Lightweight Remote Sensing Image Change Detection Based on Global Feature Fusion

Aiying Wu[1,3], Tianze Zhang[2,3], Yuanxu Zhu[1,3], Zhaole Ning[1,3], and Gang Shi[1,3(✉)]

[1] School of Computer Science and Technology, Xinjiang University, Urumqi 830017, China
shigang@xju.edu.cn
[2] Xinjiang Key Laboratory of Signal Detection and Processing, Urumqi 830046, China
[3] Faculty of Science, The University of Melbourne, Parkville, VIC 3010, Australia

Abstract. Change detection in remote sensing images is of great importance as a basis for a variety of tasks. However, traditional fully convolutional change detection networks lack global information, while transformer-based networks can extract global information, but the number of parameters and computational complexity are too large. This paper proposes a lightweight change detection network that fuses global information, with only 2.1M parameters, to address these issues. The network combines the local information extracted by fully convolutional with the global information extracted by LSTM to take full advantage of both. Experiments were carried out on three different types of datasets: LEVIR-CD, SYSU-CD and NJDS. The results show that the IOU of the model was improved by 0.94%, 4.74% and 0.9% respectively. This confirms that the model has a high accuracy with a very small number of parameters and a low computational cost, providing a new solution for efficient change detection.

Keywords: Change detection · Light weight · Global feature

1 Introduction

The successive launch of remote sensing has led to the development of a more complete Earth observation system, with a notable increase in remote sensing images. These images provide support for monitoring the environment. Change detection in remote sensing imagery employs advanced algorithms to compare images of the same location taken at different time intervals, automatically identifying regions with notable differences to detect changes on the Earth's surface [1]. This technology is crucial for applications such as urban expansion monitoring, land use planning, and disaster prevention.

Conventional change detection methods use algebraic techniques to analyze difference images [2], but they rely on manual feature extraction and threshold setting, making them unsuitable for complex scenes [3]. The rise of deep learning has shifted focus to end-to-end change detection methods that learn image features from large datasets via backpropagation, overcoming the instability of manual feature design.

D.-S. Huang et al. (Eds.): ICIC 2025, LNCS 15844, pp. 26–37, 2025.
https://doi.org/10.1007/978-981-96-9869-1_3

Remote sensing image change detection approaches based on deep learning are commonly trained using an end-to-end learning framework. Early approaches focused on fully convolutional networks, where remote sensing images from different time points are fused using simple channel-wise concatenation [4] or other specific fusion methods [5]. Change detection is then performed by leveraging the results of fully convolutional feature extraction networks and detection decoders. To enhance detection performance, attention mechanisms, such as channel-wise or spatial attention, have been integrated into the process [6].

Since the emergence of Transformers [7], numerous NLP techniques have been successfully transferred to image analysis tasks. These methods incorporate Transformer-based feature extraction and processing techniques into remote sensing image change detection [8]. Following extraction by the convolutional network, the feature map is reformulated into multiple vectors, facilitating the modeling of global dependencies via the self-attention mechanism. By combining this global information with local features extracted from the convolutional network, detection accuracy is significantly improved. However, while these Transformer-based methods deliver superior results, they come at the cost of a substantial increase in both parameters and computational complexity.

One key challenge has been improving detection accuracy while minimizing model size. Fully convolutional networks offer an advantage in this regard, as they can boost accuracy by integrating lightweight convolutional attention mechanisms, thus maintaining a relatively small model size with fewer parameters [9]. However, these models suffer from the lack of global information fusion, resulting in lower detection performance compared to Transformer-based approaches. Some methods attempt to optimize self-attention mechanisms to reduce parameter count [10], but the reduction in computational complexity is limited by the overall architecture.

Accordingly, a lightweight Siamese-structured change detection network with non-shared weights (LSNW) is developed in this work. The problem of missing global information in traditional fully convolutional networks is solved. Incorporating global information enhances the detection performance of the lightweight network across various dataset types. The primary contributions of this paper are outlined as follows:

1. We design the LSNW change detection model that incorporates global information, where essential semantic change features are derived through hierarchical fusion of five local feature layers and one global feature layer. The network design with pseudo-Siamese structure is adopted to map remote sensing image images at different times to different feature spaces to enrich the representation and improve the accuracy.
2. We introduce a lightweight multi-level feature fusion module (MLFF) that performs feature transformation through cascaded upsampling. Temporal features are fused using spatial and channel-wise attention, enhancing change features and removing interference to generate a key difference feature map.
3. The GFFM module, based on LSTM [11], is incorporated to replace self-attention and add global information to the feature map. Guarantee the fusion of global context with convolutional local features while maintaining a low parameter overhead.
4. Performance evaluations on three complex remote sensing datasets confirm that our approach delivers improved accuracy with lower model complexity compared to state-of-the-art techniques.

2 Methodology

As illustrated in Fig. 1, the LSNW network adopts EfficientNet-B4 [12] as a lightweight backbone for extracting features. To process temporally distinct images, the network employs two independent branches with unshared weights. The resulting features are concatenated and fused initially through the MLFF module. The fused change feature maps are converted into vector groups and processed by the GFFM module for global fusion and transformation. The resulting features are then progressively upsampled and combined with encoder features to restore image details. Finally, the classification module produces the upsampling and change detection results via a classifier. Figure icons

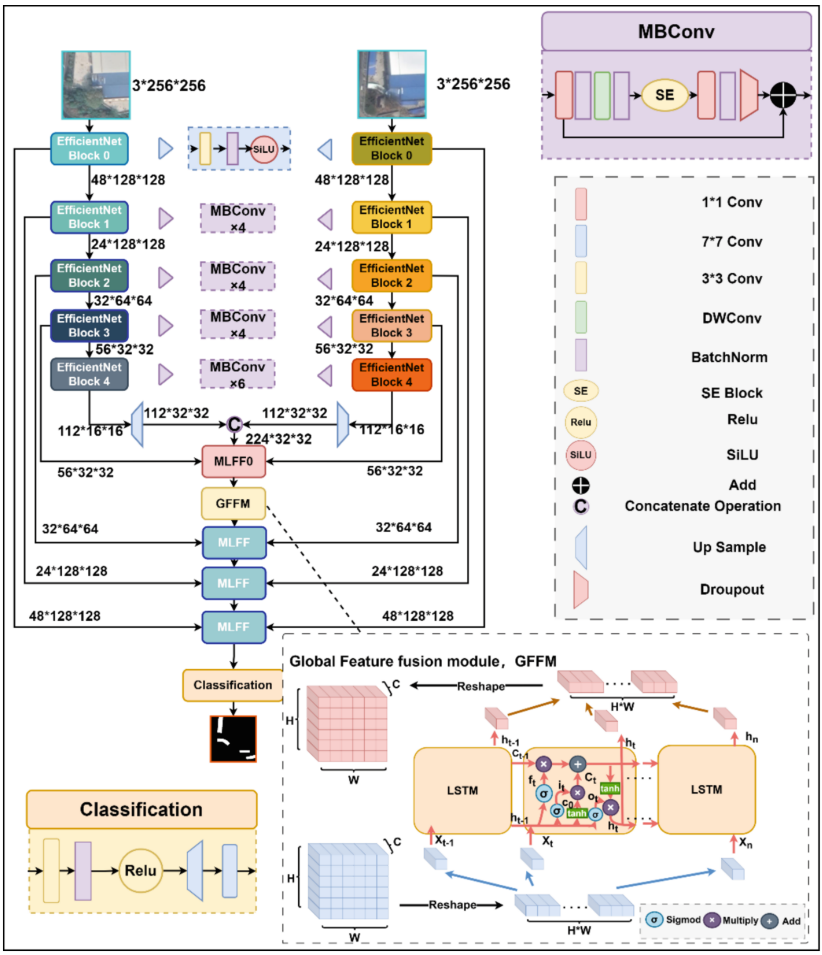

Fig. 1. An overview of the LSNW architecture is provided, with the configuration of the global feature fusion module (GFFM) depicted in the figure's bottom-right inset. 'EfficientNet Block' is the module in the feature extraction network and "Classification" is the classifier, the structure is shown in Figure.

represent the SE Block [13] (squeeze-and-excitation) and SiLU [14] activation function; other common elements are not detailed.

2.1 Global Feature Fusion Module (GFFM)

LSNW uses a fully convolutional network that captures only local features. To integrate global context, we adopts an LSTM-based method inspired by the BIT model [8], transforming convolutional feature maps into time-series data. The feature map is divided into vectors consistent with the original channel dimensions, which are then used as input tokens for the LSTM, which stores and transforms global information through its gating mechanism (memory, input, and output gates). This process reassembles the outputs into a feature map enriched with global context, effectively modeling long-range dependencies.

The 'memory gate' filters out irrelevant information while preserving essential global image features. It processes the current input x_t and previous hidden state output h_{t-1} using a learned weight matrix, followed by a sigmoid function for gating. The resulting weight is applied to the stored global information, removing noise and retaining key semantic changes. The process is described in Eq. (1), where W_f is the weight matrix in the memory gate and b_f is the bias vector.

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{1}$$

The "input gate" like the 'memory gate', receives the current input X_t and the previous transformed feature vector h_{t-1}. It controls how new information updates the 'cell' state. Using a sigmoid function (Eq. 2), it computes the update ratio, then applies a Tanh function to transform the new input (Eq. 3). The retained portion is added to the previous global state C_{t-1} to form the updated global information C_t (Eq. 4). The formula is as Eq. (4), where W_i and W_c are the weight matrices in the input gate, b_i and b_c are the bias vectors in the input gate.

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{2}$$

$$c_0 = tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \tag{3}$$

$$C_t = f_t \odot C_{\{t-1\}} + i_t \odot c_0 \tag{4}$$

Tasked with managing the transformation and delivery of information, the output gate determines the final output of the current cell's feature vector. It receives the updated global state and current input X_t and h_{t-1}, and uses a Tanh function to generate guiding information from the global state. This guide is multiplied with the processed input to produce a feature vector that embeds global context, which is then passed to the next 'cell'. The formula as shown in Eq. (5), where W_o is the weight matrix in the output gate and b_o is the bias vector.

$$h_t = \left(\sigma\left(W_o \cdot \left[h_{\{t-1\}}, x_t\right] + b_o\right)\right) \odot tanh\left(C_t\right) \tag{5}$$

The GFFM module builds global context by sequentially storing feature vectors in long-term memory. It transforms each vector using this global reference, enriching the local feature map and enhancing the ability to distinguish changes.

2.2 Multi-level Feature Fusion Module (MLFF)

To generate change detection results, the MLFF module compares semantic differences between features from different time points to produce a change feature map. This map is fused with the original feature map to restore detail. The change features are then progressively upsampled, merging large- and small-scale changes at each level to produce the final fused result. The module's structure is shown in Fig. 2.

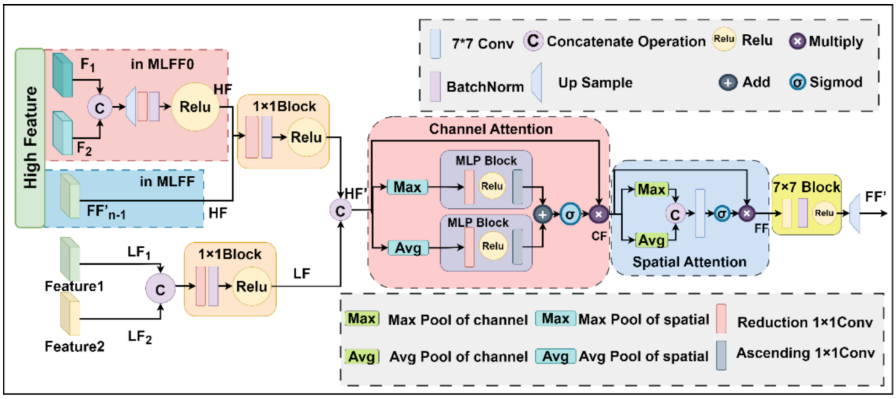

Fig. 2. Multi-Level Feature Fusion module (MLFF) structure diagram. (High Feature, HF; Low Feature, LF)

In MLFF0, two feature maps, F1 and F2, from different time are fused. The features are initially concatenated along the channel axis and then upsampled to align with the spatial dimensions of the low-level features. A single-layer perceptron with 1×1 convolutions fuses the features, and after normalization, Relu is applied to filter change regions in the high-level features (HF). In MLFF, the fusion result from the upper layer is directly used as the change region discrimination result for high-level features, HF, as shown in Eq. (6). The obtained high-level features are then transformed through 1×1 convolutions to obtain the transformed high-level features 'HF', as shown in Eq. (7).

$$HF = \begin{cases} Relu\left(BN\left(Conv_{1\times1}\left(cat\left(F_1, F_2\right)\right)\right)\right), MLFF0 \\ FF'^{n-1}, \qquad\qquad MLFF \end{cases} \tag{6}$$

$$HF' = Relu(BN(Conv_{1\times1}(HF))) \tag{7}$$

The low-level features LF1 and LF2 from different tense levels are spliced by channels and input into a 1×1 convolution block for channel processing to fit the shape of the high-level feature HF', in preparation for layer-by-layer contrastive fusion in multi-layer feature fusion, as shown in Eq. (8).

$$LF = Relu(BN(Conv_{1\times1}(cat(LF_1, LF_2))) \tag{8}$$

After processing, high-level (HF) and low-level (LF) feature maps are resized to the same shape for guided fusion using the attention mechanism. In the channel attention, max and average pooling are used to derive a weight vector. The weights are applied to the corresponding channels of the input feature, and the fused feature (CF) is generated through element-wise multiplication, as shown in Eq. (9). Spatial attention follows a similar process, pooling across the channel dimension to calculate pixel-wise weights, resulting in the final feature fusion output (FF), as shown in Eq. (10).

$$CF = HF' \odot \sigma(Add \begin{pmatrix} MaxPool_{sp}(Conv_{1\times1}(ReLU(Conv_{1\times1}(HF')))) \\ AugPool_{sp}(Conv_{1\times1}(ReLU(Conv_{1\times1}(HF')))) \end{pmatrix}) \quad (9)$$

$$FF = CF \odot \sigma(Conv_{7\times7} \begin{pmatrix} MaxPool_{ch}(CF) \\ AugPool_{ch}(CF) \end{pmatrix}) \quad (10)$$

The feature map is subsequently processed by a 7 × 7 convolution with a larger kernel to aggregate a broader range of information. After upsampling, the result is the discriminative result FF' for the change region of the current layer, which is used as the input for the next layer, as shown in Eq. (11).

$$FF' = UpSample(ReLU(BN(Conv_{7\times7}(FF)))) \quad (11)$$

The MLFF module fuses multi-scale feature maps and feature maps of different tense through attention weighting in channel and spatial dimensions. By stacking multiple MLFF modules, the model continuously enhances the change information in the image. High-level features provide large-scale information, while low-level features capture detailed information. Together, they complement each other to produce a more precise change feature map.

3 Experimental and Analysis

3.1 Datasets

LEVIR-CD [15]: The dataset contains high-resolution (0.5 m) Google Earth imagery spanning 5–14 years, with a focus on building changes. During training, images are divided into non-overlapping pairs of 256 × 256 patches. Following the dataset's standard split, there are 7120 training pairs, 1024 validation pairs, and 2048 test pairs.

SYSU-CD [16]: This dataset is based on aerial images of Hong Kong from 2007 and 2015, with a 0.5 m ground resolution. Change labels are area-based rather than tied to specific building details. The dataset consists of 12,000 pairs for training, 4,000 pairs for validation, and 4,000 pairs for testing.

NJDS [17]: This dataset includes Google Earth images of Nanjing from 2014 and 2018, with a 0.3 m resolution, focusing on changes in buildings from different perspectives. There are differences in the angle and position of the same buildings shown in the different time plots. The images were segmented into non-overlapping 256 × 256 patches for further processing. After random splitting, the dataset contains 1521 training pairs, 504 test pairs, and 504 validation pairs.

This paper analyzes the distribution of positive and negative pixels across the three datasets, as shown in Table 1. Negative samples significantly outnumber positive ones, causing the model to bias toward negative predictions during training, which reduces recall, F1 score, and IOU.

Table 1. Positive and negative sample statistics in the datasets.

Dataset	Positive Sample	Negative Sample	Positive and negative sample ratio
LEVIR-CD	95.35%	4.65%	1:19
SYSU-CD	78.17%	21.83%	7:26
NJDS-CD	96.84%	3.16%	1:24

3.2 Experimental Setup

- Evaluation metrics: Model accuracy is evaluated using four metrics: precision (Pre.), recall (Rec.), F1 score (F1.), and intersection over union (IOU). Model performance is assessed using parameter count (Params) and floating point operations (FLOPs).
- Experimental setup: The change detection model was implemented in Python 3.9 with PyTorch v11.2. AdamW optimizer was used with a 0.001 weight decay and a learning rate of 0.0002, employing a warm-up strategy. Training ran for up to 250 epochs with a batch size of 16, using an NVIDIA A40 GPU (48 GB).

3.3 Comparative Experiment

We compare our method with ten state-of-the-art (SOTA) models. These include early fusion (FE-EF), Siamese CNNs (FC-Diff, FC-Conc) [4], convolutional networks with deep supervision (IFNet [6]), self-attention-based fusion (BIT [8]), hybrid CNN-transformer models (ICIF [10]), lightweight networks (LightCD [9]), edge-aware detection (ELGC [18]), and transformer-based extraction (FTAN [19]). Results are presented in Table 2.

The results show that LSNW achieves significantly higher recall than other methods, with slightly lower precision. This indicates better handling of class imbalance, as the model emphasizes hard-to-detect and positive samples. The higher F1 score confirms its superior overall detection accuracy.

We selected three representative datasets from the three datasets for visualisation, as shown in Fig. 3. LSNW has shown good results in many types of change detection tasks. Combining change region discrimination with global information allows the model to more accurately detect the position of the change region in the middle.

To assess the balance between accuracy and efficiency, we compared each model's parameters, FLOPs, and IOU in Table 3. LSNW requires far fewer resources than self-attention-based models like BIT, ICIF, and FTAN, while outperforming fully convolutional methods in accuracy with lower parameter counts and computational cost.

Table 2. Comparison results of three CD datasets. The optimal results are denoted as bold. All indicators are shown as percentages (%).

	LEVIR-CD	SYSU-CD	NJDS
Method	Pre. /Rec. /F1	Pre. /Rec. /F1	Pre. /Rec. /F1
FC-EF(2018)	86.91/80.17/83.40	74.32/75.84/75.07	44.78/14.65/22.08
FC-Diff(2018)	89.53/83.31/86.31	**89.13**/61.21/72.58	53.25/43.61/47.95
FC-Conc(2018)	91.99/76.77/83.69	82.54/71.03/76.35	48.65/14.59/22.44
IFNet(2020)	90.37/71.27/90.82	86.96/73.37/79.59	77.25/73.37/75.26
BIT(2021)	89.24/89.37/89.30	81.14/76.48/78.74	80.24/65.70/72.25
SNUNet(2021)	89.18/87.17/88.16	78.26/76.30/77.27	69.16/72.35/70.72
ICIF(2022)	89.60/84.30/86.80	85.09/71.26/77.56	87.88/71.84/79.05
LightCD(2023)	91.30/88.00/89.60	83.01/74.90/78.75	**89.40**/73.75/80.82
ELGCNet(2024)	**92.10**/88.47/90.25	83.26/74.85/78.83	79.70/58.08/67.19
FTAN(2024)	91.32/89.66/90.48	82.05/75.18/78.47	68.67/58.10/62.94
LSNW(Ours)	92.24/**90.52/91.37**	83.01/**82.85/82.93**	84.04/**79.03/81.46**

3.4 Ablation Experiments

- Global Feature Fusion Level: The GFFM module in LSNW introduces global features to enhance convolutional outputs, with different fusion levels offering varying semantic depth and computational cost. Experiments on the NJDS dataset (Table 4) show that lower-level feature maps (layers 3 and 4) provide richer global information but require more parameters and computation. Higher-level maps (layers 0 and 1) reduce cost due to smaller sizes. Adding GFFM at layer 1 reduced parameters by 24.43% and FLOPs by 10.65%, with only a 0.23% drop in F1. Thus, LSNW incorporates GFFM at layer 1 for optimal balance.

- To assess the lightweight nature of the GFFM module, we compared it with the commonly used eight-headed self-attention module on the NJDS dataset (Table 5). The results show that GFFM significantly reduces parameters and computation compared to self-attention. Additionally, adding self-attention in a single layer does not fully capture global information, while GFFM offers superior fusion and detection performance.

- Module structure validity: To verify the validity of the non-shared weight structure design and global feature fusion design modules in the LSNW model, ablation experiments were performed on three datasets, and the results are shown in **Table 6**.

The lightweight GFFM module, with only 0.14M parameters, significantly reduces computational complexity and parameters compared to self-attention-based fusion modules. Its performance varies across datasets. In SYSU-CD, dominated by large-scale

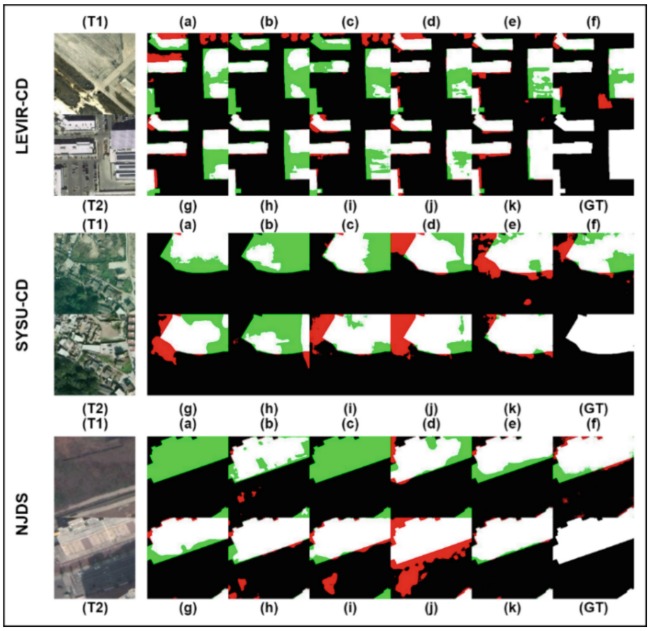

Fig. 3. Visualization results on three datasets. (T1) T1 image. (T2) T2 image. (a) FC-EF. (b) FC-Diff. (c) FC-Conc. (d) IFNet. (e) BIT. (f) SNUNet. (g) ICIF. (h) LightCD. (i) ELGCNet. (j) FTAN. (k) LSNW. (GT) Ground Truth. TP (white). TN (black). FP (red). FN (green) (Color figure online)

regional changes, global information improves regional discrimination and accuracy. In the NJDS dataset, which varies in perspective, it enhances semantic change identification. In LEVIR-CD, a small-scale building change detection dataset, the improvement is limited due to the discrete nature of the change areas.

"Different Branch" indicates whether non-shared weight branches are used. While these branches increase the model's parameters, the same feature extraction network structure ensures that computation remains unchanged. In tasks with the same viewpoint, this design allows the model to learn features separately for different times, improving detection accuracy. For datasets like NJDS with varying viewpoints, non-changing regions may suffer from semantic errors due to viewpoint deviation. The local features from non-shared weight structures amplify this semantic bias, widening differences when mapped to feature spaces, ultimately reducing detection accuracy.

When unshared weights are combined with the global feature fusion module, local feature discrepancies are corrected by the supplementary global information, improving detection results. On perspective-biased datasets like NJDS, the global features enhance the accuracy of feature maps, reducing local information errors and improving judgment accuracy. The combined F1 and IOU metrics across three datasets demonstrate the effectiveness of this design, with IOU improvements of 0.5%, 2.05%, and 2.61%, respectively.

Table 3. Comprehensive comparison of cost and accuracy calculation of different methods. The optimal results are denoted as bold.

			LEVIR-CD	SYSU-CD	NJDS
Method	Params(M)	FLOPs(G)	IOU (%)	IOU (%)	IOU (%)
FC-EF(2018)	**1.35**	3.56	71.35	60.09	12.41
FC-Diff(2018)	1.54	5.30	75.91	56.96	31.52
FC-Conc(2018)	1.35	4.70	71.96	61.75	12.64
IFNet(2020)	50.71	82.86	83.18	66.10	60.34
BIT(2021)	3.55	67.80	80.68	64.94	56.55
SNUNet(2021)	12.03	54.83	78.83	62.96	54.7
ICIF(2022)	10.10	25.41	76.80	63.35	65.36
LightCD(2023)	10.75	21.54	81.20	64.98	67.82
ELGCNet(2024)	10.57	123.59	82.23	65.06	50.59
FTAN(2024)	42.54	211.06	82.61	64.56	45.93
LSNW(Ours)	2.10	**2.35**	**84.12**	**70.84**	**68.72**

Table 4. GFFM module ablation experiments at different layers. The optimal results are denoted as bold.

Layer	Channels	Feature Map	Params(M)	FLOPs(G)	Pre./Rec./F1./IOU (%)
0	56	32 × 32	**2.038**	**2.34**	80.75/80.08/80.41/67.24
1	32	64 × 64	2.097	2.35	**84.04**/79.03/81.46/68.72
2	24	128 × 128	2.368	2.40	79.35/79.73/79.54/66.03
3	48	128 × 128	2.775	2.63	82.71/80.70/**81.69**/**69.05**
4	48	128 × 128	2.775	2.63	80.28/**81.41**/80.84/67.84

Table 5. Comparison results between GFFM module and self-attention module

Method	Params(M)	FLOPs(G)	Precision	Recall	F1	IOU
Self-Attention	69.214	4.46	67.56	63.95	65.71	48.93
GFFM	2.097	2.35	84.04	79.03	81.46	68.72

Table 6. Ablation experiments of LSNW model on three datasets. The optimal results are denoted as bold.

Dataset	Different Branch	GFFM	Params(M)	FLOPs(G)	Pre./Rec./F1./IOU
LEVIR-CD	×	×	1.020	2.32	91.94/90.23/91.08/83.62
	√	×	1.954	2.32	92.33/90.12/91.21/83.85
	×	√	1.164	2.35	**92.38**/89.83/91.09/83.63
	√	√	2.097	2.35	92.24/**90.52/91.37/84.12**
SYSU-CD	×	×	1.020	2.32	79.84/**83.24**/81.51/68.79
	√	×	1.954	2.32	**84.90**/79.30/82.00/69.50
	×	√	1.164	2.35	84.12/80.08/82.05/69.56
	√	√	2.097	2.35	83.01/82.85/**82.93/70.84**
NJDS	×	×	1.020	2.32	82.02/77.31/79.59/66.11
	√	×	1.954	2.32	78.3276.2177.2562.93
	×	√	1.164	2.35	81.17/**80.18**/80.67/67.60
	√	√	2.097	2.35	**84.04**/79.03/**81.46/68.72**

4 Conclusion

This paper presents LSNW, a lightweight change detection model that improves accuracy through non-shared weight feature extraction and a lightweight global feature fusion module (GFFM) based on LSTM. The MLFF module fuses multi-scale image features to generate change detection results. Experiments on LEVIR-CD, SYSU-CD, and NJDS datasets show promising results. However, LSNW uses a general image feature extraction network, and there is potential for reducing parameters and computational complexity. Additionally, the GFFM module currently uses fixed-length inputs, which should be made more adaptable in future work.

Acknowledgments. This research was funded by the Key R&D projects of Xinjiang Uygur Autonomous Region, grant number 2022B01006.

References

1. Lv, Z.Y., Huang, H.T., Li, X., et al.: Land cover change detection with heterogeneous remote sensing images: review, progress, and perspective. Proc. IEEE **110**(12), 1976–1991 (2022)
2. Bruzzone, L., Prieto, D.F.: Automatic analysis of the difference image for unsupervised change detection. IEEE Trans. Geosci. Remote Sens. **38**(3), 1171–1182 (2000)
3. Lei, J., Gu, Y., Xie, W., et al.: Boundary extraction constrained siamese network for remote sensing image change detection. IEEE Trans. Geosci. Remote Sens. **60**, 1–13 (2022)
4. Daudt, R.C., Le Saux, B., Boulch, A.: Fully convolutional siamese networks for change detection. In: Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 4063–4067 (2018)

5. Fang, S., Li, K., Shao, J., et al.: Snunet-CD: a densely connected siamese network for change detection of VHR images. IEEE Geosci. Remote Sens. Lett. **19**, 1–5 (2021)
6. Zhang, C., Yue, P., Tapete, D., et al.: A deeply supervised image fusion network for change detection in high-resolution bi-temporal remote sensing images. ISPRS J. Photogramm. Remote. Sens. **166**, 183–200 (2020)
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., et al.: Attention is all you need. In: Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS), pp. 1–10 (2017)
8. Chen, H., Qi, Z., Shi, Z.: Remote sensing image change detection with transformers. IEEE Trans. Geosci. Remote Sens. **60**, 1–14 (2021)
9. Yang, H., Chen, Y., Wu, W., et al.: A lightweight siamese neural network for building change detection using remote sensing images. Remote Sens. **15**(4), 928 (2023)
10. Feng, Y., Xu, H., Jiang, J., et al.: ICIF-Net: intra-scale cross-interaction and inter-scale feature fusion network for bitemporal remote sensing images change detection. IEEE Trans. Geosci. Remote Sens. **60**, 1–13 (2022)
11. Greff, K., Srivastava, R.K., Koutník, J., et al.: LSTM: a search space odyssey. IEEE Trans. Neural Networks Learn. Syst. **28**(10), 2222–2232 (2016)
12. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR, pp. 6105–6114 (2019)
13. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7132–7141 (2018)
14. Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Netw. **107**, 3–11 (2018)
15. Chen, H., Shi, Z.: A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. Remote Sens. **12**(10), 1662 (2020)
16. Shi, Q., Liu, M., Li, S., et al.: A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection. IEEE Trans. Geosci. Remote Sens. **60**, 1–16 (2021)
17. Shen, Q., Huang, J., Wang, M., et al.: Semantic feature-constrained multitask siamese network for building change detection in high-spatial-resolution remote sensing imagery. ISPRS J. Photogramm. Remote. Sens. **189**, 78–94 (2022)
18. Noman, M., Fiaz, M., Cholakkal, H., et al.: ELGC-Net: efficient local-global context aggregation for remote sensing change detection. IEEE Trans. Geosci. Remote Sens. **62**, 1–11 (2024)
19. Yu, C., Li, H., Hu, Y., et al.: Frequency-temporal attention network for remote sensing imagery change detection. IEEE Geosci. Remote Sens. Lett. **21**, 1–5 (2024)