Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
17/1 (2024), 37-48. DOI: http://dx.doi.org/10.21609/jiki.v17i1.1188

Improving Remote Sensing Change Detection Via Locality Induction on Feed-forward Vision Transformer

Lhuqita Fazry[1], Mgs M Luthfi Ramadhan[1], Wisnu Jatmiko[1]

[1] Faculty of Computer Science, University of Indonesia, Depok 16424, Indonesia

Email:lhuqita.fazry@ui.ac.id

Abstract

The main objective of Change Detection (CD) is to gather change information from bi-temporal remote sensing images. The recent development of the CD method uses the recently proposed Vision Transformer (ViT) backbone. Despite ViT being superior to Convolutional Neural Networks (CNN) at modeling long-range dependencies, ViT lacks a locality mechanism, a critical property of pixels that comprise natural images, including remote sensing images. This issue leads to segmentation artifacts such as imperfect changed region boundaries on the predicted change map. To address this problem, we propose LocalCD, a novel CD method that imposes the locality mechanism into the Transformer encoder. It replaces the Transformer's feed-forward network using an efficient depth-wise convolution between two 1×1 convolutions. LocalCD outperforms ChangeFormer by a significant margin. Specifically, it achieves an F1-score of 0.9548 and 0.9243 on CDD and LEVIR-CD datasets.

Keywords: *Change Detection, Vision Transformer, Pyramidal Vision Transformer, Local Vision Transformer, CDD, LEVIR-CD*

1. Introduction

Change Detection (CD) is one of the most important and challenging tasks in remote sensing image observation. It is a process of gathering change information from images on the same geographical area taken at different times [1–3]. In practice, CD compares two pre-registered images consisting of pre-change and post-change images. Both images represent the same spatial area while having different temporal information. The pre-change image represents an image before the change event, while the post-change image represents an image after the change event. CD has many practical applications, such as urban development analysis, disaster assessment, agricultural investigation, environment monitoring, and deforestation [4]. These applications are essential for a decision-maker organization. To make a proper decision, the change information should be accurate and can be acquired fast. However, manual observation on this challenging task is time-consuming [3]. Therefore, an automated method is needed to make the process faster and more accurate.

There are two common approaches of the CD method, pixel-based and object-based[2]. The pixel-based is the simplest one. It works by comparing each pixel from different images at the same location. On the other hand, in an object-based approach, all objects in the area are identified first and then compared to the same object from different images. The output of the CD is a binary change map (CM) that flags the changed region with a white pixel and the unchanged region with a black pixel.

Transformer architecture [5] is a recent Natural Language Processing (NLP) advance that can model long-range dependencies between word tokens. Transformer consists of two main components: self-attention (SA) and two consecutive feed-forward networks (FFNs). SA is responsible for refining the feature representation, while FFNs enrich it by expanding the feature dimension. The success of Transformer inspires the vision community to adopt the SA mechanism to enhance the image classification model. Vision Transformer (ViT) [6] is the first vision model that uses self-attention to learn long-range interaction between visual tokens. ViT uses the

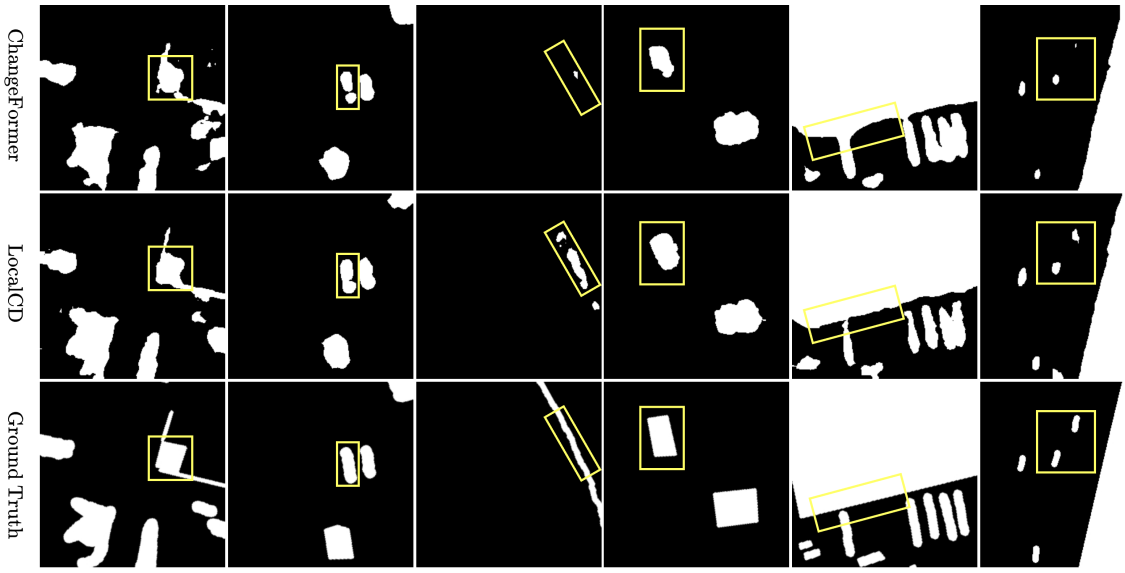


Figure 1. Examples of CD artifact on the predicted change map of ChangeFormer. The yellow boxes show the imperfect changed region boundaries.

same architecture as the original Transformer. However, instead of using a word as the token, it uses a 16×16 image patch. Despite ViT outperforming state-of-the-art Convolution Neural Network (CNN) [7] on ImageNet [8] classification, it can not be used directly to perform a downstream vision task like image segmentation and detection. One reason is that those tasks require fine-grained level information for smaller patch sizes. Smaller patch size leads to an increase in the token number. However, the SA suffers from quadratic computation concerning the number of tokens, which hinders it from utilizing many tokens.

Pyramid Vision Transformer (PVT) [9] is the first ViT variant that utilizes a down-sampling query and key matrices to reduce the computation complexity of SA. It also leverages a hierarchical structure to enhance the representation power of the model, following the success of ResNet [7]. Swin Transformer [10] is another popular approach for reducing the complexity of SA. Different from PVT, it used interactions within a constrained window. Relying on PVT as the backbone, SegFormer [11] proposed a lightweight Multi-layer Perceptron (MLP) decoder to improve the segmentation capability of the model, making it the first Transformer-based vision model specialized for segmentation tasks.

The advance of the Vision Transformer is quickly adapted to remote sensing, especially on change detection tasks. BIT [12] and ChangeFormer [13] are pioneers in Transformer-based CD methods. BIT used the Transformer encoder to enhance the feature maps produced by the CNN backbone. In contrast, ChangeFormer uses a similar architecture to SegFormer, constructed in a Siamese network. ChangeFormer had a slightly better change map compared to BIT.

Despite its superiority, the change map produced by ChangeFormer still contains artifacts. We investigated extensively and found that ChangeFormer creates imperfect boundaries in the changed region. Fig. 1 illustrates the artifacts on the change map produced by ChangeFormer. We relate this phenomenon with the lack of locality induction on the ChangeFormer, a property that is inherited from the used Transformer backbone. Natural images are grid-like data where each pixel correlates to the neighbor pixels. Relying upon this fact and inspiration from [14], we proposed LocalCD, a novel CD method to solve the boundaries problem in ChangeFormer. Specifically, we impose the locality mechanism into the FFNs inside the ChangeFormer. Our proposed method produces a change map having better boundaries of the changed region.

Finally, the following are the contributions of our work:

1) We modify the ChangeFormer by adding locality induction to the FFNs inside it. This modification is proven effective in solving the boundaries problem on the changed region of the produced change map.
2) We use Lp distance to differentiate between features from pre-image and post-image instead of simple minus operations. We incor-

porate this Lp distance in the fusion feature module. This modification is proven to be effective in increasing the model performance.

3) We use a lightweight conv decoder instead of the Multi-layer Perceptron (MLP) decoder used in ChangeFormer. This replacement is to support the locality induction in the encoder further.

2. Related Work

2.1. CNN-based CD Method

CNN is one of the most used architectures for solving CD tasks compared to other neural architectures. One of the reasons is CNN efficient and specifically designed for processing image data that matches the CD's input data. Furthermore, CNN support for transfer learning, a feature that can significantly speed up the model's convergencies. This characteristic of CNN makes it a popular vision backbone adopted in many scientific fields, including remote sensing. Many works leverage CNN to build CD methods such as Fully Connected Early Fusion (FC-EF) [15], Fully Connected Siamese Diff (FC-Siam-Diff)[15], STANet [16], DASNet [17], and SNUNet [18].

FC-EF [15] used a stack of convolution layers to extract essential features from two input images. It incorporates early fusion by concatenating pre-image and post-image along the channel dimensions. The extracted features are then processed by the pixels classifier to build the change map. In contrast, FC-Siam-Diff [15] use a different approach. Instead of using an early fusion mechanism, FC-Siam-Diff used a Siamese network containing two similar sub-networks. Each sub-network is responsible for extracting features from a single image. The final features are obtained by subtracting the feature of the pre-image from the feature of the post-image. Like FC-Siam-Diff, STANet [16] also used a Siamese network with a shared weight to extract features from pre-image and post-image. However, STANet used a Residual Network (ResNet) [7] as the backbone. Furthermore, STANet proposed sophisticated fusion modules leveraging spatial and temporal attention called Basic spatiotemporal attention module (BAM) and Pyramid spatiotemporal attention module (PAM). Both modules are responsible for enhancing the feature representation produced by the backbone in two different ways.

Similar to STANet, DASNet [17] also used a Siamese network and spatial-temporal attention modules. However, DASNet used a different approach: Spatial Attention Mechanism (SAM) and Channel Attention Mechanism (CAM). Furthermore, DASNet proposed an auxiliary loss function called Weighted Double Margin Contrastive (WDMC) loss to address the imbalance problem in the CD datasets. Unlike STANet and DASNet, SNUNet [18] used a densely connected siamese network, leveraging much skip connection. Also, SNUNet proposed an enhanced version of the attention module called the Ensemble Channel Attention Module (ECAM).

On the other hand, our proposed method uses a Transformer-based vision backbone instead of a CNN-based one. Specifically, we use a hierarchical vision Transformer, a versatile vision backbone for down-stream tasks like image segmentation and detection. We chose Transformer because it recently outperformed *state-of-the-art* CNN on the ImageNet [8] classification task.

2.2. Transformer-based CD Method

Recently, Transformers [5], a *state-of-the-art* model in Natural Language Processing (NLP), has gained increasing interest among Computer Vision (CV) researchers. The scalability of Transformers enables it to outperform CNN backbone [7] on image classification tasks. Therefore, Vision Transformer (ViT) [6] becomes a new standard model in CV. There are some previous works on CD methods that use ViT as the vision backbone, including Bitemporal Image Transformer (BIT) [12] and ChangeFormer [13].

BIT used a hybrid approach, combining CNN and ViT for solving CD tasks. Similar to the previous CNN-based CN method, BIT also uses a siamese network with a ResNet backbone for extracting salient features from bitemporal input images. The resulting feature maps are then processed by the Transformer encoder. However, the feature maps need to be converted into tokens in order for Transformer can process them. The resulting refined features are then processed by the prediction head to build the change map. Unlike BIT, ChangeFormer [13] use Pyramid Vision Transformers (PVT) [9], a specialized ViT variant for solving down-stream vision tasks. Furthermore, it used an MLP decoder to generate the change map.

Our proposed method uses a similar architecture as the ChangeFormer. However, we add a locality mechanism to replace the two consecutive MLP components after the attention calculation. This replacement is to improve the changed region boundaries on the change map. This modification relies on the fact that individual image pixels are affected

by the surrounding pixels. Furthermore, we also use Lp distance feature fusion instead of a simple subtraction operation like in ChangeFormer. Also, we replace the MLP decoder using Lightweight Convolution Decoder to support the locality induction on the encoder.

3. Proposed Method

In this paper, we propose a novel CD method called LocalCD. This section presents detailed information about the method.

3.1. Overall Architecture

LocalCD receives two input images $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{H\times W\times 3}$, which represent pre-image at time t_1 and post-image at time t_2. It then returns the change map $\mathbf{M} \in \mathbb{R}^{H\times W}$ where H and W denote the height and weight of the images. The architecture of LocalCD comprises three main modules: encoder, feature fusion, and decoder. The encoder extracts the important feature from both images in 4 different stages. Each stage extracts features at a different scale. The spatial dimension goes decreases as the stage goes deeper. In contrast, the feature dimension increases following the stage level. The encoder produces two feature maps: $\mathbf{F}_1$ and $\mathbf{F}_2$, each for input image $\mathbf{x}_1$ and $\mathbf{x}_2$. Each feature map is composed of 4 sub-feature maps representing different feature scales $\mathbf{F}_1^1$, $\mathbf{F}_1^2$, $\mathbf{F}_1^3$, and $\mathbf{F}_1^4$ for $\mathbf{F}_1$. On the other hand, $\mathbf{F}_2$ contain $\mathbf{F}_2^1$, $\mathbf{F}_2^2$, $\mathbf{F}_2^3$, and $\mathbf{F}_2^4$. Both feature maps are then fused in the feature fusion module. Finally, the fused feature maps are then decoded into the change map. Fig. 2 illustrates the overall architecture of LocalCD.

3.2. Pyramidal Transformer Encoder (PTE)

The main goal of the encoder is to extract features given two input images. To achieve this, we leverage a Siamese structure with shared weights. The PTE will process each stream in the Siamese network. This PTE incorporates the pyramidal structure within 4 stage levels, following the prior art [9]. At each stage, PTE consists of the Transformer block and patch merging. This structure has been proven to be very effective for visual backbone [7].

Consider a single input image of size $H\times W\times 3$. At the first stage, the image is partitioned into $\frac{HW}{4^2}$ patches, each having size $4\times 4\times 3$. These patches are then flattened into $\frac{HW}{4^2}\times 3$ embedding sequence. A linear projection is then applied into the feature dimension of the sequence to make feature embedding of size $\frac{HW}{4^2}\times C_1$. This embedding is then processed by attention block, and the output is then reshaped into a feature map F^1 of size $\frac{H}{4}\times\frac{H}{4}\times C_1$. This process continues for all stages, resulting in feature maps of size $\frac{H}{8}\times\frac{H}{8}\times C_2$, $\frac{H}{16}\times\frac{H}{16}\times C_3$ and $\frac{H}{32}\times\frac{H}{32}\times C_4$ for stage 2, 3, and 4 respectively.

In general, at stage-i, the feature map $\mathbf{F}^{i-1} \in \mathbb{R}^{H_{i-1}\times W_{i-1}\times C_{i-1}}$ are partitioned into $\frac{H_{i-1}W_{i-1}}{P_i^2}$ patches using patch size P_i where $P_i = 2\times P_{i-1}$. The feature dimension of each patch is then flattened and projected into an arbitrary C_i dimensional embedding; this process is called **Patch Embedding**. The embedding is then processed by the Transformer block resulting in a new feature $\mathbf{F}^i$.

Overlap Patch Embedding. Different from text, an image is grid-shaped data. Unfortunately, Transformer architecture is explicitly designed to process data sequences. Therefore, the image needs to be converted into sequences before processing. ViT [6], and PVT [9] incorporate non-overlap patch embedding for simplicity. In contrast, we use overlap patch embedding to the image into a sequence of image patches. Technically, we apply a convolution using a kernel size similar to the patch size. We use a stride size smaller than the kernel size to achieve the overlap effect. Furthermore, we project the channel dimension from 3 into an arbitrary dimension C_1 through the same convolution operation. The resulting patches are then flattened to create a token sequence.

Transformer Block. Each stage of the encoder comprises L_i layers of Transformer blocks. The Transformer block is the main component of the encoder. It consists of an attention mechanism and two consecutive feed-forward networks. This block receives feature maps as visual token sequences and then refines its representation according to token interaction. To avoid quadratic computation complexity, we apply Spatial Reduction Attention (SRA) [9], replacing the Multi-head Self Attention (MSA) in the original ViT. This type of attention effectively lowers the computation complexity by reducing the spatial dimension of the matrix query and value before the attention takes place.

Suppose the sequence of tokens $\mathbf{X} = \{X_j \in \mathbb{R}^d | j = 1, \cdots, N\}$ at stage-i where $d = C_i \times P_i^2$ is the embedded dimension and $N = \frac{H_i W_i}{P_i^2}$ is the number of tokens. It is clear that $\mathbf{X} \in \mathbb{R}^{N\times d}$. The attention score is then calculated as follows:

$$\mathbf{Z} = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}. \quad (1)$$

where $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$, $\mathbf{K} = \hat{\mathbf{X}}\mathbf{W}_K$ and $\mathbf{V} = \hat{\mathbf{X}}\mathbf{W}_V$ represent matrix query, key and value respectivelly.

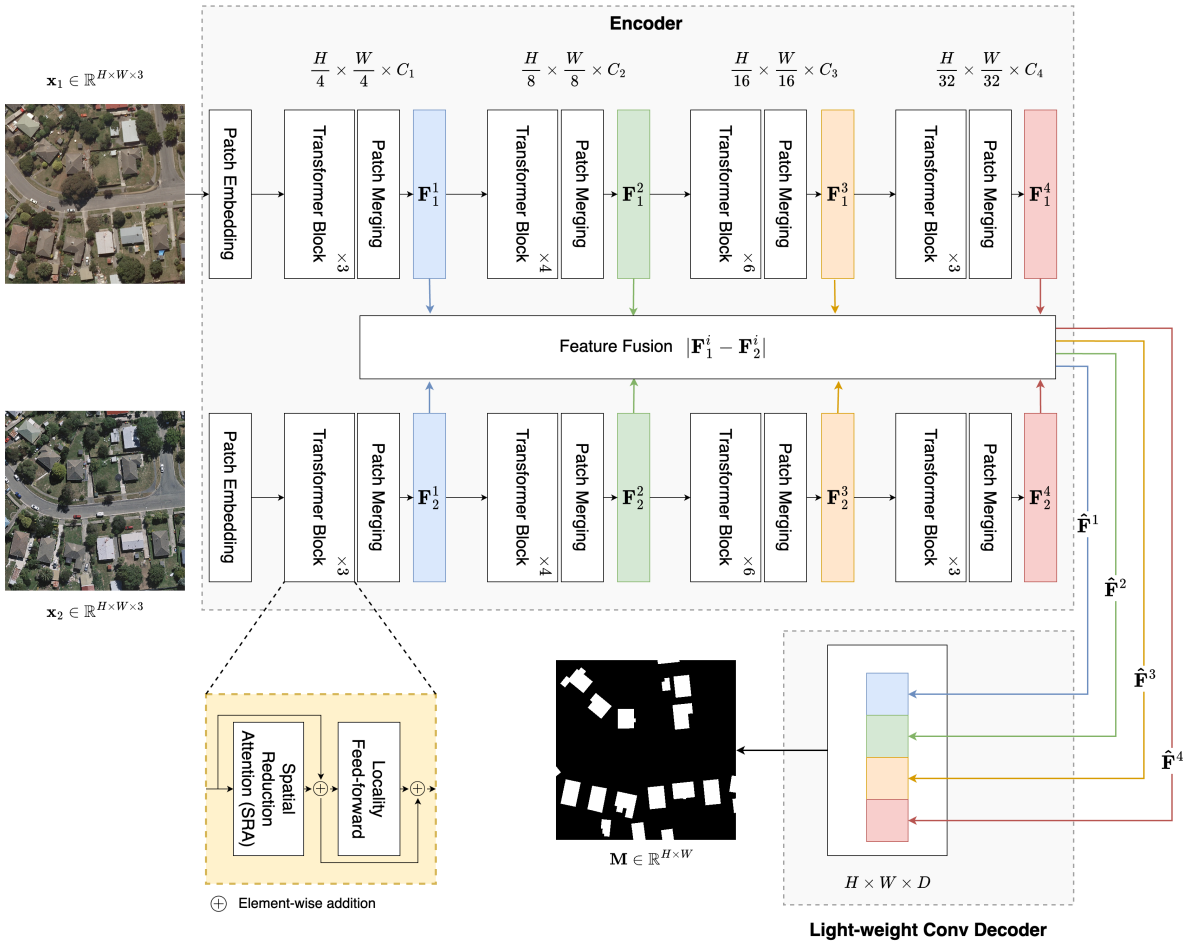


Figure 2. LocalCD consists of the encoder, feature fusion, and the decoder.

d is the number of attention heads. $\hat{\mathbf{X}} \in \mathbb{R}^{\frac{N}{R_i}\times d}$ is a reduced version of $\mathbf{X}$ after applied reshape operation with R_i reduction ratio and a linear layer. RSA computational complexity is R_i^2, which is lower than the original MSA, which has N^2.

Patch Merging. Patch Merging is responsible for decreasing the spatial dimension of the input images while increasing its feature dimension. Technically, it takes the refined visual tokens produced by the Transformer block and then reshapes them into grid-shaped data similar to the original input images. Each 2×2 image patch is concatenated along the feature dimension. Finally, a linear layer is applied to reduce the feature dimension in half. This strategy guarantees that the resulting spatial size is reduced in half while the feature dimension doubles.

3.3. Locality Feed-forward

The original Transformer [5] follows the attention operation with two layers of MLP. These layers force the Transformer to learn a richer representation by expanding the intermediate dimension. The first MLP expands the feature dimension by a factor of 4, while the second MLP restores the feature to the original dimension. This scenario improves the capability of the Transformer to model long-range interaction further. Despite being excellent at modeling long dependencies between visual tokens, Transformer lacks local connection. One reason is that it treats images as a sequence instead of grid-shaped data. However, a pixel in natural images like remote sensing images is much affected by neighbor pixels.

Inspired by [14], we impose the locality into the Transformer architecture by replacing the feed-forward network with two consecutive 1×1 convo-

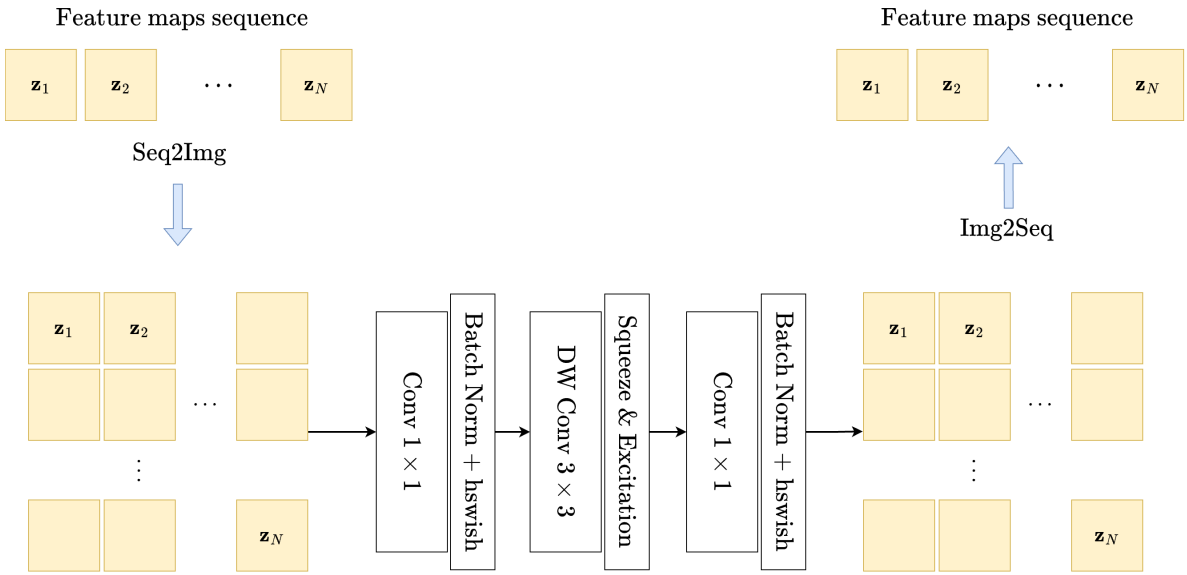


Figure 3. Locality Feed-forward imposes the locality mechanism by performing DW convolution on the reshaped feature maps

lutions. These convolutions can mimic the effect of expanding dimensions like in MLP. However, they do not impose a spatial relationship due to using the 1×1 kernel. To solve this issue, we insert an efficient depth-wise convolution (DW) with 3×3 kernel [19]. We also add a combination of `BatchNorm` [20] and `hswish` [21] activation after each 1×1 convolution to improve the locality induction further. We also add a `Squeeze & Excitation` [22] layer after the DW convolution. Figure 3 illustrates the locality feed-forward.

Suppose $\mathbf{Z} \in \mathbb{R}^{N\times d}$ is a sequence of tokens returned by the attention mechanism. First, the token sequence is reshaped into a 2D feature map to form grid-shaped data through a `Seq2Img` operation. This process is denoted as:

$$\hat{\mathbf{Z}} = \text{Seq2Img}(\mathbf{Z}), \tag{2}$$

where $\hat{\mathbf{Z}} \in \mathbb{R}^{\frac{H_i}{P_i}\times\frac{W_i}{P_i}\times C_i}$. The DW convolution is then applied to the resulting 2D feature map $\hat{\mathbf{Z}}$. Formally it could be represented as

$$\hat{\mathbf{Y}} = f(f(\hat{\mathbf{Z}} \circledast \hat{\mathbf{W}}_1) \circledast \mathbf{W}_d) \circledast \hat{\mathbf{W}}_2, \tag{3}$$

where $\hat{\mathbf{W}}_1 \in \mathbb{R}^{d\times\lambda d\times1\times1}$, $\hat{\mathbf{W}}_2 \in \mathbb{R}^{\lambda d\times d\times1\times1}$ are 1×1 convolution with λ expansion ratio. Here, $\mathbf{W}_d \in \mathbb{R}^{\lambda d\times1\times k\times k}$ is a DW convolution with $k \times k$ kernel and $f(.)$ is a nonlinear activation function. $\circledast$ is element-wise multiplication. The result is then converted back into a sequence

$$\mathbf{Y} = \text{Img2Seq}(\hat{\mathbf{Y}}). \tag{4}$$

where `Img2Seq` is simply a reshaping operation that arranges the grid-shaped data into a sequence of patches.

3.4. Feature Fusion

The feature fusion fuses the feature maps $\mathbf{F}_1^i$ and $\mathbf{F}_2^i$ produced from the encoder. Unlike ChangeFormer [13], we use Lp distance to fuse two feature maps into a single feature map. The fusion operation is defined as:

$$\hat{\mathbf{F}}^i = |\mathbf{F}_1^i - \mathbf{F}_2^i| \tag{5}$$

where $\mathbf{F}_1^i$ is feature map from image $\mathbf{x}_1$ on stage-i and $\mathbf{F}_2^i$ is feature map from image $\mathbf{x}_2$ on stage-i.

3.5. Lightweight Convolution Decoder

The decoder takes the multi-scale fused feature maps $\hat{\mathbf{F}}^1 \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C_1}$, $\hat{\mathbf{F}}^2 \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_2}$, $\hat{\mathbf{F}}^3 \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times C_3}$, and $\hat{\mathbf{F}}^4 \in \mathbb{R}^{\frac{H}{32}\times\frac{W}{32}\times C_4}$ and then convert it into the change map. Unlike ChangeFormer [13] that used an MLP decoder, we instead use Lightweight Convolution Decoder (LCD) to support the locality mechanism imposed by the encoder. Fig. 4 illustrates the LCD module.

We apply 1×1 convolution to each feature map $\hat{\mathbf{F}}^i$ to project its feature dimension into an arbitrary dimension D. We then applied upsample operation to resize the feature map to the same size as the original input image $H \times W$. We used the different

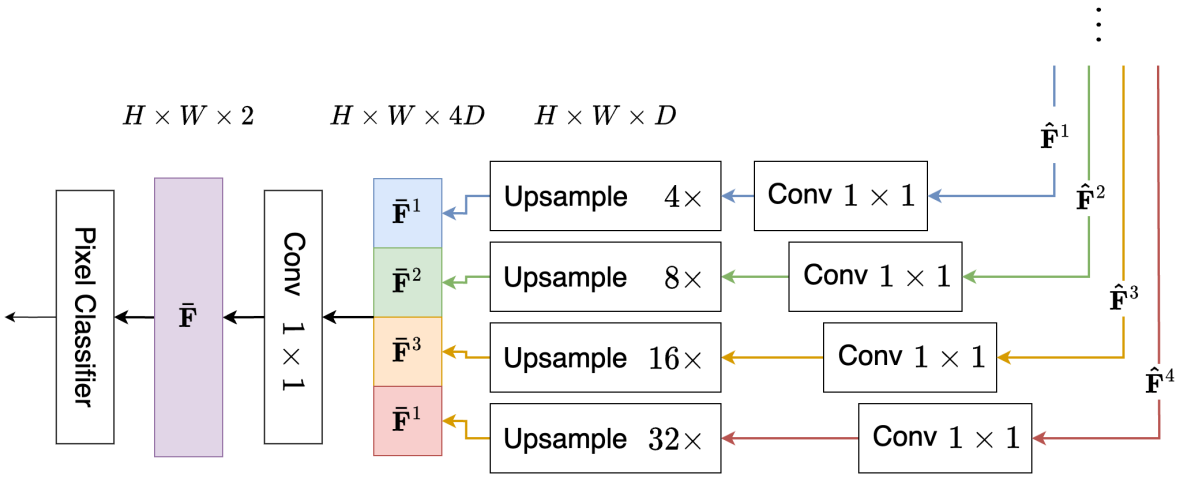


Figure 4. Lightweight Convolution Decoder (LCD) takes the feature maps from the encoder and generates the change map through several convolutions and upsample operations

upsample factor based on the scale of the feature map, $32\times$, $16\times$, $8\times$, and $4\times$ for $\hat{\mathbf{F}}^1$, $\hat{\mathbf{F}}^2$, $\hat{\mathbf{F}}^3$, and $\hat{\mathbf{F}}^4$ respectively. These operations resulting four refined feature maps $\bar{\mathbf{F}}^1, \bar{\mathbf{F}}^2, \bar{\mathbf{F}}^2, \bar{\mathbf{F}}^4 \in \mathbb{R}^{H \times W \times D}$. Formally, it can be formulated as follow:

$$\bar{\mathbf{F}}^i = \texttt{Upsample}(H, W)(\texttt{Conv}(\hat{\mathbf{F}}^i)). \tag{6}$$

We then concatenated the feature maps along feature dimensions, resulting in a single feature map with dimension $4D$. We then project the feature dimension into 2 using 1×1 resulting in the final feature map $\bar{\mathbf{F}} \in \mathbb{R}^{H \times W \times 2}$. These operations can be formulated as follows:

$$\bar{\mathbf{F}} = \texttt{Conv}(\texttt{Concat}(\bar{\mathbf{F}}^1, \bar{\mathbf{F}}^2, \bar{\mathbf{F}}^3, \bar{\mathbf{F}}^4)). \tag{7}$$

4. Experiments

4.1. Dataset

We use two public datasets in the experiments, CDD [3] and LEVIR-CD[16]. The CDD is a set of remote-sensing images used for change detection tasks. It consists of two images: a pre-image and a post-image, along with the ground truth change map. Each image has 256 × 256 pixels in size. The images are captured using various sensors, including optical and synthetic aperture radar (SAR) sensors. The CDD dataset contains several subsets with different characteristics, including urban, rural, and natural environments, and images captured at different resolutions and with other imaging conditions. The dataset includes photos with gradual and abrupt changes, such as the growth of vegetation, construction of buildings, and natural disasters. CDD provides training, validation, and testing sets. The total of images in each set is 10,000 for training, 3,000 for validation, and 3,000 for testing.

LEVIR-CD consists of 637 pairs of high-resolution images of size 1024×1024. The pairs are in the same area within 5 to 14 years. The images cover various buildings, such as villas, apartments, garages, and warehouses. The photos are taken from Google Earth in different regions that sit in several cities in Texas of the US, including Austin, Lakeway, Bee Cave, Buda, Kyle, Manor, Pflugervilletx, Dripping Springs, etc. LEVIR-CD comes with a ground truth binary change map of a pair of images. The dataset is then divided into 445, 64, and 128 image pairs for train, validation, and testing, respectively.

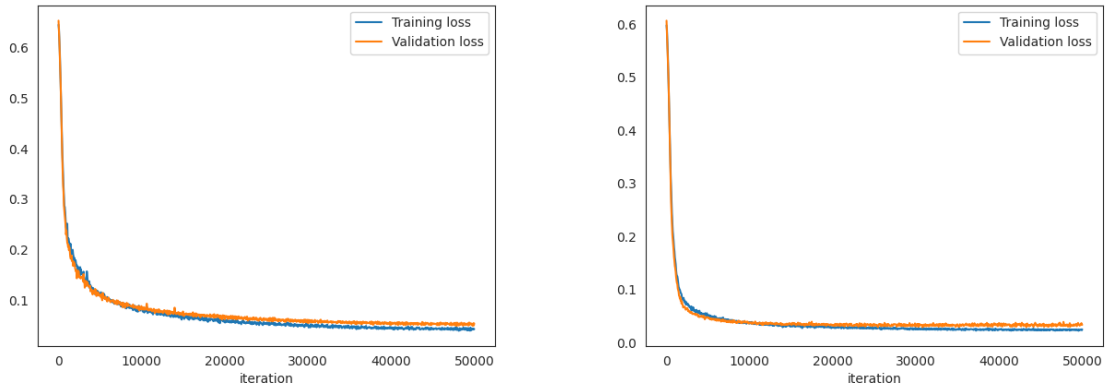
4.2. Experimental Setup

We implemented LocalCD using the OpenCD [25] toolkit. It is a CD framework built on MMSegmentation [26], a widely adopted PyTorch framework for image segmentation tasks. We used a fixed patch size of 4 with various strides in the Overlap Patch Embedding. The strides are 4, 2, 2, and 2 for stages 1, 2, 3, and 4. We used the Transformer block layer $L_i = \{3, 4, 6, 3\}$ for stage-i 1, 2, 3, and 4, respectively. Furthermore, we used the embedding size $C_i = \{64, 128, 320, 512\}$. We used the number of heads 1, 2, 5, and 8. Also, we used a reduction ratio R_i 8, 4, 2, and 1.

We augmented the input images, including the ground truth change map using the standard image augmentations such as random rotation (at maximum

Table 1. Evaluation results on CDD and LEVIR-CD datasets (all values are in percentage (%))

Method	CDD						LEVIR-CD					
	F1	IoU	OA	Acc	P	R	F1	IoU	OA	Acc	P	R
FC-EF [23]	73.09	62.52	91.22	68.29	84.61	68.29	90.18	83.36	98.47	88.98	91.47	88.98
BIT [12]	71.37	61.03	91.28	66.06	88.57	66.06	90.43	83.69	98.34	93.67	87.68	93.67
SNUNet [18]	92.1	85.95	96.82	90.08	94.43	90.08	88.16	78.83	98.82	50.00	**95.80**	50.00
STANet [16]	93.62	44.00	88.00	50.00	88.00	50.00	87.26	77.40	98.66	50.00	95.80	50.0
SwinSUNet [24]	90.52	83.51	96.27	87.71	93.97	87.71	92.42	86.69	**98.81**	91.32	93.59	91.32
ChangeFormer [13]	95.00	90.73	97.95	93.77	**96.33**	93.77	91.97	86.00	98.71	91.53	92.42	**91.53**
LocalCD (ours)	**95.48**	**91.57**	**98.12**	**94.81**	96.19	**94.81**	**92.43**	**86.70**	98.80	**91.41**	93.51	91.41



(a) **(b)**

Figure 5. Plot of training and validation loss on (a) CDD dataset, (b) LEVIR-CD dataset

$180°$), fixed-size random crop, and horizontal or vertical flip. We used these augmentations techniques at 50% probability. We normalized the image pixels using the standard normalization used for the ImageNet [8] dataset with $[123.675, 116.28, 103.53]$ as mean and $[58.395, 57.12, 57.375]$ as standard deviation, each for channel red, green, and blue respectively.

We used the AdamW [27] optimizer with the learning rate 6×10^{-5}, β_1 0.9, β_2 0.999. To avoid overfitting, we used a weight decay of 0.01. We trained the model on a single GPU V100 for 50,000 iterations using a batch size 16. Also, We used a polynomial learning rate scheduler [28] with a linear warmup for the first 1,500 iterations. To speed up the model's convergencies, we initialized the encoder's weights using pre-trained weights of SegFormer [11], a hierarchical transformers model for segmentation tasks. The models are trained to minimize the binary cross-entropy loss.

5. Result and Discussion

5.1. Quantitative Evaluation

We provide the experiment result in Table 1. We evaluate the method using various metrics commonly used to evaluate segmentation tasks like F1-score, Intersect over Union (IoU), Overall Accuracy (OA), Accuracy, Precision (P), and Recall (R). This table shows that LocalCD outperforms existing CD methods on CDD and LEVIR-CD datasets on those evaluation metrics. Specifically, it consistently outperforms the baseline CD method, ChangeFormer [13], on most evaluation metrics. The result proves the effectiveness of our proposed method.

During training, we record the training and validation loss. Fig. 5 displays the training and validation loss during the model training on CDD and LEVIR-CD datasets. From the graphic, we can see that the training losses consistently decreases as the training iteration increase. Furthermore, the validation losses are reduced smoothly following the corresponding training loss.

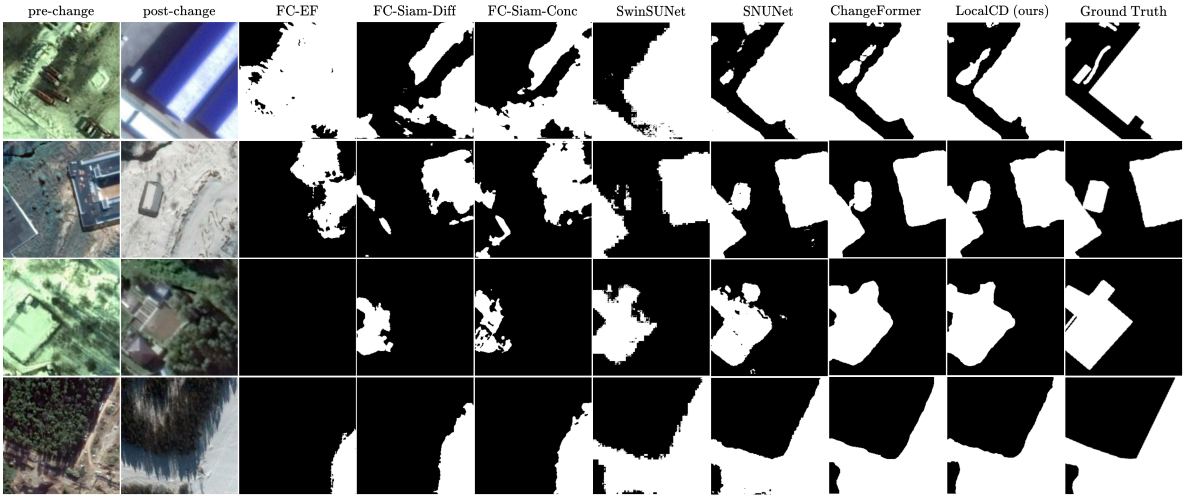


Figure 6. Comparison of predicted change map visualization between LocalCD and other CD methods

Table 2. Ablation study for verifying the effectiveness of an individual component of LocalCD (all values are in percentage (%).

No	Locality Feed-forward	Lp Distance Fusion	LCD	F1	IoU	OA	Acc	P	R
1	✗	✗	✗	87.58	79.26	95.27	83.96	92.49	83.96
2	✓	✗	✗	91.17	84.5	96.42	89.55	92.99	89.55
3	✗	✗	✓	89.82	82.46	95.96	87.45	92.66	87.45
4	✓	✓	✓	**91.92**	**85.67**	**96.77**	**89.73**	**94.48**	**89.73**

5.2. Qualitative Evaluation

We also perform a qualitative evaluation to support the quantitative one. We randomly select four samples from the test set and perform inference on them. Fig. 6 shows the visualization of the predicted change map for various CD methods on the test set. The figure shows that the predicted change maps of LocalCD have better changed-region boundaries compared to other CD methods, including ChangeFormer. Specifically, the results of LocalCD are less contain the segmentation artifacts, proving our proposed method's effectiveness.

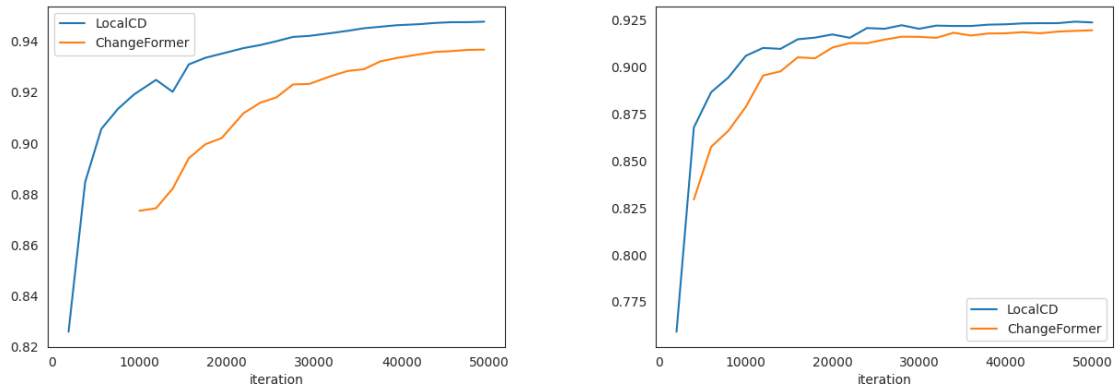
5.3. Ablation Study

We perform an ablation study to verify the effectiveness of an individual component of LocalCD. We train each model for 10,000 iterations to reduce the training time while keeping all configurations the same. Table 2 summarizes the results of ablation experiments.

Row numbers 1 & 2 on the table examine the effectiveness of the Locality Feed-forward component. It is clear from the result that Locality Feed-forward improves all evaluation metrics. In contrast, row number 3 examines the effectiveness of the LCD component. Compared to row number 1, it is also clear from the result that the LCD component improves all evaluation metrics. Finally, the last row examines the effectiveness of combined components, including the Locality Feed-forward, Lp Distance Fusion, and LCD. The combination of all these components yields the best evaluation result.

5.4. Comparison with ChangeFormer

The experiment result shown in Table 1 proves the superiority of LocalCD compared to ChangeFormer in various evaluation metrics. We also compare the number of parameters and floating-point operations for a fair comparison. Table 3 shows the number of parameters and floating-point operations for both methods. Similar to the ablation study, the results in table 3 are also trained for 10,000 iterations. Despite LocalCD having a little higher parameters and floating point operations compared to ChangeFormer, the margin is not significant compared to the increased performance of LocalCD. To verify this claim, we create a reduced version of LocalCD, called LocalCD-Reduced, by decreasing the number of Transformer blocks on each stage



(a) **(b)**

Figure 7. Plot of F1 score during training on the test set (a) CDD dataset, (b) LEVIR-CD dataset

into $\{2, 2, 2, 2\}$. Although the LocalCD-Reduced has lower parameters and floating-point operations than ChangeFormer, it has a higher F1 score.

During training, we regularly perform evaluations on the test set and record the evaluation result. Fig. 7 shows the plot of the F1 score on CDD and LEVIR-CD datasets. From the figure, we can see that LocalCD consistently outperforms ChangeFormer.

Table 3. The number of parameters and floating point operations comparison between LocalCD and ChangeFormer using $256 \times 256 \times 6$ of input size.

Method	# Parameter	GLOPs	F1
ChangeFormer	24.72 M	10.35	87.89
LocalCD (ours)	24.9 M	10.45	**90.61**
LocalCD-Reduced (ours)	**13.76 M**	**5.95**	90.12

6. Conclusion

In this paper, we proposed LocalCD, a novel CD method that addresses the issue of segmentation artifacts that arise on the prediction change map of ChangeFormer. LocalCD solves this issue by imposing the locality mechanism into the Transformer architecture used by ChangeFormer. Although the ChangeFormer can model long-range dependencies between visual tokens on the natural images, it still lacks a locality mechanism. This problem can cause imperfectness in the changed region boundaries of the predicted change map. Extensive experiments on CDD and LEVIR-CD demonstrated the effectiveness of our proposed method. It outperforms the baseline CD method with lower or comparable parameters and floating point operations.

References

[1] A. Singh, "Review Article Digital change detection techniques using remotely-sensed data," *International Journal of Remote Sensing*, vol. 10, no. 6, pp. 989–1003, Jun. 1989.

[2] M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley, "Change detection from remotely sensed images: From pixel-based to object-based approaches," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 80, pp. 91–106, Jun. 2013.

[3] M. A. Lebedev, Y. V. Vizilter, O. V. Vygolov, V. A. Knyaz, and A. Y. Rubis, "Change detection in remote sensing images using conditional adversarial networks," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2, pp. 565–571, May 2018. [Online]. Available: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2/565/2018/

[4] 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, May 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/10/1688

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, Dec.

2017, arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Jun. 2021, arXiv: 2010.11929. [Online]. Available: http://arxiv.org/abs/2010.11929

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, arXiv:1512.03385 [cs]. [Online]. Available: http://arxiv.org/abs/1512.03385

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[9] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions," Aug. 2021, arXiv:2102.12122 [cs]. [Online]. Available: http://arxiv.org/abs/2102.12122

[10] 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," Aug. 2021, arXiv:2103.14030 [cs]. [Online]. Available: http://arxiv.org/abs/2103.14030

[11] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers," Oct. 2021, number: arXiv:2105.15203 arXiv:2105.15203 [cs]. [Online]. Available: http://arxiv.org/abs/2105.15203

[12] 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, 2022, arXiv:2103.00208 [cs]. [Online]. Available: http://arxiv.org/abs/2103.00208

[13] W. G. C. Bandara and V. M. Patel, "A Transformer-Based Siamese Network for Change Detection," Sep. 2022, arXiv:2201.01293 [cs]. [Online]. Available: http://arxiv.org/abs/2201.01293

[14] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "LocalViT: Bringing Locality to Vision Transformers," *arXiv:2104.05707 [cs]*, Apr. 2021, arXiv: 2104.05707. [Online]. Available: http://arxiv.org/abs/2104.05707

[15] R. C. Daudt, B. L. Saux, A. Boulch, and Y. Gousseau, "Urban Change Detection for Multispectral Earth Observation Using Convolutional Neural Networks," Oct. 2018, arXiv:1810.08468 [cs]. [Online]. Available: http://arxiv.org/abs/1810.08468

[16] 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, May 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/10/1662

[17] 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, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9259045/

[18] 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, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9355573/

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Mar. 2019, number: arXiv:1801.04381 arXiv:1801.04381 [cs]. [Online]. Available: http://arxiv.org/abs/1801.04381

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015.

[21] A. G. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019.

[22] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 2011–2023, 2017.

[23] R. C. Daudt, B. L. Saux, and A. Boulch, "Fully Convolutional Siamese Networks for Change Detection," *arXiv:1810.08462 [cs]*, Oct. 2018, arXiv: 1810.08462. [Online]. Available: http://arxiv.org/abs/1810.08462

[24] 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. [Online]. Available: https://ieeexplore.ieee.org/document/9736956/

[25] S. Fang, K. Li, and Z. Li, "Changer: Feature Interaction is What You Need for Change Detection," Sep. 2022, arXiv:2209.08290 [cs]. [Online]. Available: http://arxiv.org/abs/2209.08290

[26] M. Contributors, "MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark," https://github.com/open-mmlab/mmsegmentation, 2020.

[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2017.

[28] P. Mishra and K. P. Sarawadekar, "Polynomial learning rate policy with warm restart for deep neural network," *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pp. 2087–2092, 2019.