ISNet: Towards Improving Separability for Remote Sensing Image Change Detection

Gong Cheng, *Member, IEEE*, Guangxing Wang, and Junwei Han, *Fellow, IEEE*

***Abstract*— Deep learning has substantially pushed forward remote sensing image change detection through extracting discriminative hierarchical features. However, as the increasingly high-resolution remote sensing images have abundant spatial details but limited spectral information, the use of conventional backbone networks would give rise to blurry boundaries between different semantics among hierarchical features. This explains why most false alarms in the final predictions distribute around change boundaries. To alleviate the problem, we pay attention to feature refinement and propose deep learning networks that deliver improved separability (ISNet). Our ISNet reaps the advantages from two strategies applied to refining bitemporal feature hierarchies: 1) margin maximization that clarifies the gap between changed and unchanged semantics and 2) targeted arrangement of attention mechanisms that direct the use of channel attention (CA) and spatial attention (SA) for highlighting semantic and positional information, respectively. Specifically, we insert CA modules into share-weighted backbone networks to facilitate semantic-specific feature extraction. The semantic boundaries in the extracted bitemporal hierarchical features are then clarified by margin maximization modules, followed by SA modules to enhance positional change responses. A top–down fusion pathway makes the final refined features cover multiscale representations and have strong separability for remote sensing image change detection. Extensive experimental evaluations demonstrate that our ISNet achieves state-of-the-art performance on the LEVIR-CD, SYSU-CD, and Season-Varying datasets in terms of overall accuracy (OA), Intersection-of-Union (IoU), and F1 score. Code is available at https://github.com/xingronaldo/ISNet.**

***Index Terms*— Attention mechanisms, change detection, deep learning.**

I. Introduction

DETECTING land-cover changes using bitemporal remote sensing images has practical uses in various applications, including land management, damage assessment, and environment monitoring [1]–[3]. Given bitemporal images showing altered spectral behavior [1], change detection aims to discriminate those spectral alterations caused by changes of interest from those brought by not exactly consistent imaging conditions. Image registration and radiometric correction are indispensable image preprocessing procedures to eliminate the negative effects of geometric and radiometric factors [4].

Traditional change detection approaches evolved in relation to the basic analysis units, i.e., from independent image pixels to segmented objects involving contextual information [5]–[7]. Most pixel-based methods work in an unsupervised manner [8]. First, a difference image is generated through simple arithmetical operations (e.g., image differencing and image rationing), simple transformations (e.g., change vector analysis and principal component analysis), or a combination of both [9]. Then, the change map of interest is obtained by thresholding or clustering analysis on the difference image [10], [11]. The rise of the postclassification comparison paradigm enabled supervised learning on large volumes of available data [9], [12], [13]. While the pixel-based approaches utilized spectral information independently, the object-based methods emerged to cope with spectral variation in very-high-resolution (VHR) remote sensing images by allowing the exploitation of spatial context in segmented objects [1]. However, the performance of classic object-based methods is still heavily limited by: 1) the normally handcrafted shadow features that encode insufficient variation and 2) the problem of error accumulation from object segmentation to change detection [14], [15]. Overall, traditional change detection approaches struggle to detect changes within the increasingly high-resolution remote sensing images.

Recent advances in deep learning have dispersed into the field of remote sensing image change detection. Convolutional neural networks (CNNs), such as ResNet series [16] and UNet series [17], are commonly leveraged as the backbone networks to help extract discriminative hierarchical features [18], [19] from bitemporal remote sensing images. One major division in deep learning-based change detection approaches is "early-fusion" versus "late-fusion" [20]. The methods that realize early fusion integrate bitemporal information at the image level, i.e., the network input. For example, Zheng *et al.* [21] stacked bitemporal images along the channel dimension and proposed a cross-layer CNN to incorporate multiscale features and multilevel contexts. By contrast, the methods that realize late fusion integrate bitemporal information at the feature level. In general, share-weighted (a.k.a., Siamese-based) backbone networks are first used to extract bitemporal features, separately. The extracted features are then processed and fused for downstream decision-making. Zhang *et al.* [22] followed this scheme and proposed a deeply supervised image fusion network to deal with the complexity of VHR images. Our

Manuscript received January 29, 2022; revised April 19, 2022; accepted May 7, 2022. Date of publication May 11, 2022; date of current version June 1, 2022. This work was supported in part by the National Science Foundation of China under Grant 62136007, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021B1515020072, and in part by the Fundamental Research Funds for the Central Universities. *(Corresponding author: Gong Cheng.)*

The authors are with the School of Automation, Northwestern Polytechnical University, Xi'an 710129, China, and also with the Research and Development Institute of Northwestern Polytechnical University in Shenzhen, Shenzhen 518057, China (e-mail: gcheng@nwpu.edu.cn).

Digital Object Identifier 10.1109/TGRS.2022.3174276

proposed change detection method also accords with the scheme.

The long-standing idea of strengthening change information and suppressing unchanged information [23] is also applicable to the current deep learning-based approaches. Attention mechanisms serve as the technical carrier [24]. Recent years have seen various methods empowered by attention mechanisms developed for enhancing the separability of deep learning features [25]–[31]. To name a few, Liu *et al.* [25] and Shi *et al.* [26] used convolutional block attention modules [32] that assemble channel attention (CA) and spatial attention (SA) to optimize hierarchical features. Chen *et al.* [27] revised vision transformer [33] that conveys self-attention to refine the features produced in the last layer of CNN-based backbones. The above methods processed bitemporal features independently and performed bitemporal feature fusion at the very end of feature refinement. To respect the feature-level temporal correlation [9], our proposed method processes and fuses bitemporal features progressively.

The development of multispectral imaging makes it conveniently accessible to VHR remote sensing images, which contains abundant spatial details to delicately describe texture, shape, and so on [3], [4]. However, the limited spectral information in VHR images brings low interclass variation (and high intraclass variation) [4] and, thus, poses great challenges to change detection. On the other hand, the use of plain convolutions in conventional backbone networks (e.g., ResNet series and UNet series) produces regular reception fields. Due to the data characteristic and the network property, the loss of detailed information during feature extraction would give rise to blurry boundaries between different semantics among hierarchical features. As a result, the predictions around change boundaries would be incredible, and a plethora of false alarms are raised. Fang *et al.* [31] stressed the importance of high-resolution low-level features that are correlated with plentiful spatial details and proposed ensemble CA for deep supervision. However, there were no direct constraints exerted on change boundaries, irrespective of the use of low-level features or attention mechanisms. In this article, we propose to strategically refine the extracted hierarchical features to alleviate the problem. In the process of feature refinement, we conduct margin maximization that constrains change boundaries directly to clarify the gap between changed and unchanged semantics in the bitemporal features produced at each stage. To promote this end, we propose a targeted arrangement of attention mechanisms to direct the use of CA and SA for highlighting semantic and positional information, respectively. We design a top–down fusion pathway, which makes the final refined features cover multiscale representations and have strong separability for remote sensing image change detection. With the above strategies and elaborate architectural designs, we propose deep learning networks that deliver improved separability (ISNet).

Our contributions are summarized as follows.

1) To tackle blurry boundaries between different semantics, we propose a margin maximization strategy to clarify the gap between changed and unchanged semantics at each level of bitemporal feature hierarchies.
2) We employ a targeted arrangement of attention mechanisms to promote the end. Specifically, we utilize CA to facilitate semantic-specific feature extraction and SA to enhance feature refinement, respectively. We demonstrate that such a targeted arrangement allows the appropriate use of several plug-and-play attention modules for improving change detection results.
3) We propose deep learning networks that deliver improved separability (ISNet) with a combination of the above two strategies and elaborate architectures. With ResNet series backbones, our ISNet achieves state-of-the-art performance on three public datasets for remote sensing image change detection. Furthermore, we show that a set of lightweight backbones equipped with our strategical and architectural designs perform acceptably.

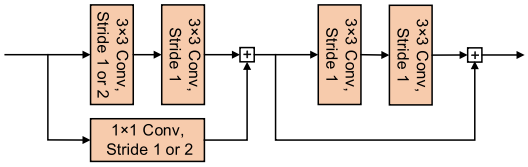

Fig. 1. Illustration of a residual block.

The rest of this article is structured as follows. Section II introduces some preliminary knowledge of residual networks (ResNets) and attention mechanisms briefly. Section III describes our algorithmic designs in detail. Section IV provides experimental evaluations. Section V concludes this article.

II. Preliminaries

A. Residual Networks

The overall architecture of ResNets [16] is built by stacking residual blocks. One forward pass of a residual block normally corresponds to one-time downsampling and, thus, is termed a stage. A residual block comprises two residual units, each of which has two convolutional layers and a skip connection, as shown in Fig. 1. Specifically, the stride of the first convolutional layer in the first residual unit and that of the convolutional layer in the first skip connection are optional. When the two strides are set to 1, the input feature maps skip the convolution operation in the skip connection, and their size will be maintained after passing through the residual block. Downsampling is achieved by setting them to 2 indeed.

Model scaling derives a series of ResNet models (including ResNet variants) with similar architectural designs but different depths or widths, such as ResNet-18/34/50, ResNeXt-50_32 $\times$ 4d [34], and Wide_ResNet-50_2 [35]. This article transfers the use of ResNet series models pretrained on ImageNet as the backbone networks for feature extraction.

B. Attention Mechanisms

Attention mechanisms lead to significant improvements in both performance and interpretability for deep learning

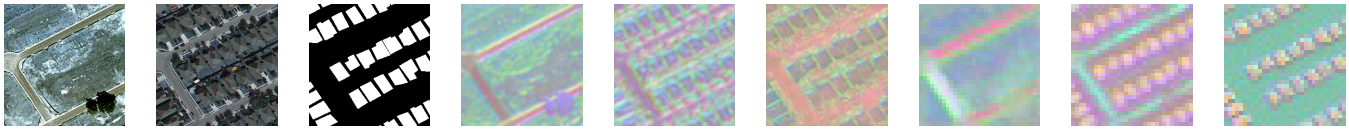

Fig. 2. (a)–(c) Illustrations of T1 instance, T2 instance, and corresponding label. (d)–(f) Visualizations of the feature of T1 instance, the feature of T2 instance, and the feature after margin maximization (MM feature), produced at stage 1. (g)–(i) Visualizations of the feature of T1 instance, the feature of T2 instance, and the feature after margin maximization, produced at stage 2. These features are obtained from a well-trained model. For visualization, we resort to principal component analysis with three bands retained.

architectures [36]. In the fields of computer vision and remote sensing, CA, SA, and self-attention prevail in algorithmic designs. CA allows deep learning models to learn what semantics should be highlighted [37]. It emphasizes specific channels in feature maps by learning different weights for each channel [29]. SA allows deep learning models to differentiate the spatial positions that should be highlighted [37]. It emphasizes specific spatial positions in feature maps by learning different weights for each spatial position [29]. Self-attention learns weights that reflect the correlation between each spatial element in feature maps to capture long-range dependencies [33], [37]–[39]. Considering that applying self-attention would introduce extensive computational burdens, we restrict our attention to the appropriate use of CA and SA in this article.

III. Change Detection With Improved Separability

We begin this section by formulating the problem of bitemporal remote sensing image change detection. Then, we elaborate on our strategical and architectural designs with bitemporal image characteristics taken into consideration. Finally, we present our loss function for training the whole framework.

A. Problem Statement

Bitemporal remote sensing image change detection analyzes a pair of remotely sensed images that record the same geographical area at different times T1 and T2. Let $\mathcal{I}_{\mathrm{T1}} \subset \mathbb{R}^{3\times H\times W}$, $\mathcal{I}_{\mathrm{T2}} \subset \mathbb{R}^{3\times H\times W}$, and $\mathcal{Y} \subset \mathbb{R}^{H\times W}$ be the domains of T1 instances [e.g., nonoverlapped patches in the T1 image; see Fig. 2(a)], T2 instances [e.g., nonoverlapped patches in the T2 image; see Fig. 2(b)], and their binary change labels [see Fig. 2(c)], respectively. H and W are the instance height and width. Formally, the task of change detection aims to learn a mapping as follows:

$$\{\mathcal{I}_{\mathrm{T1}}, \mathcal{I}_{\mathrm{T2}}\} \longrightarrow \mathcal{Y}. \tag{1}$$

In this article, we format the algorithmic pipeline of change detection as a sequential combination of three key parts.

Step i (Feature Extraction): Share-weighted backbone networks transform T1 instances and T2 instances into multilevel hierarchical features $\{f_{\mathrm{T1}}^1, f_{\mathrm{T1}}^2, f_{\mathrm{T1}}^3, f_{\mathrm{T1}}^4\}$ and $\{f_{\mathrm{T2}}^1, f_{\mathrm{T2}}^2, f_{\mathrm{T2}}^3, f_{\mathrm{T2}}^4\}$, respectively. Here, these superscripts index the hierarchical features that come from different stages of the share-weighted backbone networks.

Step ii (Feature Refinement): The bitemporal feature hierarchies are processed and fused strategically, rendering the final refined features f_{Refined} that cover multiscale representations.

Step iii (Decision-Making): A two-class (i.e., "changed" versus "unchanged") classification operating on f_{Refined} completes the mapping presented in (1). We follow this formulation and propose deep learning networks that deliver improved separability (ISNet) with ResNet series backbones, as shown in Fig. 3.

B. Strategy (i): Margin Maximization

The increasingly high-resolution remote sensing images are characterized by abundant spatial details but limited spectral information, leading to low interclass variation. In conventional backbone networks, the use of plain convolutions produces regular reception fields. However, changed objects in remote sensing images are normally oriented and have irregular shapes. Under the circumstances, the loss of detailed information during feature extraction tends to incur blurry boundaries between different semantics among hierarchical features. We can get an intuitive illustration of the problem from Fig. 2, in which (d), (e), (g), and (h) visualize the features of the T1 instance at stage 1, T2 instance at stage 1, T1 instance at stage 2, and T2 instance at stage 2, respectively.

To alleviate the problem and improve the changed/unchanged separability, we propose to maximize the margin, i.e., to clarify the gap, between changed and unchanged semantics through strategical feature refinement [see Fig. 2(f) and (i)]. We consider that it is counterintuitive to neglect the inherent temporal correlation between bitemporal hierarchical features produced at each stage. Thus, we introduce margin maximization modules to operate on feature pairs $(f_{\mathrm{T1}}^1, f_{\mathrm{T2}}^1)$, $(f_{\mathrm{T1}}^2, f_{\mathrm{T2}}^2)$, $(f_{\mathrm{T1}}^3, f_{\mathrm{T2}}^3)$, and $(f_{\mathrm{T1}}^4, f_{\mathrm{T2}}^4)$.

Fig. 4 illustrates the details of a margin maximization module. We let each feature pair in $\{(f_{\mathrm{T1}}^i, f_{\mathrm{T2}}^i), i = 1, 2, 3, 4\}$ go through margin maximization and stage-level fusion. A deformable convolutional layer [40], [41] learns to clarify the gap between changed and unchanged semantics, given the T2 feature that carries changed semantics and the learned offset from convolving the concatenation of the T1 feature and the T2 feature. Inspired by Huang *et al.* [41], Fu *et al.* [42], Xu *et al.* [43], and Wang *et al.* [44], we propagate the T1 feature using a long-range skip connection and concatenate it with the feature after margin maximization (MM feature) to complete stage-level fusion. We empirically find that it performs well in this manner.

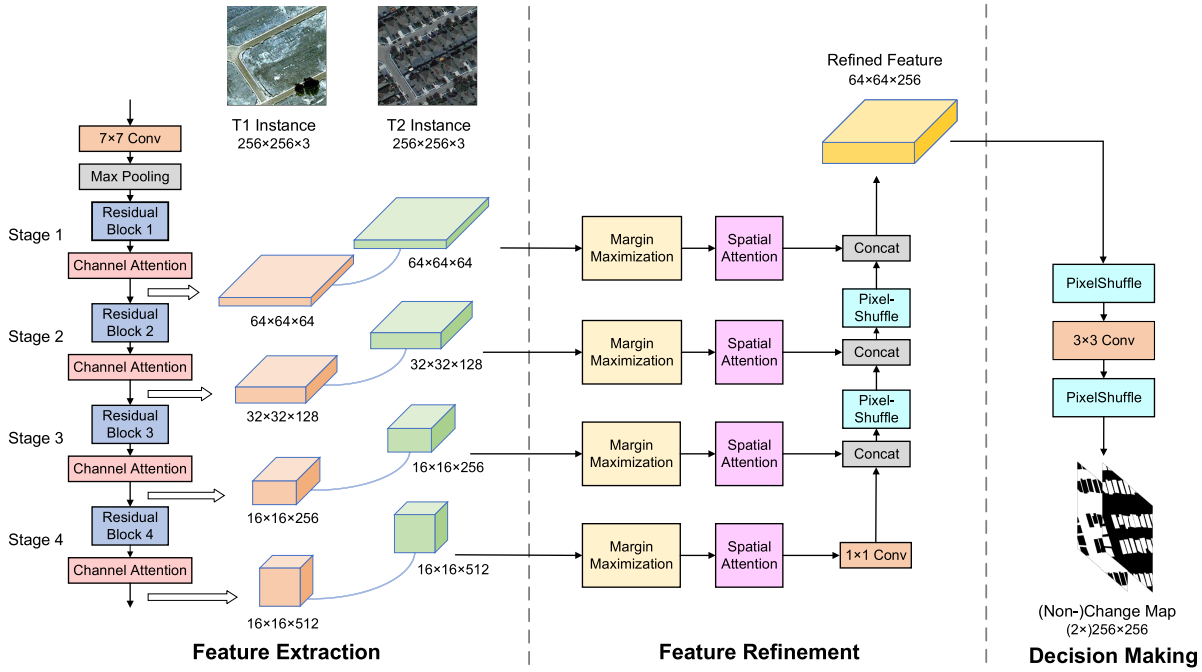

Fig. 3. **Overview of the proposed ISNet. Feature Extraction:** bitemporal instances (i.e., T1 Instance and T2 Instance) share the same ResNet series backbones and derive respective hierarchical features. CA modules are inserted into each stage to facilitate semantic-specific feature extraction. **Feature Refinement:** Top–down fusion pathway operates on bitemporal feature hierarchies to create a compact refined feature covering different scales. Two types of modules contribute to the refinement at each stage. Margin maximization modules clarify the gap between changed and unchanged semantics and realize stage-level bitemporal feature fusion. SA modules then highlight positional change responses in the stage-level fused features. We use PixelShuffle to realize upsampling in the process of top–down fusion. **Decision-Making:** Simple classifier consisting of two PixelShuffle operations and a convolution layer receives as input the refined feature and transforms it into the nonchange map and the change map of interest.

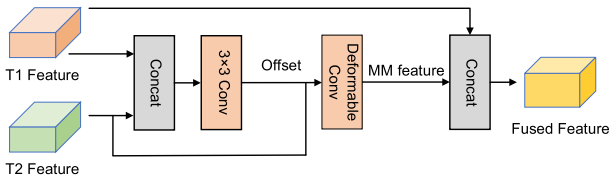

Fig. 4. Illustration of a margin maximization module.

We formally analyze why deformable convolution can serve as the margin maximization function in the following. Let $\forall\ (i, j)$, $i \in \{-1, 0, 1\}$, $j \in \{-1, 0, 1\}$ enumerate a regular 3×3 grid, which underlies a patch x in the feature map. Let w be the weights to be learned. The calculation in a plain convolution takes the following form:

$$\mathrm{Conv}\left(x_{(0,0)}\right) = \sum_{\forall(i,j)} w_{(i,j)} x_{(i,j)}. \tag{2}$$

In a deformable convolution, a horizontal offset Δi and a vertical offset Δj are introduced for each index (i, j). The calculation takes the following form:

$$\mathrm{Deformable\_Conv}\left(x_{(0,0)}\right) = \sum_{\forall(i,j)} w_{(i,j)} x_{(i+\Delta i, j+\Delta j)}. \tag{3}$$

$\forall\ (i + \Delta i, j + \Delta j)$, $i \in \{-1, 0, 1\}$, $j \in \{-1, 0, 1\}$ determines a nonrigid polygon that disorganizes the previously regular shape of reception fields. This enables learning to match the boundaries of changed semantics with different orientations and irregular shapes. The change and nonchange responses differ gradually from each other in the learning such that the ambiguity around change boundaries is largely removed. In other words, the gap between changed and unchanged semantics is enlarged and clarified. With such a direct refinement related to change boundaries, the separability of bitemporal hierarchical features is improved consequently.

C. Strategy (ii): Targeted Arrangement of Attention Mechanisms

As bitemporal instances share the use of backbone networks, their varying appearances complicate feature extraction inevitably. The resulting unmatched semantic and positional information would make feature-level boundaries incredible.

We propose to target the use of attention mechanisms to assist in margin maximization. We revisit the functions of CA and SA, and the roles of feature extraction and feature refinement. Without loss of generality, we decompose CA and SA from the convolutional block attention modules in [32] and arrange them to act in feature extraction and feature refinement, respectively. Through experimental evaluations, we further demonstrate that other off-the-shelf plug-and-play attention modules also work in this way.

1) Channel Attention in Feature Extraction: Changes are semantic-specific in most remote sensing applications, e.g., newly built buildings in land management. An in-depth insight into CA is that it highlights semantic-specific responses to

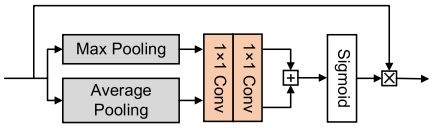

Fig. 5. Illustration of a CA module.

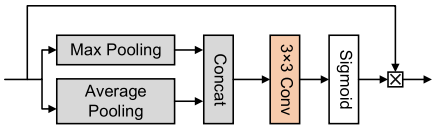

Fig. 6. Illustration of an SA module.

improve the representations of specific semantics [45]. Feature extraction in deep learning models is exactly a process of representation learning [46]. In light of this agreement, we insert CA modules into share-weighted backbone networks to facilitate semantic-specific feature extraction.

Fig. 5 illustrates the diagram of a CA module [32]. The weighting function consists of a max-pooling layer, an average pooling layer, two 1×1 convolutional layers, and a Sigmoid activation. The two pooling layers are spatialwise, reshaping the input 3-D feature maps to 1-D vectors by squeezing out the spatial dimension. The derived two branches of vectors share the two 1×1 convolutional layers, which contains full adjustable parameters. A pointwise summation integrates the learned two-branch weights, followed by the final Sigmoid activation to normalize. The learned weights are assigned correspondingly to the channels of original feature maps, and multiplication is performed.

2) Spatial Attention in Feature Refinement: Positional information is of significant importance for change detection, which obtains predictions based on bitemporal input. The positional emphasis for bitemporal features should match exactly. On the other hand, feature refinement aims at processing the extracted bitemporal features jointly to enhance their separability for facilitating the final spatialwise decision-making. Thus, we put SA modules behind margin maximization modules in the process of feature refinement to emphasize positional change responses of the stage-level fused features.

Fig. 6 illustrates the diagram of an SA module [32]. The weighting function consists of a max-pooling layer, an average pooling layer, a concatenation operation, a 3×3 convolutional layer, and a Sigmoid activation. It differs from the weighting function of CA on two sides. First, the two pooling layers are channelwise, reshaping the input 3-D feature maps to 2-D matrices by squeezing out the channel dimension. Second, the learned weights are assigned correspondingly to the spatial elements of original feature maps, and multiplication is performed.

D. Architectural Designs

We propose ISNet empowered by the above two strategies, as shown in Fig. 3. The overall architecture of our ISNet consists of: 1) share-weighted backbone networks for feature extraction; 2) a top–down fusion pathway for feature refinement; and 3) a simple classifier for decision-making.

TABLE I

ARCHITECTURE OF THE LIGHTWEIGHT BACKBONES. "BN" MEANS BATCH NORMALIZATION

Index	Filter	Stride	Padding	BN	Activation
Layer 1	3*3 Conv, $N/8$	2	1	Yes	Hardswish
Layer 2	3*3 Conv, $N/4$	2	1	Yes	Hardswish
Layer 3	3*3 Conv, $N/2$	2	1	Yes	Hardswish
Layer 4	3*3 Conv, N	2	1	Yes	Hardswish

1) Backbone Networks for Feature Extraction: We select ResNet pretrained on ImageNet as the backbone networks in our ISNet to derive bitemporal hierarchical features given bitemporal instances. The used ResNet series backbones have four residual blocks. CA modules are inserted into each stage to facilitate semantic-specific feature extraction (see the leftmost of Fig. 3 for an illustration). In our ISNet, we set the two strides in the first three residual blocks to 2 and those in the last residual blocks to 1. This renders the same spatial size for bitemporal features produced at the last two stages, avoiding that the high-level features are too small.

Besides, we introduce a set of lightweight backbone networks that allow extremely efficient feature extraction. Our intuition is to leverage the effectiveness of ImageNet pretraining to reduce the large volume of parameters in the backbone networks. To this end, we build the architecture of lightweight backbones identical to that of the convolutional part in LeViT [39]. As listed in Table I, there are simply four convolutional layers that perform continuous downsampling. Varying the basic number of filters N (i.e., the number of filters in the last layer), $N \in \{128, 192, 256, 384\}$ following LeViT, results in a set of lightweight backbones. To avoid confusion, we term our models equipped with these lightweight backbones as ISNet-lw.

2) Top–Down Fusion Pathway for Feature Refinement: We design a top–down fusion pathway to process and fuse bitemporal hierarchical features, strategically and progressively (see Fig. 3 (middle) for an illustration). As the feature maps at different stages normally have different spatial resolutions, they exhibit inherently multiscale representations [47], [48]. For bitemporal hierarchical features produced at each stage, a margin maximization module clarifies the gap between changed and unchanged semantics, and realizes stage-level fusion at the corresponding scale. In the following, an SA module highlights positional change responses in the fused bitemporal features. To fuse multiscale features between different stages, we resort to PixelShuffle [49] to realize upsampling and simple channelwise concatenation to stack cross-stage features. Recall that the last two feature maps have the same spatial size. We just use 1×1 convolution to reduce the number of channels. The downstream PixelShuffle operations rearrange elements in feature maps [49], performing upsampling and reducing the number of channels in the meanwhile. After multiscale feature fusion, the final refined features f_{Refined} cover multiscale representations and have strong separability for bitemporal change detection.

3) Classifier for Decision-Making: We cast the final decision-making as a two-class (i.e., "changed" versus "unchanged") classification. We design an extremely simple classifier to carry out this task (see Fig. 3 (right) for an illustration). Two PixelShuffle operations expand the spatial size of the refined features to that of labels, i.e., $H \times W$. We place a convolution layer in between the two PixelShuffle operations. Our primary aim is to adjust the number of channels through a parameterized layer. The classifier transforms the input refined features into the nonchange map and the change map of interest, completing the pipeline of change detection.

E. Loss Function

The data imbalance between changed pixels and unchanged pixels tends to cause classification bias (normally toward "unchanged") [50]. To alleviate the problem, we select a combination of cross-entropy loss and dice loss [51] as the objective to minimize.

Let $y \in \mathcal{Y}$ be the label for a pair of instances, and let $y_{(i,j)}$ index the coordinates of elements in the label y. Each element indicates whether a change happened in its corresponding imaging field or not. Denote that $y_{(i,j)} = 1$ if there is a change and $y_{(i,j)} = 0$ otherwise. The cross-entropy loss $\mathcal{L}_{\text{ce}}$ and the dice loss $\mathcal{L}_{\text{dice}}$ take the forms as follows:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left[y_{(i,j)} \log \hat{y}_{(i,j)} + \left(1 - y_{(i,j)}\right) \log\left(1 - \hat{y}_{(i,j)}\right) \right] \quad (4)$$

where $\hat{y}$ denotes the prediction for a pair of instances

$$\mathcal{L}_{\text{dice}} = \frac{1}{H \times W} \left[1 - \frac{\left(\sum_{i=1}^{H} \sum_{j=1}^{W} \hat{y}_{(i,j)} \times y_{(i,j)} \right) \times 2}{\left(\sum_{i=1}^{H} \sum_{j=1}^{W} \hat{y}_{(i,j)} + y_{(i,j)} \right) + \epsilon} \right] \quad (5)$$

where ϵ is a small constant (1e-7 as default) avoiding division by zero.

Our full loss function $\mathcal{L}_{\text{full}}$ is

$$\mathcal{L}_{\text{full}} = \mathcal{L}_{\text{ce}} + \lambda \mathcal{L}_{\text{dice}} \quad (6)$$

where λ controls the relative importance of the two losses and is set to 1 in this article.

IV. Experiments

A. Datasets

We conduct rigorous comparisons on three public change detection datasets composed of VHR images: the LEVIR-CD dataset [27], the SYSU-CD dataset [26], and the Season-Varying change detection dataset [52]. We run extensive ablation studies on the LEVIR-CD dataset. The LEVIR-CD dataset is semantic-specific and applies to detecting changes with regard to buildings. It contains 637 pairs of labeled bitemporal image patches collected from Google Earth. The original size of each instance/label is 1024 × 1024. To accord with the commonly used input size of state-of-the-art change detection methods, we crop each 1024 × 1024 instance/label to 16 256 × 256 ones. Consequently, there are 7120/1024/2048 pairs of instances and corresponding labels for training/validation/testing, respectively. The SYSU-CD dataset is a challenging dataset for general change detection. It contains 20 000 pairs of labeled bitemporal image patches with a size of 256 × 256. There are 12 000/4000/4000 pairs of instances and corresponding labels for training/validation/testing, respectively. The Season-Varying dataset also applies to general change detection. It contains 15 998 pairs of bitemporal instances that demonstrate distinct appearances caused by seasonal factors and associated labels. The size of each instance/label is 256 × 256. There are 10 000/2998/3000 pairs of instances and corresponding labels for training/validation/testing, respectively.

TABLE II

Comparison With State-of-the-Art Methods on the LEVIR-CD Dataset. The Method With Superscript ∗ Presents the Results Reported in the Original Article

Method/Metric(%)	precision	recall	OA	IoU	F1
FC-Siam-conc [53]	90.76	58.95	97.60	55.61	71.47
FC-Siam-diff [53]	91.97	57.84	97.60	55.06	71.02
IFN [22]	86.95	75.24	98.16	67.61	80.67
SNUNet [31]	91.80	88.53	99.01	82.04	90.14
DSAMNet [26]	80.61	88.98	98.35	73.29	84.59
BIT* [27]	89.24	**89.37**	98.92	80.68	89.31
CLNet [21]	90.07	85.70	98.79	78.30	87.83
ISNet (ours)	**92.46**	88.27	**99.04**	**82.35**	**90.32**

TABLE III

Comparison With State-of-the-Art Methods on the SYSU-CD Dataset. The Method With Superscript ∗ Presents the Results Reported in the Original Article

Method/Metric(%)	precision	recall	OA	IoU	F1
FC-Siam-conc [53]	81.33	66.40	88.48	57.62	73.11
FC-Siam-diff [53]	**90.18**	48.28	86.56	45.87	62.89
IFN [22]	82.39	73.57	**90.06**	63.57	77.73
SNUNet [31]	78.41	73.38	88.96	61.05	75.81
DSAMNet* [26]	74.81	**81.86**	-	64.18	78.18
BIT [27]	79.18	77.01	89.80	64.04	78.08
CLNet [21]	79.62	74.97	89.57	62.90	77.22
ISNet (ours)	80.27	76.41	90.01	**64.44**	**78.29**

B. Implementation Details

We implement our proposed change detection method using Python in conjunction with the PyTorch library. We initialize backbone networks with weights pretrained on ImageNet and initialize the remaining layers using normal initialization. We use ResNet-18 as backbone networks unless otherwise specified. The initial learning rate is set to 0.0001. For the LEVIR-CD dataset, the default batch size is 16. The number of routine epochs is 200 and that of decay epochs is 100. For the SYSU-CD dataset, the default batch size is 8. The number of routine epochs is 50 and that of decay epochs

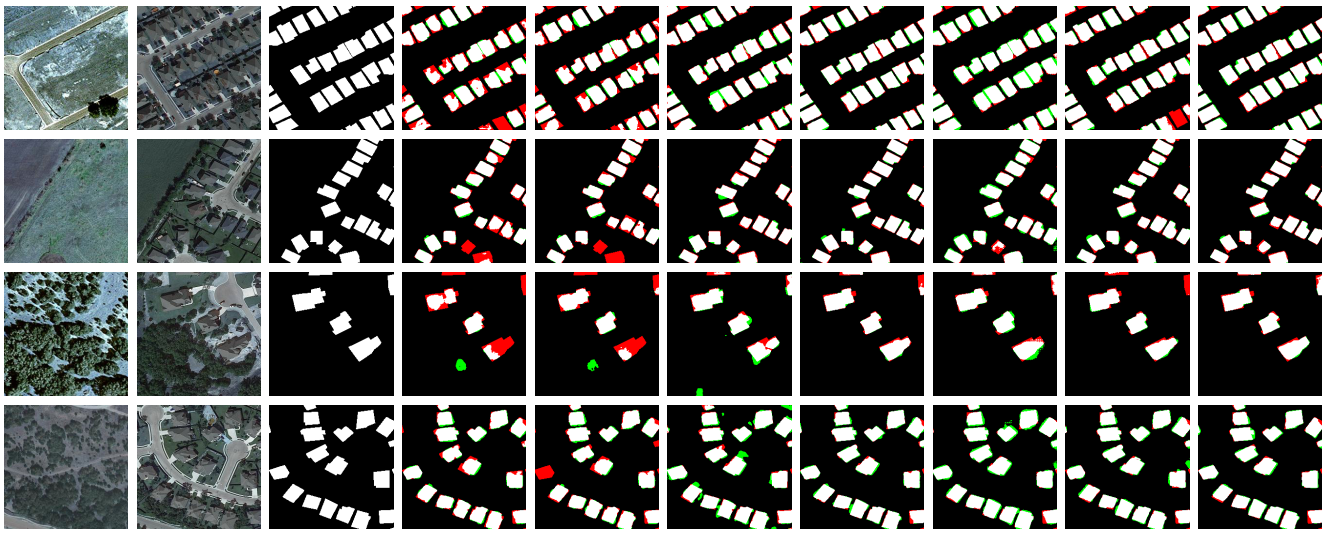

Fig. 7. Qualitative comparisons on the LEVIR-CD dataset. From (Left) to (Right): T1 instance, T2 instance, label, predictions of FC-Siam-conc [53], FC-Siam-diff [53], IFN [22], SNUNet [31], DSAMNet [26], CLNet [21], and our ISNet. Colors: white for TP (i.e., "changed"), black for TN (i.e., "unchanged"), red for FP, and green for FN.

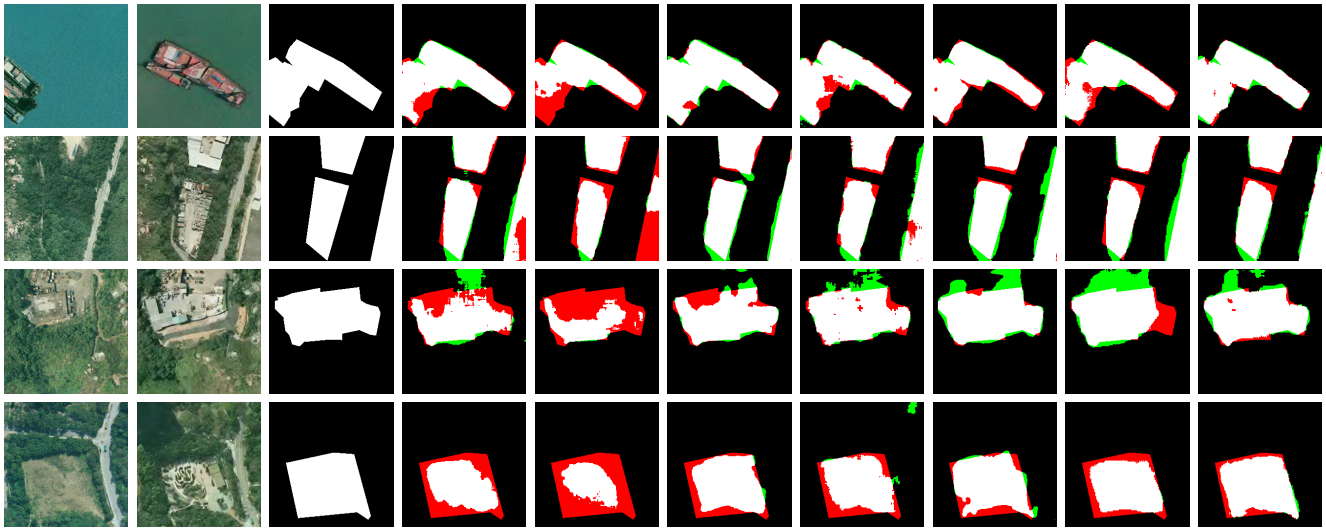

Fig. 8. Qualitative comparisons on the SYSU-CD dataset. From (Left) to (Right): T1 instance, T2 instance, label, predictions of FC-Siam-conc [53], FC-Siam-diff [53], IFN [22], SNUNet [31], BIT [27], CLNet [21], and our ISNet. Colors: white for TP (i.e., "changed"), black for TN (i.e., "unchanged"), red for FP, and green for FN.

is 50. For the Season-Varying change detection dataset, the default batch size is 8. The number of routine epochs is 50 and that of decay epochs is 150. We use random horizontal flipping, random vertical flipping, random cropping, and random blurring to realize data augmentation for the training data. We use the Adam optimizer and set $\beta_1 = 0.5$ and $\beta_2 = 0.999$. All the experiments are conducted on a TITAN X GPU. Our source code and trained models are available at https://github.com/xingronaldo/ISNet.

C. Evaluation Metrics

We adopt five generic metrics for quantitative evaluation: precision, recall, overall accuracy (OA), Intersection-of-Union (IoU), and F1 score. We report results with respect to "changed" in the two-class classification. These metrics can be derived from the following combinational calculations with the numbers of true positive (TP), false positive (FP), false

TABLE IV

COMPARISON WITH STATE-OF-THE-ART METHODS ON THE SEASON-VARYING DATASET. THE METHOD WITH SUPERSCRIPT $*$ PRESENTS THE RESULTS REPORTED IN THE ORIGINAL ARTICLE

Method/Metric(%)	precision	recall	OA	IoU	F1
FC-Siam-conc [53]	77.31	49.43	92.32	43.17	60.31
FC-Siam-diff [53]	80.80	51.16	92.80	45.62	62.65
IFN [22]	**97.15**	91.70	98.70	89.29	94.34
SNUNet [31]	94.82	92.45	98.51	88.00	93.62
DSAMNet* [26]	94.54	92.77	-	88.13	93.69
BIT [27]	95.31	87.31	98.00	83.71	91.13
CLNet [21]	93.30	89.80	98.03	84.36	91.52
ISNet (ours)	95.18	**94.43**	**98.78**	**90.12**	**94.80**

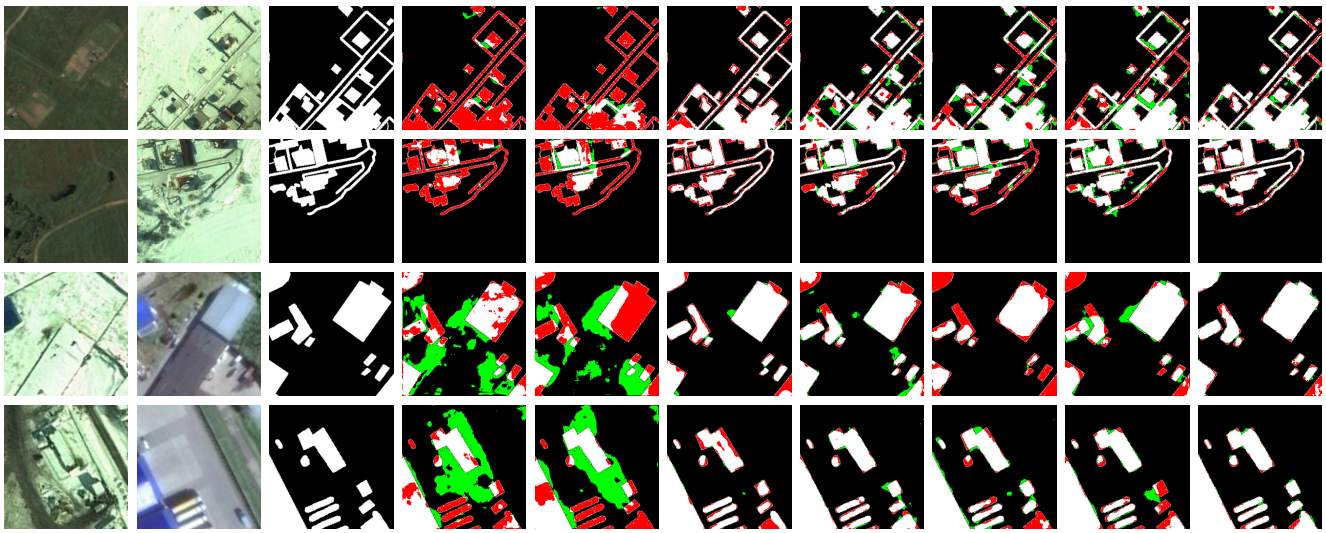

Fig. 9. Qualitative comparisons on the Season-Varying dataset. From (Left) to (Right): T1 instance, T2 instance, label, predictions of FC-Siam-conc [53], FC-Siam-diff [53], IFN [22], SNUNet [31], BIT [27], CLNet [21], and our ISNet. Colors: white for TP (i.e., "changed"), black for TN (i.e., "unchanged"), red for FP, and green for FN.

TABLE V
PERFORMANCE WITH LIGHTWEIGHT BACKBONES ON THE LEVIR-CD, SYSU-CD, AND SEASON-VARYING DATASETS

Dataset	LEVIR-CD					SYSU-CD					Season-Varying				
Model_*N*/Metric(%)	precision	recall	OA	IoU	F1	precision	recall	OA	IoU	F1	precision	recall	OA	IoU	F1
ISNet-lw_128	90.07	80.73	98.57	74.13	85.14	**79.87**	71.76	89.07	60.77	76.00	90.56	84.11	97.09	77.33	87.22
ISNet-lw_192	89.28	84.72	98.70	76.90	86.94	79.76	73.23	**89.30**	**61.75**	**76.35**	91.95	84.14	97.26	78.37	87.87
ISNet-lw_256	89.41	**86.01**	98.77	78.06	87.68	79.24	73.52	89.21	61.65	76.28	92.14	86.56	97.54	80.61	89.26
ISNet-lw_384	**90.24**	85.40	**98.78**	**78.18**	**87.75**	78.42	**74.31**	89.12	61.59	76.31	**93.44**	**88.13**	**97.87**	**82.99**	**90.70**

negative (FN), and true negative (TN)

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{7}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{8}$$

$$\text{OA} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}} \tag{9}$$

$$\text{IoU} = \frac{\text{TP}}{\text{TP}+\text{FP}+\text{FN}} \tag{10}$$

$$\text{F1} = \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}. \tag{11}$$

D. Comparison With State-of-the-Art Methods

We select seven state-of-the-art methods for comparison: FC-Siam-conc [53], FC-Siam-diff [53], IFN [22], SNUNet [31], DSAMNet [26], BIT [27], and CLNet [21].

Tables II–IV give quantitative comparisons on the LEVIR-CD, SYSU-CD, and Season-Varying test sets, respectively. It can be found that our ISNet outperforms other methods markedly on LEVIR-CD and Season-Varying, in terms of OA, IoU, and F1 score. Our ISNet also achieves the best performance overall on SYSU-CD. Our ISNet shows robustness to different datasets. SNUNet is a relatively strong competitor on LEVIR-CD and Season-Varying. However, its performance on SYSU-CD seems not competitive. While IFN achieves the second high performance on Season-Varying, it loses nearly 10% to ISNet on LEVIR-CD.

Figs. 7–9 visualize qualitative comparisons on the LEVIR-CD, SYSU-CD, and Season-Varying test sets, respectively. Four pairs of test instances are selected to exemplify each dataset. It is clear that most false alarms of high-performance methods distribute around change boundaries. Our ISNet delivers noticeably fewer false predictions in these scenes compared with other methods. In particular, our ISNet produces a small number of FNs, thanks to the use of the margin maximization modules for removing ambiguity around change boundaries.

E. Performance With Lightweight Backbones

We proceed with the evaluation of our proposed lightweight models, i.e., ISNet-lw. Table V reports the overall performance of ISNet-lw with different basic numbers of filters N, on the LEVIR-CD, SYSU-CD, and Season-Varying test sets. The results are acceptable by and large, even for the Season-Varying dataset with seasonal interference.

Furthermore, we intuitively compare the number of parameters in the main constituent parts and the F1 scores on three datasets between ISNet-lw and ISNet in Fig. 10. It reveals that ISNet exceeds ISNet-lw by a large margin in terms of parameters for both backbone networks and margin maximization (MM) modules but does not beat ISNet-lw too much

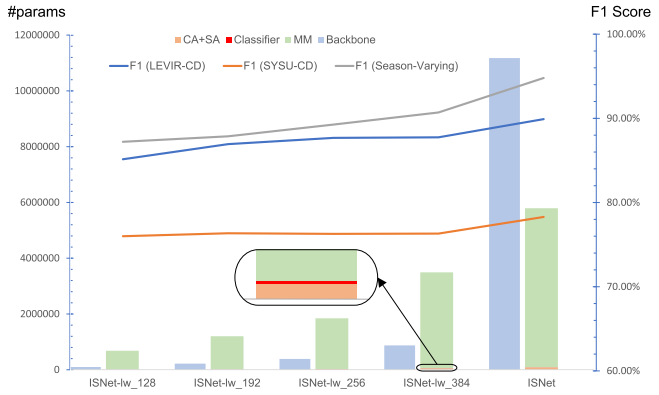

Fig. 10. Comparisons of the number of parameters in the main constituent parts (column, Left) and the F1 scores on the LEVIR-CD, SYSU-CD, and Season-Varying datasets (line, Right) between ISNet-lw and ISNet.

in terms of performance. In the meanwhile, the parameters for CA modules plus SA modules and classifier are nearly negligible. We find that the margin maximization modules dominate the volume of parameters in ISNet-lw, and the backbone network introduces much more parameters in ISNet. The performance gains between ISNet and ISNet-lw highlight the importance of utilizing advanced deep learning products to perform effective feature extraction for remote sensing image change detection. In light of the parameter numbers of different parts, the acceptable results of ISNet-lw also suggest that feature refinement carries a big weight in improving change detection results.

F. Ablation Studies

In this part, we run extensive ablation studies for our ISNet on the LEVIR-CD dataset. We slightly change the default settings to deal with possibly increased computational burdens induced by substituting some constituent parts. In all ablation studies, the batch size is decreased to 8. The routine epochs and the decay epochs are adjusted to 50 and 150, respectively.

1) Effect of Margin Maximization and Targeted Arrangement of Attention Mechanisms: Table VI evaluates the effectiveness of our proposed two strategies, i.e., margin maximization (MM) and targeted arrangement of attention mechanisms (CA and SA). Specifically, removing MM incurs the most significant drop in performance, removing SA the second, and removing CA the last. We conclude that the use of MM is both theoretically and empirically sound. Though MM induces most parameters, we argue that the volume of parameters cannot explain the fact. The parameter ratio of MM/SA/CA is 79163:1:1209 indeed. Surprisingly, the performance after removing MM merely is even worse than that after removing all three types of modules. We speculate that this reflects the synergistic effect of the two strategies. Specifically, CA, MM, and SA can be regarded as acting in a sequential manner in each stage. Accurate semantic emphasis benefits faithful margin maximization and further makes positional emphasis sensible.

2) Effect of PixelShuffle for Upsampling: Table VII compares PixelShuffle with bicubic interpolation, bilinear interpolation, and deconvolution for upsampling feature maps.

TABLE VI

EFFECT OF MARGIN MAXIMIZATION AND TARGETED ARRANGEMENT OF ATTENTION MECHANISMS

MM	CA	SA	precision	recall	OA	IoU	F1
✓	✓	✓	**91.74**	88.26	**99.00**	**81.67**	**89.97**
×	✓	✓	87.92	**88.87**	98.81	79.20	88.39
✓	×	✓	91.00	88.31	98.96	81.22	89.64
✓	✓	×	89.93	87.82	98.88	79.96	88.86
×	×	×	90.19	87.13	98.86	79.59	88.63

TABLE VII

EFFECT OF PIXELSHUFFLE FOR UPSAMPLING

Method	Δparams	precision	recall	OA	IoU	F1
PixelShuffle	-	**91.74**	88.26	**99.00**	**81.67**	**89.97**
bicubic	+0.34M	91.01	88.06	98.95	81.01	89.51
bilinear	+0.34M	90.88	88.33	98.95	81.13	89.58
deconvolution	+5.51M	90.70	**88.49**	98.95	81.13	89.58

TABLE VIII

INFLUENCE OF DIFFERENT RESNET SERIES BACKBONES

Backbone	Δparams	precision	recall	OA	IoU	F1
ResNet-18	-	91.74	88.26	99.00	81.67	89.97
ResNet-34	+10.11M	91.19	87.87	98.95	80.99	89.50
ResNet-50	+86.24M	90.92	89.80	99.02	82.41	90.36
ResNeXt-50_32x4d [34]	+85.71M	**91.38**	**90.07**	**99.06**	**83.01**	**90.72**
Wide_ResNet-50_2 [35]	+129.56M	91.06	90.06	99.04	82.75	90.56

We replace all PixelShuffle operations using the above upsampling methods in the processes of feature refinement and decision-making. To ensure that other layers and operations remain unchanged, we use additional 1×1 convolutional layers after interpolation operations to adjust channels in particular. With the least parameters, ISNet with PixelShuffle scores the best in performance.

3) Influence of Different ResNet Series Backbones: Table VIII lists the influence of different ResNet series backbones. We implement ResNet-34, ResNet-50, ResNeXt-50_32 $\times$ 4d [34], and Wide_ResNet-50_2 [35] for substituting the default ResNet-18. Residual blocks in these backbones are either deepened or widened compared with those in ResNet-18, and considerable computational burdens are introduced consequently. Specifically, these upscaled backbones except for ResNet-34 bring over 85 million increases in terms of parameters. Their enhanced learning capacity results in appreciable performance improvements. With relatively few parameter increases, ResNet-34 does not deliver performance gains over ResNet-18. This implies that ResNet-18 succeeds in serving as a decent backbone network. Besides, we speculate that a relatively small volume of parameter increases in feature extraction cannot determine the final results due to the intervention of feature refinement.

4) Influence of Different Channel and Spatial Attention Modules: Table IX evaluates the influence of different plug-and-play CA modules (CA [32], ECA [54], SE [55], and CAM [56]) and SA modules (SA [32], DA [57], SSE [58], and PAM [56]). The results demonstrate that these off-the-shelf

TABLE IX
INFLUENCE OF DIFFERENT CA AND SA MODULES

Channel-wise	Spatial-wise	precision	recall	OA	IoU	F1
CA [32]	SA [32]	91.74	88.26	99.00	81.67	89.97
CA [32]	DA [57]	**93.00**	87.25	**99.02**	**81.87**	**90.03**
ECA [54]	SA [32]	91.00	88.58	98.98	81.52	89.82
SE [55]	SA [32]	91.15	88.50	98.98	81.49	89.80
SE [55], [58]	SSE [58]	90.76	**88.94**	98.98	81.56	89.94
CAM [56]	PAM [56]	90.19	87.23	98.87	79.67	88.68

TABLE X
INFLUENCE OF DIFFERENT BATCH SIZES

Batch Size	precision	recall	OA	IoU	F1
8	91.74	88.26	**99.00**	**81.67**	**89.97**
16	**91.97**	87.95	**99.00**	**81.67**	89.91
24	89.03	89.39	98.89	80.52	89.21
32	88.00	**90.16**	98.87	80.29	89.07
40	87.09	89.33	98.77	78.88	88.19

attentions work comparably under our targeted arrangement. The use of {CAM, PAM} turns out the only "failed" case.

5) Influence of Different Batch Size: Table X reports the influence of different batch size {8, 16, 24, 32, 40}. When other settings are kept in default, batch size 8 and batch size 16 bring about matchable top performance. Interestingly, further increases in batch size lead to declines in performance instead. This reveals that increasing batch size merely to train ISNet does not necessarily result in performance improvements.

V. CONCLUSION

This article investigates the strategical and architectural designs for remote sensing image change detection. We study against the problem of blurry boundaries between different semantics among hierarchical features. We propose deep learning networks that deliver improved separability (ISNet) equipped with a combination of two strategies (i.e., margin maximization and targeted arrangement of attention mechanisms) and elaborate architectures. Our ISNet achieves state-of-the-art performance on three public datasets. Besides, we show that an effective refinement of bitemporal hierarchical features matters for accurate change detection.

REFERENCES

[1] 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.

[2] S. Liu, D. Marinelli, L. Bruzzone, and F. Bovolo, "A review of change detection in multitemporal hyperspectral images: Current techniques, applications, and challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 140–158, Jun. 2019.

[3] D. Wen *et al.*, "Change detection from very-high-spatial-resolution optical remote sensing images: Methods, applications, and future directions," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 4, pp. 68–101, Dec. 2021.

[4] Z. Lv, T. Liu, J. A. Benediktsson, and N. Falco, "Land cover change detection techniques: Very-high-resolution optical images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 44–63, Mar. 2022.

[5] 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.

[6] J. Xu, C. Luo, X. Chen, S. Wei, and Y. Luo, "Remote sensing change detection based on multidirectional adaptive feature fusion and perceptual similarity," *Remote Sens.*, vol. 13, no. 15, p. 3053, Aug. 2021.

[7] Z. Wu, W. Zhu, J. Chanussot, Y. Xu, and S. Osher, "Hyperspectral anomaly detection via global and local joint modeling of background," *IEEE Trans. Signal Process.*, vol. 67, no. 14, pp. 3858–3869, Jul. 2019.

[8] R. Touati, M. Mignotte, and M. Dahmane, "Multimodal change detection in remote sensing images using an unsupervised pixel pairwise-based Markov random field model," *IEEE Trans. Image Process.*, vol. 29, pp. 757–767, 2020.

[9] F. Bovolo and L. Bruzzone, "The time variable in data fusion: A change detection perspective," *IEEE Geosci. Remote Sens. Mag.*, vol. 3, no. 3, pp. 8–26, Sep. 2015.

[10] X. Peng, R. Zhong, Z. Li, and Q. Li, "Optical remote sensing image change detection based on attention mechanism and image difference," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7296–7307, Sep. 2021.

[11] T. Liu, L. Yang, and D. Lunga, "Change detection using deep learning approach with object-based image analysis," *Remote Sens. Environ.*, vol. 256, Apr. 2021, Art. no. 112308.

[12] Z. Wu *et al.*, "Scheduling-guided automatic processing of massive hyperspectral image classification on cloud computing architectures," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3588–3601, Jul. 2020.

[13] Z. Wu, J. Sun, Y. Zhang, Z. Wei, and J. Chanussot, "Recent developments in parallel and distributed computing for remotely sensed big data processing," *Proc. IEEE*, vol. 109, no. 8, pp. 1282–1305, Aug. 2021.

[14] A. Karpatne, Z. Jiang, R. R. Vatsavai, S. Shekhar, and V. Kumar, "Monitoring land-cover changes: A machine-learning perspective," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 8–21, Jun. 2016.

[15] H. Zhang, M. Lin, G. Yang, and L. Zhang, "ESCNet: An end-to-end superpixel-enhanced change detection network for very-high-resolution remote sensing images," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 5, 2021, doi: 10.1109/TNNLS.2021.3089332.

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

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351. Munich, Germany: Springer, 2015, pp. 234–241. [Online]. Available: https://citations.springernature.com/item?doi=10.1007/978-3-319-24574-4_28

[18] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[19] L. Zhang, Q. Zhang, B. Du, X. Huang, Y. Y. Tang, and D. Tao, "Simultaneous spectral-spatial feature selection and extraction for hyperspectral images," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 16–28, Jan. 2018.

[20] X. Hou, Y. Bai, Y. Li, C. Shang, and Q. Shen, "High-resolution triplet network with dynamic multiscale feature for change detection on satellite images," *ISPRS J. Photogramm. Remote Sens.*, vol. 177, pp. 103–115, Jul. 2021.

[21] Z. Zheng, Y. Wan, Y. Zhang, S. Xiang, D. Peng, and B. Zhang, "CLNet: Cross-layer convolutional neural network for change detection in optical remote sensing imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 175, pp. 247–267, May 2021.

[22] 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.

[23] T. Lei *et al.*, "Difference enhancement and spatial–spectral nonlocal network for change detection in VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 4507013.

[24] N. Wang, S. Ma, J. Li, Y. Zhang, and L. Zhang, "Multistage attention network for image inpainting," *Pattern Recognit.*, vol. 106, Oct. 2020, Art. no. 107448.

[25] M. Liu, Q. Shi, A. Marinoni, D. He, X. Liu, and L. Zhang, "Super-resolution-based change detection network with stacked attention module for images with different resolutions," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2022.

[26] Q. Shi, M. Liu, S. Li, X. Liu, F. Wang, and L. Zhang, "A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.

[27] H. Chen, Z. Qi, and Z. Shi, "Remote sensing image change detection with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[28] J. Huang, Q. Shen, M. Wang, and M. Yang, "Multiple attention Siamese network for high-resolution image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.
[29] L. Zhang, X. Hu, M. Zhang, Z. Shu, and H. Zhou, "Object-level change detection with a dual correlation attention-guided detector," *ISPRS J. Photogramm. Remote Sens.*, vol. 177, pp. 147–160, Jul. 2021.
[30] D. Wang, X. Chen, M. Jiang, S. Du, B. Xu, and J. Wang, "ADS-Net: An attention-based deeply supervised network for remote sensing image change detection," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 101, Sep. 2021, Art. no. 102348.
[31] S. Fang, K. Li, J. Shao, and Z. Li, "SNUNet-CD: A densely connected Siamese network for change detection of VHR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
[32] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.
[33] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy
[34] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.
[35] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, p. 87.
[36] M.-H. Guo *et al.*, "Attention mechanisms in computer vision: A survey," 2021, *arXiv:2111.07624*.
[37] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021.
[38] X. Feng, J. Han, X. Yao, and G. Cheng, "TCANet: Triple context-aware network for weakly supervised object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6946–6955, Aug. 2021.
[39] B. Graham *et al.*, "LeViT: A vision transformer in ConvNet's clothing for faster inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 12259–12269.
[40] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.
[41] S. Huang, Z. Lu, R. Cheng, and C. He, "FaPN: Feature-aligned pyramid network for dense image prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 864–873.
[42] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1715–1723.
[43] Z. Xu, K. Wu, L. Huang, Q. Wang, and P. Ren, "Cloudy image arithmetic: A cloudy scene synthesis paradigm with an application to deep learning based thin cloud removal," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5612616.
[44] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.
[45] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3141–3149.
[46] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, Dec. 2015.
[47] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
[48] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13034–13043.
[49] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.
[50] Z. Wang, C. Peng, Y. Zhang, N. Wang, and L. Luo, "Fully convolutional Siamese networks based change detection for optical aerial images with focal contrastive loss," *Neurocomputing*, vol. 457, pp. 155–167, Oct. 2021.
[51] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. Fourth Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.
[52] M. A. Lebedev, Y. V. Vizilter, O. V. Vygolov, V. A. 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.
[53] R. Caye Daudt, B. Le Saux, and A. Boulch, "Fully convolutional Siamese networks for change detection," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 4063–4067.
[54] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.
[55] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[56] T. Chen *et al.*, "ABD-Net: Attentive but diverse person re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 864–873.
[57] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "A^2-Nets: Double attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 352–361.
[58] A. G. Roy, N. Navab, and C. Wachinger, "Recalibrating fully convolutional networks with spatial and channel 'squeeze and excitation' blocks," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 540–549, Feb. 2019.

Gong Cheng (Member, IEEE) received the B.S. degree from Xidian University, Xi'an, China, in 2007, and the M.S. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, in 2010 and 2013, respectively.

He is currently a Professor with Northwestern Polytechnical University. His main research interests are computer vision, pattern recognition, and remote sensing image understanding.

Dr. Cheng is also an Associate Editor of *IEEE Geoscience and Remote Sensing Magazine* and a Guest Editor of IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING.

Guangxing Wang received the B.S. degree in electronic information science and technology from the Shandong University of Science and Technology, Qingdao, China, in 2018, and the M.S. degree in information and communication engineering from the China University of Petroleum (East China), Qingdao, in 2021. He is currently pursuing the Ph.D. degree in control science and engineering with Northwestern Polytechnical University, Xi'an, China.

His research interests include deep learning and learning with applications in remote sensing.

Junwei Han (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in pattern recognition and intelligent systems from Northwestern Polytechnical University, Xi'an, China, in 1999, 2001, and 2003, respectively.

He was a Research Fellow with Nanyang Technological University, Singapore, The Chinese University of Hong Kong, Hong Kong, Dublin City University, Dublin, Ireland, and the University of Dundee, Dundee, U.K., from 2003 to 2010. He is currently a Professor with Northwestern Polytechnical University. His research interests include computer vision and brain imaging analysis.