remote sensing

MDPI

Article

Beyond Classifiers: Remote Sensing Change Detection with Metric Learning

Yuqi Zhang, Wei Li, Yaohua Wang, Zhibin Wang * and Hao Li

DAMO Academy, Alibaba Group, Beijing 100102, China
* Correspondence: zhibin.waz@alibaba-inc.com

Abstract: For change detection in remote sensing images, supervised learning always relies on bi-temporal images as inputs and 2-class/multi-class classifiers as outputs. On the other hand, change detection can be viewed as a metric learning problem, i.e., changed areas should be dissimilar while unchanged areas should be similar. In this paper, we study several metric learning formulations for change detection. A strong baseline is achieved by training on pair-wise images with Reverted Contrastive Loss (RCL) with hard mining. Motivated by the success of triplet loss, we seek two sources of triplet pairs from the bi-temporal images, and a novel Spatial–Temporal Triplet Loss (STTL) is proposed. The proposed triplet loss is further validated on semantic change detection, where semantic labels are provided for the changed areas. The experimental results prove state-of-the-art performance on both binary and semantic change detection.

Keywords: change detection; metric learning; triplet loss

Citation: Zhang, Y.; Li, W.; Wang, Y.; Wang, Z.; Li, H. Beyond Classifiers: Remote Sensing Change Detection with Metric Learning. *Remote Sens.* **2022**, *14*, 4478. https://doi.org/10.3390/rs14184478

Academic Editor: Jon Atli Benediktsson

Received: 13 July 2022
Accepted: 6 September 2022
Published: 8 September 2022

1. Introduction

Change Detection (CD) in remote sensing aims to find surface changes between different phases. CD is essential for various real-world applications, such as environment monitoring [1], urban management [2], damage assessment [3] and crop monitoring [4]. Semantic change detection (SCD) goes further than CD with additional semantic labels. SCD predicts not only the location but the change type as well, e.g., a building changed into a playground. Generally speaking, change detection relies on bi-temporal images as input.

Current change detection methods often take the framework of deep neural networks. More specifically, Siamese Networks [5] with shared parameters are used to extract multi-scale features for bi-temporal images. Then, these bi-temporal features are concatenated and put through a decoder and a classifier to output binary or semantic maps. These Siamese networks have shown great success in recent years since they consider spatial–temporal relationships simultaneously [2,5–7].

Metric Learning aims to learn a representation function that maps objects into an embedding space. The distance in the space should preserve the similarity of objects—similar objects move close and dissimilar objects move far away [8]. In other words, metric learning pulls objects of the same class closer and objects of different classes further. Metric learning has been successful in face recognition and person re-identification to learn feature representations instead of classifiers [9,10]. In face recognition, LFW (Labeled Faces in the Wild) dataset [11] evaluates 6000 pairs of faces with annotations of the same person and different persons. Instead of the classifiers to judge the per-pixel class for change detection, we emphasize that the per-pixel class can be predicted in a metric learning way.

Metric learning-based change detection has been studied in some previous works [2,7,12,13]. The feature extraction module is similar to classifier-based methods by a Siamese Network with shared parameters. Instead of using classifiers to output class labels, an embedding space is optimized where similar (unchanged) samples are pulled closer while

dissimilar (changed) areas are pushed apart. Contrastive Loss [9] has been used with standalone optimizations [14–16] or bi-temporal optimization [7]. Triplet loss [10] is introduced in [12,13] to model the distance among triplet pairs.

However, current metric-based methods suffer from some problems. **(1) Data imbalance**. Changed areas only cover a small portion of the data. Therefore, a large number of unchanged areas are easily classified, and we name these pixels easy pixels. On the other hand, pixels that confuse the change detection methods are essential, and we name these as hard pixels. In works [2,7], the authors use a balanced contrastive loss to equally treat changed and unchanged regions. In other words, easy pixels are treated equally with hard pixels. Counting too many easy pixels in the total loss is not beneficial for the overall training. Those hard pixels with larger loss should be emphasized. **(2) Source of triplet pair**. Although some methods [12,13] utilized triplet loss for change detection, those triplet pairs still come from data augmentation of bi-temporal images. We emphasize that the triplet pair collection is essential and needs to be carefully designed.

In this paper, we study several metric learning formulations for change detection. Figure 1 illustrates the proposed different types of metric learning frameworks. We propose a strong baseline by a modified contrastive loss named Reverted Contrastive Loss (RCL) with hard mining. The baseline could ease data imbalance and make hard mining applicable to change detection. Although other methods [2,7] remove classifiers, the performance is relatively low due to data imbalance. Based on the strong baseline, we further introduce two sources of triplet pairs. Since the triplet pairs are acquired spatially and temporally, we name it Spatial–Temporal Triplet Loss (STTL). The experimental results prove state-of-the-art performance on both binary and semantic change detection.

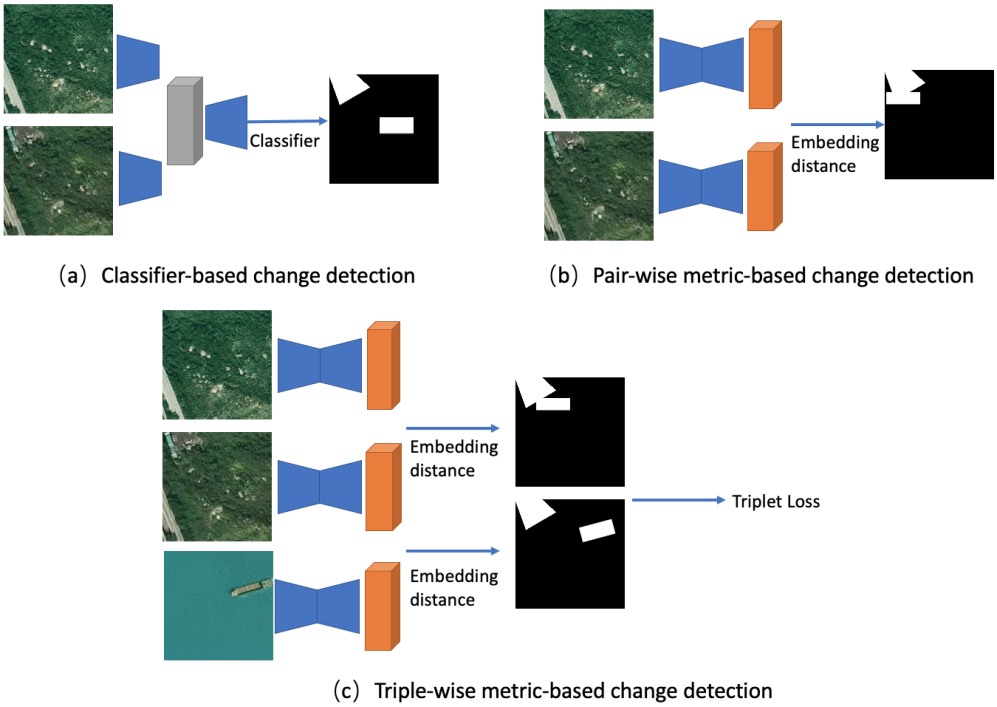

Figure 1. Illustration of classifier-based and metric-based change detection methods. (**a**) Traditional classifier-based change detection uses a classifier to output changed/unchanged regions. On the other hand, Metric-based methods use pixel-wise embedding distance as a metric, and the model training optimizes the embedding directly. (**b**) Pair-wise methods take paired images as input and compute contrastive loss. (**c**) Triple-wise methods take an additional image (the third row with different content) and have seldom been studied. How we use the extra image in triplet loss makes our method different from others. Details are explained in Section 3.4.

The main contributions of our work can be summarized as follows:

- We remove classifiers in change detection networks and propose a strong baseline with modified contrastive loss.
- We improve the contrastive loss with triplet loss by searching for triplet pairs in changed and unchanged regions. We further transfer triplet metric learning to semantic change detection. Since multiple classes are provided, we conduct more triplet pairs. To our knowledge, this is the first time triplet loss has been used in change detection both spatially and temporally.
- Extensive experiments have confirmed the effectiveness of our contrastive loss baseline and triplet loss in binary and semantic change detection.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces our improved contrastive loss as a strong baseline and the proposed Spatial–Temporal Triplet Loss (STTL). Section 4 provides the experiment analysis of our methods on the benchmark datasets. Section 5 discusses the values and potential use of our method. Section 6 concludes this work with the future direction.

2. Related Work

Change detection can be categorized into binary and semantic change detection. As a common trend in the literature [17], semantic change detection is often a combination of binary and semantic branches. Thus, binary change detection is fundamental and mainly studied in our paper. We also compare some semantic change detection methods to explain the effectiveness of our metric-based method. Furthermore, since metric learning has been less studied in change detection, we explain its wide use in the face recognition/person re-identification field to involve more background knowledge.

2.1. Binary Change Detection

Binary change detection aims at predicting changed or unchanged for every pixel [18]. With the development of modern computer techniques, binary change detection has been growing rapidly [19–22].

2.1.1. Feature Extraction

Feature extraction aims to extract spatial–temporal features from change detection images. The step is fundamental since both classifier-based and metric-based methods rely on the feature extraction module. In [20], 2D CNN is proposed to learn spectral–spatial features. To introduce temporal information, a recurrent CNN is used in [23] to learn spectral–spatial–temporal features. Additionally, methods in [21] introduce 3D CNN for change detection. However, these methods fail to extract the exact temporal relationship between images. Recent binary change detection methods often deploy Siamese Networks. FC-Siam [5] is the very first classic method. As shown in Figure 1a, FC-Siam introduces a shared encoder to extract deep features and then uses a different decoder to upsample the feature map. With the extracted features, classifiers or metric learning can be used. In this paper, we do not modify the feature extraction module and only improve the latter part, namely the metric-learning module. Thus we compare it with other classifier-based/metric-based methods.

Some works also concentrate on heterogeneous change detection to detect changes between heterogeneous images [14,24–26]. A CNN named symmetric convolutional coupling network (SCCN) for heterogeneous optical and SAR images is proposed in [14]. In [24], two novel networks are proposed with an affinity-based change prior learned from heterogeneous data. Li et al. [25] translate optical images into the SAR domain and then perform change detection with the SAR images. The heterogeneous change detection needs extra modifications on the feature extraction module. In this paper, we only focus on RGB remote sensing images and thus follow the classic Siamese Networks of FC-Siam.

2.1.2. Classifier-Based Methods

Many works treat change detection as a classification task and add classifiers after the feature extraction module. FC-Siam [5] uses two-class classifiers to judge foreground/background pixels. Peng et al. [27] improve the original UNet++ and prove beneficial for change detection. To fuse bi-temporal information, methods in [28,29] take features from Siamese Networks as input and then output the change prediction. To solve pseudo changes from seasonal transitions in change detection, the authors in [30] propose a metric learning-based generative adversarial network (MeGAN). The proposed GAN network can learn seasonal invariant feature representations. Chen et al. [6] introduce a transformer architecture for change detection. Bandara et al. [31] modify the transformer architecture to perform a pure transformer for change detection. These methods rely on a two-class classifier to clarify whether each pixel is changed or unchanged. However, hard pixels may be misled since they are close to the classifier decision boundary. On the other hand, metric-based methods remove the classifiers and better fit the change detection definition.

2.1.3. Metric-Based Methods

Metric-based methods have been less studied compared to classifier-based methods. Zhan et al. [15] propose a Siamese Network to extract bi-temporal features with a weighted contrastive loss. STANet [7] introduces batch-balanced contrastive loss (BCL) with a better feature extraction module by spatial–temporal attention. DSAMNet [2] uses the same BCL loss with deeply supervised attention in the middle layers. Although these methods remove the classifiers, they may have data imbalance issues since too many easy pixels are calculated in the loss function.

Some methods [12,13] use the triplet formulation, but the triplet pair formulation is somewhat simple. TBSRL [12] searches triplet pairs with only binary labels rather than semantic labels. For example, a building might serve as the anchor sample while a water region as the positive sample. Although these two samples differ semantically, they might be labeled as changed regions in binary change detection. HRTNet [13] introduces the difference map between images to formulate a triplet network for better temporal feature learning. We emphasize that imperfect triplet pairs may hurt the performance of triplet loss and a better sampling strategy is needed.

2.1.4. Contrastive Learning Methods

Apart from supervised change detection, there are some works on unsupervised, weakly supervised or self-supervised change detection [32–35,35–38]. Hou et al. [32] incorporate neural networks and low-rank decomposition to predict saliency maps for large change probabilities. Saha et al. [33] propose an unsupervised change vector analysis model in high-resolution images. The method is also used for SAR images after they are converted into optical-like features in [34]. Zheng et al. [35] propose the semi-supervised learning for change detection by exploiting object changes in unpaired images. Wu et al. [39] propose a fully convolutional change detection framework with generative adversarial networks. The unsupervised, weakly supervised, regionally supervised, and fully supervised change detection tasks are combined into one framework. Some self-supervised methods also share similar loss formulations as ours, e.g., contrastive learning. Despite the similar formulation, self-supervised methods focus on learning representations without labeled data. We are focusing on fully supervised methods in this paper.

2.2. Semantic Change Detection

Sometimes the change type also needs to be known, such as urbanization, deforestation, and seasonal changes. SCD [40,41] aims to detect the changes and classify the land-cover/land-use (LCLU) transition types. It has been proved that directly comparing the LCLU classification maps may omit the temporal correlation between the two temporal images. Recent dominating SCD methods decouple the problem into binary and semantic change detection branches. The binary branch outputs binary change regions and the

semantic branch outputs the semantic type of each image. Daudt et al. [42] introduce triplet branches with two branches to segment temporal images into LCLU maps and the CD branch to detect the difference information. Yang et al. [17] extend the triple-branch framework by using asymmetric decoders in the feature representations. Ding et al. [43] propose Bi-temporal Semantic Reasoning Network (Bi-SRNet) with two types of semantic reasoning blocks to reason single-temporal and cross-temporal semantic correlations.

2.3. Metric Learning

Metric learning aims to pull closer objects of the same class and push farther objects of different classes. It has been very successful in face recognition [44], person re-identification [45–48], vehicle re-identification [49–52], tracking [53–55], and image retrieval [56].

Learning features purely by softmax loss may lose discrimination [57]. Many metric learning methods [58–61] focus on modifying classic softmax loss. Wen et al. [57] simultaneously learn a center from the deep features of each class and penalize the distances between the deep features and their corresponding class centers. Liu et al. [58] propose angular softmax (A-Softmax) loss to enable CNNs to learn angularly discriminative features. The method imposes discriminative constraints on a hypersphere manifold. Deng et al. [60] propose Additive Angular Margin Loss (ArcFace) with a clear geometric interpretation to obtain highly discriminative features for face recognition. The method has exact correspondence to geodesic distance on a hypersphere. Although well studied, these methods still rely on classifiers and thus are not used in our methods.

Besides modifications on softmax loss, metric learning methods also optimize directly on distances, including contrastive loss [9], triplet loss [10], or other ranking losses. Contrastive loss [9] learns a globally coherent non-linear function that maps the data evenly to the output manifold. Only neighborhood relationships are used without needing any distance measure in the input space. Triplet loss steps further with one more sample to conduct a triplet pair. An image (anchor) of a specific class should be closer to all other images (positive) of the same class than it is to any image (negative) of any other class. Since the setting is closely related to change detection, we borrow knowledge from contrastive loss and triplet loss.

3. Change Detection by Metric Learning

3.1. Framework

Figure 2 illustrates our general framework for change detection. The standalone images go through a shared encoder to extract features. We select multi-scale feature maps with 1/4, 1/8, 1/16, and 1/32 shapes of the original image. The feature maps from each scale are concatenated and then fed into a decoder. The decoder aggregates the multi-scale features to upsample the resolution and outputs a feature embedding of dimension D. Unlike classifier-based methods, which use a two-class classification layer to map the feature embedding into two classes, we directly compute the Euclidean distance among these feature maps. For example, for an input image with a size of (H, W, 3), the output embedding is (H/4, W/4, D). This step can be viewed as semantic segmentation without label supervision for a single image. Then the supervision locates on the output embedding (H/4, W/4, D).

During training, the embeddings are pulled together or pushed away by the supervision of different loss functions (contrastive loss or triplet loss). During inference, the bi-temporal embeddings are calculated the cosine similarity and then converted into a change probability to indicate changed or unchanged. The change probability map is then upsampled four times to output per-pixel predictions. We will introduce the basic concept in the next section.

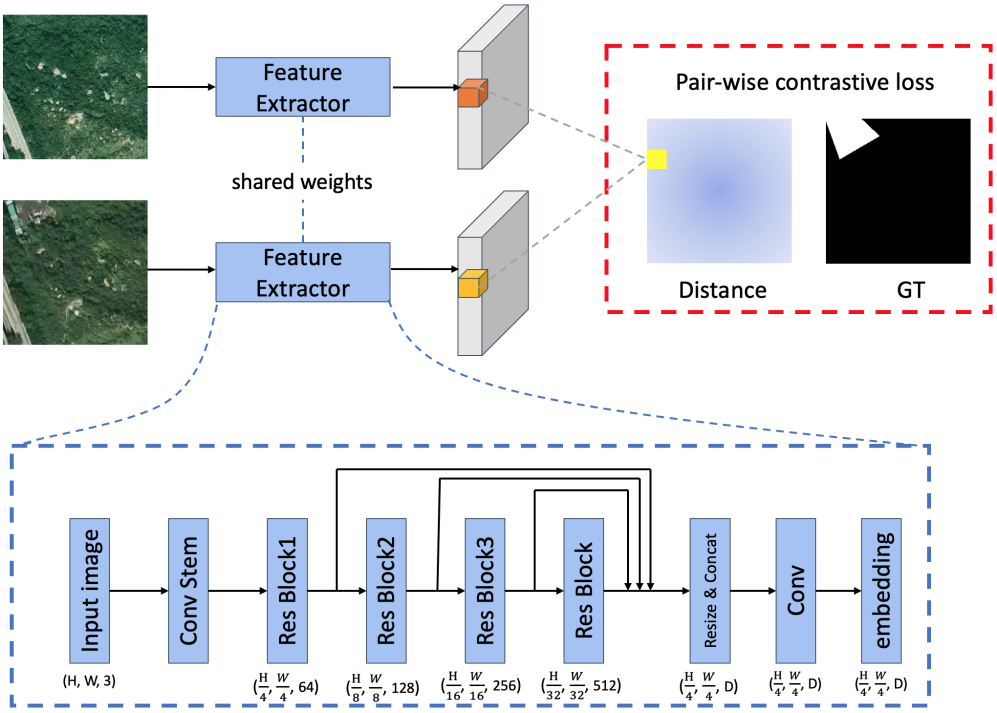

Figure 2. The structure of the proposed baseline networks with contrastive loss. With embeddings from pair-wise/triple-wise images, we compute different loss functions on them.

3.2. Basic Concept

3.2.1. Definition

For a pair of images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ with shape (H, W, 3), the feature embedding $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ have the shape (H/4, W/4, D). As a common step in metric learning, we l2-normalize these feature vectors along the channel dimension with

$$f_{k,i,j} = \boldsymbol{F}_{k,i,j} / (\sum_{p=1}^{D} \boldsymbol{F}_{p,i,j} \cdot \boldsymbol{F}_{p,i,j}) \tag{1}$$

where $\boldsymbol{F}_{k,i,j}$ stands for kth channel raw feature value at (i, j) position and $f_{k,i,j}$ for l2-normalized feature ($0 < i <$ H/4, $0 < j <$ W/4).

The pixel-wise cosine similarity can be calculated as

$$\boldsymbol{s}_{i,j}^{1,2} = \sum_{k=1}^{D} f_{k,i,j}^{1} \cdot f_{k,i,j}^{2} \tag{2}$$

where $\boldsymbol{s}_{i,j}^{1,2}$ stands for similarity at (i, j) position between two images ($0 < i <$ H/4, $0 < j <$ W/4).

With the normalized feature, the Euclidean distance can be calculated with

$$\boldsymbol{d}_{i,j}^{1,2} = \sqrt{\sum_{k=1}^{D} (f_{k,i,j}^{1} - f_{k,i,j}^{2})^2} = 2 - 2\boldsymbol{s}_{i,j}^{1,2} \tag{3}$$

The change score can then be written by linear transformation as

$$\boldsymbol{p}_{i,j}^{1,2} = 1 - 0.5(\boldsymbol{s}_{i,j}^{1,2} + 1) = 0.5(1 - \boldsymbol{s}_{i,j}^{1,2}) \tag{4}$$

The larger $\boldsymbol{p}_{i,j}^{1,2}$ indicates high a probability of change.

During inference, we use $\boldsymbol{p}_{i,j}^{1,2} > 0.5$ as changed area while $\boldsymbol{p}_{i,j}^{1,2} <= 0.5$ as unchanged. The problem then goes to optimizing $\boldsymbol{p}_{i,j}^{1,2}$ during training. There are several ways to optimize distances among data. We omit the phase number and pixel location for typical cases of paired bi-temporal images. We write $\boldsymbol{d}$, $\boldsymbol{s}$, and $\boldsymbol{p}$ in short for Euclidean distance, cosine similarity, and the prediction probability.

3.2.2. Loss Functions

Contrastive Loss [9] aims to pull similar samples closer and push dissimilar samples further. The formulation is:

$$L_{con} = \frac{1}{2} y d^2 + \frac{1}{2}(1-y)\max(m-d,0)^2 \tag{5}$$

$$y = \begin{cases} 1 & \text{similar} \\ 0 & \text{dissimilar} \end{cases} \tag{6}$$

where y indicates whether two samples are similar or dissimilar. $m > 0$ is the margin and d is the Euclidean distance between the two samples.

The above contrastive loss only considers the pair-wise relationship. Sometimes pulling samples of the same class too strictly may hurt the generalization. Triplet loss provides relaxation by introducing one more sample. A sample (anchor) to samples from the same class (positive) should be closer than those from different classes (negatives). For example, in face recognition, two samples with ID = 1 should be pulled closer, while being pushed away from images with ID = 2. Triplet loss can be written as:

$$L_{trp} = \left[d_{ap}^2 - d_{an}^2 + \alpha \right]_+ \tag{7}$$

where d_{ap} is the distance between the anchor and the positive. Similarly, d_{an} is the Euclidean distance between the anchor and the negative. α is the margin of triplet loss. The relaxation helps to improve the overall performance of inter-class and intra-class feature representation.

The symbols in the above loss functions are all scalars. We use vectors or tensors for change detection since there are multiple samples (locations). Due to the characteristic of change detection, where changed areas are often a lot smaller than unchanged, we first discuss several formulations for contrastive loss as the baseline in Section 3.3. We then explain the proposed triplet loss in Section 3.4.

3.3. Contrastive Loss Baseline

3.3.1. Naive Contrastive Loss

Following the naive observation that changed pixels are dissimilar and unchanged pixels are similar, the naive contrastive loss can be written as

$$\boldsymbol{y}_{i,j} = \begin{cases} 1 & \text{unchanged} \\ 0 & \text{changed} \end{cases} \tag{8}$$

Note that Equation (8) is a per-location explanation of label y for Equation (6).

Since unchanged regions cover a large portion of areas, the naive formulation differs from the original contrastive loss setting, where negative samples are a lot more. To solve the data imbalance, we consider several modifications.

3.3.2. Balanced Contrastive Loss

Balanced Contrastive Loss (BCL) is introduced in change detection by [7]. The label definition is the same as Equation (8). The loss for changed and unchanged regions are averaged by their pixel numbers. In other words, the two regions contribute equally to the final loss regardless of their areas.

3.3.3. Probability Contrastive Loss

Since p in Equation (4) is used to judge change detection at inference time, another formulation optimizes it directly as follows:

$$L_{con} = \frac{1}{2}\sum_{i,j} \boldsymbol{y}_{i,j}\boldsymbol{p}_{i,j}^2 + \frac{1}{2}\sum_{i,j}(1-\boldsymbol{y}_{i,j})\max(m-\boldsymbol{p}_{i,j},0)^2 \tag{9}$$

$$\boldsymbol{y}_{i,j} = \begin{cases} 1 & \text{changed} \\ 0 & \text{unchanged} \end{cases} \tag{10}$$

The label $\boldsymbol{y}$ and distance $\boldsymbol{p}$ have different meanings than the naive formulation in Equation (8). The positive samples are from changed regions, which share the same setting as the traditional contrastive loss.

3.3.4. Revert Contrastive Loss

Another way to solve the imbalance of changed and unchanged regions is to create a revert Euciledan distance as

$$\boldsymbol{R}_{i,j} = 2 - \boldsymbol{d}_{i,j} \tag{11}$$

The formulation reverts the Euclidean distance and guarantees the value range of 0–2. Then the original contrastive loss can be written as:

$$L_{con} = \frac{1}{2}\sum_{i,j} \boldsymbol{y}_{i,j}\boldsymbol{R}_{i,j}^2 + \frac{1}{2}\sum_{i,j}(1-\boldsymbol{y}_{i,j})\max(m-\boldsymbol{R}_{i,j},0)^2 \tag{12}$$

$$\boldsymbol{y}_{i,j} = \begin{cases} 1 & \text{changed} \\ 0 & \text{unchanged} \end{cases} \tag{13}$$

3.3.5. Hard Mining

Hard negative mining is effective in face recognition and person re-identification. We use hard mining to select those negative pixels with the larger loss for gradient computation. Let C be the changed pixel number in the batch. We sort the pixel-wise loss by descending order for negative pixels and only select the top-U pixels. To balance positive/negative samples, we keep U:C = 3:1. The hard mining strategy shares a similar motivation with previous balanced contrastive loss. The difference is that we ignore simple negative samples while balanced contrastive loss considers all pixels.

3.3.6. Overall Loss Function

Despite different formulations, the contrastive loss can be split into two parts: losses for changed and unchanged regions. We summarize the overall loss function as follows:

$$L_{con} = L_{con}^{c} + L_{con}^{u} \tag{14}$$

where L_{con}^{c} stands for contrastive loss in changed regions and L_{con}^{u} in unchanged regions.

The main difference from the balanced contrastive loss (BCL) in the previous literature [2,7] is how we design the labels for changed/unchanged regions. BCL follows the naive formulation to set $y = 1$ as unchanged regions. Due to the formulation, it only averages changed/unchanged pixels and fails to perform hard mining. On the other hand, the proposed contrastive loss baseline sets $y = 1$ for changed regions. This better matches the original contrastive loss formulation in Equation (5) and enables hard mining.

3.4. Triplet Loss

It is natural in metric learning to update contrastive loss into triplet loss. The motivation is that similar pairs do not need to be very close; they only need to be closer than other negative pairs by a certain margin. For face recognition or person re-identification, there

are many classes where triplet pairs are easy to collect. However, we can hardly find data with multiple classes in change detection. In the following sections, we introduce how we collect two sources of triplet pairs. Since the two sources of triplet pairs come from both spatial and temporal regions, we name it Spatial–Temporal Triplet Loss (STTL).

3.4.1. Triplet Loss in Changed Region

As illustrated in Figure 3a, the regions are likely to be in one category for the changed contour in one phase. There is intra-consistency in the changed contour in the same phase. In other words, two pixels in the changed contour with the same phase should be closer. We thus formulate triplet pairs in the changed region as:

- Anchor: pixel embedding at location (i, j) in the changed contour from phase 1.
- Positive: pixel embedding at location (e, f) in the changed contour from phase 1.
- Negative: pixel embedding at location (i, j) in the changed contour from phase 2.

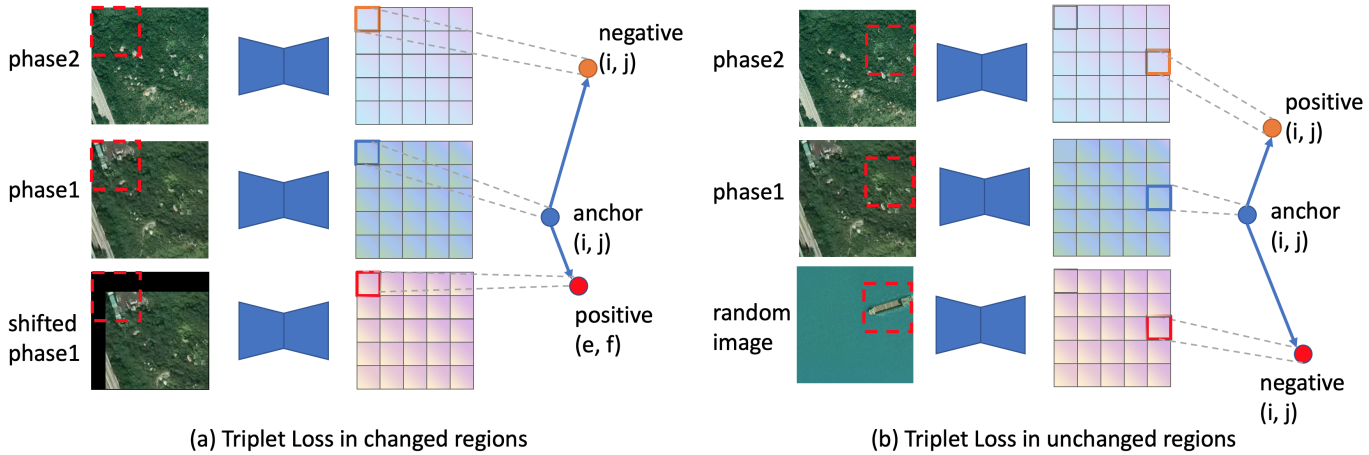

Figure 3. The two sources of triplet pairs of the proposed method.

We then explain the details of finding these triplet pairs. Positive location (e, f) is randomly sampled in the same contour with (i, j) from the same phase 1. (e, f) does not cover with (i, j). The naive implementation is to seek contours on (H/4, W/4, D) feature embedding maps. We step further by shifting the embedding map a little. Then the same position (i, j) in the shifted embedding map with changed annotation could serve as (e, f).

To cope with the revert definition in Section 3.3.4, we define the triplet loss as:

$$L_{trp}^{c} = \sum_{i,j} \left[\boldsymbol{R}_{i,j}^{2} - \boldsymbol{R}_{an}^{2} + m_c \right]_{+} \tag{15}$$

$$\boldsymbol{R}_{an} = 2 - \sqrt{\sum_{k=1}^{D} (f_{k,i,j}^{1} - f_{k,e,f}^{2})^2} \tag{16}$$

where m_c is the margin, $\boldsymbol{R}_{i,j}$ is the Anchor-Positive distance and $\boldsymbol{R}_{an}$ is the Anchor-Negative distance.

3.4.2. Triplet Loss in Unchanged Region

Since the regions in the unchanged area do not guarantee the same class in the same phase, we seek new sources of triplet pairs for unchanged areas. As illustrated in Figure 3b, we introduce a random image as the third phase to conduct triplet-temporal pairs. For the unchanged regions (red boxes), the newly added extra image is likely to be changed. Therefore, the paired pixels should be closer than the unpaired pixels for the unchanged regions. We thus formulate triplet pairs in the unchanged region as:

- Anchor: pixel embedding at location (i, j) in the unchanged region from phase 1.
- Positive: pixel embedding at location (i, j) in the unchanged region from phase 2.
- Negative: pixel embedding at location (i, j) from the random image.

To cope with the revert definition in Section 3.3.4, we define the triplet loss as:

$$L_{trp}^{u} = \sum_{i,j} \left[\boldsymbol{R}_{ap}^{2} - \boldsymbol{R}_{i,j}^{2} + m_u \right]_+ \tag{17}$$

$$\boldsymbol{R}_{ap} = 2 - \sqrt{\sum_{k=1}^{D} (f_{k,i,j}^{1} - f_{k,i,j}^{3})^2} \tag{18}$$

where m_u is the margin, $\boldsymbol{R}_{ap}$ is the Anchor-Positive distance and $\boldsymbol{R}_{i,j}$ is the Anchor-Negative distance.

3.4.3. Triplet Pairs in Semantic Change Detection

The triplet pairs in binary change detection are collected from binary change detection datasets, where no semantic labels are available. We can collect triplet pairs from different classes for semantic change detection with multiple classes as:

- Anchor: pixel embedding at location (i, j) in the changed region from phase 1.
- Positive: pixel embedding at location (e, f) in the changed region from phase 1, with the same semantic label as Anchor.
- Negative: pixel embedding at location (u, v) in the changed region from phase 2, with different semantic labels from Anchor.

Triplet loss originates from face recognition, where multiple classes are provided. The setting is the same for the semantic change detection branch. Therefore, there is no need for revert modifications for triplet pairs from multiple classes. We follow the standard triplet loss format as follows:

$$L_{trp}^{s} = \sum_{i,j} \left[\boldsymbol{d}_{ap}^{2} - \boldsymbol{d}_{an}^{2} + m_s \right]_+ \tag{19}$$

$$\boldsymbol{d}_{ap} = \sqrt{\sum_{k=1}^{D} (f_{k,i,j}^{1} - f_{k,e,f}^{2})^2} \tag{20}$$

$$\boldsymbol{d}_{an} = \sqrt{\sum_{k=1}^{D} (f_{k,i,j}^{1} - f_{k,u,v}^{2})^2} \tag{21}$$

where m_s is the margin, $\boldsymbol{d}_{ap}$ is the Anchor-Positive distance and $\boldsymbol{d}_{an}$ is the Anchor-Negative distance.

3.4.4. Overall Loss Function

In classifier-based semantic change detection, current methods [17,43] decouple binary change detection and semantic change detection. We follow this tradition and apply it to our metric-based method. The overall loss function combines losses for binary and semantic change detection as follows:

$$L = \alpha L_{trp}^{b} + \beta L_{trp}^{s} = \alpha (L_{trp}^{c} + L_{trp}^{u}) + \beta L_{trp}^{s} \tag{22}$$

where L_{trp}^{b} is the triplet loss for binary branch and can be split into losses for changed or unchanged regions. α and β determine the loss weight for the binary and semantic branches. Note that when $\beta = 0$, the formulation turns into binary change detection similar to Equation (14).

From the previous literature, TBSRL [12] also uses triplet loss for change detection. The method shares a similar formulation as Equation (7). However, the method collects the anchor, positive and negative samples with only binary labels. Some wrong triplet pairs might be selected due to the mismatch in the semantic meaning. On the other hand, our triplet pair selection strategy is carefully designed to guarantee semantically consistent triplet pairs.

4. Experiments

4.1. Datasets

We experiment on both binary change detection and semantic change detection datasets. Figure 4 shows some sample images of these datasets.

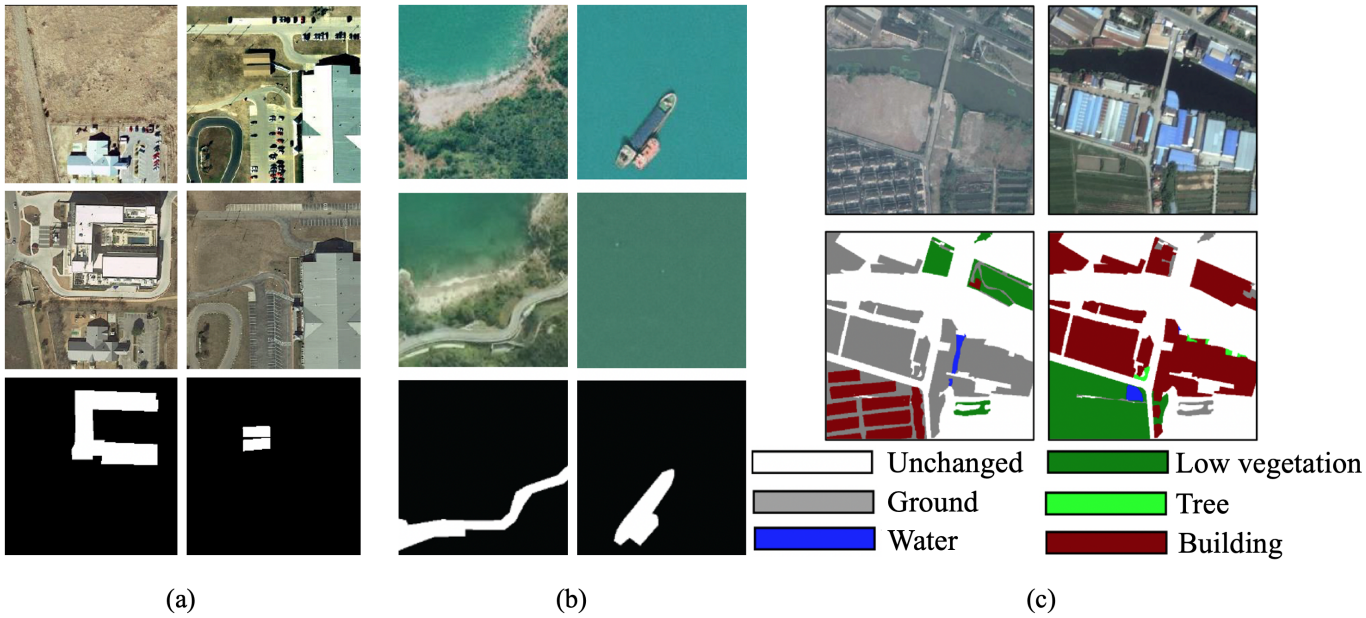

Figure 4. Sample images from the change detection datasets. The images are in the format of RGB three channels. (**a**) LEVIR-CD dataset mainly focuses on building updates and decline. (**b**) Apart from building construction, SYSU-CD dataset also focuses on roads and the sea. (**c**) SECOND dataset also provides the change type to provide more details of change detection.

Learning, Vision and Remote Sensing Laboratory (LEVIR)-CD: LEVIR-CD is collected from Google Earth platform with 637 bi-temporal remote sensing image pairs. The image has a size of 1024 × 1024 with a spatial resolution of 0.5 m. The dataset focuses on change labels of building instances rather than common semantic change types. The building change includes not only building appearing but also building disappearing. The dataset is officially split into train, validation, and test, three parts of which include 445, 64, and 128 pairs. We further crop non-overlapping 256 × 256 patches, which results in 7120 pairs for training. We keep the testing size unchanged as 1024 × 1024 in our experiments.

Sun Yat-Sen University(SYSU)-CD: SYSU-CD contains 20,000 pairs of 0.5 m aerial images taken between 2007 and 2014 in Hong Kong, with a total land area of 1106.66 square kilometers. The change includes the construction and maintenance of ports, sea routes, oceanic and coastal projects in Hong Kong, and major shipping hubs in international and Asia–Pacific areas. The dataset greatly complements change instances of high-rise buildings, which are very difficult to mark in high-resolution images because of the influence of deviation and shadow, and the change information related to the port compared to previous datasets. The image has a size of 256 × 256, and the official split for train/validation/test is 12,000:4000:4000.

SEmantic Change Detection Dataset (SECOND): SECOND is a large-scale benchmark dataset for SCD. The dataset is constructed with bitemporal high-resolution optical

images collected by several aerial platforms and sensors. Several cities in China, including Hangzhou, Chengdu and Shanghai are included in the dataset. The image size is 512 × 512 pixels and the spatial resolution varies from 0.5 m to 3 m (per pixel). The semantic change class includes unchanged, non-vegetated ground surface (ground), tree, low vegetation, water, buildings and playgrounds, resulting in 30 change types. Since the validation set from the original paper is not available, we follow the experimental setting of [43] with a train/validation = 4:1 split. There are 2375 image pairs for training and 593 for testing.

4.2. Implementation Details

We implement our methods based on Pytorch [62]. The backbone is HRNet18 [63] with FCN [64] as the decoder. We use the AdamW [65] optimizer with an initial learning rate of 0.001. CosineAnnealing scheduler is used with a maximum iteration of 1000. The training image size is 256 × 256 for all datasets. If the datasets provide larger images than 256 × 256, we crop non-overlapping images before the experiments. The test images keep the same resolution as the original ones. We use a random horizontal flip and random color manipulation for data augmentation. The color operations include random brightness, contrast, saturation, and hue. Images are used with different random color operations for our implementation. The random image scale is set to 0.8–1.2. The margin m_{con} for contrastive loss is 1.0. As for triplet loss, the margin for changed region $m_c = 4.0$, margin for unchanged region $m_u = 0.5$ and margin for semantic loss $m_s = 0.5$. The loss weight α and β in Equation (22) for different losses is 1.0.

We use precision, recall, intersection over union (IoU), and F1-score ($F1$) of changed regions to evaluate binary change detection. We set $q_{i,j}$ as the number of pixels of class j predicted as class i. The total class number is C and we use label 1 for changed regions in binary change detection. Then we compute the metrics as

$$precision = q_{11} / \sum_{j=0}^{1} q_{1j} \tag{23}$$

$$recall = q_{11} / \sum_{i=0}^{1} q_{i1} \tag{24}$$

$$F1 = (2 \cdot precision \cdot recall) / (precision + recall) \tag{25}$$

$$IoU = q_{11} / (q_{11} + q_{10} + q_{01}) \tag{26}$$

The higher the values are, the better the performance is.

We mainly use Separated Kappa (SeK) [17] as a metric for semantic change detection. SeK is a modified Kappa metric for semantic change detection tasks, which separates the non-change class from other change categories to reduce the effects of label imbalance. We set $\hat{Q} = \{\hat{q}_{ij}\}$ where $\hat{q}_{ij} = q_{ij}$ except that $\hat{q}_{00} = 0$. We compute SeK as follows:

$$\mathrm{IoU}_2 = \sum_{i=2}^{C} \sum_{j=2}^{C} q_{ij} / \left(\sum_{i=1}^{C} \sum_{j=1}^{C} q_{ij} - q_{00} \right) \tag{27}$$

$$\rho = \sum_{i=0}^{N} \hat{q}_{ii} / \sum_{i=0}^{N} \sum_{j=0}^{N} \hat{q}_{ij} \tag{28}$$

$$\eta = \sum_{i=0}^{N} \left(\sum_{j=0}^{N} \hat{q}_{ij} * \sum_{j=0}^{N} \hat{q}_{ji} \right) / \left(\sum_{i=0}^{N} \sum_{j=0}^{N} \hat{q}_{ij} \right)^2 \tag{29}$$

$$SeK = e^{IoU_2 - 1} \cdot (\rho - \eta) / (1 - \eta) \tag{30}$$

Since semantic change detection lacks widely used metrics, we also use Overall Accuracy (OA), mean intersection over union (mIoU) and F1-score over all foreground classes F_{scd}. The higher values indicate better performance. We recommend the reference [43] for a detailed explanation of these metrics.

4.3. Binary Change Detection Results

We compare our method with the following: **FC-EF** [5] concatenates bi-temporal images and processes them through a ConvNet to detect changes. **FC-Siam-Di** [5] extracts multi-level features of bi-temporal images from a Siamese ConvNet. The feature difference is used to detect changes. **FC-Siam-Conc** [5] extracts multi-level features of bi-temporal images from a Siamese ConvNet. The feature concatenation is used to detect changes. **DTCDSCN** [66] utilizes a dual attention module (DAM) to exploit the inter-dependencies between channels and spatial positions of CNN features to detect changes. **STANet** [7] is a Siamese-based spatial–temporal attention network for change detection. **IFNet** [28] is a multi-scale feature concatenation method with multi-level deep features of bi-temporal images. The image difference features are obtained by attention modules for change map reconstruction. **SNUNet** [67] is a multi-level feature concatenation method with a densely connected (NestedUNet) Siamese network for change detection. **BIT** [6] uses a transformer-based method to enhance the context-information of CNN features via semantic tokens. Feature differencing is followed to obtain the change map. **DSAMNet** [2] is a deeply supervised attention metric-based network to learn change maps by means of deep metric learning with convolutional block attention modules.

As shown in Table 1, the proposed method achieves better performance in terms of F1 and IoU. Notably, the IoU improves by 3.24% in LEVIR-CD and 2.89% for SYSU-CD. In the table, FC-EF, Fc-Siam-Di, FC-Siam-Conc, DTCDSCN, IFNet, SNUNet, and BIT all deploy two-class classifiers for change prediction. We beat these methods by the modified baseline. Although STANet [7] and DSAMNet [2] also use metric learning, we show that the discrimination during optimization is of vital importance, e.g., the data imbalance and triplet pair selection. TBSRL [12] is not open-sourced and thus not included in the table. Figure 5 also compares the visual quality of different state-of-the-art (SOTA) methods on test images on SYSU-CD. The proposed metric-based method produces fewer false positives and false negatives. The quantitative and qualitative comparisons confirm the superiority of our method over the existing methods.

Table 1. Comparison with different SOTA binary change detection methods.

Method	LEVIR-CD				SYSU-CD			
	Precision	**Recall**	**F1**	**IoU**	**Precision**	**Recall**	**F1**	**IoU**
FC-EF [5]	86.91	80.17	83.40	71.53	74.32	75.84	75.07	60.09
FC-Siam-Di [5]	89.53	83.31	86.31	75.92	**89.13**	61.21	72.57	56.96
FC-Siam-Conc [5]	91.99	76.77	83.69	71.96	82.54	71.03	76.35	61.75
DTCDSCN [66]	88.53	86.83	87.67	78.05	83.45	73.77	78.31	64.35
STANet [7]	83.81	91.00	87.26	77.40	70.76	**85.33**	77.37	63.09
IFNet [28]	**94.02**	82.93	88.13	78.77	84.30	72.69	78.06	64.02
SNUNet [67]	89.18	87.17	88.16	78.83	83.72	73.74	78.42	64.50
BIT [6]	89.24	89.37	89.31	80.68	84.15	74.25	78.89	65.14
DSAMNet [2]	91.70	86.77	89.17	80.45	74.81	81.86	78.18	64.18
ours baseline	90.03	90.77	90.40	82.48	81.94	77.28	79.54	65.67
ours	90.64	**91.89**	**91.26**	**83.92**	82.75	79.27	**80.97**	**68.03**

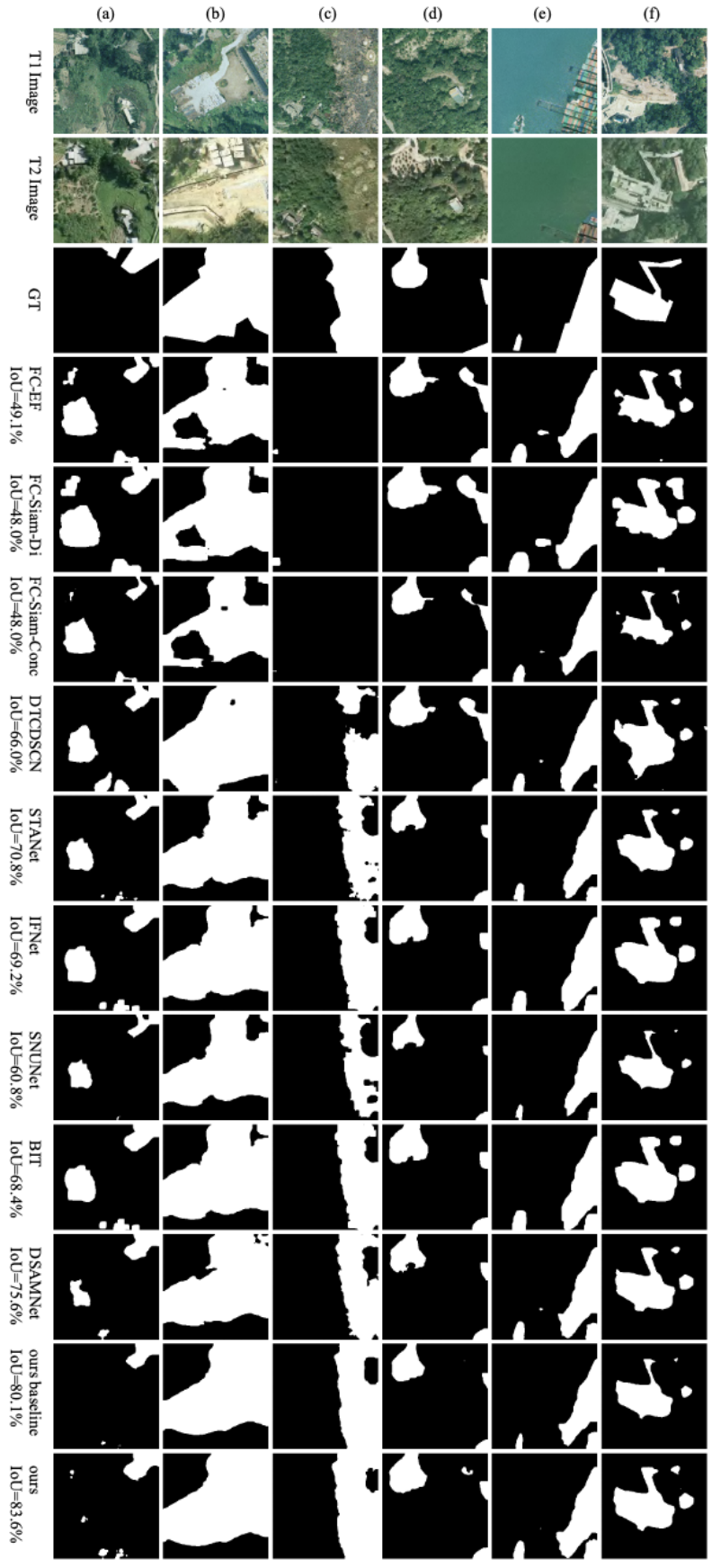

Figure 5. Visual comparison with other methods on binary change detection. We also list IoU values of each method over the sample images. Subfigures (**a**–**f**) are six samples from SYSU-CD dataset.

4.4. Semantic Change Detection Results

Following classifier-based semantic change detection methods, we add one extra triplet loss. The embeddings are supervised by both binary and semantic signals. During training, we collect extra triplet pairs based on semantic labels. We compute the average embedding for each class after training. During inference, we first obtain the changed region as binary change detection. These foreground regions are then compared with the pre-computed class embeddings to obtain the prediction class.

We compare this with methods in semantic change detection in Table 2. The listed methods all belong to classifier-based methods. Unet++ [27] is a variant of Unet [68] which enables more connections among multi-scale features. Resnet-GRU [23] and Resnet-LSTM [23] extract features with Resnet [69] encoder and then use Gated Recurent Units (GRU) [70] or Long Short-Term Memory units (LSTM) [71] for change detection. FC-EF, UNet++, ResNet-GRU, ResNet-LSTM, FC-Siam-conv, FC-Siam-diff, and IFNet methods are originally designed for binary change detection. The authors in [43] modify the last layer to multiple classes and report these results. Since these methods couple binary and semantic change detection together, the results are relatively low. HRSCD (str2, str3, str4) [42] decouples binary and semantic branches and is widely used in the later literature. However, the model structure is simple and thus the performance is limited on the complex semantic change detection datasets. Bi-SRNet obtains the highest accuracy among these methods. It contains the skip connections between the temporal branches and the CD branch. With shared embeddings learned from binary and semantic supervision, the proposed method outperforms SOTA methods in all the metrics. Figure 6 illustrates visualizations of different methods on semantic change detection. As can be seen, the proposed method performs well in finding the changed locations. Additionally, for the changed classes, our method classifies more correctly. The quantitative and qualitative comparisons confirm that metric-based methods also apply to SCD.

Table 2. Comparison of the proposed methods with literature methods for the SCD. F_{scd} is a new metric introduced by [43].

Method	Accuracy			
	OA (%)	**mIoU (%)**	**Sek (%)**	**F_{scd} (%)**
FC-EF [5]	85.18	64.25	9.98	48.45
UNet++ [27]	85.18	63.83	9.90	48.04
HRSCD-str.2 [42]	85.49	64.43	10.69	49.22
ResNet-GRU [23]	85.09	60.64	8.99	45.89
ResNet-LSTM [23]	86.77	67.16	15.96	56.90
FC-Siam-conv. [5]	86.92	68.86	16.36	56.41
FC-Siam-diff [5]	86.86	68.96	16.25	56.20
IFNet [28]	86.47	68.45	14.25	53.54
HRSCD-str.3 [42]	84.62	66.33	11.97	51.62
HRSCD-str.4 [42]	86.62	71.15	18.80	58.21
Bi-SRNet [43]	87.84	73.41	23.22	62.61
ours	**88.16**	**73.77**	**23.84**	**63.15**

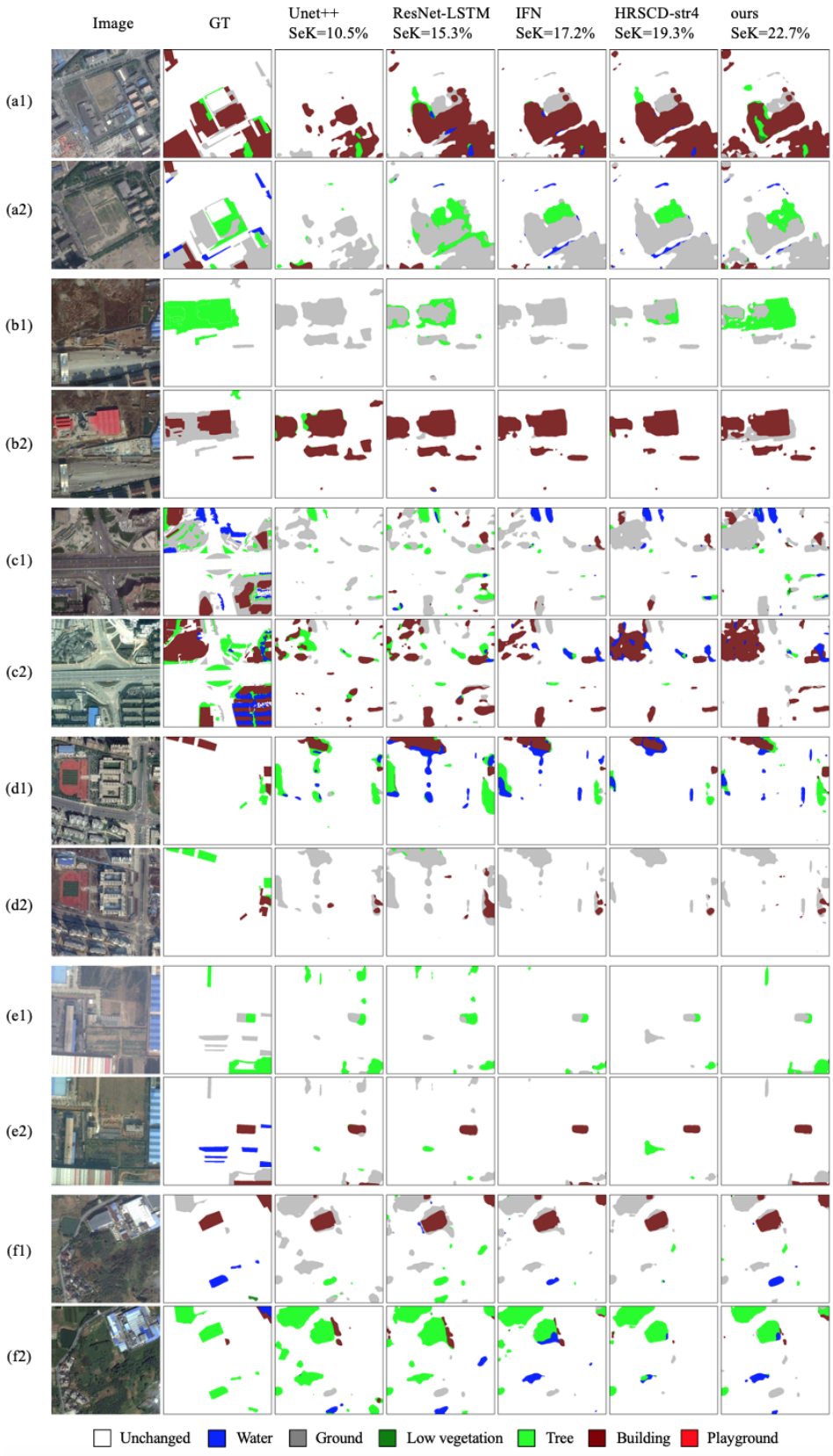

Figure 6. Visualization of several methods on semantic change detection. We also list SeK values of each method over the sample images. Subfigures (**a**–**f**) are six samples from SECOND dataset

4.5. Ablation Study

4.5.1. Contrastive Loss Baseline

We compare different formulations of contrastive loss on binary change detection in Table 3. As can be seen, naive implementation suffers from imbalanced data distribution. On the other hand, cosine modification and balanced modification perform relatively well. When using revert contrastive loss, we achieve better performance. Equipped with hard mining, we further improve the accuracy of revert contrastive loss.

Table 3. The accuracy on different modifications of contrastive loss. Naive, Balanced, Probability, Revert, and Revert+Mining correspond to the contrastive loss modifications in Section 3.

Method	LEVIR-CD		SYSU-CD	
	F1	IoU	F1	IoU
Naive	88.77	79.23	77.78	63.67
Balanced	88.76	80.15	78.14	64.11
Probability	90.01	82.18	79.24	65.17
Revert	90.20	82.28	79.14	65.04
Revert+Mining	90.40	82.48	79.54	65.67

4.5.2. Two Sources of Triplet Pairs

We then study how each source of triplet pair attributes for the final loss in binary change detection. As shown in Equations (14) and (22), we replace the contrastive loss with triplet loss in the changed area and unchanged region separately. Thus we test with $L = L^c_{con} + L^u_{trp}$ and $L = L^c_{trp} + L^u_{con}$ respectively.

As shown in Table 4, triplet loss from changed and unchanged regions is beneficial for performance. Among the two sources, triplet loss in unchanged regions contributes more, indicating the importance of temporal information. The best performance is achieved with the combination of the two losses. Figure 7 illustrates change detections from different loss functions. Some false alarms might mislead the baseline. On the contrary, the performance boosts with triplet loss in the changed/unchanged regions. False alarm regions are reduced due to better discriminative learning from triplet loss. The best performance is achieved with triplet loss on both changed and unchanged regions, confirming our conclusion.

Table 4. The accuracy on the two sources of triplet pairs.

Method	LEVIR-CD		SYSU-CD	
	F1	IoU	F1	IoU
baseline	90.40	82.48	79.54	65.67
Triplet in Changed	90.62	82.71	79.85	65.89
Triplet in Unchanged	90.99	83.48	80.51	67.87
Triplet from two sources	91.26	83.92	80.97	68.03

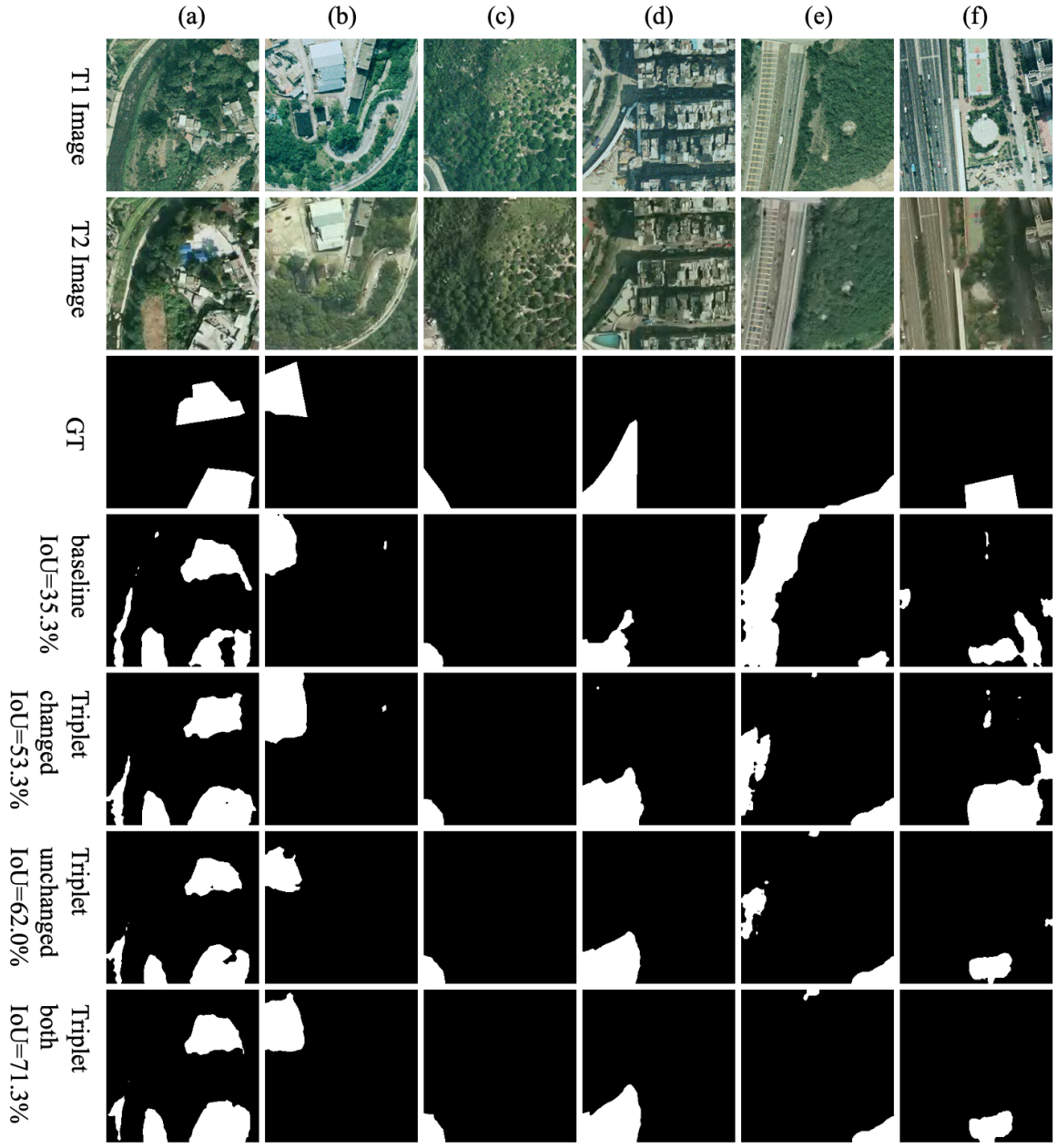

Figure 7. Comparison of different loss functions for binary change detection. We also list IoU values of each method over the sample images. Subfigures (**a**–**f**) are six samples from SYSU-CD dataset.

4.5.3. Embedding Visualization

We visualize feature embeddings in Figure 8. As can be seen, pixel embedding at certain points in red boxes produces larger activations than the others. Changed regions are often those pixels with strong activations. We also visualize the triplet pairs by computing pixel-wise change probabilities. The brighter value means a higher probability of change. The triplet loss in unchanged regions provides extra supervision since the extra unpaired probability map is reasonable.

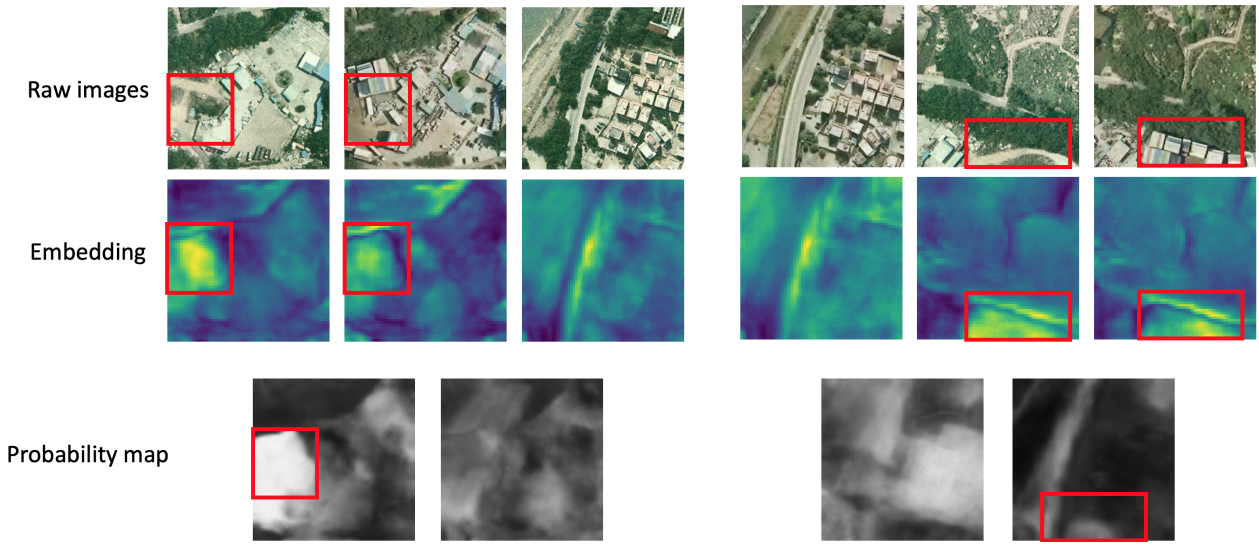

Figure 8. Embedding visualization on SYSU-CD test set.

5. Discussion

In the change detection field, most methods rely on classifiers to output predictions. Some metric-based methods suffer from data imbalance or imperfect triplet pair selection. On the other hand, metric learning seems very suitable for change detection with carefully designed loss functions and strategies. Our work bridges metric learning and change detection and may bring insights to later works. The value of our method is the modification of contrastive loss to build a strong baseline. We propose a revert version of contrastive loss for change detection and enable hard mining simultaneously. Then triplet pairs from multiple sources are collected, which improves the contrastive loss. Some naive sampling strategies might collect imperfect triplet pairs since the changed contours in one image are not guaranteed to have the same semantic meaning. We try to collect semantically consistent triplet pairs by carefully designed strategies. The experiments on binary and semantic change detection prove our effectiveness.

Our methods can be applied in environmental monitoring, urban management, damage assessment, and crop monitoring. Since we compute embedding distances among images, a potential use might be multi-temporal change detection. For example, we could average three embedding maps from January, February, and March to obtain one single embedding map. This strategy could avoid cloud or poor light conditions from a certain month. Additionally, the strategy could save inference time if multiple images are provided. For example, for a sequence of 12 images of a year, traditional classifier-based methods have to make paired inputs and $12 \times 11/2 = 66$ times of inference are needed. Our method can potentially infer only 12 times to cache feature embeddings. Then we can directly compute distances among these embeddings efficiently.

Due to limited categories of change detection, e.g., SECOND dataset only has six basic land-use types, the problem becomes simple for some methods. Complex change detection datasets with more evaluation samples and change types could be beneficial, which would be our future work.

6. Conclusions

In this paper, we replace classifiers with embedding distance for change detection. Several metric learning formulations have been studied, including contrastive loss and triplet loss. First, a strong baseline is achieved by training on pair-wise images with Reverted Contrastive Loss (RCL) with hard mining. Then, motivated by the success of triplet loss, we introduce two sources of triplet pairs and propose a novel Spatial–Temporal Triplet Loss (STTL). We also show that the triplet loss can be extended to semantic change

detection, where semantic labels are provided for the changed area. The experimental results prove state-of-the-art performance on both binary and semantic change detection.

Author Contributions: Conceptualization, Y.Z. and W.L.; methodology, Y.Z., W.L. and Y.W.; software, Y.Z. and W.L.; validation, Y.Z. and W.L.; formal analysis, Z.W. and H.L.; investigation, Y.Z. and W.L.; resources, Z.W. and H.L.; data curation, Z.W.; writing—original draft preparation, Y.Z., W.L. and Y.W.; writing—review and editing, Z.W. and H.L.; visualization, Y.W.; supervision, Z.W. and H.L.; project administration, Z.W. and H.L.; funding acquisition, Z.W. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Zhejiang Science and Technology Program under Grant 2021C01017.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hussain, M.; Chen, D.; Cheng, A.; Wei, H.; Stanley, D. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS J. Photogramm. Remote Sens.* **2013**, *80*, 91–106. [CrossRef]
2. Shi, Q.; Liu, M.; Li, S.; Liu, X.; Wang, F.; Zhang, L. A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5604816. [CrossRef]
3. Mahdavi, S.; Salehi, B.; Huang, W.; Amani, M.; Brisco, B. A PolSAR change detection index based on neighborhood information for flood mapping. *Remote Sens.* **2019**, *11*, 1854. [CrossRef]
4. Woodcock, C.E.; Loveland, T.R.; Herold, M.; Bauer, M.E. Transitioning from change detection to monitoring with remote sensing: A paradigm shift. *Remote Sens. Environ.* **2020**, *238*, 111558. [CrossRef]
5. 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), Athens, Greece, 7–10 October 2018; pp. 4063–4067.
6. Chen, H.; Qi, Z.; Shi, Z. Remote sensing image change detection with transformers. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5607514. [CrossRef]
7. Chen, H.; Shi, Z. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote Sens.* **2020**, *12*, 1662. [CrossRef]
8. Sun, Y.; Fu, K.; Wang, Z.; Zhang, C.; Ye, J. Road Network Metric Learning for Estimated Time of Arrival. In Proceedings of the 2020 25th International Conference On Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 1820–1827.
9. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 1735–1742.
10. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
11. Huang, G.B.; Mattar, M.; Berg, T.; Learned-Miller, E. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In Proceedings of the Workshop on faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Tuscany, Italy, 28 July–3 August 2008; Springer: Berlin/Heidelberg, Germany, 2008; p. 1.
12. Zhang, M.; Xu, G.; Chen, K.; Yan, M.; Sun, X. Triplet-based semantic relation learning for aerial remote sensing image change detection. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 266–270. [CrossRef]
13. Hou, X.; Bai, Y.; Li, Y.; Shang, C.; Shen, Q. High-resolution triplet network with dynamic multiscale feature for change detection on satellite images. *ISPRS J. Photogramm. Remote Sens.* **2021**, *177*, 103–115. [CrossRef]
14. Liu, J.; Gong, M.; Qin, K.; Zhang, P. A deep convolutional coupling network for change detection based on heterogeneous optical and radar images. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *29*, 545–559. [CrossRef]
15. Zhan, Y.; Fu, K.; Yan, M.; Sun, X.; Wang, H.; Qiu, X. Change detection based on deep siamese convolutional network for optical aerial images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1845–1849. [CrossRef]
16. Wang, M.; Tan, K.; Jia, X.; Wang, X.; Chen, Y. A deep siamese network with hybrid convolutional feature extraction module for change detection based on multi-sensor remote sensing images. *Remote Sens.* **2020**, *12*, 205. [CrossRef]
17. Yang, K.; Xia, G.S.; Liu, Z.; Du, B.; Yang, W.; Pelillo, M. Asymmetric siamese networks for semantic change detection. *arXiv* **2020**, arXiv:2010.05687.
18. Khelifi, L.; Mignotte, M. Deep Learning for Change Detection in Remote Sensing Images: Comprehensive Review and Meta-Analysis. *IEEE Access* **2020**, *8*, 126385–126400. [CrossRef]
19. Ji, S.; Wei, S.; Lu, M. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 574–586. [CrossRef]
20. Zhang, W.; Lu, X. The Spectral-Spatial Joint Learning for Change Detection in Multispectral Imagery. *Remote Sens.* **2019**, *11*, 240. [CrossRef]

21. Song, A.; Choi, J.; Han, Y.; Kim, Y. Change detection in hyperspectral images using recurrent 3D fully convolutional networks. *Remote Sens.* **2018**, *10*, 1827. [CrossRef]
22. Liu, M.; Shi, Q.; Marinoni, A.; He, D.; Liu, X.; Zhang, L. Super-resolution-based change detection network with stacked attention module for images with different resolutions. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 4403718. [CrossRef]
23. Mou, L.; Bruzzone, L.; Zhu, X.X. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 924–935. [CrossRef]
24. Luppino, L.T.; Kampffmeyer, M.; Bianchi, F.M.; Moser, G.; Serpico, S.B.; Jenssen, R.; Anfinsen, S.N. Deep image translation with an affinity-based change prior for unsupervised multimodal change detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 4700422. [CrossRef]
25. Li, X.; Du, Z.; Huang, Y.; Tan, Z. A deep translation (GAN) based change detection network for optical and SAR remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2021**, *179*, 14–34. [CrossRef]
26. Sun, Y.; Lei, L.; Li, X.; Sun, H.; Kuang, G. Nonlocal patch similarity based heterogeneous remote sensing change detection. *Pattern Recognit.* **2021**, *109*, 107598. [CrossRef]
27. Peng, D.; Zhang, Y.; Guan, H. End-to-end change detection for high resolution satellite images using improved UNet++. *Remote Sens.* **2019**, *11*, 1382. [CrossRef]
28. Zhang, C.; Yue, P.; Tapete, D.; Jiang, L.; Shangguan, B.; Huang, L.; Liu, G. A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2020**, *166*, 183–200. [CrossRef]
29. Zhang, M.; Shi, W. A feature difference convolutional neural network-based change detection method. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7232–7246. [CrossRef]
30. Zhao, W.; Mou, L.; Chen, J.; Bo, Y.; Emery, W.J. Incorporating Metric Learning and Adversarial Network for Seasonal Invariant Change Detection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 2720–2731. [CrossRef]
31. Bandara, W.G.C.; Patel, V.M. A Transformer-Based Siamese Network for Change Detection. *arXiv* **2012**, arXiv:2201.01293.
32. Hou, B.; Wang, Y.; Liu, Q. Change detection based on deep features and low rank. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2418–2422. [CrossRef]
33. Saha, S.; Bovolo, F.; Bruzzone, L. Unsupervised deep change vector analysis for multiple-change detection in VHR images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 3677–3693. [CrossRef]
34. Saha, S.; Bovolo, F.; Bruzzone, L. Building change detection in VHR SAR images via unsupervised deep transcoding. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 1917–1929. [CrossRef]
35. Zheng, Z.; Ma, A.; Zhang, L.; Zhong, Y. Change is Everywhere: Single-Temporal Supervised Object Change Detection in Remote Sensing Imagery. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 15193–15202.
36. Akiva, P.; Purri, M.; Leotta, M. Self-Supervised Material and Texture Representation Learning for Remote Sensing Tasks. *arXiv* **2021**, arXiv:2112.01715.
37. Chen, Y.; Bruzzone, L. Self-supervised Remote Sensing Images Change Detection at Pixel-level. *arXiv* **2021**, arXiv:2105.08501.
38. Chen, H.; Zao, Y.; Liu, L.; Chen, S.; Shi, Z. Semantic decoupled representation learning for remote sensing image change detection. *arXiv* **2022**, arXiv:2201.05778.
39. Wu, C.; Du, B.; Zhang, L. Fully Convolutional Change Detection Framework with Generative Adversarial Network for Unsupervised, Weakly Supervised and Regional Supervised Change Detection. *arXiv* **2022**, arXiv:2201.06030.
40. Bruzzone, L.; Serpico, S.B. An iterative technique for the detection of land-cover transitions in multitemporal remote-sensing images. *IEEE Trans. Geosci. Remote Sens.* **1997**, *35*, 858–867. [CrossRef]
41. Liu, S.; Bruzzone, L.; Bovolo, F.; Zanetti, M.; Du, P. Sequential spectral change vector analysis for iteratively discovering and detecting multiple changes in hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4363–4378. [CrossRef]
42. Daudt, R.C.; Le Saux, B.; Boulch, A.; Gousseau, Y. Multitask learning for large-scale semantic change detection. *Comput. Vis. Image Underst.* **2019**, *187*, 102783. [CrossRef]
43. Ding, L.; Guo, H.; Liu, S.; Mou, L.; Zhang, J.; Bruzzone, L. Bi-Temporal Semantic Reasoning for the Semantic Change Detection in HR Remote Sensing Images. *arXiv* **2021**, arXiv:2108.06103.
44. Wang, Y.; Zhang, Y.; Zhang, F.; Wang, S.; Lin, M.; Zhang, Y.; Sun, X. Ada-nets: Face clustering via adaptive neighbour discovery in the structure space. *arXiv* **2022**, arXiv:2202.03800.
45. Zhang, Y.; Huang, Y.; Wang, L.; Yu, S. A comprehensive study on gait biometrics using a joint CNN-based method. *Pattern Recognit.* **2019**, *93*, 228–236. [CrossRef]
46. Luo, H.; Gu, Y.; Liao, X.; Lai, S.; Jiang, W. Bag of tricks and a strong baseline for deep person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.
47. Zhang, Y.; Qian, Q.; Liu, C.; Chen, W.; Wang, F.; Li, H.; Jin, R. Graph convolution for re-ranking in person re-identification. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 2704–2708.
48. Zhang, Y.; Liu, C.; Chen, W.; Xu, X.; Wang, F.; Li, H.; Hu, S.; Zhao, X. Revisiting instance search: A new benchmark using Cycle Self-Training. *Neurocomputing* **2022**, *501*, 270–284. [CrossRef]

49. He, S.; Luo, H.; Chen, W.; Zhang, M.; Zhang, Y.; Wang, F.; Li, H.; Jiang, W. Multi-domain learning and identity mining for vehicle re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 582–583.
50. Liu, C.; Zhang, Y.; Luo, H.; Tang, J.; Chen, W.; Xu, X.; Wang, F.; Li, H.; Shen, Y.D. City-scale multi-camera vehicle tracking guided by crossroad zones. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 4129–4137.
51. Luo, H.; Chen, W.; Xu, X.; Gu, J.; Zhang, Y.; Liu, C.; Jiang, Y.; He, S.; Wang, F.; Li, H. An empirical study of vehicle re-identification on the AI City Challenge. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 4095–4102.
52. Liu, C.; Zhang, Y.; Chen, W.; Wang, F.; Li, H.; Shen, Y.D. Adaptive Matching Strategy for Multi-Target Multi-Camera Tracking. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 2934–2938.
53. Zhang, Y.; Huang, Y.; Wang, L. Multi-task deep learning for fast online multiple object tracking. In Proceedings of the 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), Nanjing, China, 26–29 November 2017; pp. 138–143.
54. Du, F.; Xu, B.; Tang, J.; Zhang, Y.; Wang, F.; Li, H. 1st place solution to eccv-tao-2020: Detect and represent any object for tracking. *arXiv* **2021**, arXiv:2101.08040.
55. Zhang, Y.; Huang, Y.; Wang, L. What makes for good multiple object trackers? In Proceedings of the 2016 IEEE International Conference on Digital Signal Processing (DSP), Beijing, China, 16–18 October 2016; pp. 467–471.
56. Yuqi, Z.; Xianzhe, X.; Weihua, C.; Yaohua, W.; Fangyi, Z.; Fan, W.; Hao, L. 2nd Place Solution to Google Landmark Retrieval 2021. *arXiv* **2021**, arXiv:2110.04294.
57. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the European conference on computer vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 499–515.
58. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. Sphereface: Deep hypersphere embedding for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 212–220.
59. Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; Liu, W. Cosface: Large margin cosine loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5265–5274.
60. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.
61. Zhang, Y.; Huang, Y.; Yu, S.; Wang, L. Cross-view gait recognition by discriminative feature learning. *IEEE Trans. Image Process.* **2019**, *29*, 1001–1015. [CrossRef]
62. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
63. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364. [CrossRef] [PubMed]
64. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 2015; pp. 3431–3440.
65. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
66. Liu, Y.; Pang, C.; Zhan, Z.; Zhang, X.; Yang, X. Building Change Detection for Remote Sensing Images Using a Dual-Task Constrained Deep Siamese Convolutional Network Model. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 811–815. [CrossRef]
67. Fang, S.; Li, K.; Shao, J.; Li, Z. SNUNet-CD: A densely connected siamese network for change detection of VHR images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 8007805. [CrossRef]
68. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015.
69. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
70. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
71. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]