e., building growth and decline. The number of changed/unchanged pixels is 30 913 975 and 637 028 937. The creator randomly split the dataset into three parts, i.e., 70% samples for training, 10% for validation, and 20% for testing. Due to the limitation of GPU memory, we cropped the original images into smaller image tiles with a size of 512 × 512 pixels for model training and evaluation. Specifically, 4016 and 1024 samples were generated for training and validation, respectively, using a sliding window with a stride of 256 overlapping pixels. In all, 512 nonoverlapping samples were generated for testing using a sliding window with a stride of 512 pixels.

*2) WHU Building Dataset[2]:* The dataset consists of a pair of coregistered aerial images (TA-2011 and TA-2016) with a size of 15 354×32 507 pixels and the reference change masks. The study area is in Christchurch, New Zealand, which had undergone an earthquake in 2011. The study area covers large amounts of building growth. The ground sampling distance of these images is 0.2 m/pixel. As shown in Fig. 6, we divided the original images into two parts. Specifically, the part inside the red box with a size of 7182×32 507 pixels was used for model training, and the remaining part with a size of 8172×32 507 pixels was used for testing. The number of changed/unchanged pixels is 21 352 815 and 477 759 663. Similarly, we cropped the original images into smaller image tiles with a size of 512×512 pixels for model training and evaluation. Specifically, 3528 samples were generated for training using a sliding window with a stride of 256 overlapping pixels. In all, 1024 nonoverlapping samples were generated for testing using a sliding window with a stride of 512 pixels.

*3) Season-Varying Dataset[3]:* The dataset contains seven pairs of coregistered season-varying Google Earth (DigitalGlobe) images with a size of 4725×2700 pixels and the reference change masks. The spatial resolution of these images is from 3 to 100 cm/pixel. The change types are mainly related to land changes, building changes, road changes, and car changes. Each pair of images was cropped into randomly rotated fragments (0–2$\pi$) with a size of 256×256 pixels and at least a fraction of changed pixels. The number of changed/unchanged pixels is 134 068 750 and 914 376 178. Finally, Season-Varying contains 16 000 pairs of image tiles, of which 10 000 and 3000 tiles were used for training and validation, respectively, and extra 3000 tiles were used for testing.

*4) WV2 and ZY3 Dataset[4]:* Images of WV2 and ZY3 datasets were acquired by the satellite WorldView-2 and Ziyuan 3, respectively. WV2 dataset contains two pairs of coregistered images with a size of 1431×1431 pixels and a spatial resolution of 2 m/pixel. WV2 images were acquired in 2010 and 2015, respectively. WV2 Site 1 and WV2 Site 2 come from two different regions in Shenzhen, China. The number of changed/unchanged pixels is 270 438/1 777 323 in WV2 Site 1, and that of changed/unchanged pixels is 376 610/1 671 151 in WV2 Site 2. ZY3 dataset contains two coregistered images with a size of 559×458 pixels and a spatial resolution of 5.8 m/pixel. Bitemporal images were acquired in 2014 and 2016, and the study area is Wuhan, Hubei, China. The number of

[1] Online. [Available]: https://justchenhao.github.io/LEVIR/

[2] Online. [Available]: https://study.rsgis.whu.edu.cn/pages/download/building_dataset.html

[3] Online. [Available]: https://drive.google.com/file/d/1GX656JqqOyBi_Ef0w65kDGVto-nHrNs9

[4] Online. [Available]: https://github.com/MinZHANG-WHU/FDCNN

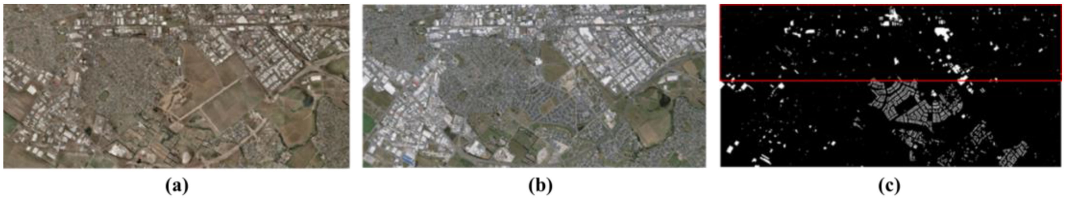


Fig. 6. WHU building CD dataset. (a) TA-2011, image acquired in 2011. (b) TA-2016, image acquired in 2016. (c) Reference change map, a binary change label. The red box is used to mark the training part, and the remaining part is used for testing. Best viewed in color.

changed/unchanged pixels is 27 195/228 827. WV2 Site 1 was cropped and randomly augmented into 2888 image tiles with a size of 256×256 pixels for model training using a sliding window with a stride of 32 pixels. WV2 Site 2 and ZY3 were used for testing.

### B. Evaluation Metrics

Metrics for the quantitative evaluations, including precision (***Pr***), recall (***Re***), false alarm rate (***FA***), miss alarm rate (***MA***), overall accuracy/error (***OA***)/(***OE***), Kappa coefficients (***Kappa***), intersection of union (***IoU***), and F1-score ($\boldsymbol{F_1}$), can be formulated as shown in (12). In binary CD, false positive (***FP***)/true positive (***TP***) indicates the number of pixels misclassified/correctly classified as changed. False negative (***FN***)/true negative (***TN***) indicates the number of pixels misclassified/correctly classified as unchanged. $\boldsymbol{F_1}$ and ***IoU*** are comprehensive indicators; the higher the value, the better the performance. ***Kappa*** indicates the consistency between the change map and the reference, where OA indicates the percentage of correct classifications and PRE indicates that of expected agreements

$$\begin{aligned}
&\boldsymbol{Pr} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}\\
&\boldsymbol{Re} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}\\
&\boldsymbol{FA} = \frac{\boldsymbol{FP}}{\boldsymbol{TN+FP}}\\
&\boldsymbol{MA} = \frac{\boldsymbol{FN}}{\boldsymbol{TP+FN}}\\
&\boldsymbol{OE} = \frac{\boldsymbol{FP+FN}}{\boldsymbol{TP+FP+TN+FN}}\\
&\boldsymbol{Kappa} = \frac{\boldsymbol{OA-PRE}}{\boldsymbol{1-PRE}}\\
&\boldsymbol{PRE} = \frac{\boldsymbol{(TP+FN)\cdot(FP+TP)+(TN+FP)\cdot(TP+TN)}}{\boldsymbol{(TP+FP+TN+FN)^2}}\\
&\boldsymbol{OA} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+FP+TN+FN}}.
\end{aligned} \tag{12}$$

### C. Experiment Settings

The proposed AGCDetNet was implemented using *PyTorch* [70] framework and optimized through the AdamW optimizer [71] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, of which a fixed learning rate and weight decay were set to 0.00125 and 0.0001. During training, AGCDetNet was trained from scratch, of which the weights were initialized by He initialization [72] and biases were initialized as zeros. The batch size was set to 16. The coefficients of loss function were empirically set to $\lambda_1 = 0.7$ and $\lambda_c = 0.4$ for all the experiments. $\lambda_s$ was set to 0.1 and 0.3 for the LEVIR-CD dataset and other datasets. During testing, the threshold for binarization was set to 0.5. Experiments were performed using two *NVIDIA RTX2080Ti GPUs* with 11 GB memory.

### D. Experiment Results

Ablation studies were conducted to verify the contributions of AGCDetNet's core components. Comprehensive evaluations and comparisons with other state-of-the-art methods were performed on different public datasets.

*1) Ablation Study:* The challenging WHU dataset was selected for ablation study. The building appearance varies from training data to testing data, which brings more challenges to the generalization ability. Table II reports the contribution of each component of AGCDetNet as well as the number of parameters (*M*) and computational costs (*FLOPs*) of each model, where "w/" means "with." The ***Baseline*** means the pure FCN-based encoder–decoder architecture without using any attention-based modules. The backbone network $\mathbf{DilatedResNet}50$with $\boldsymbol{OS} = 16$ costs about 23.63 *M* parameters and 39.94 *GFLOPs*. Compared to the ***Baseline***, models "w/ SPAM" and "w/ CIFU" achieve better performance, whereas only introducing very few additional parameters and negligible computations. The model with SPAM gains an improvement with 0.58% of $F_1$ and 0.91% of *IoU*. The model with CIFU gains an improvement with 0.46% of $F_1$ and 0.73% of *IoU*. The model with CG-ASPP achieves an improvement with 0.94% of $F_1$ and 1.49% of *IoU*. Moreover, AGCDetNet achieves the best $F_1$ and *IoU*, which are increased by 1.47% and 2.33% compared with the *baseline*, respectively.

*2) Comparisons on LEVIR-CD:* Several state-of-the-art methods were selected as benchmarks, which include three U-shape-based variants, i.e., FC-EF-Res [28], Peng *et al.* [29], and W-Net [34] and four attention-based methods, i.e., DANet [58], FarSeg [57], STANet [6], and DDCNN [37]. In particular, STANet was proposed by creators of the LEVIR-CD dataset. The results of STANet were obtained by using the model trained by the authors. Table III reports the quantitative results and suggests that AGCDetNet outperforms other benchmarks and achieves the best *Pr* (92.12%), *FA* (0.41%), $F_1$ (0.9076), and *IoU* (0.8309). Compared with STANet, AGCDetNet gains a considerable improvement with 2.68% of $F_1$ and 4.40% of *IoU*. STANet achieves a slightly higher recall but lower precision than AGCDetNet, whereas AGCDetNet makes a better tradeoff between precision and recall. Although DDCNN/Peng *et al.* achieve very close results to AGCDetNet, AGCDetNet reduces the computational costs. Compared with DDCNN, AGCDetNet achieves an improvement with 1.09% of *IoU* and 0.71% of $F_1$ and requires fewer parameters. The number of parameters of DDCNN (60.21 *M*) is about 1.4 times that of AGCDetNet (43.05 *M*). The computational cost of DDCNN (855.19 *GFLOPs*) is

TABLE II
ABLATION RESULTS ON THE WHU DATASET

| *Method* | *Pr (%)* | *Re (%)* | *MA (%)* | *FA (%)* | *IoU* | $F_1$ | *Computational Costs (GFLOPs)* | *Number of Parameters (M)* | *Testing time (ms)* |
|---|---|---|---|---|---|---|---|---|---|
| Baseline | 87.17 | 86.35 | 13.65 | 0.68 | 0.7661 | 0.8676 | 63.27 | 26.23 | 5.73 |
| w/ SPAM | 86.36 | 88.34 | 11.66 | 0.75 | 0.7752 | 0.8734 | 64.83 | 27.88 | 6.85 |
| w/ CIFU | 88.89 | 85.62 | 14.38 | 0.57 | 0.7734 | 0.8722 | 63.27 | 26.24 | 6.03 |
| w/ CG-ASPP | 87.92 | 87.49 | 12.51 | 0.65 | 0.7810 | 0.8770 | 78.03 | 41.38 | 6.72 |
| AGCDetNet | 88.27 | 88.20 | 11.80 | 0.63 | 0.7894 | 0.8823 | 79.60 | 43.05 | 8.09 |

TABLE III
COMPARISON OF RESULTS ON THE LEVIR-CD DATASET

| *Method* | *Pr(%)* | *Re(%)* | *MA(%)* | *FA(%)* | *IoU* | $F_1$ |
|---|---|---|---|---|---|---|
| W-Net [34] | 90.37 | 85.92 | 14.08 | 0.49 | 0.7871 | 0.8809 |
| FC-EF-Res [28] | 90.22 | 88.25 | 11.75 | 0.51 | 0.8055 | 0.8923 |
| Peng *et al.* [29] | 91.53 | 88.70 | 11.30 | 0.44 | 0.8197 | 0.9009 |
| DANet [58] | 84.31 | 87.26 | 12.97 | 0.84 | 0.7507 | 0.8591 |
| FarSeg [57] | 89.14 | 88.80 | 11.20 | 0.58 | 0.8014 | 0.8897 |
| STANet [6] | 85.01 | 91.38 | 8.62 | 0.87 | 0.7869 | 0.8808 |
| DDCNN [37] | 91.88 | 88.29 | 11.71 | 0.42 | 0.8190 | 0.9005 |
| AGCDetNet | 92.12 | 89.45 | 10.55 | 0.41 | 0.8309 | 0.9076 |

TABLE IV
COMPARISON OF RESULTS ON THE WHU DATASET

| *Method* | *Pr(%)* | *Re(%)* | *MA(%)* | *FA(%)* | *IoU* | $F_1$ |
|---|---|---|---|---|---|---|
| Ji *et al.* [11] | 93.10 | 89.20 | — | — | 0.8370 | 0.9111 |
| FC-EF-Res [28] | 83.88 | 81.33 | 18.75 | 0.84 | 0.7034 | 0.8259 |
| Peng *et al.* [29] | 89.55 | 82.67 | 17.41 | 0.52 | 0.7539 | 0.8597 |
| DANet [58] | 83.36 | 84.60 | 15.40 | 0.91 | 0.7238 | 0.8398 |
| STANet [6] | 86.10 | 81.79 | 16.51 | 0.93 | 0.7225 | 0.8389 |
| DDCNN [37] | 87.11 | 83.18 | 16.82 | 0.66 | 0.7406 | 0.8510 |
| FarSeg [57] | 90.96 | 83.05 | 16.95 | 0.44 | 0.7672 | 0.8682 |
| AGCDetNet | 88.27 | 88.20 | 11.80 | 0.63 | 0.7894 | 0.8823 |

about 10.75 times that of AGCDetNet (79.60 *GFLOPs*) when they take as an input of 6 × 512 × 512 size, and that of Peng *et al.* (125.35 *GFLOPs*) is about 1.57 times that of AGCDetNet.

For intuitive comparisons, some CD results are presented in Fig. 7. The black regions indicate the changed objects, and white regions indicate the background. The comparison results on the large-scale change objects indicate that AGCDetNet performs better than other benchmarks. Compared with other approaches, AGCDetNet achieves more complete boundaries and higher internal compactness. In addition, change maps generated by other methods suffer from false alarms in the background region. Comparisons on the challenging case of crowded small-scale building groups show that change maps generated by AGCDetNet are closer to the reference than other methods. Not like the buildings sticking together in the results obtained by other approaches, AGCDetNet successfully detected most individual building instances and identified the tiny gap among the crowded building groups. The above analysis demonstrated that AGCDetNet has the advantage of overcoming scale variations in RS images-based building CD.

The penult row of Fig. 7 illustrates the false alarms caused by the spectral pseudochanges. Other approaches except for FarSeg exhibit poor results with many false alarms because they misclassified the roofs with different colors as a changed building while they are unchanged. The last row of Fig. 7 shows that STANet and W-Net produced isolated noise in the background region, especially at the roof's misaligned boundary. Instead, attention-based AGCDetNet and FarSeg overcome these pseudochanges and generated change maps without false alarms. Attention helps to alleviate the distractions of the pseudochanges and improves accuracy. Compared with the state-of-the-art approaches, AGCDetNet can obtain promising change maps in the case of some challenging practical applications.

*3) Comparisons on WHU:* For comparison, four attention-based methods, i.e., DANet [58], FarSeg [57], STANet [6], DDCNN [37] and two U-shape-based variants, i.e., FC-EF-Res [28] and Peng *et al.* [29], as well as the postclassification-based method proposed by Ji *et al.* [11] were selected as benchmarks. Ji *et al.* [11] are exactly the creators of the WHU dataset.

Table IV reports the quantitative results. Ji *et al.* achieve the best results among these methods using the excellent building extraction network Mask R-CNN. AGCDetNet achieves better performance in *Re* (88.20%), *MA* (11.80%), $F_1$ (0.8823), and *IoU* (0.7894) than the remaining approaches. Compared with FarSeg, AGCDetNet improved $F_1$ (1.41%) and *IoU* (2.22%), respectively. Moreover, AGCDetNet makes a good tradeoff between precision and recall and achieves competitive performance. CD results on the WHU dataset are given in Fig. 8. It suggests that the detection results of AGCDetNet on buildings with various scales are closer to the reference. In contrast, other methods achieve poor results with broken boundaries and low internal compactness.

Even in the challenging case of distractions caused by other geospatial objects such as vehicles and containers, AGCDetNet correctly identified the real changed building, whereas other methods suffer from false alarms of varying degrees. For example, DDCNN and Peng *et al.* detected the missed changed objects but produced many false alarms because they misclassified some containers as changed buildings. The above analysis demonstrated that AGCDetNet overcomes the distractions and achieves a better generalization ability than some existing methods on the challenging data. Meanwhile, AGCDetNet gains competitive performance compared with the postclassification-based method.

*4) Comparisons on Season-Varying:* Supplemental experiments were conducted on the Season-Varying dataset to evaluate the generalization ability of AGCDetNet in general CD. For comparison, five attention-based methods, i.e., DANet [58], IFN [35], BA$^2$Net [64], DASNet [38], DDCNN [37] and two U-shape-based variants, i.e., FC-EF-Res [28] and Peng *et al.* [29] were selected as benchmarks. To our best knowledge, DDCNN [37] achieves the most state-of-the-art performance

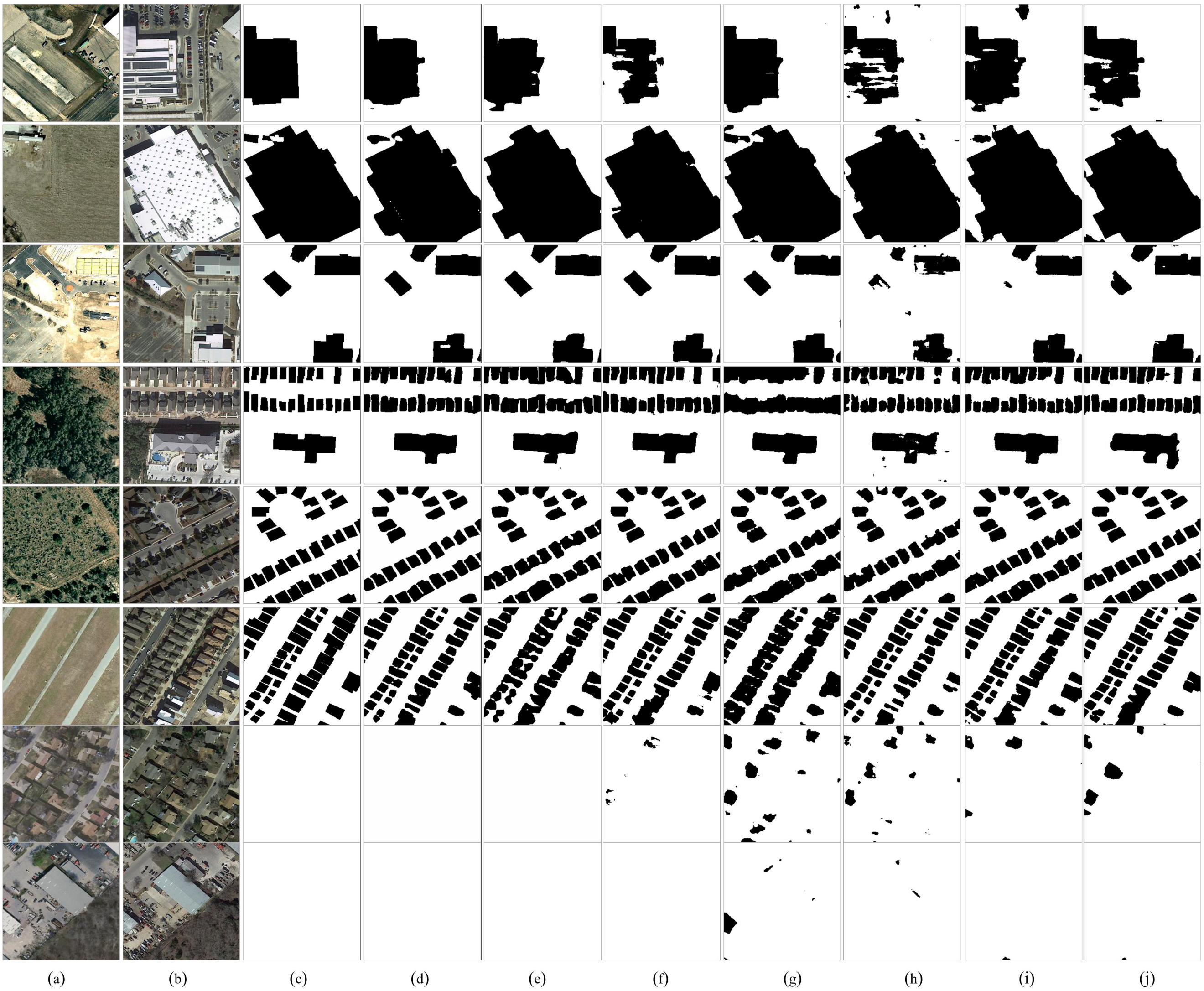


Fig. 7. CD results of the proposed AGCDetNet and other four benchmarks on the LEVIR-CD dataset. Zoom-in for an improved view. (a) Image T1. (b) Image T2. (c) Reference change map. (d) AGCDetNet. (e) FarSeg. (f) DDCNN. (g) STANet. (h) W-Net. (i) FC-EF-Res. (j) Peng *et al*. Best viewed in color.

TABLE V
Comparison of Results on the Season-Varying Dataset

| ***Method*** | ***OA*** **(%)** | ***Pr*** **(%)** | ***Re*** **(%)** | ***F₁*** |
|---|---|---|---|---|
| Peng *et al.* [29] | 96.73 | 89.54 | 87.11 | 0.8756 |
| FC-EF-Res [28] | 97.25 | 89.91 | 87.37 | 0.8862 |
| DANet [58] | 94.31 | 68.73 | 98.92* | 0.8111 |
| IFN [35] | 97.71 | 94.96 | 86.08 | 0.9030 |
| BA$^2$Net [64] | 98.94 | 88.12 | 95.28 | 0.9136 |
| DASNet [38] | 98.20 | 92.20 | 93.20 | 0.9270 |
| DDCNN [37] | 98.64 | 96.71* | 92.32 | 0.9446 |
| AGCDetNet | 99.13* | 95.03 | 98.10 | 0.9654* |

on this dataset. The $\boldsymbol{OS}$ of AGCDetNet was set to 8 due to the small size of samples.

Table V presents the quantitative results. AGCDetNet consistently outperforms other benchmarks in terms of $F_1$ and *OA*. In particular, AGCDetNet achieved the best $F_1$ (0.9654) and improved approximately 2.08% compared with state-of-the-art DDCNN with $F_1$ (0.9446). Some challenging CD results are shown in Fig. 9. The black areas indicate changed objects, and the white areas indicate the unchanged regions. The change types mainly consist of building change and road change. The change maps generated by AGCDetNet are closely consistent with the reference. Specifically, AGCDetNet achieves fine-grained road change and high internal compactness for building change. Even in the case of season variation (summer-to-winter/autumn, see the last three rows in Fig. 9), AGCDetNet accurately identified the real changed objects of various scales and appearance.

*5) Comparisons on WV2 and ZY3:* Experiments were conducted using satellite images acquired by different satellite sensors with different spatial resolutions. WV2 Site1/Site2 data were acquired by Worldview-2 with a spatial resolution of 2 m/pixel, and ZY3 data were acquired by Ziyuan 3 with a spatial resolution of 5.8 m/pixel. For comparison, the supervised FDCNN [10] and unsupervised IR-MAD [17] and PCA-kMeans [15] were selected as benchmarks. The maximum number of iterations in IR-MAD was set to 30, and the k-Means algorithm was used for binarization. The block size of PCA-kMeans was

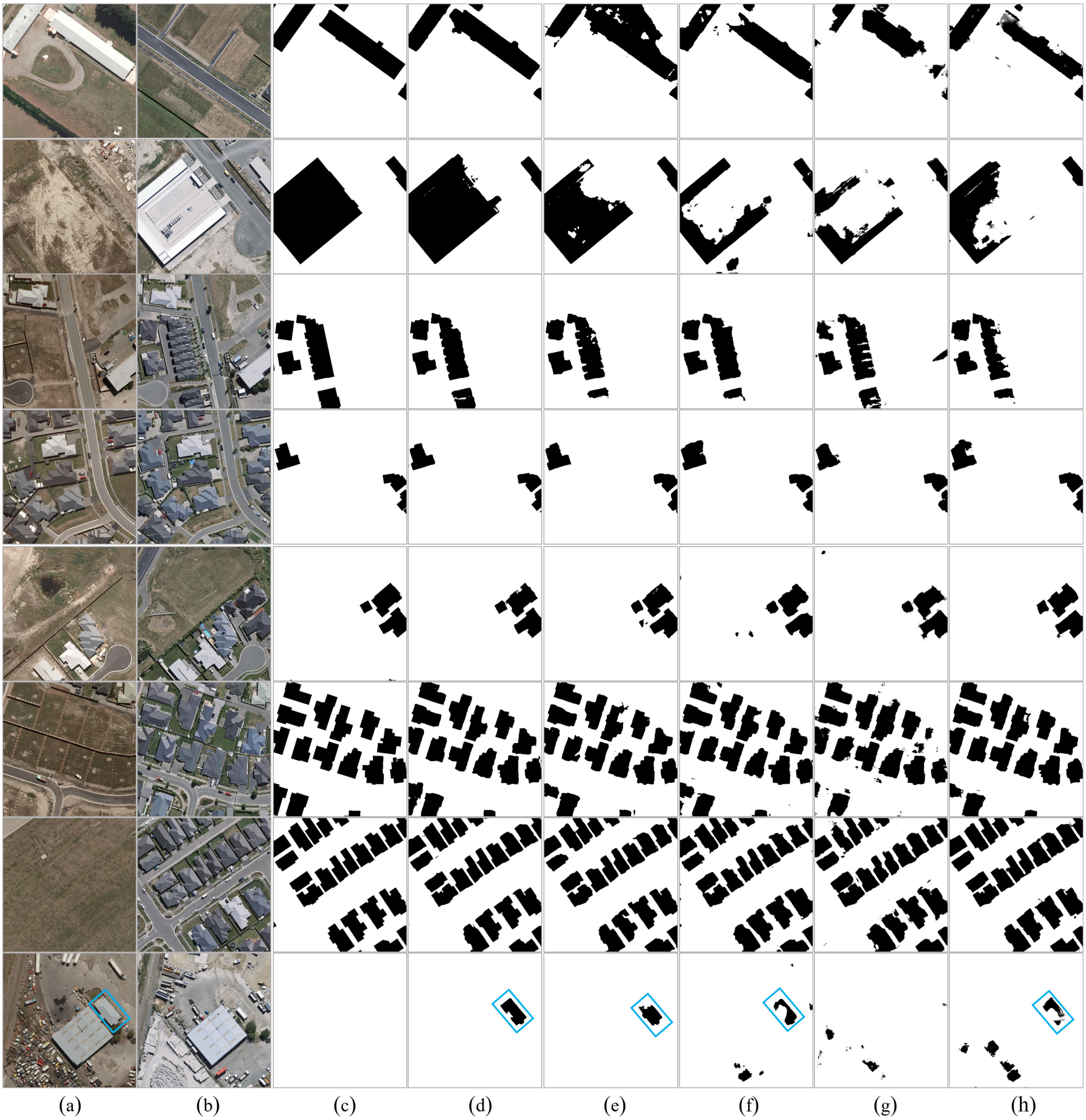


Fig. 8. CD results on the WHU dataset. Zoom-in for an improved view. The blue box marked the missed changed objects by manual interpretation. (a) Image T1. (b) Image T2. (c) Reference change map. (d) AGCDetNet. (e) FarSeg. (f) DDCNN. (g) FC-EF-Res. (h) Peng *et al*. Best viewed in color.

set to 10, and the contribution rate was set to 0.9. The supervised models were trained only using WV2 Site 1 data. To our best knowledge, FDCNN achieves the most state-of-the-art performance on this dataset.

Table VI presents the quantitative results, and CD results are shown in Fig. 10. Change types in the WV2 Site 2 and ZY3 data mainly involve building and construction land changes. We can observe that AGCDetNet exhibits a better generalization ability in images acquired by different satellite sensors with different resolutions. Compared with the other approaches, the change maps generated by AGCDetNet are closer to the reference. AGCDetNet achieves a higher *Kappa* on both test sets. AGCDetNet achieves the lowest false alarm rate and miss alarm rate on WV2 Site 2 data and gains a better false alarm rate and higher *Kappa* on ZY3 data.

## V. Discussion

For an improved understanding of why AGCDetNet works well, this section presents the visualization of the feature maps generated by the SPAM and CIFU. The visualization interpretation using heatmaps can intuitively explain what attention weights the network learns and how to emphasize change information representations.

TABLE VI
COMPARISON OF RESULTS ON THE WV2 SITE 2 AND THE ZY3 DATASET

| *Method* | *WV2 Site 2* | | | | *ZY3* | | | |
|---|---|---|---|---|---|---|---|---|
| | *FA(%)* | *MA(%)* | *OE(%)* | *Kappa* | *FA(%)* | *MA(%)* | *OE(%)* | *Kappa* |
| IR-MAD [17] | 10.62 | 46.08 | 17.14 | 0.43 | 9.33 | 32.45 | 11.79 | 0.48 |
| PCA-kMeans [15] | 8.21 | 41.83 | 14.39 | 0.51 | 4.66 | 47.51 | 9.21 | 0.50 |
| FDCNN [10] | 9.13 | 27.08 | 12.43 | 0.61 | 12.54 | 18.01* | 13.13 | 0.50 |
| AGCDetNet | 8.16* | 18.72* | 10.10* | 0.68* | 3.48* | 40.83 | 7.45* | 0.59* |

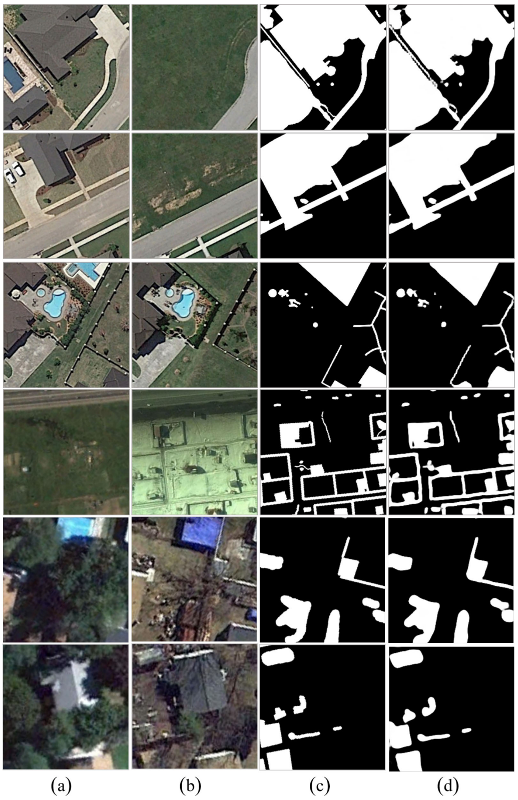


Fig. 9. CD results of AGCDetNet on the Season-Varying dataset. Zoom-in for an improved view. (a) Image T1. (b) Image T2. (c) Reference change map. (d) AGCDetNet. Best viewed in color.

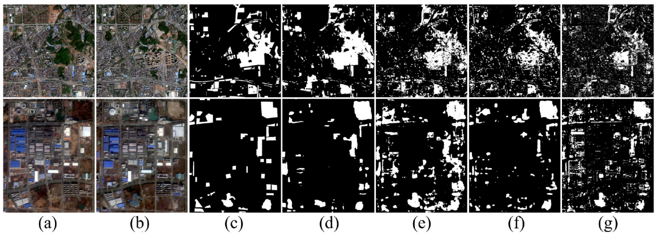


Fig. 10. CD results on the WV2 Site2 and ZY3 data. Zoom-in for an improved view. (a) Image T1. (b) Image T2. (c) Reference change map. (d) AGCDetNet. (e) FDCNN. (f) PCA-kMeans. (g) IR-MAD. Best viewed in color.

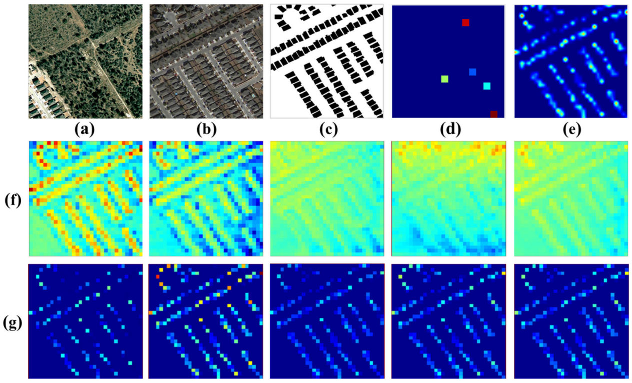


Fig. 11. Visualization of the SPAM. Zoom-in for an improved view. (a) Image T1. (b) Image T2. (c) Reference change map. (d) Sparse vector. (e) Spatial attention map. (f) Original deep features. (g) Augmented deep features. Best viewed in color.

## *A. Visualization Analysis on the SPAM*

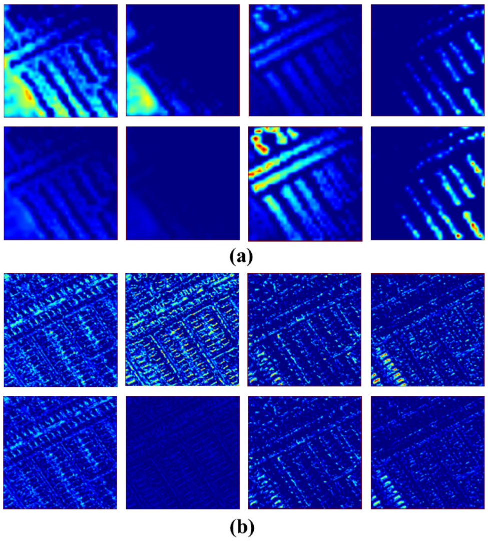
As shown in Fig. 11(d), SPAM learns a sparse vector $\hat{\boldsymbol{R}}_{\mathbf{f}} \in \mathbb{R}^{\boldsymbol{C}\times 1}$ from original deep features by incorporating the prior knowledge from the coarse change map. (To present the sparseness, the *C*-dimension feature vector $\hat{\boldsymbol{R}}_{\mathbf{f}}$ is reshaped to a 2-D feature map.) The sparse representation $\hat{\boldsymbol{R}}_{\mathbf{f}} \in \mathbb{R}^{\boldsymbol{C}\times 1}$ helps select the most important feature channels of $\boldsymbol{X}_{\boldsymbol{inner}} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$, i.e., those feature channels that gain a high response to the changed objects. Afterward, the spatial attention map $\boldsymbol{SA} \in \mathbb{R}^{1\times\boldsymbol{H}\times\boldsymbol{W}}$ encodes a more reliable change probability of each pixel in deep features $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{C}\times\boldsymbol{H}\times\boldsymbol{W}}$. The attention map highlights most pixel locations where the changed buildings are located [see Fig. 11(e)]. Besides, some feature channels of the original deep features and the corresponding augmented ones were selected to show how SPAM helps promote the discrimination between the changed objects and background. As shown in Fig. 11(f) and (g), the discrimination is substantially improved in the augmented deep features (g) compared with the original ones (f). The pixels where the changed buildings are located are emphasized in the augmented deep features using the learned spatial attention map. Meanwhile, some redundant context in deep features is effectively suppressed.

In summary, the above analysis demonstrated the effectiveness of SPAM for improving the discrimination of deep features.

## *B. Visualization Analysis on the CIFU*

Instead of visualizing the channelwise attention vectors, some refined channels and the corresponding original ones were selected to see whether channel attention highlights the task-relevant channels and suppresses the task-irrelevant ones, as shown in Fig. 12. The lighter areas in the heatmaps highlight the response of specific semantics. For example, the highlighted regions gain a high response to the changed buildings [see the second row and third column of Fig. 12(a)].

As for the channels from the high-level features $\boldsymbol{X_H}$ and $\hat{\boldsymbol{X}}_{\boldsymbol{H}}$ [see the first and second rows of Fig. 12(a)], the channels


(a)

(b)

Fig. 12. Visualization of the CIFU. Zoom-in for an improved view. (a) Channels of $\boldsymbol{X}_{\boldsymbol{H}}$ and $\hat{\boldsymbol{X}}_{\boldsymbol{H}}$. (b) Channels of $\boldsymbol{X}_{\boldsymbol{L}}$ and $\hat{\boldsymbol{X}}_{\boldsymbol{L}}$. Best viewed in color.

that gain a high response to the changed objects are enhanced to highlight the change areas after applying the CIFU (see the third and fourth columns). In contrast, the background-related channels are suppressed with low activation (see the first and second columns). As for the channels from the low-level features $\boldsymbol{X}_{\boldsymbol{L}}$ and $\hat{\boldsymbol{X}}_{bmL}$ [see the first and second rows of Fig. 12(b)], the case of suppression is more commonly observed compared with the case of enhancement because the spatial details encoded in the low-level features are abundant enough. For example, the first and second columns indicate that the response to the road between buildings is effectively suppressed. The second, third, and fourth columns suggest that the response to the unchanged buildings that lie in the bottom left corner is successfully suppressed.

In summary, CIFU tends to filter the redundant context information in the low-level features and enhance the high-level semantic features.

## VI. Conclusion

The attention-guided AGCDetNet was proposed for building CD in high-resolution RS images. AGCDetNet learns to enhance the feature representations of the change information through the attention-based modules. AGCDetNet achieves accuracy improvements in two ways. One is by incorporating the SPAM to increase the discrimination between the change objects and background in deep features. Another is by applying CGASPP and CIFU to enhance the representations of multiscale context and multilevel features. Ablation studies have verified the contribution of the core components. Extensive experiments on four datasets have shown that AGCDetNet exhibited better performance compared with existing approaches. AGCDetNet exhibits the advantage of alleviating the adverse effects of scale variations and the false alarms caused by pseudochanges. Specifically, AGCDetNet achieved the best $F_1$ on two datasets, i.e., LEVIR-CD (0.9076) and Season-Varying (0.9654).

## References

[1] A. Singh, “Review article digital change detection techniques using remotely-sensed data,” *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, Jun. 1989.

[2] M. Bouziani, K. Goïta, and D.-C. He, “Automatic change detection of buildings in urban environment from very high spatial resolution images using existing geodatabase and prior knowledge,” *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 1, pp. 143–153, Jan. 2010.

[3] X. Huang, L. Zhang, and T. Zhu, “Building change detection from multitemporal high-resolution remotely sensed images based on a morphological building index,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 1, pp. 105–115, Jan. 2014.

[4] M. Pesaresi, D. Ehrlich, I. Caravaggi, M. Kauffmann, and C. Louvrier, “Toward global automatic built-up area recognition using optical VHR imagery,” in *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 4, pp. 923–934, Dec. 2011.

[5] P. Xiao, X. Zhang, D. Wang, M. Yuan, X. Feng, and M. Kelly, “Change detection of built-up land: A framework of combining pixel-based detection and object-based recognition,” *ISPRS J. Photogramm. Remote Sens.*, vol. 119, pp. 402–414, Sep. 2016.

[6] H. Chen and Z. Shi, “A spatial-temporal attention-based method and a new dataset for remote sensing image change detection,” *Remote Sens.*, vol. 12, no. 10, May 2020, Art. no. 1662.

[7] A. Fujita, K. Sakurada, T. Imaizumi, R. Ito, S. Hikosaka, and R. Nakamura, “Damage detection from aerial images via convolutional neural networks,” in *Proc. 15th IAPR Int. Conf. Mach. Vision Appl.*, 2017, pp. 5–8.

[8] S. Ji, S. Wei, and M. Lu, “Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 574–586, Jan. 2019.

[9] R. Gupta *et al.*, “Creating xBD: A dataset for assessing building damage from satellite imagery,” in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. Workshops*, 2019, pp. 10–17.

[10] M. Zhang and W. Shi, “A feature difference convolutional neural network-based change detection method,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 10, pp. 7232–7246, Oct. 2020.

[11] S. Ji, Y. Shen, M. Lu, and Y. Zhang, “Building instance change detection from large-scale aerial images using convolutional neural networks and simulated samples,” *Remote Sens.*, vol. 11, no. 11, Jun. 2019, Art. no. 1343.

[12] L. Bruzzone and D. F. Prieto, “Automatic analysis of the difference image for unsupervised change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1171–1182, May 2000.

[13] R. D. Johnson and E. S. Kasischke, “Change vector analysis: A technique for the multispectral monitoring of land cover and condition,” *Int. J. Remote Sens.*, vol. 19, no. 3, pp. 411–426, Nov. 1998.

[14] A. Yavariabdi and H. Kusetogullari, “Change detection in multispectral Landsat images using multiobjective evolutionary algorithm,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 414–418, Mar. 2017.

[15] T. Celik, “Unsupervised change detection in satellite images using principal component analysis and k-means clustering,” *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 4, pp. 772–776, Oct. 2009.

[16] J. S. Deng, K. Wang, Y. H. Deng, and G. J. Qi, “PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data,” *Int. J. Remote Sens.*, vol. 29, no. 16, pp. 4823–4838, 2008.

[17] A. A. Nielsen, “The regularized iteratively reweighted MAD method for change detection in multi- and hyperspectral data,” *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 463–478, Feb. 2007.

[18] M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley, “Change detection from remotely sensed images: From pixel-based to object-based approaches,” *ISPRS J. Photogramm. Remote Sens.*, vol. 80, pp. 91–106, Jun. 2013.

[19] P. Lv, Y. Zhong, J. Zhao, and L. Zhang, “Unsupervised change detection based on hybrid conditional random field model for high spatial resolution remote sensing imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 4002–4015, Jul. 2018.

[20] M. Volpi, D. Tuia, F. Bovolo, M. Kanevski, and L. Bruzzone, “Supervised change detection in VHR images using contextual information and support vector machines,” *Int. J. Appl. Earth Observ. Geoinf.*, vol. 20, pp. 77–85, Feb. 2013.

[21] H. Nemmour and Y. Chibani, "Multiple support vector machines for land cover change detection: An application for mapping urban extensions," *ISPRS J. Photogramm. Remote Sens.*, vol. 61, no. 2, pp. 125–133, Nov. 2006.

[22] D. Lu, P. Mausel, E. Brondízio, and E. Moran, "Change detection techniques," *Int. J. Remote Sens.*, vol. 25, no. 12, pp. 2365–2401, Jun. 2004.

[23] A. P. Tewkesbury, A. J. Comber, N. J. Tate, A. Lamb, and P. F. Fisher, "A critical synthesis of remotely sensed optical image change detection techniques," *Remote Sens. Environ.*, vol. 160, pp. 1–14, Apr. 2015.

[24] W. Shi, Z. Min, R. Zhang, S. Chen, and Z. Zhan, "Change detection based on artificial intelligence: State-of-the-art and challenges," *Remote Sens.*, vol. 12, May 2020, Art. no. 1688.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3431–3440.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[28] R. C. Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Multitask learning for large-scale semantic change detection," *Comput. Vision Image Understanding*, vol. 187, Oct. 2019, Art. no. 102783.

[29] D. Peng, Y. Zhang, and H. Guan, "End-to-end change detection for high resolution satellite images using improved UNET++," *Remote Sens.*, vol. 11, no. 11, Jun. 2019, Art. no. 1382.

[30] R. C. Daudt, B. L. Saux, and A. Boulch, "Fully convolutional Siamese networks for change detection," in *Proc. 25th IEEE Int. Conf. Image Process.*, Athens, Greece, 2018, pp. 4063–4067.

[31] H. Chen, C. Wu, B. Du, L. Zhang, and L. Wang, "Change detection in multisource VHR images via deep Siamese convolutional multiple-layers recurrent neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2848–2864, Apr. 2020.

[32] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change detection based on deep Siamese convolutional network for optical aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1845–1849, Oct. 2017.

[33] J. Liu, M. Gong, K. Qin, and P. Zhang, "A deep convolutional coupling network for change detection based on heterogeneous optical and radar images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 545–559, Mar. 2018.

[34] B. Hou, Q. Liu, H. Wang, and Y. Wang, "From W-Net to CDGAN: Bitemporal change detection via deep learning techniques," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1790–1802, Mar. 2020.

[35] C. Zhang *et al.*, "A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 166, pp. 183–200, Aug. 2020.

[36] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 714–722.

[37] X. Peng, R. Zhong, Z. Li, and Q. Li, "Optical remote sensing image change detection based on attention mechanism and image difference," in *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TPAMI.2019.2933510.

[38] J. Chen *et al.*, "DASNet: Dual attentive fully convolutional Siamese networks for change detection of high resolution satellite images," in *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1194–1206, Nov. 2020, doi: 10.1109/JSTARS.2020.3037893.

[39] M. Lebedev, Y. V. Vizilter, O. Vygolov, V. Knyaz, and A. Y. Rubis, "Change detection in remote sensing images using conditional adversarial networks," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 42, no. 2, pp. 565–571, Jun. 2018.

[40] M. Gong, Y. Yang, T. Zhan, X. Niu, and S. Li, "A generative discriminatory classified network for change detection in multispectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 1, pp. 321–333, Jan. 2019.

[41] X. Niu, M. Gong, T. Zhan, and Y. Yang, "A conditional adversarial network for change detection in heterogeneous images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 1, pp. 45–49, Jan. 2019.

[42] W. Zhao, L. Mou, J. Chen, Y. Bo, and W. J. Emery, "Incorporating metric learning and adversarial network for seasonal invariant change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2720–2731, Apr. 2020.

[43] 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.*, vol. 57, no. 2, pp. 924–935, Feb. 2019.

[44] H. Lyu, H. Lu, and L. Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sens.*, vol. 8, no. 6, Jun. 2016, Art. no. 506.

[45] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised deep change vector analysis for multiple-change detection in VHR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3677–3693, Jun. 2019.

[46] B. Hou, Y. Wang, and Q. Liu, "Change detection based on deep features and low rank," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2418–2422, Dec. 2017.

[47] M. Zhang, G. Xu, K. Chen, M. Yan, and X. Sun, "Triplet-based semantic relation learning for aerial remote sensing image change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 266–270, Feb. 2018.

[48] M. Yang, L. Jiao, F. Liu, B. Hou, and S. Yang, "Transferred deep learning-based change detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6960–6973, Sep. 2019.

[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015.

[50] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 2961–2969.

[51] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Las Vegas, NV, USA, 2015, pp. 4353–4361.

[52] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[53] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 7794–7803.

[54] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 7132–7141.

[55] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vision*, 2020, pp. 173–190.

[56] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 3–19.

[57] Z. Zheng, Y. Zhong, J. Wang, and A. Ma, "Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2020, pp. 4095–4104.

[58] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 3141–3149.

[59] G. Cheng, Y. Si, H. Hong, X. Yao, and L. Guo, "Cross-scale feature fusion for object detection in optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 431–435, Mar. 2021.

[60] J. Han, X. Yao, G. Cheng, X. Feng, and D. Xu, "P-CNN: Part-based convolutional neural networks for fine-grained visual categorization," in *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2019.2933510.

[61] X. Li, A. Yuan, and X. Lu, "Vision-to-language tasks based on attributes and attention mechanism," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 913–926, Feb. 2021.

[62] X. Lu, B. Wang, and X. Zheng, "Sound active attention framework for remote sensing image captioning," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1985–2000, Mar. 2020.

[63] T. Bao, C. Fu, S. Sirajidin, T. Fang, and H. Huo, "Dual pyramid attention network for high-resolution remotely sensed image change detection," in *Proc. 12th Int. Conf. Mach. Learn. Comput.*, 2020, pp. 259–265.

[64] Y. Zhang, S. Zhang, Y. Li, and Y. Zhang, "Coarse-to-fine satellite images change detection framework via boundary-aware attentive network," *Sensors*, vol. 20, no. 23, Nov. 2020, Art. no. 6735.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.

[66] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.

[67] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 636–644.

[68] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 801–818.

[69] V. Iglovikov, S. S. Seferbekov, A. Buslaev, and A. Shvets, “TernausNetV2: Fully convolutional network for instance segmentation,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, p. 237.

[70] A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[71] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *Proc. 7th Int. Conf. Learn. Represent.*, 2019.

[72] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1026–1034.

**Kaiqiang Song** received the B.S. degree in measurement and control technology and instrument from the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China, in 2018. He is currently working toward the M.S. degree in instrument science and technology with the School of Instrumentation and Optoelectronic Engineering, Key Laboratory of Precision Opto-Mechatronics Technology, Ministry of Education, Beihang University, Beijing, China.

His research interests include deep learning and multitemporal remote sensing image change detection.

**Jie Jiang** received the B.S. and Ph.D. degrees in optoelectronics engineering from Tianjin University, Tianjin, China, in 1995 and 2000, respectively.

She is currently a Professor with the School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing, China. She has authored more than 50 articles and 30 inventions. Her research interests include image processing and star sensors.