

# Similarity Learning for Land Use Scene-Level Change Detection

Jinglei Liu , Weixun Zhou , *Member, IEEE*, Haiyan Guan , *Senior Member, IEEE*, and Wenzhi Zhao 

**Abstract**—Scene-level change detection (SLCD) can provide semantic change information at image level, thus it is of great significance for monitoring land use changes. Supervised SLCD approaches tend to outperform unsupervised ones. However, the existing supervised methods rely on postclassification, which often results in unsatisfactory performance due to classification error accumulation. We therefore formulate SLCD as a similarity learning task, and propose a scene similarity learning network (SSLN) for land use SLCD. To be specific, SSLN is a two-branch network with ResNet as the backbone for feature extraction, where the global feature difference and the multiscale local feature fusion modules are considered in order to better mine the temporal correlation between bitemporal scenes. Then, the trained SSLN is further exploited to obtain the similarity of scene pairs for determining the similarity threshold via threshold traversal algorithm. Finally, the land use scenes are categorized into changed or unchanged by comparing scene similarity with the threshold. Experimental results on the publicly available MtS-WH dataset and our newly released land use scene change detection dataset show that the proposed approach achieves better performance than comparison methods, indicating that our approach is a simple yet effective solution to land use SLCD.

**Index Terms**—Land use, scene change detection, scene similarity, similarity learning.

## I. INTRODUCTION

REMOTE sensing change detection (RSCD) aims to monitor the changes of ground objects in a region through repeated observation [1]. At present, RSCD has been widely used in many fields [2], [3], such as urban planning [4], natural resource management [5], [6], [7], and disaster assessment

[8]. With the development of earth observation technology, the spatial resolution of remote sensing images has shown a trend of development from medium and low resolution to high resolution. Compared with medium- and low-resolution images, high-resolution images provide more detailed information of ground objects, and thus it is a main data source for RSCD.

RSCD can be divided into pixel-level, object-level, and scene-level change detection. Pixel-level change detection is focused on independent pixels as detection units. There has been a large number of works in this field [9], [10], [11], [12], [13], [14], [15]. Shi et al. [12] proposed a deep supervised based attention metric network to reduce pseudovariation and noise. In [13], an iterative training sample enhancement strategy was proposed, which was combined with deep learning neural networks to improve the performance of land cover change detection. In contrast, object-level change detection is to detect the changes based on semantic objects [16], [17]. Zhang et al. [16] proposed an object-level change detection framework that detects changing geographic entities such as new buildings or changing artificial structures by paying more attention to the overall characteristics and contextual associations of changing object instances. Doi et al. [17] designed a network that can detect object-level changes in image pairs, and can capture different scene changes in image pairs with different viewpoints.

But different from pixel- and object-level change detection, scene-level change detection (SLCD) is able to detect and identify changes at image-level, as shown in Fig. 1. It is observed that the changes of pixels or objects cannot directly reflect the changes of land use types. For example, though the bare land (BL) has changed to industrial house in pixel-level and object-level change detection, the land use type remains to be industrial land (IL). With the continuous refinement of urban functional areas, SLCD is of great significance for monitoring the change of land use types and further planning of the city. Therefore, land use SLCD has become a necessary and important research direction in the field of RSCD.

The key of SLCD is to extract powerful scene feature representations. Many existing approaches rely on handcrafted features to perform change detection [18], [19]. Wu et al. [18] presented a supervised scene change detection framework that combined the bag-of-visual-word (BoVW) model with SVM classifier. Specifically, the multitemporal scene images were encoded by BoVW model and then fed into SVM to obtain classification results. Du et al. [19] proposed an unsupervised SLCD method based on latent Dirichlet allocation (LDA) and multivariate alteration detection (MAD), where LDA and MAD

Manuscript received 24 November 2023; revised 8 February 2024; accepted 1 March 2024. Date of publication 5 March 2024; date of current version 18 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 42001285, in part by the Open Fund of State Key Laboratory of Remote Sensing Science under Grant OFSLRSS202215, and in part by the Key Laboratory of Land Satellite Remote Sensing Application, Ministry of Natural Resources of the People's Republic of China under Grant KLSMNR-G202202. (Corresponding author: Weixun Zhou.)

Jinglei Liu and Haiyan Guan are with the School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: 859507654@qq.com; guanhy.nj@nuist.edu.cn).

Weixun Zhou is with the School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with the State Key Laboratory of Remote Sensing Science, Beijing Normal University, Beijing 100875, China (e-mail: zhouwx@nuist.edu.cn).

Wenzhi Zhao is with the State Key Laboratory of Remote Sensing Science, Beijing Normal University, Beijing 100875, China (e-mail: wenzhi.zhao@bnu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3373401

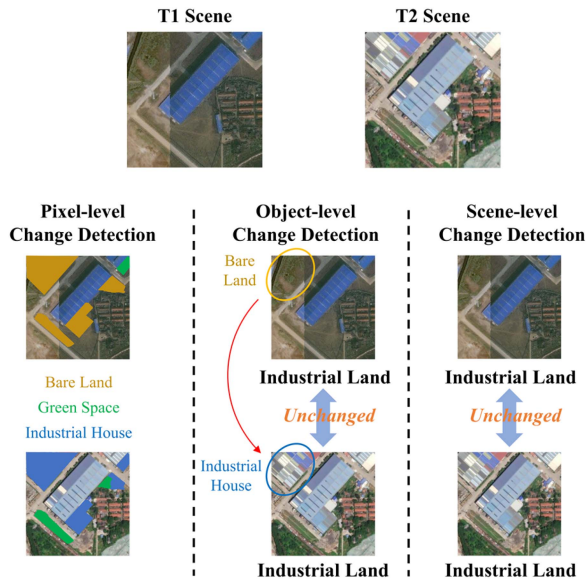


Fig. 1. Relationship between pixel-level, object-level, and scene-level change detection.

were used to identify feature topics and detect scene-level changes, respectively. However, the handcrafted features are not feasible for image scenes with high complexity. In recent years, inspired by the remarkable performance of deep learning methods particularly convolutional neural networks (CNNs) on various image recognition tasks [20], [21], [22], remote sensing community has explored them for various tasks, including scene classification, semantic segmentation, as well as change detection [23], [24], [25].

To advance land use SLCD, scholars have proposed various methods, which mainly include unsupervised methods and supervised methods. Fang et al. [26] presented a novel automatic binary SLCD approach based on deep learning. It combined the direct predetection with postclassification-based predetection to ensure the stability for generating change detection training samples. However, it is often difficult for unsupervised methods to achieve satisfactory performance due to the lack of labeled samples. For supervised approaches, Fang et al. [27] proposed a new framework for fusing differential aggregation networks and class probability-based fusion strategies to fully capture temporal change features and generate final binary change detection results by fusing three predicted class-based probability vectors. In their recent work, Fang et al. [28] also presented a multibranch fusion network capable of accepting both high-resolution images and a kernel density map as input, helping to fully mine depth features. Ru et al. [29] proposed a CorrFusion module by fusing the highly correlated information between bitemporal features. Though SLCD performance has been improved, CorrFusion belongs to classification-based methods (CBM). For simple land use scenes, the CBM can achieve good performance, which is, however, not the case for land use scenes with high complexity. This is because the change detection performance depends on the classification results of bitemporal scenes. The incorrect classification results of land use scenes in any phase will affect the final change detection performance. The comparisons between the

existing works along with their main distinctive characteristics are summarized in Table I.

To avoid the influence of incorrect classification results, scholars have investigated similarity learning for image classification. For example, Pinheiro [30] proposed a classification method based on similarity learning, which performs classification by calculating the similarity between prototype representations of each category. Moreover, similarity learning is also used in other fields [31], [32]. Zagoruyko and Komodakis [31] proposed a method of learning general patch similarity function directly from image data and its performance is much better than the optimal method at that time. Simo-Serra et al. [32] presented a feature description method based on deep learning, where the Siamese network was used to extract features from blocks, and L2 distance was selected to measure the differences between features, in order to shorten the distance between matched block features, and to increase the distance between unmatched block features.

Apart from powerful scene representations, large-scale datasets are also indispensable for developing supervised deep learning-based land use SLCD methods. There are currently two existing benchmark datasets, i.e., MtS-WH [18] and WH-MAVS [33] in the literature. However, MtS-WH dataset contains 190 training samples and 1920 testing samples, respectively, in each phase, thus is not suitable for deep learning-based land use SLCD due to its small volume. Compared to MtS-WH, WH-MAVS dataset is a large-scale dataset with 47 134 samples in total, but it is not publicly available at present. Therefore, an open-source large-scale benchmark is necessary in order to advance SLCD research.

To solve the above-mentioned issues, we propose a simple yet effective land use SLCD approach by addressing SLCD from a different perspective. To be specific, we regard SLCD as a similarity learning task and propose a scene similarity learning network (SSLN). Unlike the existing CBM, our proposed approach performs SLCD based on scene similarity, and thus is independent of classification results. To conduct binary change detection, the similarity threshold is first determined via traversal of similarity values between bitemporal scene pairs obtained by the trained SSLN, and the final binary SLCD results are achieved by comparing the scene pair similarity values with the similarity threshold. To evaluate the performance of our proposed approach, we construct a large-scale benchmark dataset, which is publicly available for research purposes. In summary, our main contributions are as follows.

- 1) We formulate land use binary SLCD as a similarity learning problem, and propose a simple yet effective SLCD approach based on SSLN. To the best of our knowledge, it is the first work that addresses the SLCD task from the perspective of similarity learning.
- 2) To demonstrate the proposed approach, we collect and release a large-scale land use scene change detection (LUSCD) dataset, which is currently the largest publicly available dataset for land use SLCD.
- 3) Our proposed approach achieves state-of-the-art performance on MtS-WH and LUSCD datasets, outperforming

TABLE I  
COMPARISONS BETWEEN THE EXISTING WORKS ALONG WITH THEIR MAIN DISTINCTIVE CHARACTERISTICS

Change detection works		Main distinctive characteristics
Pixel-level change detection [9], [10], [11], [12], [13], [14], [15]		Only changes at pixel level can be identified, and semantic changes at image level cannot be detected.
Object-level change detection [16], [17]		Only changes at object level can be identified, and semantic changes at image level cannot be detected.
SLCD	Handcrafted feature-based methods [18], [19]	It is laborious to design powerful handcrafted features and their generalization ability is limited.
	CBM [26]-[29]	The performance will be severely affected by the classification results due to the accumulation of classification errors.

TABLE II  
SOME IMPORTANT ITEMS AND CORRESPONDING ABBREVIATIONS

Abbreviations	Full name
RSCD	remote sensing change detection
SLCD	scene-level change detection
SSLN	scene similarity learning network
LUSCD	land use scene change detection
GFD	global feature difference
MSLFF	multiscale local feature fusion
CBM	classification-based methods
OFBM	offline similarity-based methods

the CBM and offline similarity-based methods (OSBM), proving the effectiveness of the proposed method.

The rest of the article is organized as follows. Section II depicts the new LUSCD dataset in detail. Section III introduces the proposed land use SLCD approach. Section IV discusses the experimental results. Finally, Section V concludes the article. In addition, we list some important items and their corresponding abbreviations, as shown in Table II, for the readers' convenience.

## II. LARGE-SCALE LUSCD BENCHMARK DATASET

As shown in Table III, our LUSCD dataset is compared to the two commonly used datasets, i.e., MtS-WH [18] and MH-MAVS [33], in terms of the number of samples, image size, spatial resolution, the number of categories, and the availability. It is obvious that our newly constructed LUSCD dataset has the advantages of large volume and open source. To make LUSCD a more challenging dataset, the land use scenes are collected from five China cities including Hangzhou, Shanghai, Wuhan, Hefei, and Nanjing. Based on China's "Code for classification of urban land use and planning standards of development land (GB50137-2011)," and the land use classes used in MtS-WH [17] and MH-MAVS [33], the land use scenes are distributed in the following 10 categories in our LUSCD dataset: residential land (RL), public service and commercial land (PSCL), educational land (EL), IL, transportation land (TL), agricultural land (AL), water body (WB), green space (GS), woodland (WL), and BL. Regarding the training set and validation set, the land use

scenes of each category of phase 1 and phase 2 of the three cities, i.e., Hangzhou, Shanghai, and Wuhan are combined to obtain the scenes of phase 1 and phase 2 of each city. Then, the land use scenes of each city are randomly divided into training set and validation set with the ratio of 80% and 20%, respectively, whereas for the testing set, the land use scenes of Hefei and Nanjing are taken as testing set A and testing set B, respectively. It is notable that the two testing datasets are collected from different cities, thus is able to demonstrate the transferability of deep learning methods.

Table IV illustrates the division of the training set, validation set, and testing set in LUSCD. It can be seen that the training set and validation set contain 18 108 and 4526 land use scene pairs, respectively, whereas the testing sets A and B contain 5078 and 5062 land use scene pairs, respectively. Therefore, LUSCD is a large-scale dataset that is appropriate for land use SLCD. Fig. 2 presents some changed and unchanged land use scenes of each category.

## III. PROPOSED METHOD

### A. Problem Definition

Let  $X_1$  and  $X_2$  be the bitemporal scenes of the same region,  $Y_1, Y_2 \subset \{0, 1, \dots, L - 1\}$  be their corresponding labels, where  $L$  is the number of scene classes. For the classification-based SLCD methods, the binary change information (changed or unchanged) between  $X_1$  and  $X_2$  is obtained based on the predicted labels  $Y_1^*$  and  $Y_2^*$ . Specifically, the bitemporal scenes  $X_1$  and  $X_2$  are changed if  $Y_1^* \neq Y_2^*$ , otherwise  $X_1$  and  $X_2$  are not changed if  $Y_1^* = Y_2^*$ . As we know, scenes from the same class tend to have higher similarity than that from different classes. For example, the similarity between two RL scenes is generally higher than the similarity between RL and BL scenes. Based on this premise, the scene pair  $X$  consisting of  $X_1$  and  $X_2$  can be associated with the label  $Y = \{0, 1\}$ , where "0" means  $X_1$  and  $X_2$  are dissimilar ( $X_1$  and  $X_2$  are from different classes) and "1" means  $X_1$  and  $X_2$  are similar ( $X_1$  and  $X_2$  are from the same class). Therefore, the binary SLCD can be regarded as a similarity learning problem, and the key is to find a mapping function  $g(\cdot)$  defined as follows:

$$g(X|(X_1, X_2)) = s \quad (1)$$



TABLE III  
COMPARISON BETWEEN LUSCD AND THE EXISTING DATASETS

Dataset	Number of Images	Image Size	Resolution (m)	Number of Categories	Data Source	Open Source
MtS-WH [17]	4220	150×150	1	8	one city	Yes
MH-MAVS [33]	47134	200×200	1.2	14	one city	No
LUSCD	65548	300×300	1	10	five cities	Yes

TABLE IV  
DATA SPLITS OF TRAINING SET, VALIDATION SET, AND TESTING SET IN LUSCD DATASET

Category	Label	Training Set		Validation Set		Testing Set			
						Test A		Test B	
		Time 1	Time 2	Time 1	Time 2	Time 1	Time 2	Time 1	Time 2
RL	1	3696	4073	873	963	855	1017	1187	1339
PSCL	2	513	671	138	178	248	274	65	79
EL	3	551	644	134	155	204	293	206	247
IL	4	1058	925	252	228	396	411	500	436
TL	5	3402	3825	860	978	1357	1466	758	910
AL	6	2788	1968	728	508	1016	593	489	176
WB	7	2512	2446	630	614	412	402	521	493
GS	8	1193	1243	293	321	250	334	373	398
WL	9	1754	1766	457	466	76	141	814	794
BL	10	641	547	161	115	264	147	149	190
<b>Number of Images</b>		18108	18108	4526	4526	5078	5078	5062	5062
<b>Number of Images in LUSCD dataset: 65548</b>									

where  $s \in (0, 1)$  is the learned similarity value between  $X_1$  and  $X_2$ , which is also the predicted label of  $X$ . There is a substantial chance that  $X_1$  and  $X_2$  are not changed if  $s$  closes to 1, otherwise  $X_1$  and  $X_2$  are changed if  $s$  closes to 0.

It is worth noting that  $s$  is essentially the probability of whether  $X_1$  and  $X_2$  are similar, which is different from the similarity (or distance) of feature vectors calculated in the Euclidean space. However, we can regard  $s$  as the learned similarity because it has the same range (i.e.,  $[0, 1]$ ) as the similarity value, and a higher  $s$  indicates the higher similarity score between  $X_1$  and  $X_2$ .

### B. Process of the Proposed SLCD Approach

As shown in Fig. 3, our proposed SLCD approach contains two phases. In Phase 1, the land use scene pairs in training set are fed into SSLN to learn scene similarity. To ensure the performance of SSLN, MSLFF module and GFD module are designed and integrated into SSLN. In Phase 2, the trained SSLN is exploited for searching similarity threshold  $T$  based on the similarity values of bitemporal scenes in validation set. The final binary scene-level change information of scenes in testing set is obtained by comparing scene similarity value with threshold  $T$ .

### C. Scene Similarity Learning With SSLN

1) *Architecture of the Proposed SSLN*: SSLN is a two-branch weight-shared network with the pretrained ResNet50 [34] as

the backbone for feature extraction (the fully connected layer is removed), as shown in Fig. 3. It is worth noting that some layers in ResNet50 are omitted for convenience. SSLN can be coarsely divided into feature extraction part and similarity learning part consisting of two modules, i.e., global feature difference (GFD) module and multiscale local feature fusion (MSLFF) module.

With respect to feature extraction, the scenes of Time 1 and Time 2 are fed into one of the two branches, respectively. The outputs of the last pooling layer are integrated into GFD module to obtain global feature difference. The outputs of the last layer in the last bottleneck of ResNet50 are integrated into MSLFF module to obtain multiscale fused local features. Considering that it is difficult to compare land use scenes from two phases due to temporal effects, the outputs of modules GFD and MSLFF (i.e.,  $F_G$  and  $F_L$ ) are combined to better learn scene similarity. Specifically,  $F_G$  and  $F_L$  are converted into feature vectors, which are fused and connected to a fully connected layer with a single output. Here, a sigmoid function is used to convert the single output into a value between  $(0, 1)$ , indicating the similarity score between two temporal scenes.

2) *MSLFF Module*: The features from shallow layers have higher resolution, and thus contain more location and detail information [35]. To take into account the scale difference of ground objects within the scene, we designed the MSLFF module. Local feature  $F_L$  is achieved by MSLFF module, as shown in Fig. 3.  $A^1$  and  $A^2$  are the output feature maps extracted from the last layer of the last bottleneck of ResNet50 in the



Fig. 2. Some changed and unchanged example images of each category in LUSCD dataset.

two branches, respectively. To take objects at different scales in the scene into account, three filter kernels with different sizes and pooling operation are used to obtain feature maps  $A_j^1$  and  $A_j^2$  ( $j = 1, 2, 3$ ), which are then concatenated to obtain  $A_4^i$  as follows:

$$A_4^i = f(A_1^i, A_2^i, A_3^i) \quad (i = 1, 2) \quad (2)$$

where  $f(\cdot)$  represents the concatenation operation. In order to reduce the channel dimensions,  $A_4^i$  ( $i = 1, 2$ ) is converted to  $B^i$  ( $i = 1, 2$ ) by a  $1 \times 1$  convolution, followed by a product operation to achieve  $F^i$  ( $i = 1, 2$ ). The process is defined by

$$F^i = A^i \odot B^i \quad (i = 1, 2) \quad (3)$$

where  $\odot$  stands for elementwise product operation. Finally, local feature  $F_L$  is achieved by concatenating  $F^1$  and  $F^2$  as follows:

$$F_L = f(F^1, F^2) \quad (4)$$

where  $f(\cdot)$  also represents the concatenation operation.

3) *GFD Module*: The difference features between the bitemporal images can clearly display the change information of the images [2], [36]. In order to obtain global difference features effectively and enhance the similarity learning ability of the network, we designed a GFD module. Global feature difference  $F_G$  is achieved by GFD module in Fig. 3.

$F_G$  is defined by

$$F_G = |D_1 - D_2| \quad (5)$$

where  $D_1$  and  $D_2$  are the output feature maps of the two branches, respectively.  $|\cdot|$  is the absolute difference between the two feature maps.

To train the proposed SSLN, the binary cross-entropy loss function is selected, which is defined by

$$L = -\frac{1}{N} \sum_{i=1}^N [y^i \cdot \log(p^i) + (1 - y^i) \cdot \log(1 - p^i)] \quad (6)$$

where  $N$  is the number of scene pairs,  $y^i$  is the label of scene pair  $X^i$ , and  $p^i$  is the output of SSLN after sigmoid function, which is also the learned similarity between the two temporal scenes in  $X^i$ .

#### D. SLCD Using the Learned Scene Similarity

Once SSLN is trained, the learned scene similarity is exploited for land use SLCD. Phase 2 in Fig. 3 depicts the process of the similarity-based SLCD. During the process, one can observe that the key step is to find the similarity threshold  $T$ . To this end, we propose a threshold search approach, as shown in Algorithm 1. It is worth noting that the optimal similarity threshold  $T$  is determined on the validation set to ensure the transferability of SSLN. In addition, considering the fact that Kappa coefficient can better

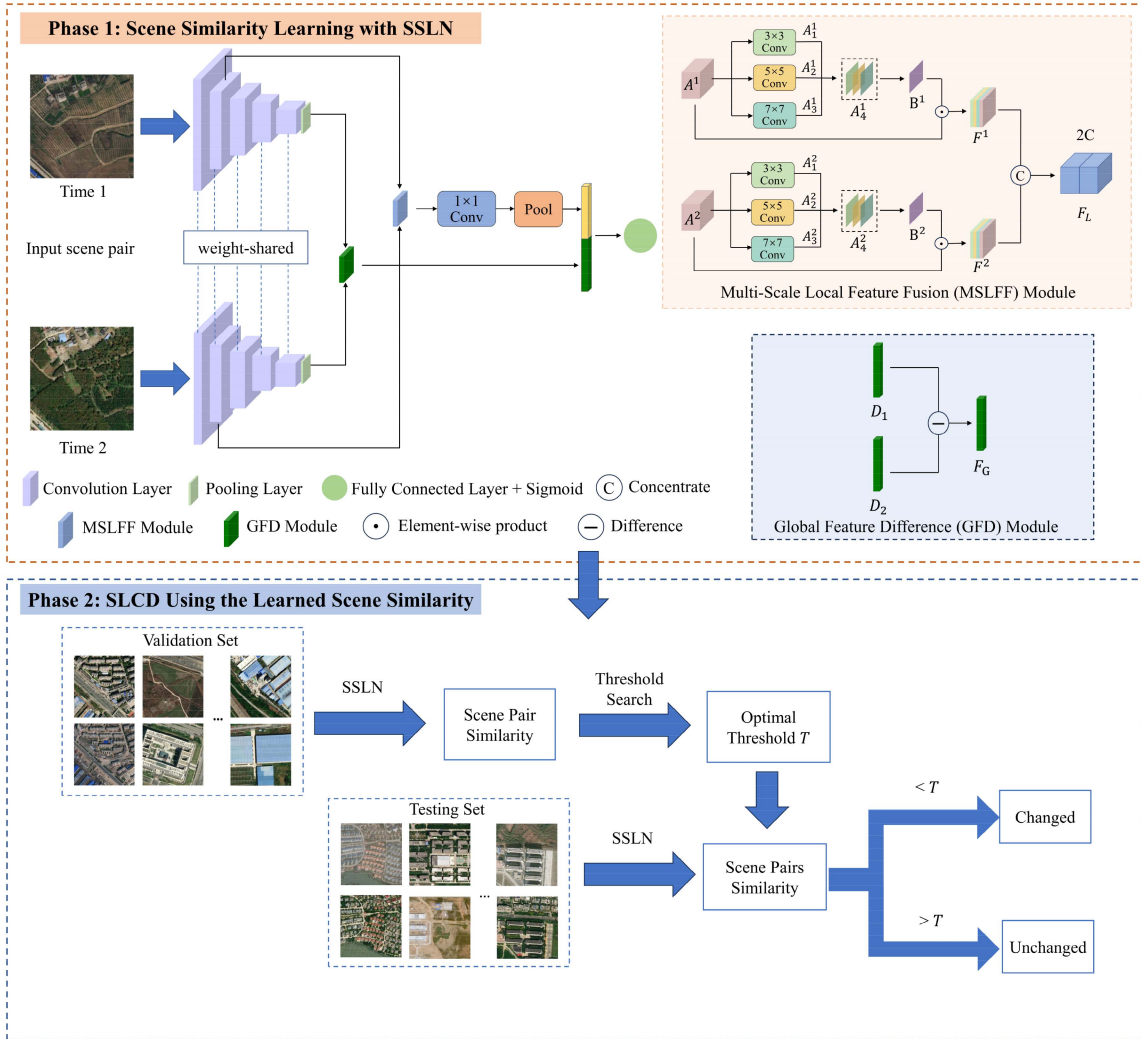


Fig. 3. Flowchart of the proposed SLCD approach based on similarity learning.

#### Algorithm 1: Similarity Threshold Search Approach.

**Input:** Land use scene pair  $X^i$  and corresponding label  $Y^i$  in validation set; The trained SSLN  $g_{w,b}(x)$

**Output:** The optimal similarity threshold  $T$

- 1: For threshold  $t = 0$  to 1, step size = 0.0001
- 2: Calculate the similarity value of each scene pair in validation set with  $s_i = g_{w,b}(X^i)$
- 3: Take  $t$  as the current threshold, if  $t > s_i$ , then the bi-temporal scenes in pair  $X^i$  are considered to have changed, otherwise, no change is considered
- 4: Calculate the overall accuracy (OA) and Kappa coefficient based on the results in step 3 and  $Y^i$
- 5: The best threshold  $T$  is determined as  $t$  when  $t$  achieves the best OA and Kappa
- 6: End

adapt to the imbalance in the number of scene categories,  $T$  is determined with Kappa coefficient when the best values of OA and Kappa conflict with each other.

TABLE V  
NUMBER OF THE CHANGED AND UNCHANGED SAMPLES OF THE LUSCD DATASET

	LUSCD			
	Train	Validation	Test A	Test B
<b>Changed</b>	2359	588	933	878
<b>Unchanged</b>	15749	3938	4145	4184
<b>Sum</b>	18108	4526	5078	5062

## IV. EXPERIMENTAL RESULTS

### A. Dataset

We select MtS-WH [18] and LUSCD to evaluate our proposed approach. MtS-WH is a publicly available dataset consisting of 1050 scene pairs and eight categories: parking lot, WB, sparse house, dense house, residential area, BL, farmland, and industrial area. For our LUSCD dataset, the number of changed and unchanged samples of each subset (i.e., training set, validation set, testing set) is presented in Table V.



## B. Experimental Settings

The batch size is set as 8 and the learning rate is set as  $2e-4$  in our experiments. The number of training epochs is 1050. In addition, the parameters used in comparison methods are consistent with the original literatures. Considering the differences in the color and brightness between the two temporal scenes, data augmentation is used during training, including horizontal and vertical flipping, to avoid overfitting and improve SLCD performance.

To evaluate the performance of our proposed SLCD approach, the unchanged scene pairs and the changed scene pairs are regarded as positive class and negative class, respectively, to obtain the binary confusion matrix. Based on the confusion matrix, several commonly used metrics, including overall accuracy (OA), Kappa coefficient, Precision, Recall (also known as sensitivity), F1-score (F1), and true negative rate (TNR) (also known as specificity), are calculated to evaluate the performance. These six metrics can be calculated as follows:

$$OA = \frac{TP + TN}{TP + FN + FP + TN} \quad (7)$$

$$Kappa = \frac{OA - PRE}{1 - PRE} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (11)$$

$$TNR = \frac{TN}{FP + TN} \quad (12)$$

$$PRE = \frac{(TP + FN)(TP + FP)}{(TP + TN + FP + FN)^2} + \frac{(TN + FP)(TN + FN)}{(TP + TN + FP + FN)^2} \quad (13)$$

where TP, FP, TN, and FN refer to true positives, false positives, true negatives, and false negatives, respectively. PRE represents the sum of the “truth and the product of predicted values” for all categories, divided by the “square of the total number of samples.” Moreover, the numbers of parameters (Params) and floating point operations (FLOPs) are also selected to evaluate computational complexity.

Our proposed approach (SSLN) is compared with three types of methods including CBM, deep learning based approach CorrFusion [29], and OSBM, respectively. For CBM, the nine pretrained CNNs, including AlexNet [37], VGG19 [38], GoogLeNet [39], ResNet101 [34], DenseNet [40], EfficientNet [41], SENet [42], SwinT [43], ViT [44], are trained using the land use scenes in two datasets (bitemporal scenes are combined to constitute the classification dataset), which are then used for extracting CNN features. It is noted that the features are extracted from the first fully connected layer with respect to AlexNet and VGG19, and the last pooling layer with respect to GoogLeNet and ResNet101, which are then used to train SVM classifier.

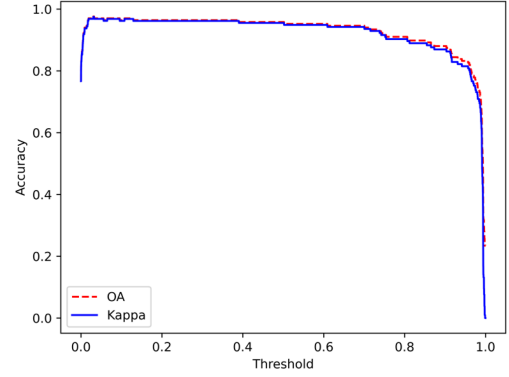


Fig. 4. Accuracy-threshold curve on MtS-WH dataset.

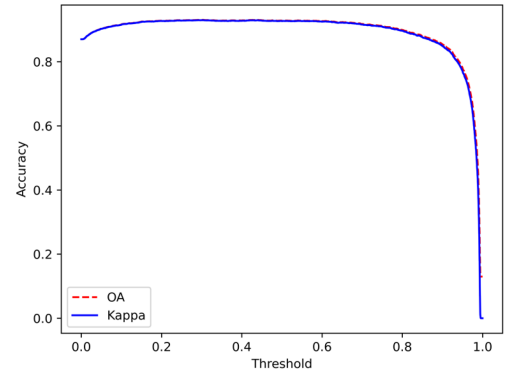


Fig. 5. Accuracy-threshold curve on LUSCD dataset.

For the rest CNNs, the built-in softmax classifier is used for classification. Regarding OSBM, the scene similarity is achieved by calculating the Euclidian distance between the CNN features of bitemporal scenes.

## C. Results and Discussion

According to Algorithm 1, we first draw the accuracy-threshold curves of the two datasets (i.e., MtS-WH and LUSCD), as shown in Figs. 4 and 5, to achieve the optimal similarity threshold of our approach. It is observed that the optimal thresholds  $T$  for MtS-WH and LUSCD are 0.0304 and 0.3297, respectively.

1) *Results on MtS-WH Dataset:* Table VI shows the comparison performance on MtS-WH dataset. As can be observed, our proposed method achieved the best performance in terms of OA, Kappa, Precision, and F1 values. Specifically, the OA, Kappa, Precision, and F1 values are 0.9760, 0.9746, 0.9844, and 0.9844, respectively. It is also obvious that our method and OSBM outperform CBM and CorrFusion in terms of most metrics. This is because change detection performance depends heavily on the classification results. In addition, our approach achieves slightly better performance than OSBM, indicating that the similarity learning process has a positive influence on scene similarity. In terms of Params and FLOPs, our method has higher Params value due to the introduction of more fully connected layers in feature fusion, but has lower FLOPs value than most methods.

TABLE VI  
PERFORMANCE COMPARISON OF DIFFERENT APPROACHES ON MTS-WH DATASET

Method		OA	Kappa	Precision	Recall	F1	TNR	Params (M)	FLOPs (G)
<b>CBM</b>	AlexNet	0.9190	0.8383	0.9505	0.8889	0.9187	<b>0.9510</b>	0.69	57.04
	VGG19	0.9190	0.8918	0.9505	0.8889	0.9187	<b>0.9510</b>	19.17	139.60
	GoogLeNet	0.9190	0.8383	0.9505	0.8889	0.9187	<b>0.9510</b>	1.47	5.61
	ResNet101	0.9381	0.9197	0.9279	0.9537	0.9406	0.9216	7.68	42.52
	DenseNet	0.9286	0.8568	0.9043	0.9630	0.9327	0.8922	5.66	6.97
	EfficientNet	0.9333	0.8666	0.9352	0.9352	0.9352	0.9314	<b>0.18</b>	<b>0.38</b>
	SENet	0.8667	0.8166	0.9348	0.7963	0.8600	0.9412	3.81	26.06
	SwinT	0.9000	0.8663	0.9307	0.8704	0.8995	0.9314	4.27	27.50
	ViT	0.9048	0.8096	0.9231	0.8889	0.9057	0.9216	32.94	171.30
<b>CorrFusion</b>		0.9143	0.8289	0.8704	0.9592	0.926	0.8750	68.42	55.92
<b>OSBM</b>	AlexNet	0.9333	0.9133	0.9273	0.9444	0.9358	0.9216	0.69	57.04
	VGG19	0.9190	0.8992	0.9640	<b>1.0000</b>	0.9270	0.8333	19.17	139.60
	GoogLeNet	0.9286	0.9090	0.8974	0.9722	0.9333	0.8824	1.47	5.61
	ResNet101	0.9143	0.8918	0.8750	0.9722	0.9211	0.8529	7.68	42.52
	DenseNet	0.9190	0.8969	0.8889	0.9630	0.9244	0.8725	5.66	6.97
	EfficientNet	0.9323	0.9381	0.9455	0.9630	0.9541	0.9411	<b>0.18</b>	<b>0.38</b>
	SENet	0.8714	0.8409	0.8189	0.9630	0.8851	0.7745	3.81	26.06
	SwinT	0.9190	0.8956	0.9027	0.9444	0.9231	0.8922	4.27	27.50
	ViT	0.8183	0.7814	0.7379	0.9907	0.8458	0.6275	32.94	171.30
<b>Ours</b>		<b>0.9760</b>	<b>0.9746</b>	<b>0.9844</b>	0.9844	<b>0.9844</b>	0.9487	34.74	28.42

The bold means the best results in terms of each performance metric.

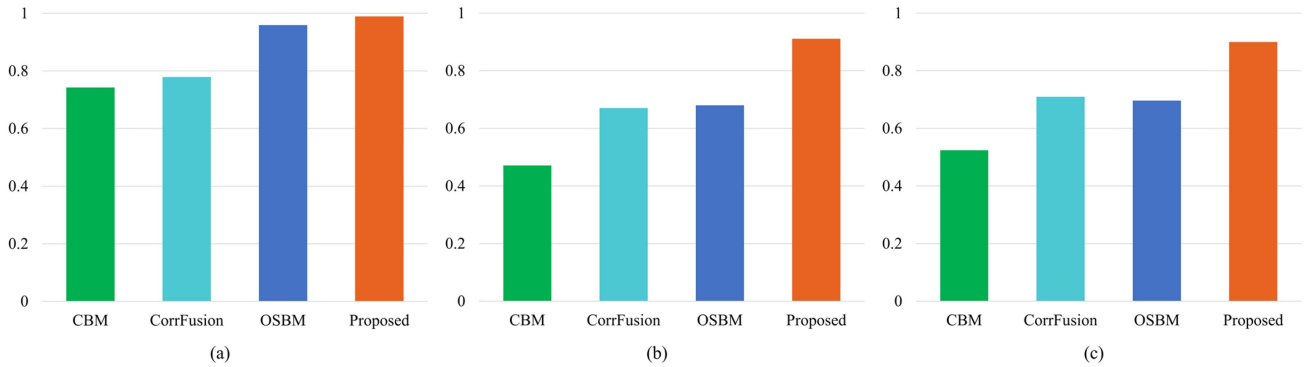


Fig. 6. AUC values on three testing sets. (a) MtS-WH dataset. (b) Test A of LUSCD dataset. (c) Test B of LUSCD dataset.

To further evaluate our proposed approach, the area under curve (AUC) [45] and receiver operating characteristic (ROC) curve are selected for performance evaluation on MtS-WH dataset, as shown in Figs. 6(a) and 7(a), respectively. It can be observed that our method achieves the best AUC value of 0.9891 compared with CBM, CorrFusion, and OSBM. With respect to ROC, the curve of our approach is located on upper left corner, indicating its better performance than other comparison methods.

2) *Results on LUSCD Dataset:* Table VII shows the comparison performance on LUSCD dataset. As can be observed, our proposed similarity learning approach achieves the best performance on LUSCD dataset in terms of OA, Kappa, Recall, and F1. Specifically, OA, Kappa, Recall, and F1 values on Test A are 0.8919, 0.8894, 0.9708, and 0.9359, and are 0.8964, 0.8938, 0.9699, and 0.9383 on Test B, respectively. In addition, as the results presented on MtS-WH dataset, our proposed approach and OSBM also outperform CBM and CorrFusion for most of the



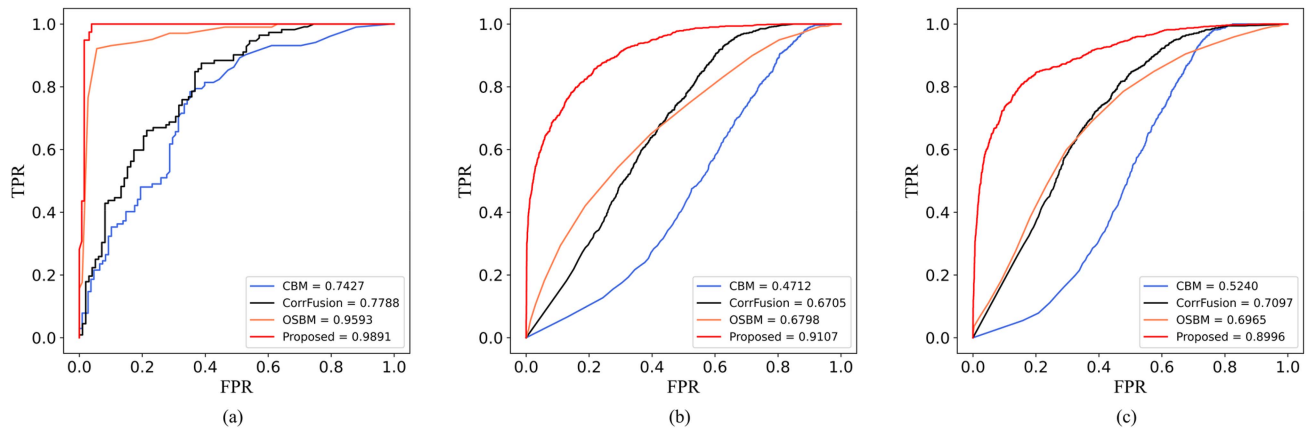


Fig. 7. ROC curves on three testing sets. (a) Mts-WH dataset. (b) Test A of LUSCD dataset. (c) Test B of LUSCD dataset.

TABLE VII  
PERFORMANCE COMPARISON OF DIFFERENT APPROACHES ON LUSCD DATASET

Method		Test A						Test B						Params (M)	FLOPs (G)
		OA	Kappa	Precision	Recall	F1	TNR	OA	Kappa	Precision	Recall	F1	TNR		
CBM	AlexNet	0.6122	0.2494	0.9378	0.7279	0.8197	0.7702	0.6304	0.2357	0.9504	0.6895	0.7992	0.8403	0.69	57.04
	VGG19	0.6268	0.2508	0.9540	0.7530	0.8417	0.8271	0.6557	0.2633	<b>0.9632</b>	0.7505	0.8437	0.8725	19.17	139.60
	GoogLeNet	0.6327	0.2623	0.9403	0.7079	0.8077	0.7861	0.6367	0.2320	0.9546	0.6951	0.8044	0.8531	1.47	5.61
	ResNet101	0.6788	0.3209	0.9516	0.7387	0.8318	0.8214	0.6993	0.3233	0.9629	0.7083	0.8162	<b>0.8789</b>	7.68	42.52
	DenseNet	0.7637	0.7464	0.9606	0.7409	0.8366	0.8650	0.7778	0.7640	0.9553	0.7669	0.8508	0.8294	5.66	6.97
	EfficientNet	0.7363	0.7155	0.9585	0.7076	0.8142	0.8639	0.7513	0.7345	0.9535	0.7349	0.8300	0.8294	<b>0.18</b>	<b>0.38</b>
	SENet	0.7420	0.7221	0.9538	0.7264	0.8247	0.8435	0.7452	0.7275	0.9558	0.7251	0.8246	0.8407	3.81	26.06
	SwinT	0.7759	0.4588	<b>0.9656</b>	0.7522	0.8457	<b>0.8810</b>	0.7596	0.4116	0.9577	0.7418	0.8361	0.8441	4.27	27.50
	ViT	0.6004	0.2174	0.9311	0.5513	0.6925	0.8189	0.6586	0.2529	0.9284	0.6359	0.7548	0.7668	32.94	171.30
<b>CorrFusion</b>		0.6823	0.5361	0.9258	0.5242	0.6720	0.8403	0.6906	0.5444	0.9439	0.5314	0.6800	0.8498	68.42	55.92
OSBM	AlexNet	0.7808	0.2313	0.8282	0.9059	0.8653	0.7039	0.8248	0.3687	0.8167	0.8990	0.8559	0.6957	0.69	57.04
	VGG19	0.7674	0.1669	0.8603	0.9290	0.8933	0.2821	0.7966	0.2475	0.8559	0.9385	0.8953	0.2980	19.17	139.60
	GoogLeNet	0.8062	0.3661	0.8264	0.9049	0.8638	0.7156	0.7861	0.2747	0.8163	0.8988	0.8556	0.7024	1.47	5.61
	ResNet101	0.8620	0.4516	0.8193	0.9035	0.8593	0.7261	0.8469	0.3707	0.8249	0.8852	0.8540	0.7086	7.68	42.52
	DenseNet	0.8163	0.8163	0.8163	0.8529	0.8342	0.7265	0.8251	0.8251	0.8261	0.8642	0.8447	0.7812	5.66	6.97
	EfficientNet	0.8163	0.8163	0.8159	0.8351	0.8254	0.6943	0.8264	0.8264	0.8264	0.8336	0.8300	0.7168	<b>0.18</b>	<b>0.38</b>
	SENet	0.8322	0.8266	0.8936	0.9018	0.8977	0.5230	0.8556	0.8515	0.9052	0.9218	0.9134	0.5404	3.81	26.06
	SwinT	0.8775	0.8735	0.9319	0.9190	0.9254	0.6803	0.8820	0.8774	0.9404	0.9134	0.9267	0.7428	4.27	27.50
	ViT	0.8163	0.8163	0.9425	0.9086	0.9252	0.7208	0.8264	0.8164	0.9315	0.9047	0.9179	0.7610	32.94	171.30
<b>Ours</b>		<b>0.8919</b>	<b>0.8894</b>	0.9035	<b>0.9708</b>	<b>0.9359</b>	0.5391	<b>0.8964</b>	<b>0.8938</b>	0.9086	<b>0.9699</b>	<b>0.9383</b>	0.5358	34.74	28.42

The best results are marked in bold.

metrics on LUSCD dataset. The interesting phenomenon is that both CBM and CorrFusion outperform similarity-based methods (i.e., OSBM and our approach) with respect to the precision and TNR metrics. A possible explanation is that the number of positive scene pairs in both Test A and Test B is much larger than that of the negative scene pairs, thus the change detection results tend to incline toward positive class. Meanwhile, our proposed approach has improved the performance of OSBM by a significant margin, indicating that similarity learning is able to measure scene similarity more accurately. Regarding Params and FLOPs, though our approach has higher Params

and FLOPs values than some of the other comparison methods, it is acceptable considering its improvement of performance.

To further evaluate the performance of our approach on LUSCD, we also draw the AUC histograms and ROC curves on Test A and Test B, as shown in Figs. 6(b) and (c) and 7(b) and (c). The AUC values of our proposed method on Test A and B are 0.9107 and 0.8996, respectively. In addition, as shown in Fig. 7(b) and (c), compared with other comparison methods, the proposed method has more obvious curves on upper left corner on Tests A and B, confirming its validity. For ROC curves, our proposed method is closer to the upper left corner than

TABLE VIII  
CHANGE DETECTION EXAMPLES OF LUSCD DATASET

















Test A	Time 1				
		AL	GL	IL	AL
	Time 2				
		AL	GL	BL	BL
	Ground Truth	Unchanged	Unchanged	Changed	Changed
	Method	CBM (DenseNet)	×	×	✓
	CBM (SwinT)	✓	✓	×	×
	OSBM (DenseNet)	✓	×	✓	×
	OSBM (SwinT)	✓	×	✓	✓
	Ours	✓	✓	✓	✓
Test B	Time 1				
		BL	EL	BL	BL
	Time 2				
		BL	EL	AL	EL
	Ground Truth	Unchanged	Unchanged	Changed	Changed
	Method	CBM (DenseNet)	×	✓	×
	CBM (SwinT)	×	✓	×	✓
	OSBM (DenseNet)	✓	×	✓	✓
	OSBM (SwinT)	✓	✓	×	✓
	Ours	✓	✓	✓	✓

TABLE IX  
ABLATION EXPERIMENTS ON MTS-WH DATASET

Method	OA	Kappa	Precision	Recall	F1	TNR	Params (M)	FLOPs (G)
<b>Base</b>	0.8905	0.8652	0.8295	<b>0.9907</b>	0.9030	0.7843	<b>33.33</b>	<b>27.05</b>
<b>Base + MSLFF</b>	0.8952	0.8653	0.8772	0.9259	0.9009	0.8627	34.48	28.41
<b>Base + GFD</b>	0.9476	0.9333	0.9145	<b>0.9907</b>	0.9511	0.9020	33.42	27.06
<b>SSLN</b>	<b>0.9760</b>	<b>0.9746</b>	<b>0.9844</b>	0.9844	<b>0.9844</b>	<b>0.9487</b>	34.74	28.41

The best results are marked in bold.

the other three methods when FPR is around 0.2, indicating its better change detection performance. Table VIII presents some change detection examples achieved by CBM, OSBM, and our approach. It can be observed that our similarity learning approach performs the best.

According to the above results, we can conclude that our approach has the following advantages.














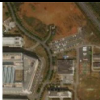
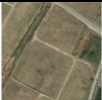

- 1) The similarity is accurately described by fusing local features and global feature difference via MSLFF and GFD modules.
- 2) The similarity threshold is used to directly determine whether one scene pair has changed or not based on scene similarity, thereby avoiding the dependence on the classification results.

TABLE X  
ABLATION EXPERIMENTS ON LUSCD DATASET

Method	Test A						Test B						Params (M)	FLOPs (G)
	OA	Kappa	Precision	Recall	F1	TNR	OA	Kappa	Precision	Recall	F1	TNR		
<b>Base</b>	0.8169	0.8164	0.8188	0.9913	0.8983	0.0418	0.8167	0.8163	0.8218	0.9907	0.9038	0.0466	<b>33.33</b>	<b>27.05</b>
<b>Base + MSLFF</b>	0.8174	0.8172	0.8213	<b>0.9969</b>	0.8991	0.0204	0.8248	0.8246	0.8274	0.9957	0.9043	0.0114	34.48	28.41
<b>Base + GFD</b>	0.8291	0.8284	0.8293	0.9954	0.9048	0.0900	0.8380	0.8375	0.8381	<b>0.9964</b>	0.9104	0.0841	33.42	27.06
<b>SSLN</b>	<b>0.8919</b>	<b>0.8894</b>	<b>0.9035</b>	0.9708	<b>0.9359</b>	<b>0.5391</b>	<b>0.8964</b>	<b>0.8938</b>	<b>0.9086</b>	0.9699	<b>0.9383</b>	<b>0.5358</b>	34.74	28.41

The best results are marked in bold.

TABLE XI  
EXAMPLES OF THE ABLATION EXPERIMENT ON THE PROPOSED METHOD

<b>Test A</b>	<b>Time 1</b>					
			AL	GL	IL	AL
	<b>Time 2</b>					
			AL	GL	BL	BL
<b>Ground Truth</b>		Unchanged	Unchanged	Changed	Changed	
<b>Method</b>	Base	×	✓	✓	×	
	Base + MSLFF	×	✓	✓	×	
	Base + GFD	✓	✓	✓	×	
	SSLN	✓	✓	✓	✓	
<b>Test B</b>	<b>Time 1</b>					
			BL	EL	BL	BL
	<b>Time 2</b>					
			BL	EL	AL	EL
<b>Ground Truth</b>		Unchanged	Unchanged	Changed	Changed	
<b>Method</b>	Base	✓	✓	×	×	
	Base + MSLFF	✓	✓	✓	×	
	Base + GFD	✓	✓	×	✓	
	SSLN	✓	✓	✓	✓	

- 3) Taking objects at different scales into account will contribute to extract powerful features and further learn scene similarity more accurately.

#### D. Ablation Analysis

To analyze the effectiveness of two modules in SSLN, we conduct ablation experiments on both the MSLFF and GFD modules on MtS-WH and LUSCD datasets. In the following experiments, “Base” represents the basic model without MSLFF module and GFD module. “Base + MSLFF” represents the

“Base” with MSLFF module, and “Base + GFD” represents “Base” with GFD module.

As can be seen from Tables IX and X, the introduction of both MSLFF and GFD module can significantly improve the performance (i.e., our approach) on both datasets. More specifically, compared with “Base” model, the OA, Kappa, Precision, F1, and TNR values of our approach have been improved by 8.55%, 10.94%, 15.49%, 8.14%, 16.44% on MtS-WH dataset, whereas the OA, Kappa, Precision, F1, and TNR values of our approach have been improved by 7.50%, 7.30%, 8.47%, 3.76%, 49.73%



on Test A, and 7.97%, 7.75%, 8.68%, 3.45%, 48.92% on Test B of LUSCD dataset, respectively. The interesting phenomenon is that “Base + MSLFF” and “Base + GFD” achieve comparable performance to “Base” on both datasets, indicating that a single module contributes little to the improvement of model performance. Notably, the great improvement of SSLN not only further validates the effectiveness of MSLFF module and GFD module, but also proves the gain effect of their combination.

In addition, the Params and FLOPs values are also provided in Tables IX and X to analyze the computational complexity of different modules. As we can see, the Params of the MSLFF and GFD modules are 1.15M and 0.09M, respectively, indicating that the performance can be effectively improved without introducing a large number of parameters.

Table XI presents some change detection examples of the ablation experiments on the two testing sets of LUSCD dataset. With both MSLFF and GFD modules, SSLN can largely enhance the accuracy of change detection results.

## V. CONCLUSION

In this article, we proposed SSLN for land use SLCD by combining the multiscale local features and GFD, which can overcome the limitations of the existing classification-based SLCD approaches, and improve their performance by a remarkable margin. SSLN takes the bitemporal scene images as input and outputs the learned similarity between scene pairs. The change detection results are achieved by comparing scene similarity and the optimal similarity threshold determined by our threshold search algorithm. Furthermore, we also collect a new benchmark dataset termed LUSCD for performance evaluation, which largely complements the existing SLCD datasets in terms of image resolution, the number of images and categories. The experimental results on LUSCD and one publicly available dataset demonstrated that our similarity learning-based approach is a simple yet effective method for SLCD, providing RSCD literature a promising perspective for developing intelligent SLCD methods. Our future work will focus on exploring scene-level semantic change detection to achieve the change types of land use scenes.

## REFERENCES

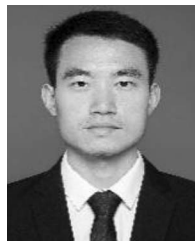
- [1] M. Hussain et al., “Change detection from remotely sensed images: From pixel-based to object-based approaches,” *ISPRS J. Photogramm. Remote Sens.*, vol. 80, pp. 91–106, Jun. 2013, doi: [10.1016/j.isprsjprs.2013.03.006](https://doi.org/10.1016/j.isprsjprs.2013.03.006).
- [2] T. Lei et al., “Difference enhancement and spatial-spectral nonlocal network for change detection in VHR remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Mar. 2022, Art. no. 4507013, doi: [10.1109/TGRS.2021.3134691](https://doi.org/10.1109/TGRS.2021.3134691).
- [3] P. P. De Bem, O. A. de Carvalho Junior, R. Fontes Guimarães, and R. A. Trancoso Gomes, “Change detection of deforestation in the Brazilian Amazon using Landsat data and convolutional neural networks,” *Remote Sens.*, vol. 12, no. 6, Mar. 2020, Art. no. 901, doi: [10.3390/rs12060901](https://doi.org/10.3390/rs12060901).
- [4] Q. Zheng, Q. Weng, and K. Wang, “Characterizing urban land changes of 30 global megacities using nighttime light time series stacks,” *ISPRS J. Photogramm. Remote Sens.*, vol. 173, pp. 10–23, 2021, doi: [10.1016/j.isprsjprs.2021.01.002](https://doi.org/10.1016/j.isprsjprs.2021.01.002).
- [5] X. P. Song et al., “Global land change from 1982 to 2016,” *Nature*, vol. 560, no. 7720, pp. 639–643, 2018, doi: [10.1038/s41586-018-0411-9](https://doi.org/10.1038/s41586-018-0411-9).
- [6] Q. Zhu et al., “Oil spill contextual and boundary-supervised detection network based on marine SAR images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jan. 2022, Art. no. 5213910, doi: [10.1109/TGRS.2021.3115492](https://doi.org/10.1109/TGRS.2021.3115492).
- [7] M. Mahdianpari et al., “A large-scale change monitoring of wetlands using time series Landsat imagery on Google Earth Engine: A case study in Newfoundland,” *GISci. Remote Sens.*, vol. 57, pp. 1102–1124, 2020, doi: [10.1080/15481603.2020.1846948](https://doi.org/10.1080/15481603.2020.1846948).
- [8] F. Bovolo and L. Bruzzone, “A split-based approach to unsupervised change detection in large-size multitemporal images: Application to tsunami-damage assessment,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1658–1670, Jun. 2007, doi: [10.1109/TGRS.2007.895835](https://doi.org/10.1109/TGRS.2007.895835).
- [9] S. Saha, F. Bovolo, and L. Bruzzone, “Unsupervised deep change vector analysis for multiple-change detection in VHR images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3677–3693, Sep. 2019, doi: [10.1109/TGRS.2018.2886643](https://doi.org/10.1109/TGRS.2018.2886643).
- [10] D. Pirrone, F. Bovolo, and L. Bruzzone, “A novel framework based on polarimetric change vectors for unsupervised multiclass change detection in dual-pol intensity SAR images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4780–4795, Feb. 2020, doi: [10.1109/TGRS.2020.2966865](https://doi.org/10.1109/TGRS.2020.2966865).
- [11] J. Wang, Y. Zhong, and L. Zhang, “Change detection based on supervised contrastive learning for high-resolution remote sensing imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Jan. 2023, Art. no. 5601816, doi: [10.1109/TGRS.2023.3236664](https://doi.org/10.1109/TGRS.2023.3236664).
- [12] Q. Shi, M. Liu, S. Li, X. Liu, F. Wang, and L. Zhang, “A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jan. 2022, Art. no. 5604816, doi: [10.1109/TGRS.2021.3085870](https://doi.org/10.1109/TGRS.2021.3085870).
- [13] Z. Lv, H. Huang, W. Sun, M. Jia, J. A. Benediktsson, and F. Chen, “Iterative training sample augmentation for enhancing land cover change detection performance with deep learning neural network,” *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: [10.1109/TNNLS.2023.3282935](https://doi.org/10.1109/TNNLS.2023.3282935).
- [14] J. Barber, “A generalized likelihood ratio test for coherent change detection in polarimetric SAR,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1873–1877, Sep. 2015, doi: [10.1109/LGRS.2015.2433134](https://doi.org/10.1109/LGRS.2015.2433134).
- [15] D. Ciunzo, V. Carotenuto, and A. D. Maio, “On multiple covariance equality testing with application to SAR change detection,” *IEEE Trans. Signal Process.*, vol. 65, no. 19, pp. 5078–5091, Oct. 2017, doi: [10.1109/TSP.2017.2712124](https://doi.org/10.1109/TSP.2017.2712124).
- [16] L. Zhang, X. Hu, M. Zhang, Z. Shu, and H. Zhou, “Object-level change detection with a dual correlation attention-guided detector,” *ISPRS J. Photogramm. Remote Sens.*, vol. 177, pp. 147–160, Jul. 2021, doi: [10.1016/j.isprsjprs.2021.05.002](https://doi.org/10.1016/j.isprsjprs.2021.05.002).
- [17] K. Doi et al., “Detecting object-level scene changes in images with viewpoint differences using graph matching,” *Remote Sens.*, vol. 14, no. 17, Aug. 2022, Art. no. 4225, doi: [10.3390/rs14174225](https://doi.org/10.3390/rs14174225).
- [18] C. Wu, L. Zhang, and L. Zhang, “A scene change detection framework for multi-temporal very high resolution remote sensing images,” *Signal Process.*, vol. 124, pp. 184–197, Oct. 2015, doi: [10.1016/j.sigpro.2015.09.020](https://doi.org/10.1016/j.sigpro.2015.09.020).
- [19] B. Du, Y. Wang, C. Wu, and L. Zhang, “Unsupervised scene change detection via latent Dirichlet allocation and multivariate alteration detection,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 4676–4689, Dec. 2018, doi: [10.1109/JSTARS.2018.2869549](https://doi.org/10.1109/JSTARS.2018.2869549).
- [20] L. Ma et al., “Deep learning in remote sensing applications: A meta-analysis and review,” *ISPRS J. Photogramm. Remote Sens.*, vol. 152, pp. 166–177, Jun. 2019, doi: [10.1016/j.isprsjprs.2019.04.015](https://doi.org/10.1016/j.isprsjprs.2019.04.015).
- [21] L. Zhang, L. Zhang, and B. Du, “Deep learning for remote sensing data: A technical tutorial on the state of the art,” *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016, doi: [10.1109/MGRS.2016.2540798](https://doi.org/10.1109/MGRS.2016.2540798).
- [22] X. Zhu et al., “Deep learning in remote sensing: A comprehensive review and list of resources,” *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017, doi: [10.1109/MGRS.2017.2762307](https://doi.org/10.1109/MGRS.2017.2762307).
- [23] S. Wang, Y. Guan, and L. Shao, “Multi-granularity canonical appearance pooling for remote sensing scene classification,” *IEEE Trans. Image Process.*, vol. 29, pp. 5396–5407, Apr. 2020, doi: [10.1109/TIP.2020.2983560](https://doi.org/10.1109/TIP.2020.2983560).
- [24] B. Pan, X. Xu, Z. Shi, N. Zhang, H. Luo, and X. Lan, “DSSNet: A simple dilated semantic segmentation network for hyperspectral imagery classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 11, pp. 1968–1972, Nov. 2020, doi: [10.1109/LGRS.2019.2960528](https://doi.org/10.1109/LGRS.2019.2960528).
- [25] Z. Lv et al., “Land cover change detection with heterogeneous remote sensing images: Review, progress, and perspective,” *Proc. IEEE*, vol. 110, no. 12, pp. 1976–1991, Nov. 2022, doi: [10.1109/JPROC.2022.3219376](https://doi.org/10.1109/JPROC.2022.3219376).

- [26] H. Fang, S. Guo, X. Wang, S. Liu, C. Lin, and P. Du, "Automatic urban scene-level binary change detection based on a novel sample selection approach and advanced triplet neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Jan. 2023, Art. no. 5601518, doi: [10.1109/TGRS.2023.3235917](https://doi.org/10.1109/TGRS.2023.3235917).
- [27] H. Fang et al., "Scene change detection by differential aggregation network and class probability-based fusion strategy," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Sep. 2023, Art. no. 5406918, doi: [10.1109/TGRS.2023.3317701](https://doi.org/10.1109/TGRS.2023.3317701).
- [28] H. Fang et al., "Scene-level change detection by integrating VHR images and POI data using a multiple-branch fusion network," *Remote Sens. Lett.*, vol. 14, no. 8, pp. 808–820, Jul. 2023, doi: [10.1080/2150704X.2023.2242588](https://doi.org/10.1080/2150704X.2023.2242588).
- [29] L. Ru, B. Du, and C. Wu, "Multi-temporal scene classification and scene change detection with correlation based fusion," *IEEE Trans. Image Process.*, vol. 30, pp. 1382–1394, Nov. 2020, doi: [10.1109/TIP.2020.3039328](https://doi.org/10.1109/TIP.2020.3039328).
- [30] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 8004–8013, doi: [10.1109/CVPR.2018.00835](https://doi.org/10.1109/CVPR.2018.00835).
- [31] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 4353–4361, doi: [10.1109/CVPR.2015.7299064](https://doi.org/10.1109/CVPR.2015.7299064).
- [32] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 118–126, doi: [10.1109/ICCV.2015.22](https://doi.org/10.1109/ICCV.2015.22).
- [33] J. Yuan, L. Ru, S. Wang, and C. Wu, "WH-MAVS: A novel dataset and deep learning benchmark for multiple land use and land cover applications," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 1575–1590, Jan. 2022, doi: [10.1109/JSTARS.2022.3142898](https://doi.org/10.1109/JSTARS.2022.3142898).
- [34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778, doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [35] S. Wang, C. Yuan, and C. Zhang, "LPE-Unet: An improved UNet network based on perceptual enhancement," *Electronics*, vol. 12, no. 12, Jun. 2023, Art. no. 2750, doi: [10.3390/electronics12122750](https://doi.org/10.3390/electronics12122750).
- [36] X. Zhang et al., "ADHR-CDNet: Attentive differential high-resolution change detection network for remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Nov. 2022, Art. no. 5634013, doi: [10.1109/TGRS.2022.3221492](https://doi.org/10.1109/TGRS.2022.3221492).
- [37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, no. 6, pp. 1097–1105, Jan. 2012, doi: [10.1145/3065386](https://doi.org/10.1145/3065386).
- [38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, Apr. 2015, doi: [10.48550/2015.arXiv.1409.1556](https://doi.org/10.48550/2015.arXiv.1409.1556).
- [39] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 1–9, doi: [10.1109/CVPR.2015.7298594](https://doi.org/10.1109/CVPR.2015.7298594).
- [40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 4700–4708, doi: [10.1109/CVPR.2017.243](https://doi.org/10.1109/CVPR.2017.243).
- [41] M. Tan and L. Quoc, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2019, pp. 6105–6114, doi: [10.48550/2019.arXiv.1905.11946](https://doi.org/10.48550/2019.arXiv.1905.11946).
- [42] J. Hu, S. Li, and S. Gang, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141, doi: [10.1109/CVPR.2018.00745](https://doi.org/10.1109/CVPR.2018.00745).
- [43] Z. Liu et al., "Swin transformer: Hierarchical Vis. transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, Mar. 2021, pp. 10012–10022, doi: [10.1109/ICCV48922.2021.00986](https://doi.org/10.1109/ICCV48922.2021.00986).
- [44] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, doi: [10.48550/2021.arXiv.2010.11929](https://doi.org/10.48550/2021.arXiv.2010.11929).
- [45] J. A. Hanley and B. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982, doi: [10.1148/radiology.143.1.7063747](https://doi.org/10.1148/radiology.143.1.7063747).



**Jinglei Liu** received the B.S. degree in surveying and mapping engineering from the School of Information Science and Engineering, Shandong Agricultural University, Tai'an, China, in 2022. He is currently working toward the M.S. degree in resources and environment with the School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing, China.

His current research focuses on the change detection with deep learning.



**Weixun Zhou** (Member, IEEE) received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2019.

He is currently a Lecturer with the School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing, China. His research interests include intelligent extraction of remote sensing information and urban remote sensing application.



**Haiyan Guan** (Senior Member, IEEE) received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2009, and the Ph.D. degree in geomatics from the University of Waterloo, Waterloo, ON, Canada, in 2014.

She is currently a Professor with the School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing, China. She has authored/coauthored more than 50 research papers in refereed journals, books, and proceedings, including IEEE TRANSACTIONS ON

GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, *ISPRS Journal of Photogrammetry and Remote Sensing*, and IEEE International Geoscience and Remote Sensing Symposium & International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Her research interests include information extraction from LiDAR point clouds and from earth observation images.



**Wenzhi Zhao** received the Ph.D. degree in cartography and geography information system from Peking University, Beijing, China, in 2018.

He is currently an Associate Professor with Beijing Normal University, Beijing. His research interests include remote sensing Big Data, spatial-temporal data mining, and machine learning, especially deep networks and their applications in remote sensing.