Changer: Feature Interaction Is What You Need for Change Detection

Sheng Fang, Kaiyu Li, Graduate Student Member, IEEE, and Zhe Li

Abstract—Change detection is an important tool for long-term Earth observation missions. It takes bi-temporal images as input and predicts “where” the change has occurred. Different from other dense prediction tasks, a meaningful consideration for change detection is the interaction between bi-temporal features. With this motivation, in this article we propose a novel general change detection architecture, MetaChanger, which includes a series of alternative interaction layers in the feature extractor. To verify the effectiveness of MetaChanger, we propose two derived models, ChangerAD and ChangerEx with simple interaction strategies: aggregation-distribution (AD) and feature “exchange.” AD is abstracted from some complex interaction methods, and feature “exchange” is a completely parameter and computation-free operation by exchanging bi-temporal features. In addition, for better alignment of bi-temporal features, we propose a flow-based dual-alignment fusion (FDAF) module which allows interactive alignment and feature fusion. Crucially, we observe Changer series models achieve competitive performance on different scale change detection datasets. Further, our proposed ChangerAD and ChangerEx could serve as a starting baseline for future MetaChanger design. Code and weights are made available at https://github.com/likyoo/open-cd.

Index Terms—Change detection, deep neural network, feature interaction, high-resolution remote sensing (RS) image.

I. INTRODUCTION

C

HANGE detection is one of the most widely used fundamental technologies in Earth observation. Compared to remote sensing (RS) image segmentation, change detection has two advantages in application.

1) In long-term Earth observation, rather than predicting all pixels of the whole image, we often only need to focus on the land-cover category in the changed area.

2) In some semiautomated applications, change detection is more tolerant of some misdetection.

Specifically, change detection is a pixel-to-pixel task, which takes bi-temporal images as input and predicts “where” the change has occurred. Driven by large amounts of RS data [1], [2], [3], change detection models based on ConvNet or Vision Transformer achieve very competitive performance in many complex scenarios. Recently, most deep learning (DL)-based change detection methods have been designed in close relation to segmentation models, and focus on some common problems, for instance, the misdetection caused by edges, small targets, and various scales.

The goal of image segmentation is to determine a network to fit the target \( Y \) as much as possible, which can be described by minimizing the empirical loss as: \( \min L(F_\theta(X_0), Y) \). Different from this, change detection takes two inputs, i.e., the bi-temporal images \( X_0 \) and \( X_1 \), and it can be described as \( \min L(F_\theta(X_0, X_1), Y) \), where \( F_\theta \) denotes neural networks.

Therefore, a worthwhile consideration is whether the correlation between \( X_0 \) and \( X_1 \) should be explored, and if so, how to implement it. For the first issue, there are many studies illustrating the important effects of interaction between homo/hetero-geneous features [4], [5]. And specifically for bi-temporal images, there are style differences between different temporal image domains, which are due to climate change, preprocessing corrections, etc. The domain differences affect the target of interest (e.g., building) and the background in different degrees, which, together with the use of the Siamese network, makes the understanding of the “change of interest” ambiguous for the model.

For the second issue, in this article, firstly, we propose a general change detection architecture, MetaChanger, which aims to emphasize the effect of feature interactions during feature extraction in change detection, as shown in Fig. 1. Specifically, there are a series of alternative interaction layers and fusion layers designed in MetaChanger.

Then, to verify MetaChanger, we try two simple interaction strategies: aggregation-distribution (AD) and feature “exchange.” Specifically, the AD interaction is abstracted from some co-/cross-attention mechanisms, coming from tasks related to multimodality, tracking, etc. And the “exchange”
interaction is a completely parameter and computation-free operation, which is achieved by exchanging bi-temporal feature maps in the spatial or channel dimension, with the exchanged features being mixed as they pass through subsequent convolution or token mixer. Astonishingly, these derived models, termed ChangerAD and ChangerEx, achieve very competitive performance, and even consistently outperforms other well-tuned change detection models. Moreover, we make further exploration of ChangerEx in view of its excellent performance. The results in multiple datasets show that MetaChanger, even with naive interaction layers, can still deliver promising performance.

In addition to the interaction during feature extraction, we propose the flow-based dual-alignment fusion (FDAF) module for the interactive fusion of dual-branch features to overcome the problem of side-looking and misalignment in multitemporal RS images. All the interaction and fusion components abstracted in MetaChanger are not limited to these specific types. We hope our findings inspire more future research dedicated to improving MetaChanger.

Our main contributions can be summarized as follows.
1) We propose MetaChanger as a general change detection framework that focuses on a series of alternative interaction layers and can serve as a strong baseline.
2) We propose two simple interaction strategies, AD and feature “exchange,” and extensive experiments demonstrate that they can greatly improve MetaChanger’s performance, especially feature “exchange,” even when embedded in complex networks or applied to challenging datasets.
3) We propose an interactive fusion module, called FDAF, which alleviates misalignment problems in bi-temporal images in a flow-based manner.

II. RELATED WORK

A. Binary Change Detection

We roughly divide DL-based change detection methods into two types, according to how the change map is acquired: metric-based and classification-based. In general, the metric-based method transforms the two input images into a feature space whose feature representation becomes more consistent [6], [7]. In this feature space, the ultimate change map is obtained by the distance metric [8]. To achieve this goal, metric learning-related losses, such as contrast loss [9] are leveraged to pull unchanged pairs together and push changed pairs apart [10], [11].

The classification-based method, however, takes change detection as a dense classification task directly [12], [13], [14], [15], [16]. The bi-temporal features are fused at a certain stage and the ultimate change map is generated by a classifier at the head of the network. Usually, a simple cross-entropy (CE) loss is sufficient to optimize the model stably. Daoudt et al. [17] propose three typical change detection models: FC-EF, FC-Siam-Conc, and FC-Siam-Diff. Among them, FC-EF uses the early fusion strategy and the latter two use the medium fusion strategy with different fusion policies. In addition to CNN models, some transformer-based models achieve competitive performance in change detection. BiT [18] builds an efficient change detection model with very limited parameters by mixing CNN and transformer. ChangeFormer [19] is a pure transformer model, which is a siamese variant of SegFormer [20], and is fine-tuned in network depth.

Different from the above methods, our proposed MetaChanger, shown in Fig. 1, abstracts and simplifies the processes for change detection and especially focuses on the feature interaction between bi-temporal images.

B. Feature Fusion

Feature fusion is a fundamental process in many DL tasks, such as in classification [21], [22], segmentation [23], [24], multimodal tasks [25], change detection [16], [26], [27], etc. They include the fusion of multiple levels, multiple scales, heterogeneous features, etc.

For the specific operation of feature fusion, some simple parameter-free operations such as concat, weighted sum or bilinear pooling [28] can build stable baseline performance. Additionally, several attention-based methods make feature fusion more flexible and learnable [29], [30], [31], [32], [33]. Moreover, alignment-based fusion methods focus on feature alignment, and they typically use flow field or deformable convolution [34], [35] to align features of different levels in the spatial dimension [36], [37], [38], [39], [40].

C. Feature Interaction

In some studies, feature interaction is included in feature fusion. However, here, we define feature interaction in change detection as the correlation or communication of homo/heterogeneous features during feature extraction before fusion. Co-attention mechanism [41] is frequently used in feature interaction, and similar formats have been applied with great success in many research fields like multimodal related tasks (e.g., visual question answering (VQA) [42], RGB-D segmentation [43]), registration [44], matching [45] and network architecture design [46]. This format of interaction aggregates several features and then distributes them, respectively, as attention maps. Hence, we abstract this format as AD interaction.

In addition, there are some other effective methods for interaction. For example, [47] uses channel-wise cross-attention to learn mutual information from dual branches for object tracking. Wu et al. [44] generate the spatial affinity matrix between source and target point clouds. MixFormer [48] makes bi-directional interaction across self-attention [49] and DWConv [50], providing complementary cues in the channel and spatial dimensions. Different from these complex methods, our proposed feature “exchange” method is extremely simple and does not introduce any extra computation.

More relevant to the feature “exchange” is channel exchanging network (CEN) [4], which uses channel exchange for interaction, where channels corresponding to the smaller batch normalization (BN) scaling factors will be replaced by channels from other modalities. Yet, unlike CEN which is applied to multimodal problem, we focus on identifying positions where semantic differences exist for change detection. Hence,
strict semantic maintenance and semantic correspondence must be kept between both domains.

### III. Method

#### A. MetaChanger

MetaChanger is not a departure from classification-/metric-based models but is a purposeful framework for exploring interaction strategies. In MetaChanger, we use CE loss like the classification-based method to avoid the extra hyper-parameters in the metric-based method. In addition, to more directly demonstrate the effect of feature interactions, we use an entire Siamese encoder-decoder, like the metric-based models but is a purposeful framework for exploring interaction strategies. Let \( F_i,j \) denote the output of a hierarchy where \( i \) indexes the hierarchy along the encoder and \( j \) indexes the temporal dimension \( (i \in \{0, 1, 2, 3\} \) typically, and \( j \in \{0, 1\} \) for change detection). A hierarchy of MetaChanger consists of two main steps. First, multilevel features \( F_i,j \) go through a network stage with sharing weights, and features \( F'_{i+1,j} \) are generated. Then, \( F'_{i+1,j}, \forall j \) feed into the interaction layer to get the correlated \( F_{i+1,j} \). The hierarchy \( i \) can be formulated as

\[
F'_{i+1,j} = \text{Stage}_i(F_{i,j}) \quad \forall j
\]
\[
F_{i+1,j} = \text{InterAct}(F'_{i+0,j}, F'_{i+1,j}) \quad \forall j
\]  

(2)

where InterAct(\( \cdot \)) refers to the interaction layer. We denote \( F'_{i+1,j} \) as \( F'_{i+0,j} \) and \( F'_{i+1,j} \) for clearer description in (2). And to ensure MetaChanger is efficient in practice, we use a lightweight multilayer perceptron (MLP) decoder, like SegFormer [20]. Our decoder can be formulated as

\[
F_{i,j} = \text{Upsample}(\text{Linear}(C_i, C(J_i)), \forall i, j
\]
\[
\hat{F}_j = \text{Linear}(4C_i, C)\text{(Concat}(F_{i,j})) \quad \forall i
\]  

(3)

where Linear(\( C_{in}, C_{out}, \cdot \)) refers to a linear layer with \( C_{in} \) and \( C_{out} \) as input and output dimensions, respectively, and Upsample(\( \cdot \)) refers to upsampling features to \( 1/4 \)th.

Here we obtain the feature maps \( \hat{F}_0 \) and \( \hat{F}_1 \) from each bi-temporal image. And then \( \hat{F}_0 \) and \( \hat{F}_1 \) are aggregated and projected to the ultimate change map \( \hat{Y} \in \mathbb{R}^{2H \times W} \), which can be formulated as

\[
\hat{F} = \text{Fuse}(\hat{F}_0, \hat{F}_1) \\
\hat{Y} = \text{Project}(\hat{F})
\]  

(4)

where Fuse(\( \cdot \)) and Project(\( \cdot \)) indicate the abstraction of the fusion layer and projection layer, respectively. Specifically, in our implementation, the projection layer consists of two convolutional layers.

#### B. ChangerVanilla

For better comparisons, we first build a baseline model, ChangerVanilla. ChangerVanilla has no interaction layer and uses a simple concat operation as the fusion layer. Let \( x \) denote the feature map, we formulate the interaction and fusion layer of ChangerVanilla as

\[
\text{InterAct}_{\text{vanilla}}(x_i) = x_i
\]
\[
\text{Fuse}_{\text{vanilla}}(x_0, x_1) = \text{Concat}([x_0, x_1]).
\]  

(5)

Then, for the specific exploration of feature interaction layers, there are many complex strategies that can be adapted. Here, however, we want to demonstrate that using only simple modules, even parameter-free interaction operations, can effectively improve the performance of change detection models, which has rarely been discussed in previous related studies. Specifically, we throw in two embarrassingly simple interaction methods: AD and feature “exchange.”

#### C. ChangerAD

We abstract the AD style feature interaction from co-attention and some similar mechanisms [41], [42], [43], [46], as shown in Fig. 2(c). We refer to this variant model as ChangerAD. The basic idea of AD is to project the bi-temporal features into a feature space and get the global co-feature, then, use the distributed attention maps adaptively reweight each channel of the bi-temporal features.
Specifically, we first aggregate features from Siamese branches via an element-wise summation. Then we use the global average pooling to obtain more global information. Furthermore, we take a MLP to extract the cofeature, and employ sigmoid to obtain the ultimate two attention maps. The interaction layer of ChangerAD is formulated as

$$\text{InterAct}_{\text{AD}}(x_i) = x_i \cdot \sigma(\hat{x}_i)$$

where $x_0$ and $x_1$ refer to the bi-temporal features, $\sigma(\cdot)$ refers to the Sigmoid function, GAP refers to global average pooling, and the $\text{MLP}(C_i, 2C_i)(\text{GAP}(x_0 + x_1))$ refers to a two-layer MLP, with the first layer squeezing and the second layer expanding channel.

### D. ChangerEx

Feature “exchange” refers to partial exchange between bi-temporal features, during feature extraction. Hence, the natural question is why exchange and why it is feasible.

1) Why Exchange?: On the one hand, contextual information of bi-temporal features can be perceived by mutual learning through feature exchange and the subsequent mix layers (e.g., convolution or token mixer). On the other hand, through feature exchange and the subsequent layers, the distribution between the features of the two branches is more similar and automatic domain adaptation between the bi-temporal domains is achieved to some extent.

2) Why is the Exchange Feasible?: While some studies have emphasized temporal information, we do not believe that bi-temporal change detection is strongly constrained by “time.” The core of bi-temporal change learning is to train a change detector for images with the same spatial position but at different times. The temporal order is only to ensure that the appearance and disappearance of the target is interpretable, and in most definitions and applications in change detection, we are not concerned with whether it appears or disappears but simply changes. In brief, the semantic correspondence constraint for bi-temporal images in the definition of change detection makes feature exchange feasible.

Then, an important issue is where to exchange. Because of the logical operations involved in determining whether or not to exchange certain features, we need to be cautious to avoid nondifferentiable problem. For this, we considered two solutions. One is to convert the soft exchange mask to attention maps to guarantee the continuity of the gradient chain, and the other is to use an unlearnable (fixed) way to exchange, i.e., predefined a hard exchange mask. In practice, we find that simple learnable exchange does not perform better than the parameter-free unlearnable exchange, and therefore we tend to use the latter.

Let $x_0, x_1$ denote the bi-temporal features. InterActEx for $x_{0/1}$ can be formulated as

$$x_{0/1}(n,c,h,w) = \begin{cases} x_{0/1}(n,c,h,w), & M(n,c,h,w) = 0 \\ x_{1/0}(n,c,h,w), & M(n,c,h,w) = 1 \end{cases}$$

where $n$, $c$, and $hw$ index the batch, channel, and space, respectively. $M$ refers to the exchange mask consisting of 1 and 0, indicating exchange and nonexchange. Furthermore, we have tried feature exchange in two dimensions separately, the channel dimension and the spatial dimension, with the details described as follows.

3) Channel Exchange: Channel exchange refers to exchanging features in channel dimension, which is shown in Fig. 3(c). Specifically, for (7), the predefined mask remains spatially consistent. After channel exchange, gradients are detached from the exchanged channel and back-propagated through the other ones.

4) Spatial Exchange: Relatively, spatial exchange [shown in Fig. 3(b)] refers to exchanging features in the spatial dimension, and the predefined mask remains consistent on the channel. Considering that subsequent layers may be fully channel-wise MLP, a mix layer is optional.

The pseudo-code of channel exchange and spatial exchange are in Algorithms 1 and 2. Finally, we combine these two feature exchange and propose ChangerEx, as shown in Fig. 3(a). In addition, feature exchange involves issues like how many
Algorithm 1 Channel Exchange for ChangerEx, PyTorch-Like Code

```python
import torch
import torch.nn as nn

class ChannelExchange(nn.Module):
    def __init__(self, p=2):
        super().__init__()
        self.p = p  # 1/p of the features will be exchanged.

    def forward(self, x0, x1):  # x0, x1: the bi-temporal feature maps.
        N, C, H, W = x0.shape
        exchange_mask = torch.arange(C) % self.p == 0
        exchange_mask = exchange_mask.unsqueeze(0).expand((N, -1))
        out_x0, out_x1 = torch.zeros_like(x0), torch.zeros_like(x1)
        out_x0[~exchange_mask, ...] = x0[~exchange_mask, ...]
        out_x1[~exchange_mask, ...] = x1[~exchange_mask, ...]
        out_x0[exchange_mask, ...] = x1[exchange_mask, ...]
        out_x1[exchange_mask, ...] = x0[exchange_mask, ...]
        return out_x0, out_x1
```

features to exchange, at which stage to exchange, etc. which we will discuss later.

E. Flow-Based Dual-Alignment Fusion

Registration error is one of the most common challenges in change detection. Image registration is an essential part of change detection preprocessing. However, there are always more or less misalignment or side-looking problems, as shown in Fig. 4. Some work has tried implicit alignment, such as the use of global attention in [11]. In this article we use an explicit bi-directional alignment with optical flow, as shown in Fig. 2(e). The FDAF introduces a task prior for change detection, i.e., discovering differences between images. Specifically, the bi-temporal feature maps are resampled through a deformable field to obtain their respective corrected features. Then we take the distance to the other original feature map separately and feed it into the subsequent forward propagation.

In other words, the FDAF implements an explicit task transformation function where the object extraction task is converted into the change detection task. Mathematically, the features $x_0, x_1$ feed into $\text{Fuse}_{\text{align}}$ can be written as

$$x = \text{Concat}([x_0(p + \Delta p_0) - x_1, x_1(p + \Delta p_1) - x_0])$$  \hspace{1cm} (8)

where $p$ enumerates the locations in $x_0$ or $x_1$, and the offset field $\Delta p$ is obtained by the FlowNet (two-layer Conv)

$$\Delta p = \text{FlowNet}(\text{Concat}([x_0, x_1])).$$  \hspace{1cm} (9)

The bi-linear interpolation [54] is used to compute the exact value of the features. We refer to MetaChanger with FDAF as ChangerAlign.

IV. EXPERIMENTS

A. Dataset

S2Looking dataset [2] contains 5000 image pairs (1024 × 1024, 7:1:2 for train, eval, and test) and more than 65,920

---

Fig. 4. Some visualization comparisons among Changer models on the S2Looking. The rendered colors represent true positives (TP), false positives (FP), and false negatives (FN). Buildings with misregistration are framed out in the first row. (a) $t_0$ image. (b) $t_1$ image. (c) Ground truth. (d) ChangerVanilla. (e) ChangerAlign. (f) ChangerAD. (g) ChangerEx.
Algorithm 2 Spatial Exchange for ChangerEx, PyTorch-Like Code

```python
class SpatialExchange(nn.Module):
    def __init__(self, p=2):
        super().__init__()
        self.p = p  # 1/p of the features will be exchanged.

    def forward(self, x0, x1):  # x0, x1: the bi-temporal feature maps.
        N, C, H, W = x0.shape
        exchange_mask = torch.arange(W) % p == 0
        out_x0, out_x1 = torch.zeros_like(x0), torch.zeros_like(x1)
        out_x0[exchange_mask] = x0[exchange_mask]
        out_x1[exchange_mask] = x1[exchange_mask]
        out_x0[~exchange_mask] = x1[~exchange_mask]
        out_x1[~exchange_mask] = x0[~exchange_mask]
        return out_x0, out_x1
```

annotated instances of changes extracted from side-looking rural area satellite images, which are collected by optical satellites around the world. The images span one-to-three years with a resolution of 0.5–0.8 m/pixel.

LEVIR-CD dataset [11] contains 637 bitemporal RS image pairs and more than 31 333 annotated instances of changes, which are collected from Google Earth. Each image in the pairs is 1024 × 1024 with an image resolution of 0.5 m/pixels. Following the official division, we have 445/64/128 image pairs for training/validation/test, respectively.

B. Implementation Detail

We develop a change detection toolbox, Open-CD, based on PyTorch and OpenMMLab related tools [55]. During training, we use the CE Loss and AdamW optimizer. The weight decay is set to 0.05 always. The poly schedule with an initial learning rate of 0.001 is adopted. Particularly for Mix Transformer (MiT) [20] as the backbone, we use an initial learning rate of 0.0001. We use NVIDIA RTX A6000 and Tesla V100 GPUs for training and the batch size is set to 8. We train all Changer models for 80k and 40k iterations for S2Looking and LEVIR-CD dataset. For data augmentation, we use random crop, flip, and photometric distortion. And we randomly exchange the order of the bi-temporal images.

C. Evaluation Metrics

We use the F1-score with regard to the change category as the main evaluation indices. F1-score is calculated as follows:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$  

(10)

In addition, precision and recall are also reported, which are defined as follows:

$$\text{Precision} = \frac{TP}{(TP + FP)}$$  

(11)

$$\text{Recall} = \frac{TP}{(TP + FN)}$$  

(12)

where TP, FP, and FN are calculated on the change category, indicating true positive, false positive, and false negative. Some metrics such as receiver operating characteristic (ROC) [56], overall accuracy, etc., are not reported due to the high sample imbalance in the used datasets and these metrics are difficult to show algorithmic differences significantly.

D. Compared Methods

We make a comparison to several state-of-the-art (SOTA) methods, which are described as follows.

FC-EF, FC-Siam-Conc and FC-Siam-Diff [17] are three classification-based UNet-like models. FC-EF uses early fusion to directly concatenate bi-temporal images, FC-Siam-Conc and FC-Siam-Diff use siamese encoders and use concatenation and difference to fuse features, respectively.

DTCDSCN [51] is a multitask model, which can accomplish both change detection and semantic segmentation at the same time. It also introduces a dual-attention module to exploit the interdependencies between channels and spatial positions, improving the feature representation.

STANet [11] is a Siamese network with spatial-temporal attention designed to explore spatial-temporal relationships for change detection. It includes a base model (STANet-Base) that uses a weight-sharing CNN feature extractor, and optimizes models through the metric method. STANet-BAM and STANet-PAM equip the basic spatial-temporal attention module (like self-attention) and the pyramid spatial-temporal attention module on top of STANet-Base.

CDNet [52] is a well-tuned siamese CNN model. CDNet is used with an instance-level data augmentation, which can generate bi-temporal images that contain changes involving plenty and diverse synthesized building instances by leveraging generative adversarial training.

BiT [18] is a hybrid model of CNN and transformer. It uses the convolutional blocks at the shallow layers and the transformer blocks (with cross-attention) at the deeper
layers, which can effectively model contexts within the spatial-temporal domain.

IFN [53] continues the design of FC-Siam-Conc on the macro level, and on the micro level, it uses a more complex backbone, attention mechanism, and adds multiple auxiliary heads for deep supervision in the decoder to speed up the convergence of the model.

ChangeFormer [19] is a transformer-based siamese network. ChangeFormer combines a hierarchical transformer encoder with an MLP decoder in a siamese network to effectively render long-range details.

SNUNet [13] is a nested U-Net [24] structure. It uses dense skip connections between Siamese encoder and multiple sub-decoders to alleviate the loss of spatial position information in the deep decoder layers.

E. Main Results

In Table I, we show the results of some SOTA and typical change detection methods, including ConvNets and Vision Transformers. Our baseline ChangerVanilla achieves more competitive performance than previous change detection methods, which demonstrates MetaChanger’s effectiveness in the overall architecture. The FDAF brings a boost...
in F1-score, and its effect can also be observed in Fig. 4. ChangerAD and ChangerEx are built on top of ChangerAlign and make feature interactions during feature extraction. Compared to ChangerAlign, ChangerAD and ChangerEx achieved significant improvements with only slight or no increases in parameters and computational cost. Furthermore, the Changer models achieve more promising gains on the more challenging dataset S2Looking. In terms of model inference efficiency, although ChangerEx (with ResNet-18) has more parameters than BiT and SNUNet and more MACs than ChangeFormer-b1, it has the fastest inference speed on our testing GPU device (NVIDIA RTX A6000), taking only 35 ms to infer a 1024 × 1024 image pair.

The visualization comparison of the methods on the two datasets is shown in Fig. 5. In scenes where small targets exist, ChangerEx not only detects the small-sized change regions better than the other models, but also works better on details and edges, as shown in (a), (b), and (e) of Fig. 5. In scenes where there are dense targets, as shown in Fig. 5(d), ChangerEx can better separate the building instances, i.e., the edges of the regions segmented by ChangerEx are finer, which is helpful for post-processing of change instances. Furthermore, in Fig. 5(g), ChangerEx shows advantages for scenes where both “disappearances” and “appearances” occur in adjacent buildings, which is consistent with what feature exchange can theoretically achieve. SNUNet is second only to ChangerEx, which still has misdetected pixels at the connection of change region. In the S2Looking dataset, compared with in the LEVIR-CD dataset, there are larger style differences between the bi-temporal images. After using feature exchange, the feature distribution between the bi-temporal branches is closer, and domain adaptation between the bi-temporal domains is achieved to a certain extent, which makes it easier to detect changes. Therefore, the improvement of ChangerEx on the S2Looking dataset is more significant.

| Method          | Backbone | #Param (M) | MACs (G) | Latency (ms) | S2Looking Precision | S2Looking Recall | S2Looking F1 | LEVIR-CD Precision | LEVIR-CD Recall | LEVIR-CD F1 |
|-----------------|----------|------------|----------|--------------|---------------------|------------------|---------------|-------------------|----------------|-------------|
| FC-EF [17]      | -        | 1.35       | 12.48    | -            | 81.36               | 8.95             | 7.65          | 86.91             | 80.17          | 83.40       |
| FC-Siam-Conc [17]| -        | 1.54       | 19.47    | -            | 68.27               | 18.52            | 13.54         | 91.99             | 76.77          | 83.69       |
| FC-Siam-Diff [17]| -        | 1.35       | 17.06    | -            | 83.29               | 15.82            | 13.19         | 89.53             | 83.31          | 86.31       |
| DTCNet [51]     | SE-ResNet45 | 41.07     | 60.87    | -            | 68.58               | 49.16            | 57.27         | 88.53             | 86.83          | 87.67       |
| STANet-Base [11] | ResNet18  | -         | -        | 39           | 25.75               | 56.29            | 35.34         | 79.20             | 89.10          | 83.90       |
| STANet-Base [11]| ResNet18  | 12.18     | 49.16    | 398          | 31.19               | 52.91            | 39.24         | 81.50             | 90.40          | 85.70       |
| STANet-Base [11]| ResNet18  | 12.21     | 50.21    | 581          | 38.75               | 56.49            | 45.97         | 83.81             | 91.00          | 87.26       |
| CDNet [52]      | ResNet18  | 14.33     | -        | -            | 67.48               | 54.93            | 60.56         | 91.60             | 86.50          | 89.00       |
| BiT* [18]       | ResNet18  | 2.99      | 35.00    | 53           | 74.80               | 55.56            | 63.76         | 91.97             | 88.62          | 90.26       |
| IFN* [19]       | VGG-16    | 36.00     | 316.52   | 209          | 66.46               | 61.95            | 64.13         | 91.17             | 90.51          | 90.83       |
| ChangeFormer* [19]| M1-b1    | 13.94     | 26.42    | 72           | 72.82               | 56.13            | 63.39         | 92.59             | 89.68          | 91.11       |
| SNUNet [13]     | -         | 3.01      | 8.52     | 183          | 71.94               | 56.92            | 63.19         | 92.45             | 90.17          | 91.30       |
| ChangerVanilla  | ResNet18  | 11.39     | 23.74    | -            | 72.59               | 58.25            | 64.63         | 92.66             | 89.60          | 91.10       |
| ChangerAlign    | ResNet18  | 11.39     | 23.82    | -            | 71.62               | 60.06            | 65.33         | 93.30             | 89.59          | 91.41       |
| ChangerAD       | ResNet18  | 11.46     | 23.82    | -            | 74.21               | 58.97            | 65.72         | 93.34             | 90.12          | 91.70       |
| ChangerEx       | ResNet18  | 11.39     | 23.82    | 35           | 73.59               | 60.15            | 66.20         | 92.97             | 90.61          | 91.77       |
| ChangerEx†      | M1-b1     | 3.46      | 8.52     | 62           | 73.01               | 62.04            | 67.08         | 93.61             | 90.56          | 92.06       |

### Table II

**F1-Score (%) on Applying AD/Channel Exchange/Spatial Exchange on Different Stages on S2Looking. ✓ Means That Interaction Layer Is Used at This Stage**

| Stages w/ Feature Interaction | Stage1 | Stage2 | Stage3 | Stage4 | AD Exchange | Channel Exchange | Spatial Exchange |
|-------------------------------|--------|--------|--------|--------|-------------|------------------|-----------------|
|                               | ✓      | ✓      | ✓      | ✓      | ✓           | ✓                | ✓               |
|                               | 65.63  | 65.71  | 65.38  |        | ✓           | ✓                | ✓               |
|                               | ✓ ✓ ✓ ✓ | ✓ ✓ ✓ ✓ | ✓ ✓ ✓ ✓ | ✓ ✓ ✓ ✓ | ✓ ✓ ✓ ✓        | ✓ ✓ ✓ ✓          | ✓ ✓ ✓ ✓          |
|                               | 65.72  | 65.53  | 66.11  |        | ✓ ✓ ✓ ✓        | ✓ ✓ ✓ ✓          | ✓ ✓ ✓ ✓          |

### F. Ablation Studies

To delve into MetaChanger and its variants, especially ChangerEx, we conducted comprehensive experiments for the following questions. If not specified, ResNet18_V1c (without pretrained) is used as the backbone network.

1) **Which Stage to Interact:** We insert interaction layers including AD and channel/spatial exchange at different stages. As shown in Table II, inserting AD layer in the last three stages achieves the best performance, while inserting the interaction layer in the first stage degrades the performance. A similar situation occurs in channel exchange, the best results are obtained with the interaction layer in the latter two stages only. In spatial exchange, however, the situation is different, with the best two settings interacting in the earlier stages, as listed in Table II. An intuition is that in the shallow layers of the network, there is a high spatial resolution and a lower channel dimension, which is more suitable for spatial interactions; and vice versa. Based on this observation, we use spatial exchange in the shallow layers and channel exchange in the deeper layers for ChangeFormerEx, as shown in Fig. 3(a).

2) **What Ratio of Features Should Be Exchanged:** As shown in Table III, we try various exchange ratios, from 1/32 to 1/2. We find that the difference in the performance of ChangerEx at different exchange ratios is relatively slight. Thus, in feature exchange, the ratio of features exchanged is not the
Table III: Ablation Study on Exchange Ratios Used in Spatial and Channel Exchange on S2Looking (%)

| Exchange Ratio | Precision | Recall | F1   |
|----------------|-----------|--------|------|
| 1/32           | 72.31     | 60.72  | 66.01|
| 1/16           | 72.68     | 60.50  | 66.03|
| 1/8            | 72.81     | 60.43  | 66.05|
| 1/4            | 72.58     | 60.40  | 66.04|
| 1/2            | 73.59     | 60.15  | 66.20|

Table IV: Ablation Study on Different Window Size Options for Spatial Exchange on S2Looking (%)

| Window Size | Precision | Recall | F1   |
|-------------|-----------|--------|------|
| 8 x 8       | 72.83     | 60.38  | 66.02|
| 4 x 4       | 72.75     | 60.41  | 66.01|
| 2 x 2       | 72.98     | 60.40  | 66.10|
| 1 x 1       | 73.59     | 60.15  | 66.20|

Table V: Ablation Study on Whether Use Learnable Exchange for Spatial/Channel Exchange on S2Looking (%)

| Exchange | Learnable | Precision | Recall | F1   |
|----------|-----------|-----------|--------|------|
| Channel  | ✓         | 73.77     | 59.61  | 65.94|
|          | ✓         | 74.19     | 59.29  | 65.91|
| Spatial  | ✓         | 73.41     | 56.53  | 63.88|
|          | ✓         | 74.51     | 59.41  | 66.11|

determining factor; the emphasis is on the presence or absence of exchange. This ablation also suggests that feature exchange is robust, even when some additional hyper-parameters are introduced.

3) How to Choose the Size of the Exchange Window in Spatial Exchange: A worthwhile consideration is whether a small exchange patch in spatial exchange would disrupt the original spatial structure. We, therefore, tried using different window sizes in spatial exchange, from 1 x 1 to 8 x 8, as listed in Table IV. We find that the window size is an insensitive hyper-parameter and that a more complete original spatial structure does not result in a performance gain.

4) Learnable Exchange: We try the learnable exchange method in the channel and spatial dimensions, respectively. In our implementation, we generate an exchange map based on the distance between the soft attention maps of the two branches, with the smaller half being exchanged and the larger half retained. And the two attention maps also need to be exchanged if their corresponding features are exchanged. As Table V lists, the learnable channel exchange shows a slight improvement over the unlearnable one, but the performance of the learnable spatial exchange drops dramatically.

We have only tested simple learnable exchanges here, and this part deserves further exploration in future work.

5) MetaChanger With More Complex Backbones: To further illustrate the generalizability of ChangerEx to different network architectures, we replaced the backbone with the more complex networks, ResNeSt50 and ResNeSt101. As listed in Table VI, the networks with feature exchange outperform all the baselines significantly, demonstrating that the ChangerEx can generalize well on various models, especially in more challenging dataset.

We find that ChangerEx can deliver higher gains to complex models in large-scale datasets, which is promising. Specifically, ChangerEx with ResNeSt101 leads to a performance gain of 1.68% (achieving a F1-Score of 67.61%).

6) Why Exchange Work?: To further explore why ChangerEx is effective, we visualize the Changer models with and without “exchange” separately, using grad-CAM [57]. As shown in Fig. 6, most of the buildings in the upper half of the image disappear from $t_0$ to $t_1$. In the $t_0$-heat map with no exchange, only very few building areas are activated. In the $t_0$-heat map with exchange, the situation improves but is similar overall. However, an interesting phenomenon is that the areas with buildings in $t_0$ are activated in the $t_1$-heat map with feature exchange. In other words, the perceptual targets that are lost in $t_0$ are reactivated in $t_1$. Another possible explanation is that feature exchange increases the diversity of the samples, achieving a kind of intranetwork data augmentation. The order of appearance and disappearance of targets of interest in the bi-temporal features is shuffled, but still keeping the strict semantic preservation and semantic correspondence.
V. Conclusion

In this article, we propose MetaChanger to explore the effect of feature interactions in change detection. To verify the effectiveness of feature interaction, we deliberately specify the interaction layer as extremely simple AD and feature “exchange” for MetaChanger. It is found that the derived ChangerAD and ChangerEx can achieve competitive performance on multiple change detection datasets. Extensive ablation studies demonstrate the robustness and extensibility of ChangerEx.

In the future, we will further evaluate MetaChanger under more different learning settings and related tasks, such as semantic change detection. We hope this work can inspire more future research devoted to improving the MetaChanger, especially the interaction methods.

REFERENCES

[1] A. Toker et al., “DynamicEarthNet: Daily multi-spectral satellite dataset for semantic change segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 21158–21167.

[2] C. Shi et al., “S2Looking: A satellite side-looking dataset for building change detection,” Remote Sens., vol. 13, no. 24, p. 5094, Dec. 2021.

[3] S. Verma, A. Panigrahi, and S. Gupta, “QFabric: Multi-task change detection dataset,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2021, pp. 1052–1061.

[4] Y. Wang, W. Huang, F. Sun, T. Xu, Y. Rong, and J. Huang, “Deep multimodal fusion by channel exchanging,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 4835–4845.

[5] Y. Zhang, T. Yao, Z. Qiu, and T. Mei, “Lightweight and progressively-scalable networks for semantic segmentation,” 2022, arXiv:2207.13600.

[6] 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, 2022, Art. no. 5064816.

[7] O. Manas, A. Lacoste, X. Giro-i-Nieto, D. Vazquez, and P. Rodriguez, “Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 9414–9423.

[8] 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.

[9] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2, Jun. 2006, pp. 1735–1742.

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

[11] H. Chen and Z. Shi, “A spatial–temporal attention-based method and a Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, “Change detection based on deep Siamese network for optical aerial images,” IEEE Geosci. Remote Sens. Lett., vol. 14, no. 10, pp. 1845–1849, Oct. 2017.

[12] Z. Zheng, A. Ma, L. Zhang, and Y. Zhong, “Change is everywhere: Single-temporal supervised object change detection in remote sensing imagery,” in Proc. Inf. Inf. Vis. Conf. Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 15193–15202.

[13] S. Fang, K. Li, J. Shao, and Z. Li, “SNUNet-CF: A densely connected Siamese network for change detection of VHR images,” IEEE Geosci. Remote Sens. Lett., vol. 19, pp. 1–5, 2022.

[14] Z. Zheng, Y. Zhong, S. Tian, A. Ma, and L. Zhang, “ChangeMask: Deep multi-task encoder-transformer-decoder architecture for semantic change detection,” IPSRS J. Photogramm. Remote Sens., vol. 183, pp. 228–239, Jan. 2022.

[15] P. Chen, B. Zhang, D. Hong, Z. Chen, X. Yang, and B. Li, “FCCDN: Feature constraint network for VHR image change detection,” IPSRS J. Photogramm. Remote Sens., vol. 187, pp. 101–119, May 2022.

[16] Y. Deng et al., “Feature-guided multitask change detection network,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 15, pp. 9667–9679, 2022.

[17] R. C. Daundt, 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.

[18] 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, doi: 10.1109/TGRS.2021.3095166.

[19] W. G. C. Bandara and V. M. Patel, “A transformer-based Siamese network for change detection,” in Proc. IEEE Int. Geosci. Remote Sens. Symp., Jul. 2022, pp. 207–210.

[20] Z. Zheng, A. Ma, L. Zhang, and Y. Zhong, “Change is everywhere: Semantic change detection. We hope this work can inspire more different learning settings and related tasks, such as semantic change detection. We hope this work can inspire more future research devoted to improving the MetaChanger, especially the interaction methods.

[21] REFERENCES

[1] A. Toker et al., “DynamicEarthNet: Daily multi-spectral satellite dataset for semantic change segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 21158–21167.

[2] C. Shi et al., “S2Looking: A satellite side-looking dataset for building change detection,” Remote Sens., vol. 13, no. 24, p. 5094, Dec. 2021.

[3] S. Verma, A. Panigrahi, and S. Gupta, “QFabric: Multi-task change detection dataset,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2021, pp. 1052–1061.

[4] Y. Wang, W. Huang, F. Sun, T. Xu, Y. Rong, and J. Huang, “Deep multimodal fusion by channel exchanging,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 4835–4845.

[5] Y. Zhang, T. Yao, Z. Qiu, and T. Mei, “Lightweight and progressively-scalable networks for semantic segmentation,” 2022, arXiv:2207.13600.

[6] 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, 2022, Art. no. 5064816.

[7] O. Manas, A. Lacoste, X. Giro-i-Nieto, D. Vazquez, and P. Rodriguez, “Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 9414–9423.

[8] 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.

[9] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2, Jun. 2006, pp. 1735–1742.

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

[11] 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, p. 1662, May 2020.

[12] Z. Zheng, A. Ma, L. Zhang, and Y. Zhong, “Change is everywhere: Single-temporal supervised object change detection in remote sensing imagery,” in Proc. Inf. Inf. Vis. Conf. Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 15193–15202.

[13] S. Fang, K. Li, J. Shao, and Z. Li, “SNUNet-CF: A densely connected Siamese network for change detection of VHR images,” IEEE Geosci. Remote Sens. Lett., vol. 19, pp. 1–5, 2022.

[14] Z. Zheng, Y. Zhong, S. Tian, A. Ma, and L. Zhang, “ChangeMask: Deep multi-task encoder-transformer-decoder architecture for semantic change detection,” IPSRS J. Photogramm. Remote Sens., vol. 183, pp. 228–239, Jan. 2022.

[15] P. Chen, B. Zhang, D. Hong, Z. Chen, X. Yang, and B. Li, “FCCDN: Feature constraint network for VHR image change detection,” IPSRS J. Photogramm. Remote Sens., vol. 187, pp. 101–119, May 2022.

[16] Y. Deng et al., “Feature-guided multitask change detection network,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 15, pp. 9667–9679, 2022.
[40] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, “TDAN: Temporally-deformable alignment network for video super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 3360–3369.
[41] J. Lu, J. Yang, D. Batra, and D. Parikh, “Hierarchical question-image co-attention for visual question answering,” in Proc. Adv. Neural Inf. Process. Syst., vol. 29, 2016, pp. 1–9.
[42] D.-K. Nguyen and T. Okatani, “Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6087–6096.
[43] X. Chen et al., “Bi-directional cross-modality feature propagation with separation-and-aggregation gate for RGB-D semantic segmentation,” in Computer Vision—ECCV 2020 (Lecture Notes in Computer Science), vol. 12356, A. Vedaldi, H. Bischof, T. Brox, and J. M. Frahm, Eds. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-58621-8_33.
[44] B. Wu, J. Ma, G. Chen, and P. An, “Feature interactive representation for point cloud registration,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 5530–5539.
[45] X. Wei, T. Zhang, Y. Li, Y. Zhang, and F. Wu, “Multi-modality cross attention network for image and sentence matching,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10941–10950.
[46] X. Li, W. Wang, X. Hu, and J. Yang, “Selective kernel networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 510–519.
[47] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, “Deformable Siamese attention networks for visual object tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6728–6737.
[48] Q. Chen et al., “MixFormer: Mixing features across windows and dimensions,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 5249–5259.
[49] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 1–11.
[50] A. G. Howard et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861.
[51] Y. Pan, C. Pang, Z. Zhan, X. Zhang, and X. Yang, “Building change detection for remote sensing images using a dual-task constrained deep Siamese convolutional network model,” IEEE Geosci. Remote Sens. Lett., vol. 18, no. 5, pp. 811–815, May 2021.
[52] H. Chen, W. Li, and Z. Shi, “Adversarial instance augmentation for building change detection in remote sensing images,” IEEE Trans. Geosci. Remote Sens., vol. 60, 2022, Art. no. 5603216.
[53] 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.
[54] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 28, 2015, pp. 1–9.
[55] OpenMMLab Contributors. (2018). MMVC: OpenMMLab Computer Vision Foundation. [Online]. Available: https://github.com/openmmlab/mmvc
[56] C.-I. Chang, “Comprehensive analysis of receiver operating characteristic (ROC) curves for hyperspectral anomaly detection,” IEEE Trans. Geosci. Remote Sens., vol. 60, 2022, Art. no. 5541124.
[57] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 618–626.

**Sheng Fang** received the B.S. degree in applied mathematics and the B.E. degree in material science from Shanghai Jiao Tong University, Shanghai, China, in 1994, the M.S. degree in computer science from Tsinghua University, Beijing, China, in 2002, and the Ph.D. degree in computer science from the Shandong University of Science and Technology, Qingdao, China, in 2008. He is currently a Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology. He has published more than 30 articles. His research interests are in the fields of video coding, understanding, and processing. Dr. Fang is a Senior Member of the China Computer Federation (CCF), a member of the CCF Multimedia Technical Committee, and an Association for Computing Machinery (ACM) Special Interest Group on Multimedia Systems (SIGMM) China Chapter Member.

**Kaiyu Li** (Graduate Student Member, IEEE) received the B.E. degree from the Shandong University of Science and Technology, Qingdao, China, in 2020, where he is currently pursuing the M.E. degree under the supervision of Dr. S. Fang. His major research interests are in remote sensing visual perception and Earth vision.

**Zhe Li** received the B.E. degree in computer science and technology, and the M.E. degree in computer application technology from the Shandong University of Science and Technology, Qingdao, China, in 2005 and 2009, respectively, and the Ph.D. degree in computer application technology from Beihang University, Beijing, China, in 2014. He is currently an Assistant Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology. His research interests include remote sensing image processing, video coding, and video surveillance.