Remote Sensing Image Change Detection based on Lightweight Transformer and Multi-scale Feature Fusion

Jingming Li, Panpan Zheng and Liejun Wang

***Abstract*—As deep learning demonstrates excellent performance in remote sensing image change detection (CD), early methods that mostly used Convolutional Neural Networks (CNNs) have limitations in the accuracy due to their insufficient global feature representation, an inherent shortcoming of CNNs. The lack of global feature can lead to notable issues, such as the inability to detect small targets and loss of edge information. In recent years, vision transformers (ViTs) have been employed in CD owing to their powerful global feature representation capabilities. However, pure transformer methods lack effective local feature extraction, which also restricts the performance of CD, while the original transformer models require a large amount of computing resources. To address these issues and improve CD performance, we propose a Lightweight Transformer-based Multi-scale Feature Fusion network (LTMFFNet). By integrating CNN structures both before and after the multi-head self-attention in each layer of the main backbone, we enhance the encoder's local feature extraction ability and reduce the computational complexity through convolution and linear operations. For the siamese encoding outputs at different scales, we design two distinct fusion modules based on depth-wise convolution for bitemporal information fusion in deep layers and shallow layers, respectively. Our model employs a multi-layer cascaded structure with a deep supervision strategy applied to multiple outputs. Experiments on four public CD datasets demonstrate that our network achieves better performance while maintaining relatively smaller computational complexity compared to other state-of-the-art methods for CD.**

***Index Terms*—Remote sensing (RS), change detection (CD), convolutional neural network (CNN), transformer.**

I. Introduction

REMOTE sensing (RS) image change detection (CD) is a methodology which monitors changes in the identical location over various periods by processing dual temporal images to obtain change maps. As shown in Fig. 1, the inputs of the CD task are two RS images from different times and the output is a change map where black denotes unchanged position while white indicates regions that have undergone changes. Change detection of RS images is widely used in fields such as land use [1], [2], agricultural and forestry monitoring [3], [4], urban and environmental planning [5], [6] and disaster assessment [7], [8]. It plays a crucial role in these applications. Therefore, automated CD has received increasing attention and research interest.

Machine learning technologies, particularly deep learning, have been heavily applied to CD, achieving significant breakthroughs in recent years [9]. In general, the accuracy of edge detection has always been a problem in CD tasks. Meanwhile, smaller change areas are often missed, which is also a major challenge in CD [10], [11]. So the focus of CD research is to reduce such kinds of of error to improve the final accuracy. The initial CD methods were mainly based on Convolutional Neural Networks (CNNs) [12], which demonstrated good performance in numerous tasks of computer vision (CV), including image segmentation, target recognition, and object detection. Effective CNN-based methods for CD have been proposed, such as the Full Convolutional Early Fusion Network (FC-EF), the Full Convolutional Siamese Difference Network (FC-Siam-Diff), and the Full Convolutional Siamese Concatenation Network (FC-Siam-Conc) [13]. They use different fusion methods based on simple convolutional residual networks for CD tasks. However, due to the inherent shortcomings of CNN in global feature extraction, the rich features contained in dual temporal RS images have not been fully extracted or utilized. Therefore, some methods have enhanced pure CNN networks. For example, SNUNet [14] used dense skip connections to extract information from different scale and introducing improved attention of channels for multi-scale feature fusion. Similarly, STANet [15] introduced self-attention to enhance the interactivity of dual temporal image encoding. Yu et al. [16] introduced an interactive attention module and a multi-dimensional convolutional frequency attention module to construct a dual-branch encoding backbone. Although the introduction of attention mechanisms enhances CNN's recognition ability, the global feature extraction remains insufficient. In CD task, large areas of change require effective global feature extraction and the capture of dependency relationships between different positions, so the comprehensive performance of CNN methods in the CD direction has certain limitations.

With the proposal of transformer, based on multi-head self-attention (MHSA) [17], which shows strong performance in the field of NLP, the CV field has also started using transformer to solve related problems. The vision transformer (ViT) [18], which divides the input image into small pieces and converts each one into a fixed-length vector for transformer blocks to process, was proposed for image classification tasks. Its excellent performance in image classification tasks has

This research was funded in part by Scientific and Technological Innovation 2030 major project under Grant 2022ZD0115800, in part by the Xinjiang Uygur Autonomous Region Tianshan Talent Program Project under grant 2022TSYCLJ0036, and in part by the National Natural Science Foundation of China (Regional Project) under Grant No.62466056. *(Corresponding author: Liejun Wang.)*

Jingming Li, LieJun Wang, and Panpan Zheng are with the School of Computer Science and Technology, Xinjiang University, 57 Urumqi 830046, China (e-mail: ljmxju@stu.xju.edu.cn; wljxju@xju.edu.cn; 007652@xju.edu.cn).

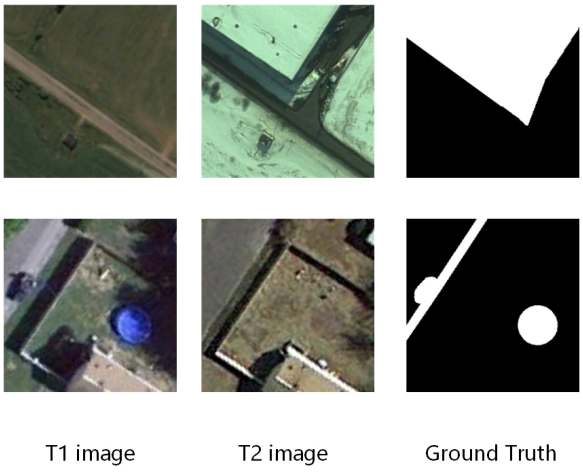

Fig. 1. Diagram of the CD task, where T1 and T2 images are images from different temporal.

demonstrated effectiveness of transformer in CV. As a result, it has been widely adopted for other CV tasks. In CD, ViT and its variants have been applied to several models, showing excellent performance. Yan et al. [19] developed a pure transformer-based CD network. Chen et al. [20] enhanced the feature extraction ability of CNN backbone by incorporating ViT. Bandara et al. [21] proposed ChangeFormer, which encodes RS images with transformer module and generates the final change map through MLP. Zhang et al. [22] proposed a CD model composed entirely of swin transformer modules which introduces a window-based mechanism.

The MHSA structure enables transformer to effectively represent global features and capture dependency relationships between different positions, which is lacking in pure CNN methods. Moreover, transformer can process sequences in parallel, improving training speed. However, transformer lack the ability to perceive local features, making it difficult to capture precise positional information in remote sensing images. This limitation reduces their effectiveness in detecting small targets and edge details in changing areas. Additionally, due to global modeling, the multi-head self-attention mechanism demands significant computing resources, especially for high-resolution images. As a result, although transformer benefits from faster training speeds through parallel computation, it exhibits high computational complexity and consume substantial memory.

To address the limitations of pure transformer methods, a key research focus in the field of CV is the integration of transformer with CNNs to balance their strengths and mitigate their weaknesses. Chen et al. [20] used CNN-based encoding to generate tokens, which were then fed into a transformer to produce the change map. Li et al. [23] introduced transformer layers following a CNN backbone for CD. These works directly combine independent CNN backbones with transformer networks, leveraging the feature extraction capabilities of both. However, they fail to fully utilize image information across multiple scales. Guo et al. [24] proposed a hybrid network named CMT, which integrates ViT with CNN. CMT introduces low-computation depthwise convolution operations [25] before and after multi-head self-attention in the transformer module, enhancing the perception of local features while reducing computational overhead through convolutional downsampling in MHSA. At the same time, it incorporates convolutional preprocessing layers into the overall structure. Yun et al. [26] also proposed a transformer-based method with efficient operators, and it only uses a few tokens by random initialization to resprensent global features for more efficient calculations. Li et al. [27] developed a novel hybrid architecture by stacking new types of convolutional and transformer blocks, effectively integrating these modules at each stage of the encoder while maintaining an optimal balance between them. Lu et al. [28] proposed SBCFormer, which combines the attention mechanism of transformer with convolutional operations. SBCFormer enhances the output of attention using standard CNN components, and applies pointwise convolution operations to all components, replacing traditional linear transformations for query and key vectors. These methods demonstrate state-of-the-art (SOTA) performance with reduced parameter counts and computational complexity in image classification tasks, proving the feasibility of combining transformer with convolutional structures in CV applications. Consequently, we speculate that the hybrid approach of CNN and transformer could also be effectively applied to CD.

Inspired by above observations, we propose a Lightweight Transformer-based Multi-scale Feature Fusion network (LTMFFNet). It consists a siamese encoder, four fusion modules, and a decoder. The encoder is mainly composed of lightweight transformer (LWT) blocks which integrates with convolutional structure, compensating for the shortcomings of pure transformer models and reducing a certain number of parameters and computational complexity. In each branch of the encoder, remote sensing image is first fed into multi-layer convolution preprocessing for initial feature extraction. The preprocessed image will be processed through four LWT layers with patch embedding modules to generate multi-scale feature maps. The patch embedding module is located before the LWT block, which will convert the image into tokens and perform downsampling. The encoder's shallow and deep layers generate feature maps containing coarse-grained spatial information and fine-grained semantic information, respectively. So we input the generated multi-scale dual-time domain features into two different fusion modules based on depthwise convolution: the Convolutional Fusion Module (CFM) and the Convolutional Differentiation Module (CDM), to extract different levels of disparate features. The decoder similarly uses four LWT layers to process the fused multi-scale feature map, where each LWT layer has an upsampling module after the LWT blocks and combines convolution operations to gradually restore the original size of the image and integrate features of different scales by skip connections. Finally, we use an output module that combines multi-layer convolution and linear upsampling to generate change maps. To further enhance the utilization of feature information at different scales during training, we introduce an auxiliary branch in the third layer of the decoder. This branch generates auxiliary feature maps for deep supervision, improving training efficiency and accuracy.The main contributions of our paper can be summarized as follows:

1) We propose a cascaded twin U-shaped network based on LWT block for the CD task. Combining convolutional operations with the transformer backbone enhances local perception and therefore improves overall performance while the computational complexity is reduced.
2) We designed two kinds of convolutional fusion modules named CFM and CDM in different layer of our model to fuse dual temporal features for processing the differential information at different levels in a targeted manner.
3) We designed an output module in decoder with a stem module in encoder to enhance the utilization of local information in LWT and use a strategy of deep supervision in the last two layers of the decoder, achieving full utilization of multi-level feature extraction.
4) We conduct experiments on four public datasets for change detection in which the results shows our model exceeds other SOTA methods on four mainstream CD datasets and reduces params and computation compared to pure transformer networks.

The remaining part of this paper is organized as follows. In Section II, we describe the relevant work of CD. In Section III, we elaborate on the overall framework and various parts of LTMFFNet. In section IV, we record the relevant experiment with analysis. Finally, in section V, we will have a final discussion and summary.

II. Related Work

In this section, we will review CNN-based and transformer-based CD approaches in these years, respectively..

A. CNN Methods for CD

As deep learning becomes more and more powerful and popular, more and more deep learning methods are introduced into CD, and many early CD methods were based on CNN. Through the powerful image feature extraction capabilities of convolutional networks such as Fully Convolutional Neural Networks (FCN) [29], Unet [30], and Residual Networks (Resnet) [31], feature of bitemporal RS images can be fully extracted for generating differential features. The CNN-based CD method is usually a siamese structure [32], which can enhance the semantic representation and feature differentiation of RS images by modifying the encoder or decoder, modifying the way fusion is performed, and combining attention mechanisms, while achieve better training results by optimizing the loss function.

In the direction of CD, Zhan et al. [33] first applied siamese convolutional networks to change detection task, processing input dual temporal RS images simultaneously through a dual branch network structure. Most of the subsequent CD methods are then based on the siamese network structure. Daudt et al. [13] designed three change detection architectures based on fully connected neural networks, which directly connect dual temporal images or process them through siamese structures. Varghese et al. [34] used resnet backbone to extract information from different levels and integrated it through FCN. Chen et al. [35] proposed a siamese convolutional network for CD based on resnet, which has stronger feature extraction ability when combined with dual attention. Fang et al. [14] proposed a dense connected sparse network (SNUNet) which combines Unet++ [36] and extracts multi-scale information through multi-level decoding combined with dense skip connections, and finally fuses through an ensemble channel attention module. It simultaneously uses a mixed loss function combining bce and dice loss for optimization. To reduce the number of parameters, Xing et al. [37] proposed a lightweight network which utilizes early fusion through a deep supervised fusion module to achieve good performance with few parameters.

The traditional CNN methods can effectively extract local features of RS images and detect changing regions using positional information [38] with a relatively small number of parameters. However, convolutional operation has an inherent flaw of insufficient attention to global information, so pure CNN networks cannot make full use of long-term global information, thereby significantly limiting the performance of CD networks. Therefore, some CD methods introduce attention mechanisms into CNN networks, such as the ensemble channel attention proposed by Fang et al. [14] to fuse multi-scale convolutional outputs, and Chen et al. [15] 's pyramid attention model for acquiring features with different scales. Chen et al. [35] and Liu et al. [39] also introduced dual attention mechanisms [40] into change detection tasks. The attention mechanism has shown corresponding performance in CD tasks and enhances the feature representation of CNN. Furthermore, with the application of transformer based on multi-head self-attention in CV field with its powerful performance, transformer has also been introduced in CD tasks.

B. Transformer Methods for CD

With the popularity of ViT in the field of CV in 2020 due to its powerful modeling ability for global image features, the CD methods have also begun to use ViT and its variants. Chen et al. [20] firstly introduced transformer into the CD field and proposed bitemporal image transformer (BiT). It extracts the semantics of bitemporal RS images through CNN and converts them into tokens, which are then input into transformer for encoding and decoding operations. Li et al. [23] also introduced transformer to extract global features for supplementing after the CNN-based encoder.

The above methods supplement CNN with transformer, and there are also CD methods mainly using transformer. Bandara et al. [21] proposed a siamese network architecture based on transformer, which designs a dual-branch transformer for feature extraction and downsampling operations, extracts multi-scale features, and then fuses multi-scale differential features through an MLP decoder to generate the final prediction map. Zhang et al. [22] proposed a Unet structure CD model called SwinSUNet based on swin transformer [41] which is a ViT variant based on sliding window mechanism, which is the first fully transformer-based method used for CD. The main backbone is totally based on the swin transformer block. Yan et al. [19] proposed a completely swin transformer based change detection network, which combines multi-level features from the swin transformer based on a feature pyramid and performs deep supervision through multiple sets of outputs.

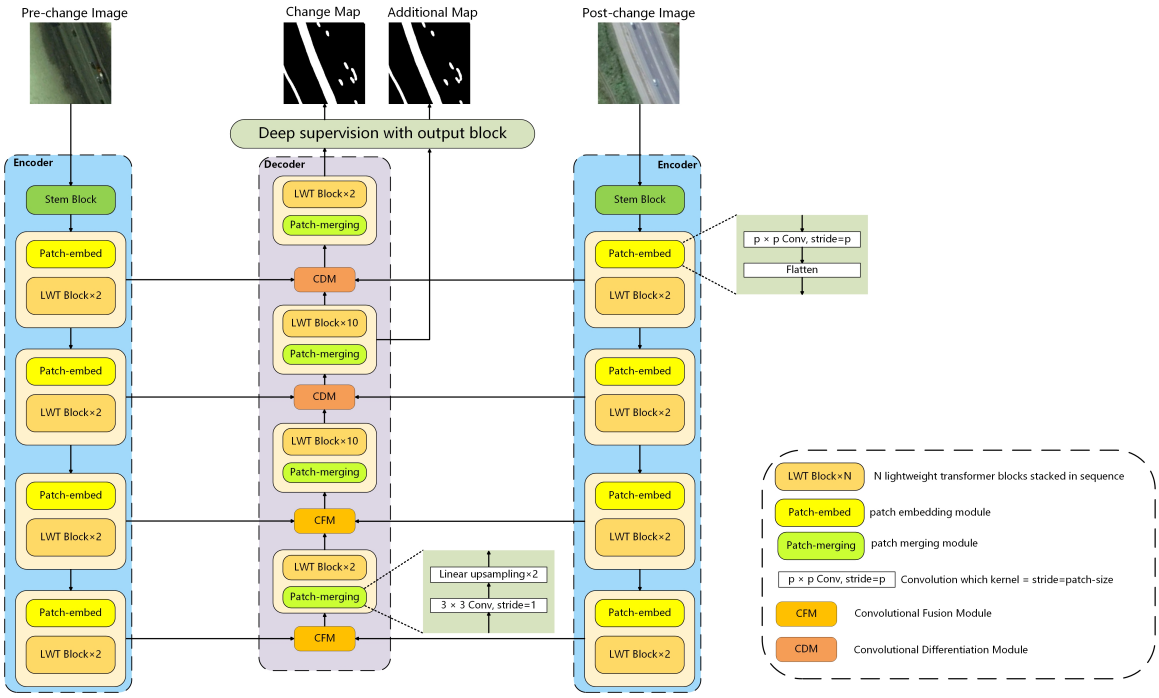

Fig. 2. Architecture of proposed LTMFFNet. It mainly consists of a dual branch encoder, fusion modules, and a decoder. The dual temporal RS image is input into the encoder, which extracts features and downsamples through four layers of LWT modules. The outputs of the first two layers and the last two layers are integrated through different fusion modules and input into a decoder composed of four layers of LWT modules for upsampling. The outputs of the last two layers are then subjected to depth supervision to generate the final change map.

Compared to pure convolutional networks, transformer has a strong global perception field that can model long-distance dependencies with a constant number of layers, and can perform parallel operations [42]. A major problem that exists and needs to be solved in these CD models using transformer is the lack of detection for small targets and edge detection, which is also a major challenge in the current direction of change detection. Due to the transformer's self- attention focusing more on global feature capture, there is a lack of effective extraction of local features in the image, and there are small change regions and irregular edge information in change detection, which is difficult for pure transformer method to pay attention to. A fixed size transformer is also not conducive to CD tasks that require multi-scale feature extraction. What's more, multi-layer stacked transformer often has mass parameters and computation, which affect the training efficiency and deployment of the model.

In response to problems in transformer, Guo et al. [24] uses a transformer architecture that combines CNN and hierarchical structure, which performs fine-grained feature extraction through convolution operations and multi-level feature extraction while reducing computational consumption, in order to reduce computation while enhancing model performance. Taking inspiration from this, we introduce a multi-scale CNN and transformer combined model into the change detection task and designed a model with siamese U-Net structure.

III. Methodology

Given two optical RS images, representing the observation of the same area before and after changes, the task of CD is to generate the predicted change map, which includes estimated actual change areas and unchanged areas. In this part, we offer an detailed description of LTMFFNet. In Section III-A, we provide an general introduction for the overall framework of LTMFFNet. In section III-B to III-E, we respectively introduce the LWT block we used in our network, encoder and decoder, fusion module, prediction part and loss function calculation.

A. Framework Overview

The structure of LTMFFNet is displayed in Fig. 2. As we can see, LTMFFNet consists of an encoder network, two kinds of fusion block and a decoder network. The decoder network uses two different kinds of prediction head to generate two change maps, one of which is used in deep supervise while the other one is the final result.

The inputs of LTMFFNet are a pair of bitemporal images, and the outputs are predicted maps of CD. The encoder network is a siamese structure and each branch contains four layers, each of which is the combination of a patch-embed module and several LWT blocks. Before the four layers, there is a convolution stem block to firstly extract fine-grained feature of the input picture. Then, each layer will convert the image into image tokens by patch-embed module and extract its feature by LWT blocks which combine convolutional operations with Multi-head Self-Attention.

The outputs of a branch are four feature maps of different scale from corresponding layer. For the third and fourth layers, we design a fusion block to integrate the deep feature of each branch. As for the First two layer, we use a feature differentiation module to extract the change information of

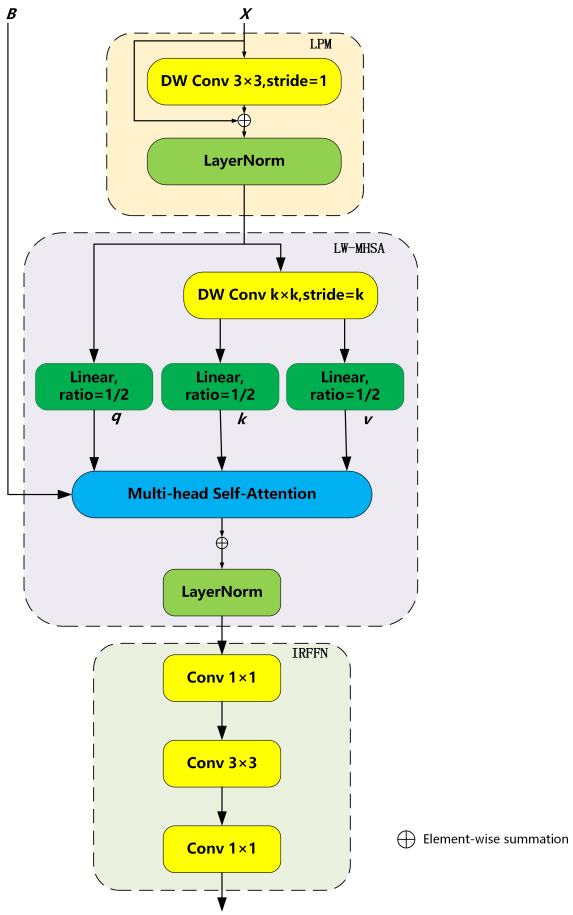

Fig. 3. Structure of a LWT block. It consists of LPM, LW-MHSA and IRFFN from top to bottom.

bitemporal feature maps. Next, we input them into the decoder which similarly consists of four levels. Every layer is a stack of several LWT blocks and a upsample module with convolution. The final output of encoder after fusion firstly enters the decoder and sequencely restore the change information and change its size to fuse feature maps of other scale. Finally, we use a output module to gain the ultimate change map. And we also generate a auxiliary Change map by upsample and convolution operation for deep supervision.

B. LWT Block

In our LTMFFNet, We use the LWT block which enhances local feature extraction capabilities while reducing computational complexity and params. As shown in Fig. 3, this block contains a local perception module (LPM), a lightweight multi-head self-attention (LW-MHSA) module and an inverted residual feed-forward network (IRFFN).

(1) LPM: To keep the translation-invariance in data augmentation and focus on local correlations and the structure information, this block inserts a LPM before the multi-head attention. It uses a depth-wise convolution with a residual connection to firstly extract the input image's local information.

2) LW-MHSA: In previous MHSA module, the input $X \in R^{n\times d}$ is projected to query $Q \in R^{n\times d_k}$, key $K \in R^{n\times d_k}$ and value $V \in R^{n\times d_v}$ by linear operation where $n = H \times W$ reflects the number of patches while the notation d, d_k and d_v respectively denote the number of dimensions for X, query (key) and value. The self-attention represented by SA is calculated in this way:

$$SA(q,k,v) = Softmax(\frac{QK^T}{\sqrt{d_k}}) \times V \tag{1}$$

To save computational costs, the LWT block uses a $k \times k$ depth-wise convolution with stride k which is denoted by $DwConv$ to reduce the size of K and V before the MHSA. Moreover, We also reduce the dimensions of Q and K by half in linear mapping from input to query and key so we get $Q \in R^{n\times \frac{d_k}{2}}$ and key $K \in R^{n\times \frac{d_k}{2}}$ firstly in this block. In this way, we can get $K^{'} = DwConv(K) \in R^{n\times \frac{d_k}{2}}$ and $V^{'} = DwConv(V) \in R^{\frac{n}{k^2}\times d_v}$ as lightweght key and value where DwConv means depth-wise convolution. We use different k which is the ratio of k and v reduction in different layer of our model and we list them in Table I. So the Besides, the LWT block also uses a relative position bias B which can be learned on each SA module. Overall,the lightweight MHSA which is abbreviated as $LightSA$ is applied as:

$$LightSA(Q,K,V) = Softmax(\frac{QK^{'T}}{\sqrt{d_k}} + B)V^{'} \tag{2}$$

where the bias $B \in R^{n\times \frac{n}{k^3}}$ is randomly initialized for each LWT block. Combine with the learnt relative position bias, the MHSA can better learn the local correlations of the image.In the end, with heads' number h, each head in MHSA outputs a sequence with size $n \times \frac{d}{h}$, and h sequence are connected into the final sequence with size $n \times d$.

3) IRFFN: In the ViT, the FFN contains two linear units with a GELU layer in the middle. One extends the dimension of map by a multiple of 4 while the other restore the original dimension. To get better performance, lightweight block changes the location of skip connection with convolutional processing in IRFFN [43]:

$$F(x) = DwConv(x) + x \tag{3}$$

$$IRFFN(x) = Conv(Act(F(Act(Conv(x))))) \tag{4}$$

where $Act(\cdot)$ denotes the activation layer followed with a batch normalization. The middle convolution is a depth-wise convolution which can capture regional feature with little expense. Besides, a shortcut is used between the first 1×1 convolution and the last 1×1 convolution to improve propagation capabilities.

To summarize, the complete calculation process of LWT block are shown in Algorithm 1.

Algorithm 1 LWT

Input: X_{i-1} (input of the i-th block), B (nitial bias)
Output: X_i (output of the i-th block)

// step1: preliminary feature extraction through LPM based on deep convolution
$Y_i = DWCONV(X_{i-1})$
// step2: extraction of Multi-Head Self-Attention with randomly initialized Bias
$Z_i = LW - MHSA(LN(Y_i) + B)$
$Z_i = Z_i + Y_i$
// step3: output by IRFFN
$X_i = IRFFN(LN(Z_i))$
$X_i = X_i + Z_i$

$$Y_i = LPM(X_{i-1}) \tag{5}$$

$$Z_i = LW - MHSA(LN(Y_i)) + Y_i \tag{6}$$

$$X_i = IRFFN(LN(Z_i)) + Z_i \tag{7}$$

where LN indicates the layer normalization. And X_i denotes the input of the i-th block while Y_i and Z_i denote the the output features of LPM and LW-MHSA module in corresponding block. In each layer in our LTMFFNet, we use different numbers of lightweight blocks in each layer of encoder and decoder shown as Fig. 2.

C. Encoder and Decoder

In our LTMFFNet, before we use the LWT block to extract the bitemporal images' feature, there is a stem block in the encoder to firstly extract the fine-grained feature. As shown in Fig. 4(a), the stem block contains one 3×3 convolution and two 3×3 depth-wise convolutions with GELU activation [44] and batch normalization. The first convolution changes the input channel into a stem channel and do a down sampling with a scale factor of 2. And we add two skip connections in next convolutions. So the calculation of the stem block is given as:

$$X_1 = Act(DownConv(X)) \tag{8}$$

$$X_2 = Act(DwConv(X_1)) + X_1 \tag{9}$$

$$Out = Act(DwConv(X_2)) + X_2 \tag{10}$$

where $Act(\cdot)$ denotes GELU activation layer with batch normalization and $DownConv$ is the initial 3×3 convolution for down-sampling while $DwConv$ denotes depth-wise convolutions. X, X_1, X_2 and Out denote the input, the two convolution layer's output and the final output, respectively. The input channel is 3 for RGB image and we choose 16 as stem channel here.

After stem convolution, the input image need to be transformed into image tokens like ViT. For each layer in encoder, there is a patch embedding block to convert the image into several tokens and map each token's channel number to a embedding dimension C. In our model, each patch have a size of 2×2 so that there are $(H/2) \times (W/2)$ patches and we can consider each patch as a token. Meanwhile, we change the dimension into C. Then we get a feature map with size of $(H/2) \times (W/2) \times 2$. We implement them by a convolution with kernel size 2 and stride 2. By doing the convolution, we flatten each patch to size of 1×1 so each spatial point can be seen as a token. It's equivalent to doing a down sampling with a scale factor of 2 for the input image. Finally we reshape each token to 1-D tensor so that the output size of patch embedding block is $(H/2) \times (W/2) \times 2$.

TABLE I
THE NUMBERS OF BLOCKS AND EMBEDDING DIMENSION IN EACH LAYER OF LTMFFNET. E-LAYER AND D-LAYER INDIVIDUALLY REPRESENT DIFFERENT LAYER IN ENCODER AND DECODER.

Layer	Numbers of blocks	Embedding dimension	Ratio of k and v reduction
E-Layer 1	LWT block x 2	46	8
E-Layer 2	LWT block x 2	92	4
E-Layer 3	LWT block x 10	184	2
E-Layer 4	LWT block x 2	368	1
D-Layer 1	LWT block x 2	368	1
D-Layer 2	LWT block x 10	184	2
D-Layer 3	LWT block x 2	92	4
D-Layer 4	LWT block x 2	46	8

After patch embedding in each layer, the image tokens are processed through stacked LWT blocks. After each layer in encoder, we merge the patch by reshaping so that we can get a image with embedding dimension and half the size. Table I lists the quantity of LWT blocks with embedding dimension in each layer of the encoder. It also lists the reduction ratio of k and v in each layer.

Each branch of encoder outputs four feature maps from four layers. After we fuse bitemporal feature maps from the same layer in siamese encoder by fusion module, we get four fusion feature maps. Then we can use decoder for multi-scale feature aggregation and generating change graph.

The decoder of LTMFFNet similarly consists of four layers. Each layer has the same numbers of LWT blocks and embedding dimensions as the encoder has in reverse order, just like Table I displays. We put the fourth fusion feature map into the first layer of decoder firstly, and combine third to first feature maps sequentially. In each layer, we firstly restore the size of input into 3-D data as it used to be in encoder so that we can pred change information with LWT blocks. After LWT blocks process the corresponding feature map, it is sent into a patch merging module, containing a linear upsampling that doubles the size and a convolution that halves the number of channels. Then it adds with the fusion feature map which has the same size and dimensions. In this way, we can extract multi-scale change information for generation of final change map. Finally, we use a output block based on convolution similarly to get the output which is the remote sensing change map. In the output block, we also use a depth-wise convolution and a residual connection in the middle to enhance the restoration of local features as Fig. 4(b) shows.

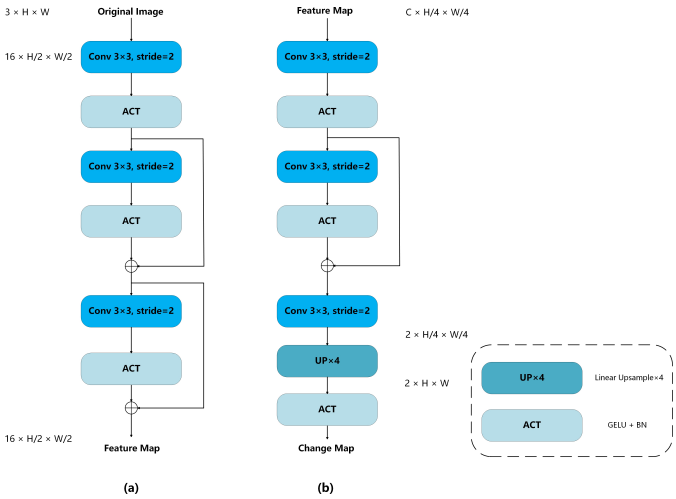

Fig. 4. Structure of stem block and ouput block in LTMFFNet. is the stem block used in encoder of LTMFFNet. (b) is the output block used in decoder of LTMFFNet.

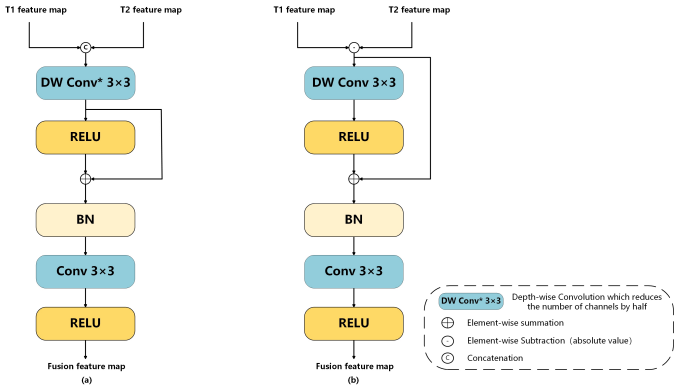

Fig. 5. Structure of two fusion modules in LTMFFNet. (a) is CFM. (b) is CDM. C, H and W signify channel numbers, height and width of each input image.

D. Fusion Module

In our LTMFFNet, we use two kinds of fusion module which are CDM and CFM. Both of them can be seen in Fig. 5. For the outputs of third layer and fourth layer in the siamese encoder, we use CFM to fuse the semantic feature from double branch. It concatenates the outputs of the corresponding layers from two branches and uses a 3×3 convolution with stride 1 to reduce the dimension by half. Then there is a RELU layer for activating with a batch normalization to do. Next there is a 3×3 depth-wise convolution which outputs the same dimention, and use a residual connection to add the output from initial convolution which has the same dimension. Finally we get the fusion output after a RELU activation.

As for the first two layers of the encoder with fine-grained positional information, we use CDM to highlight the differential features. In the CDM, two feature maps corresponding to bitemporal images are performed an absolute value subtraction. Similarly, we put the new matrix into a 3×3 convolution but it doesn't change the channels. The remain operations are the same as CFM and we also add a shortcut between the second convolution and the result of subtracting two images. For both CDM and CFM, the map's resolution keep the same while the dimension after fusion equals to each map input into fusion module.

E. Output and Deep Supervision

To make the final prediction after four layers of decoder in LTMFFNet, we design a output block for more efficient utilization of the change information extracted by decoder. As Fig. 4(b) depicts, it consist of two 3×3 depth-wise convolutions and a 3×3 convolution which changes the dimension to 2 with a upsampling module by interpolation and GELU layer. The first convolution reduces the dimension by half, then the next one changes the numbers of channel to 2. Finally we expand image size four times through linear interpolation. After GELU activation and batch normalization, we obtain the final predicted map with the same size as original dual temporal images.

To better train deep networks and fully utilize features of different scale, we disign an auxiliary branch after third layer of decoder to generate another change map for deep supervision. As shown in Fig. 2, the auxiliary branch is simply consists of a linear interpolation to upsample by a quarter with a convolution operation which changes the channel numbers to 2. In this way, a auxiliary change map is generated which has identical size with the final change map. We use both of them to calculate the loss with the same ground truth and add them up after each training epoch.

The loss funciton we use in LTMFFNet is the hybrid function which is mixture of weighted cross-entropy (WCE) loss [45] and dice loss [46]. It can be defined as:

$$L = \lambda \cdot L_{\mathrm{wce}} + (1 - \lambda) \cdot L_{dice} \tag{11}$$

Where λ is the weight parameter while L_{wce} and L_{dice} denote WCE loss and dice loss respectively, which can be expressed as follows:

$$L_{wce} = \frac{1}{H \times W} \sum_{k=1}^{H \times W} w[c] \cdot \log \frac{\exp(\widehat{y}[k][c])}{\sum_{l=0}^{1} \exp(\widehat{y}[k][l])} \tag{12}$$

$$L_{dice} = 1 - \frac{2 \cdot Y \cdot sm(\widehat{Y})}{Y + sm(\widehat{Y})} \tag{13}$$

where w represents weight, $\widehat{y}$ indicates a point in the generated change map which is the k-th point and c means its class where class 0 represents the unchanged and 1 denotes changed pixels in ground truth map, respectively. l denotes two kinds of classes. Y denotes the ground truth and $\widehat{Y}$ denotes the predicted map output from LTMFFNet where height and width are H and W. $sm(\cdot)$ means softmax operation. L_{wce} uses different weight for each class to resolve category disequilibrium problem while L_{dice} evaluates images for similarity through softmax function. We calculate hybrid loss for both of two change maps and add them up in training process.

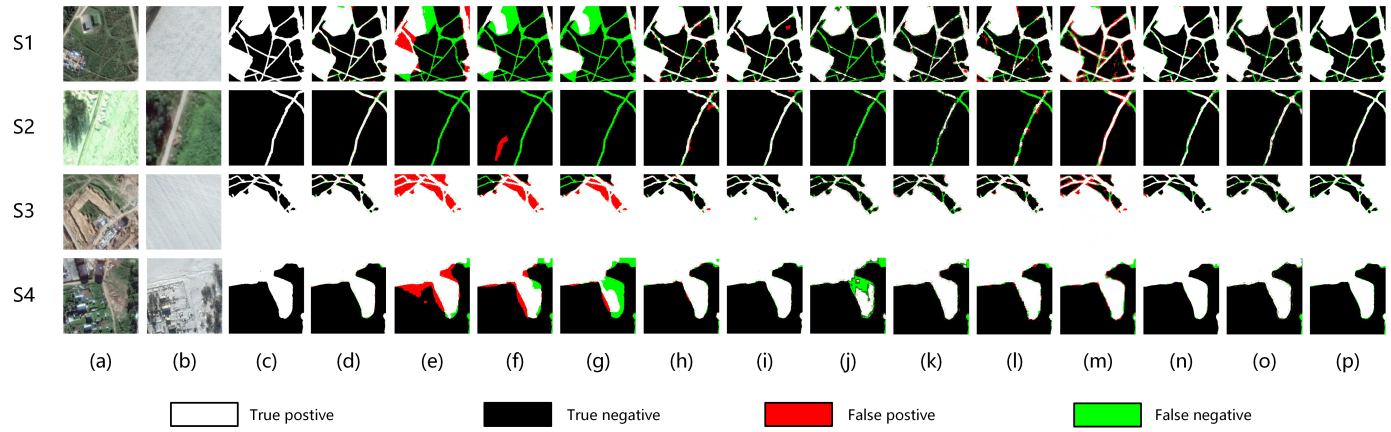

Fig. 6. Experimental results on CDD dataset. S1-S4 are four samples selected from CDD test set. (a) T1 image. (b) T2 image. (c) Ground truth. (d) LTMFFNet (ours). (e) FC-EF. (f) FC-Siam-Conc. (g) FC-Siam-Diff. (h) SNUNet/32. (i) SNUNet/48. (j) BIT. (k) ChangeFormer. (l) SwinSUNet. (m) ChangeViT. (n) FTAN. (o) DMINet. (p) SEIFNet.

IV. Experiment Results

A. Datasets

We choose four public CD datasets in our experiment: CDD dataset, LEVIR-CD dataset, SYSU-CD dataset and GZ-CD dataset.

The CDD dataset [47] is one of the most commonly used evaluation sets in CD. It includes 11 pairs of bitemporal RGB images from different seasons taken by Google Earth. This dataset covers three types of data: composite images without relative object motion, composite images with minor object movement, and real RS images with seasonal variations. These images have a spatial resolution of 3cm/px to 100cm/px. We crop these images to a size of 256×256 so that we get 16 000 pairs of images finally. 10 000 pairs of them are used as training set while 3000 pairs are utilized for validation. The remaining 3000 pairs are regarded as testing set.

The LEVIR-CD dataset [15] is a enormous CD dataset mainly in building area. It consists of 637 couples of RS images taken by Google Earth with a length and width of 1024. LEVIR-CD comprises varied kinds of structures, such as tall buildings, motorhomes, sea view room and warehouses. We crop them into 13,072 patches of size 256 × 256 while 7120 couples of them are training dataset, 1024 couples are validation dataset and 2048 couples are left for testing dataset.

The SYSU-CD dataset [48] contains 20 000 couples of photographs captured from the air with size 256×256. All of the images are captured on two different years in Hong Kong, a international port city located in the southern China. It mainly contains several change types, such as new urban buildings, suburban extensions, pre-construction groundwork and so on. We use 12 000 couples of photoes as training set, 4000 couples for validation and 4000 couples left for evaluation.

The GZ-CD dataset [49] is taken by the Google Earth service of BIGEMAP software whicn covers the suburbs of Guangzhou in China. It collects 19 seasonally changing image pairs with a length and width of 1006×1168 pixels to 4936×5224. We crop them into 256×256 size and use 2504 images to be training set and 313 images for testing and sets validation respectively.

B. Evaluation Metrics

To quantitatively analysis our LTMFFNet, we choose five common indicators: Precision (Pre), Recall (Rec), $F1$-score ($F1$), intersection over union (IoU) and the overall accuracy (OA). Pre is the correct proportion of the generated positive pixels from model, Rec is the percentage of correct predictions from model in all the pixels that are actually positive of the dataset. $F1$ is the arithmetic-geometric mean of them and IoU is the proportion of correct positive pixels to the sum of correct positive pixels and incorrect pixels from the method while OA is the correct proportion of both true and false samples. They are defined as:

$$Pre = \frac{TP}{TP + FP} \tag{14}$$

$$Rec = \frac{TP}{TP + FN} \tag{15}$$

$$F1 = \frac{2 \times \mathrm{Rec} \times \mathrm{Pr}\, e}{\mathrm{Rec} + \mathrm{Pr}\, e} \tag{16}$$

$$IoU = \frac{TP}{TP + FP + FN} \tag{17}$$

$$OA = \frac{TP + TN}{TP + FP + TN + FN} \tag{18}$$

where TP, FP and FN respectively denote the quantity of true positive pixels, false positive pixels, and false negative pixels. Generally, $F1$ can generally reflect performance of a model so that a higner $F1$ shows a better accuracy of model in CD task. And IoU usually reflects the resemblance between the generated map and the real picture. So we consider $F1$ and IoU as main quantitative evaluation metrics in our experiments.

C. Implementation Details

Our LTMFFNet is deployed by the PyTorch framework in our experiment. We train the model in NVIDIA Tesla T4 GPU for 200 epochs. We use AdamW algorithm [50] as

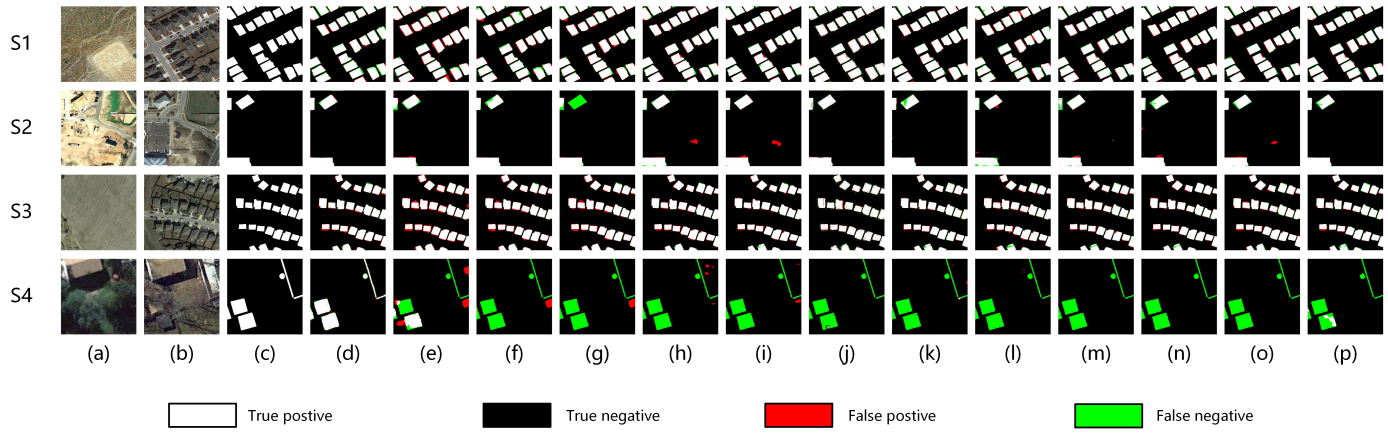

Fig. 7. Experimental results on LEVIR-CD dataset. S1-S4 are four samples selected from LEVIR-CD test set. (a) T1 image. (b) T2 image. (c) Ground truth. (d) LTMFFNet (ours). (e) FC-EF. (f) FC-Siam-Conc. (g) FC-Siam-Diff. (h) SNUNet/32. (i) SNUNet/48. (j) BIT. (k) ChangeFormer. (l) SwinSUNet. (m) ChangeViT. (n) FTAN. (o) DMINet. (p) SEIFNet.

our Optimization algorithm. The weight decay is set to 0.01 and the initial learning rate is 0.0001. We apply Kaiming initialization [51] to weights of each layer and train with batch size of 5 from scratch. For the loss function, we set λ in Eq.(11) to 0.5 in our experiment.

For each epoch in the training process, we calculate the $F1$-score of trained model on validation set. We refer to the $F1$ to choose the best one and use it to evaluate in the test set. The result is used for comparison.

D. Comparison With SOTA Methods

We choose different excellent CD models with deep learning for comparison which cover pure CNN methods, pure transformer methods and methods based on both of CNN and transformer, including FC-EF, FC-Siam-Diff, FC-Siam-Conc [13], SNUNet [14], bitemporal image transformer (BIT) [20], ChangeFormer [40], SwinSUNet [22], ChangeViT [52], Frequency-Temporal Attention Network (FTAN) [16], dual-branch multilevel intertemporal network (DMINet) [53] and Spatiotemporal Enhancement and Interlevel Fusion Network (SEIFNet) [54].

FC-EF is a U-shaped CD simply based on CNN. It concatenates bitemporal images as the inputs firstly and use a a single fully Convolutional Networks(FCN) with residual layers to process it.

FC-Siam-Conc uses a dual-branch encoder to simultaneous processing images from different temporal and concatenates outputs from each branch.

FC-Siam-Diff similarly uses a siamese structure but uses a absolute substraction to fuse features from different branchs.

SNUNet combines UNet++ with siamese network. It uses a densely connected CNN-based network with dual branchs to extract features of bitemporal images. The Ensemble Channel Attention Module(ECAM) proposed by it is used for refining the most useful semantic features from different scale to generate the eventual change map. Since SNUNet has different initial number of channels with different performance, we choose SNUNet with 32 and 48 channels (SNUNet/32 and SNUNet/48) to compare.

BIT uses two branchs of resne to initially capture features of inputs and encode outputs into semantic tokens. Next, it uses a transformer structure to enhance features and predicts through two transformer decoder. The final result is generated by absolute substraction and convolution operation after decoder, which is a late fusion strategy.

ChangeFormer which doesn't use CNN-based encoder, directly extracts bitemporal features through siamese hierarchical transformer and gets difference feature by four differentiation modules based on convolution operation. Finally, it uses a lightweight MLP decoder to aggregate multi-scale features and predict the CD mask.

SwinSUNet is a pure transformer network which uses stacked swin transformer blocks and corresponding downsampling or upsampling modules as encoders and decoders to extract dual temporal image features and fuse them by swin transformer block similarly. It doesn't use convolutional module and shows the high ability of transformer in CD task.

ChangeViT is a framework that improves the performance of large-scale changes by using a common ViT backbone. The network is supplemented by a detail capture module that generates detailed spatial features and a feature injector that efficiently integrates fine-grained spatial information into high-level semantic learning.

FTAN introduces multi-dimensional convolutional frequency attention module and interactive attention module as a dual temporal encoder, and generates the final change map through an MLP decoder. It aggregates different scales of remote sensing image features based on multi-dimensional convolution and handles cross-attention information from different time stages through interactive attention.

DMINet presents a inter-temporal joint-attention(JointAtt) block which unifies SA and cross-attention to achieve interplay between dual-temporal images and better extraction of differential information. It uses siamese resnet as encoder and encodes intralevel bitemporal features in each stage by JointAtt module. The coupled multilevel bitemporal features are processed in different ways base on CNN to gain the change map.

TABLE II
EVALUATION ON CDD DATASET (%).

Method	Pre	Rec	F1	IoU	OA
FC-EF	71.83	66.46	69.04	52.72	92.97
FC-Siam-Conc	79.74	72.76	76.09	61.41	94.61
FC-Siam-Diff	86.03	64.99	74.04	58.79	94.63
SNUNet/32	95.14	94.34	94.76	89.91	98.77
SNUNet/48	96.82	97.00	96.91	93.92	99.27
BIT	97.10	89.26	93.01	86.94	98.42
ChangeFormer	95.50	93.02	94.25	89.11	98.66
SwinSUNet	94.72	93.12	93.91	88.30	98.51
ChangeViT	95.28	94.70	94.99	90.46	98.71
FTAN	95.05	95.06	95.06	90.58	98.83
DMINet	96.88	96.45	96.67	93.55	99.22
SEIFNet	96.65	95.86	96.25	92.78	99.12
LTMFFNet(ours)	**98.36**	**98.34**	**98.35**	**96.75**	**99.47**

TABLE III
EVALUATION ON LEVIR-CD DATASET (%).

Method	Pre	Rec	F1	IoU	OA
FC-EF	83.95	83.74	83.84	72.19	98.36
FC-Siam-Conc	87.48	83.97	85.69	74.96	98.57
FC-Siam-Diff	89.05	81.90	85.32	74.41	98.57
SNUNet/32	90.14	87.38	88.74	79.74	98.87
SNUNet/48	91.25	89.18	90.21	82.14	99.01
BIT	92.09	87.95	89.98	81.78	99.00
ChangeFormer	91.55	88.61	90.06	81.91	99.00
SwinSUNet	90.89	88.78	89.82	80.00	98.82
ChangeViT	92.03	89.90	90.90	83.32	99.08
FTAN	89.77	88.89	89.33	80.72	98.92
DMINet	**92.99**	88.06	90.45	82.50	99.05
SEIFNet	91.48	87.92	89.66	81.26	98.97
LTMFFNet(ours)	92.40	**90.23**	**91.30**	**83.80**	**99.53**

SEIFNet uses a temporal-spatial difference enhancement module based on multi-layer convolution and coordinated attention to highlight the difference information between the same location at different time, and combines it with an element-wise addition and multiplication adaptive context fusion module based on convolution operation to form a progressive decoder, integrating inter-layer features under different semantic guidance to generate the final change map.

We conducted comparative experiments between our model and the selected SOTA methods on four datasets. The experimental results will be presented for quantitative analysis in the following paragraphs.

1) Results on CDD dataset: We display the results of all methods on CDD test set in Table II which shows Pre, Rec and $F1$-score of them. The numerical results reflect that the overall performance of LTMFFNet on CDD dataset is obviously stronger than others for comparison. Particularly, the $F1$-score of LTMFFNet on CDD set is 98.35% that outperforms SNUNet/48 by 1.44% and the IoU of our method is 96.75% that outperforms SNUNet/48 by 2.83%. And our method also performs best in other indicators among these models.

To enhance the performance comparison between different methods, we choose four pairs of images from CDD test set and obtain change maps through the above methods. They are shown in Fig. 6 in visual form with ground truth and we use different colour to mark areas for incorrect detections including error and missed change area. Fig. 6(a) and Fig. 6(b) concludes changes caused by roads and infrastructures, Fig. 6(b) and Fig. 6(c) mainly have changes caused by plants.

We can see our LTMFFNet achieves better result which is more similar to ground truth. LTMFFNet can significantly catch the changes caused by roads which can't be recognized by other methods and make a balance of large regions and small targets according to these comparative images.

2) Results on LEVIR Dataset: Similarly, we show the outcomes of all the models on LEVIR test set in Table III from which we can see LTMFFNet get the best $F1$-score over these comparative models and outperforms DMINet which has best performance in comparative models by 0.85% while our IoU is also best that outperforms DMINet by 1.3%. By the way, our model also has the highest Rec and OA among these methods.

The visualization are displayed in Fig. 6. We also choose four groups of bitemporal images and labels from LEVIR-CD dataset, and generate the change map with different methods. By analyzing these images, we can see that there is a problem of imbalanced samples on this dataset, while other models lack the ability to detect small change regions on this dataset. Comparing with these methods, we conclude that our LTMFFNet has the ability to catch both small and large change regions with excellent edge. These changes are mainly the result of buildings some of which can't be recognized by other methods while our model can catch them.

3) Results on SYSU Dataset: The results of our experiment on SYSU test set are present in Table IV. In such a big set, our LTMFFNet get best performance compared with other methods as the table shows. Fig. 7 displays the results of comparative experiments on SYSU dataset and it obviously shows that our LTMFFNet can better capture changes in large areas with more accurate edges than others mainly by comparing $F1$ and IoU.

4) Results on GZ-CD Dataset: GZ-CD is the smallest dataset in four datasets and it is usually hard to obtain satisfactory results on its test set. As shown in Table V, our model has best $F1$-score and IoU among all comparison methods. The visualization results on GZ-CD test set are put in fig. 8. It's challenging to catch some boundary and fine-grained features through training in limited set and our method has smaller deviation than others by comparing these results.

5) Analysis of parameters and computation: We compare efficiency of different models in Table VI by calculating the parameter quantity (Params) and floating point operations per second (FLOPs) which reflects computational speed. By analyzing these data, we can conclude that LTMFFNet has fewer Params than SwinSUNet and ChangeFormer which are mainly based on transformer. And the Params of our method is also less than SNUNet/48 which is a CNN-based method.

As for computation, our method has small FLOPs which is only larger than FC-EF and FC-Siam-Diff. The computational complexity of LTMFFNet is lower than all transformer meth-

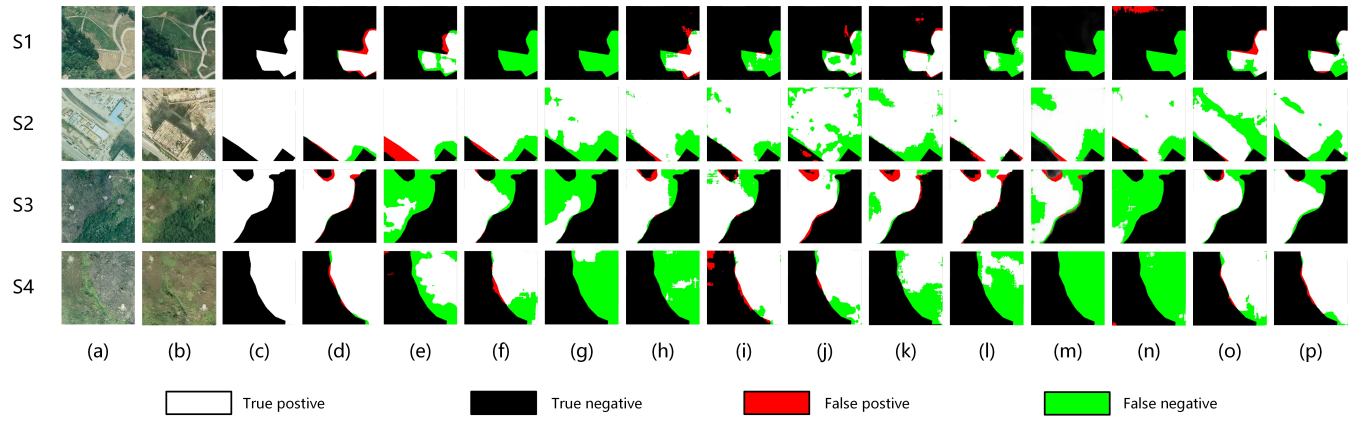

Fig. 8. Experimental results on SYSU-CD dataset. S1-S4 are four samples selected from SYSU-CD test set. (a) T1 image. (b) T2 image. (c) Ground truth. (d) LTMFFNet (ours). (e) FC-EF. (f) FC-Siam-Conc. (g) FC-Siam-Diff. (h) SNUNet/32. (i) SNUNet/48. (j) BIT. (k) ChangeFormer. (l) SwinSUNet. (m) ChangeViT. (n) FTAN. (o) DMINet. (p) SEIFNet.

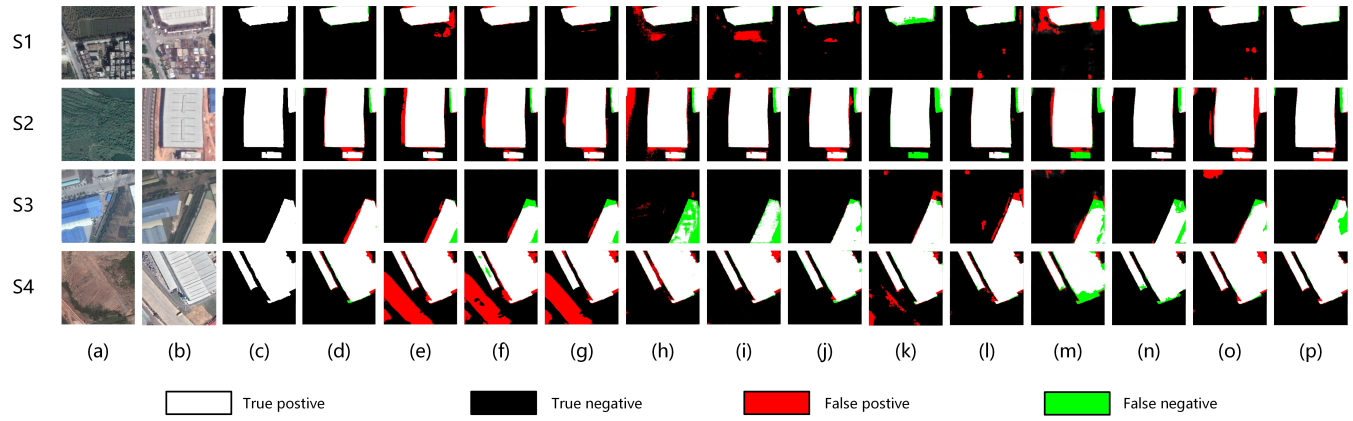

Fig. 9. Experimental results on GZ-CD dataset. S1-S4 are four samples selected from GZ-CD test set. (a) T1 image. (b) T2 image. (c) Ground truth. (d) LTMFFNet (ours). (e) FC-EF. (f) FC-Siam-Conc. (g) FC-Siam-Diff. (h) SNUNet/32. (i) SNUNet/48. (j) BIT. (k) ChangeFormer. (l) SwinSUNet. (m) ChangeViT. (n) FTAN. (o) DMINet. (p) SEIFNet.

TABLE IV
EVALUATION ON SYSU-CD DATASET (%).

Method	Pre	Rec	F1	IoU	OA
FC-EF	76.50	77.96	77.22	62.90	89.16
FC-Siam-Conc	79.28	78.13	78.70	64.78	89.88
FC-Siam-Diff	88.42	62.52	73.25	57.80	89.23
SNUNet/32	81.53	71.63	76.26	62.15	89.54
SNUNet/48	81.52	75.34	78.31	64.35	90.52
BIT	76.84	73.42	75.09	60.12	88.51
ChangeFormer	80.69	75.19	77.84	63.72	89.91
SwinSUNet	80.61	74.83	77.61	63.56	89.56
ChangeViT	80.93	**81.45**	81.19	68.34	91.10
FTAN	81.66	72.21	76.64	62.13	89.62
DMINet	87.04	76.53	81.45	68.29	91.39
SEIFNet	85.11	77.10	80.91	67.94	91.42
LTMFFNet(ours)	**89.28**	80.56	**84.70**	**73.78**	**91.43**

TABLE V
EVALUATION ON GZ-CD DATASET (%).

Method	Pre	Rec	F1	IoU	OA
FC-EF	82.50	75.69	78.93	65.20	95.84
FC-Siam-Conc	77.29	75.71	76.49	61.94	95.20
FC-Siam-Diff	76.91	70.86	73.76	58.11	94.83
SNUNet/32	86.83	82.99	84.86	73.72	96.95
SNUNet/48	88.07	86.13	87.09	77.13	97.37
BIT	86.18	74.76	80.06	73.20	97.02
ChangeFormer	86.18	74.76	80.06	66.75	96.16
SwinSUNet	88.70	85.60	87.12	76.11	97.48
ChangeViT	88.70	85.80	87.22	77.34	97.41
FTAN	**91.10**	81.45	86.00	75.44	97.27
DMINet	89.54	86.56	88.02	78.61	97.57
SEIFNet	89.07	84.85	86.91	76.85	97.37
LTMFFNet(ours)	88.25	**92.22**	**90.19**	**84.22**	**97.74**

ods and several methods based on CNN in our experiments. It proves the efficiency of LWT block in LTMFFNet for reducing model complexity compared with pure transformer methods and some complex CNN-based methods while improve performance in CD task. By using efficient convolution operations and simplifying the calculation of self attention, our method effectively reduces redundant calculation of transformer for CD.

(6) *Analysis of training process*: We draw the line chart generated during training process of LTMFFNet in Fig. 9. We

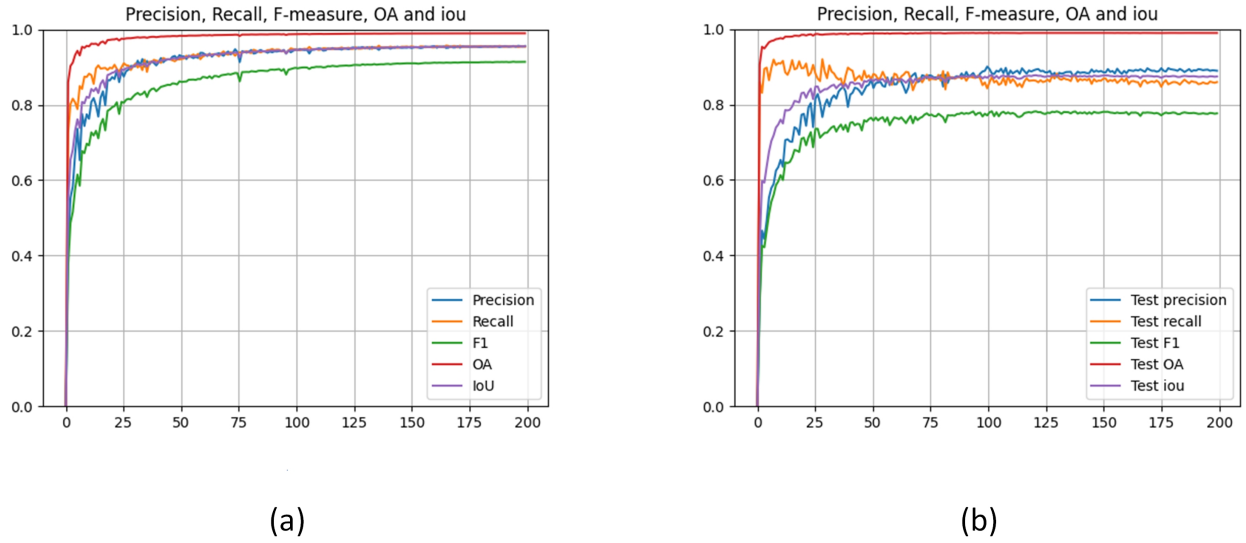

Fig. 10. Line chart of different indicators on validation set. (a) is the process on CDD dataset. (b) is the process on LEVIR-CD dataset

TABLE VI
THE PARAMS (M) AND FLOPS (G) OF EACH MODEL.

Method	Params	Flops
FC-EF	1.35	3.58
FC-Siam-Diff	1.35	4.73
FC-Siam-Conc	1.55	5.33
BIT	3.50	10.63
DMINet	6.24	14.55
SNUNet/32	12.03	54.83
SNUNet/48	27.07	123.14
SEIFNet	27.91	8.37
ChangeViT	32.14	38.81
ChangeFormer	41.03	202.79
FTAN	42.27	211.06
SwinSUNet	42.94	15.71
LTMFFNet(ours)	17.64	5.24

choose CDD and LEVIR-CD dataset as example and show the growth process of indicators on their validation set. As we can see from the curves, our model converges in a set number of epochs and tends to be relative stable. After too many epochs, there may be undulation but no significant changes. So we choose 200 epochs in our experiments and choose the best result on test set.

V. DISCUSSION

In general, there are numbers of factors can affect results of the model. To confirm the contributions of different parts in our method, we design several ablation experiments on benchmark dataset and analyze their effectiveness. We will also discuss the limitations of the model and the directions for future development of CD in this section.

A. Effects of Different Modules

1) Effect of Fusion Module: We use two kinds of fusion module separately in shallow and deep layers of LTMFFNet to combine deep syntactic feature and superficial positional information separately. To verify the effectiveness of two fusion module which are CFM and CDM, we do experiments about fusion module on CDD dataset and analysis results as listed in Table VII shows. We do the experiments by removing CFM or CDM in corresponding layer individually and replace with simple concatenation as second line and third line of Table VII. And we also remove both of them and similarly do experiments in CDD dataset. The results show that both of CFM and CDM has its effectiveness in our model, and each of the fusion module can improve overall performance of LTMFFNet.

2) Effect of Deep Supervision: In our LTMFFNet, we use a auxiliary branch in the third layer for deep supervision. We calculate loss of it and adds with the loss of final output so that we can fully utilize information at different scales from LWT blocks. To verify the effect of deep supervision, we do ablation experiment on CDD dataset by training without the auxiliary branch. The result can be seen in the first line of Table VIII and it shows that the effectiveness of auxiliary branch in the training process of LTMFFNet where the $F1$-score of LTMFFNet with deep supervision is 1.52% higner than the model without auxiliary branch. Thus the deep supervision is important for the training of LTMFFNet.

3) Effect of Stem and Output Block: In LTMFFNet, we use a stem block to preliminarily process the input image in each branch of encoder and use a output block in the end of decoder for gaining the result. In order to prove its effectiveness, we replace stem block with a 1×1 convolution which decreases the dimensions of input image while reduce its size by half and substitute output block for a simple 1×1 convolution and linear upsample to restore its original size. We run the

TABLE VII
EFFECT OF FUSION MODULES IN LTMFFNET ON CDD DATASET.

Method	Pre	Rec	F1	IoU	OA
LTMFFNet (without CDM and CFM)	97.40	98.08	97.74	95.58	99.27
LTMFFNet (without CDM)	97.85	98.09	97.96	96.02	99.35
LTMFFNet (without CFM)	97.84	98.05	97.94	95.98	99.34
LTMFFNet	**97.88**	**98.52**	**98.20**	**96.47**	**99.42**

TABLE VIII
EFFECT OF AUXILIARY BRANCH, STEM BLOCK AND OUTPUT BLOCK IN LTMFFNET ON CDD DATASET.

Method	Pre	Rec	F1	IoU	OA
LTMFFNet (without auxiliary branch)	96.24	97.11	96.68	93.71	99.07
LTMFFNet (without stem block and uutput block)	**98.82**	95.18	96.95	93.94	99.18
LTMFFNet	97.84	**98.05**	**97.94**	**95.98**	**99.34**

TABLE IX
EFFECT OF MODEL DEPTH.

Depth	Pre	Rec	F1	IoU	OA
(2, 2, 2, 2)	97.77	98.39	98.08	96.23	99.38
(2, 2, 10, 2)	98.36	98.34	98.35	96.75	99.47
(2, 2, 12, 2)	98.27	98.54	98.40	96.85	99.49

experiments on CDD set similarly and consequential data is displayed as the second row of Table VIII. Comparative data clearly indicate that the stem and output block is useful for the performance of LTMFFNet.

B. Effect of Model Depth

In general, the model depth will affect the performance. We use three configurations of encoder depths in our experiment which are (2, 2, 2, 2), (2, 2, 10, 2) and (2, 2, 12, 2). The values are the numbers of LWT blocks in each layer of encoder and the values in decoder is the same in opposite order which are (2, 2, 2, 2), (2, 10, 10, 2) and (2, 12, 12, 2). We list results at different depths in Tabel 9 and it obviously shows the performance of model with various numbers of blocks. Obviously, the performance of model gradually enhances as the quantity of LWT blocks grows. What's more, the growth rate of the performance becomes less and less as the depth increases. When the depth exceeds a certain number, the performance increase is limited while the computation become larger. Considering about both performance and computational complexity, we choose (2, 2, 10, 2) as our model depth in our paper.

C. Limitations and Future Work

There are still limitations for the performance of our model: The situation where small targets are missed and the edges of the changed areas are inaccurate still exists. What's more, how to further reduce the params of model which based on transformer combined with CNN is a issue to be studied. Future work need to focus more on efficient integration of CNN and transformer to achieve model performance improvements with less computation and parameters.

VI. CONCLUSION

With the aim of enhancing the local sensing ability of transformer in CD to better capture change areas of different scales and get better change region edges while reducing computational complexity, we propose a siamese U-shape network LTMFFNet based on modules which combines the advantages of CNN and transformer for CD. It uses LWT block which integrates CNN structure with MHSA as the basic module in our network. Compared to traditional transformer-based methods which mainly focus on global information, LWT block enhances local perception ability for input images while reduces computational complexity. Therefore, LTMFFNet can effectively balance both global and local features from RS images and capture more precise change information than pure transformer methods with less computation as a result. We conduct experiments on four public CD datasets and compare with other methods. The experimental result shows that LTMFFNet achieve better comprehensive performance relative to other SOTA methodologies which cover CNN-based model and transformer-based model while our model has small computational complexity especially compared with these transformer methods.

REFERENCES

[1] O. Abd El-Kawy, J. Rød, H. Ismail, and A. Suliman, "Land use and land cover change detection in the western nile delta of egypt using remote sensing data," *Applied Geography*, vol. 31, pp. 483–494, 2011.
[2] A. Alqurashi and L. Kumar, "Investigating the use of remote sensing and gis techniques to detect land use and land cover change: A review," *Advances in Remote Sensing*, 2013.
[3] S. Kumar, M. Anouncia, S. Johnson, A. Agarwal, and P. Dwivedi, "Agriculture change detection model using remote sensing images and gis: Study area vellore," in *2012 International Conference on Radar, Communication and Computing (ICRCC)*. IEEE, 2012, pp. 54–57.
[4] V. Agone and S. Bhamare, "Change detection of vegetation cover using remote sensing and gis," *Journal of research and development*, vol. 2, no. 4, 2012.
[5] R. K. Gupta, "Change detection techniques for monitoring spatial urban growth of jaipur city," *Inst Town Planners India J*, vol. 8, no. 3, pp. 88–104, 2011.
[6] I. R. Hegazy and M. R. Kaloop, "Monitoring urban growth and land use change detection with gis and remote sensing techniques in daqahlia governorate egypt," *International Journal of Sustainable Built Environment*, vol. 4, no. 1, pp. 117–124, 2015.
[7] Z. Zheng, Y. Zhong, J. Wang, A. Ma, and L. Zhang, "Building damage assessment for rapid disaster response with a deep object-based semantic change detection framework: From natural disasters to man-made disasters," *Remote Sensing of Environment*, vol. 265, p. 112636, 2021.
[8] Y. Ma, F. Chen, J. Liu, Y. He, J. Duan, and X. Li, "An automatic procedure for early disaster change mapping based on optical remote sensing," *Remote Sensing*, vol. 8, no. 4, p. 272, 2016.
[9] L. Khelifi and M. Mignotte, "Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis," *Ieee Access*, vol. 8, pp. 126 385–126 400, 2020.
[10] W. Shi, M. Zhang, R. Zhang, S. Chen, and Z. Zhan, "Change detection based on artificial intelligence: State-of-the-art and challenges," *Remote Sensing*, vol. 12, no. 10, p. 1688, 2020.
[11] S. Liu, D. Marinelli, L. Bruzzone, and F. Bovolo, "A review of change detection in multitemporal hyperspectral images: Current techniques, applications, and challenges," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 140–158, 2019.
[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 4063–4067.

[14] S. Fang, K. Li, J. Shao, and Z. Li, "Snunet-cd: A densely connected siamese network for change detection of vhr images," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[15] H. Chen and Z. Shi, "A spatial-temporal attention-based method and a new dataset for remote sensing image change detection," *Remote Sensing*, vol. 12, no. 10, p. 1662, 2020.

[16] C. Yu, H. Li, Y. Hu, Q. Zhang, M. Song, and Y. Wang, "Frequency-temporal attention network for remote sensing imagery change detection," *IEEE Geoscience and Remote Sensing Letters*, 2024.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[19] T. Yan, Z. Wan, and P. Zhang, "Fully transformer network for change detection of remote sensing images," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 1691–1708.

[20] H. Chen, Z. Qi, and Z. Shi, "Remote sensing image change detection with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[21] W. G. C. Bandara and V. M. Patel, "A transformer-based siamese network for change detection," in *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2022, pp. 207–210.

[22] C. Zhang, L. Wang, S. Cheng, and Y. Li, "Swinsunet: Pure transformer network for remote sensing image change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[23] Q. Li, R. Zhong, X. Du, and Y. Du, "Transunetcd: A hybrid transformer network for change detection in optical remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–19, 2022.

[24] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "Cmt: Convolutional neural networks meet vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 175–12 185.

[25] Y. Guo, Y. Li, L. Wang, and T. Rosing, "Depthwise convolution is all you need for learning multiple visual domains," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8368–8375.

[26] S. Yun and Y. Ro, "Dynamic mobile-former: Strengthening dynamic convolution with attention and residual connection in kernel space," *arXiv preprint arXiv:2304.07254*, 2023.

[27] J. Li, X. Xia, W. Li, H. Li, X. Wang, X. Xiao, R. Wang, M. Zheng, and X. Pan, "Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios," *arXiv preprint arXiv:2207.05501*, 2022.

[28] X. Lu, M. Suganuma, and T. Okatani, "Sbcformer: Lightweight network capable of full-size imagenet classification at 1 fps on single board computers," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 1123–1133.

[29] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[32] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 539–546.

[33] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change detection based on deep siamese convolutional network for optical aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1845–1849, 2017.

[34] A. Varghese, J. Gubbi, A. Ramaswamy, and P. Balamuralidhar, "Changenet: A deep learning architecture for visual change detection," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[35] J. Chen, Z. Yuan, J. Peng, L. Chen, H. Huang, J. Zhu, Y. Liu, and H. Li, "Dasnet: Dual attentive fully convolutional siamese networks for change detection in high-resolution satellite images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 1194–1206, 2020.

[36] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*. Springer, 2018, pp. 3–11.

[37] Y. Xing, J. Jiang, J. Xiang, E. Yan, Y. Song, and D. Mo, "Lightcdnet: Lightweight change detection network based on vhr images," *IEEE Geoscience and Remote Sensing Letters*, 2023.

[38] M. A. Islam, S. Jia, and N. D. Bruce, "How much position information do convolutional neural networks encode?" *arXiv preprint arXiv:2001.08248*, 2020.

[39] Y. Liu, 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 Geoscience and Remote Sensing Letters*, vol. 18, no. 5, pp. 811–815, 2020.

[40] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3146–3154.

[41] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[42] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.

[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[44] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[45] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[46] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. Ieee, 2016, pp. 565–571.

[47] M. Lebedev, Y. V. Vizilter, O. Vygolov, V. A. Knyaz, and A. Y. Rubis, "Change detection in remote sensing images using conditional adversarial networks," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 565–571, 2018.

[48] M. Liu and Q. Shi, "Dsamnet: A deeply supervised attention metric based network for change detection of high-resolution images," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 2021, pp. 6159–6162.

[49] D. Peng, L. Bruzzone, Y. Zhang, H. Guan, H. Ding, and X. Huang, "Semicdnet: A semisupervised convolutional neural network for change detection in high resolution remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5891–5906, 2020.

[50] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[52] D. Zhu, X. Huang, H. Huang, Z. Shao, and Q. Cheng, "Changevit: Unleashing plain vision transformers for change detection," *arXiv preprint arXiv:2406.12847*, 2024.

[53] Y. Feng, J. Jiang, H. Xu, and J. Zheng, "Change detection on remote sensing images using dual-branch multilevel intertemporal network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–15, 2023.

[54] Y. Huang, X. Li, Z. Du, and H. Shen, "Spatiotemporal enhancement and interlevel fusion network for remote sensing images change detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.