land

MDPI

Article

Feature-Differencing-Based Self-Supervised Pre-Training for Land-Use/Land-Cover Change Detection in High-Resolution Remote Sensing Images

Wenqing Feng [1,*], Fangli Guan [1], Chenhao Sun [2] and Wei Xu [3]

[1] Computer & Software School, Hangzhou Dianzi University, Hangzhou 310018, China; fangli.guan@hdu.edu.cn
[2] Electrical & Information Engineering School, Changsha University of Science & Technology, Changsha 410114, China; chenhaosun@csust.edu.cn
[3] Information System and Management College, National University of Defense Technology, Changsha 410015, China; weixu@nudt.edu.cn
* Correspondence: wq_feng@hdu.edu.cn

Abstract: Land-use and land-cover (LULC) change detection (CD) is a pivotal research area in remote sensing applications, posing a significant challenge due to variations in illumination, radiation, and image noise between bi-temporal images. Currently, deep learning solutions, particularly convolutional neural networks (CNNs), represent the state of the art (SOTA) for CD. However, CNN-based models require substantial amounts of annotated data, which can be both expensive and time-consuming. Conversely, acquiring a large volume of unannotated images is relatively easy. Recently, self-supervised contrastive learning has emerged as a promising method for learning from unannotated images, thereby reducing the need for annotation. However, most existing methods employ random values or ImageNet pre-trained models to initialize their encoders and lack prior knowledge tailored to the demands of CD tasks, thus constraining the performance of CD models. To address these challenges, we introduce a novel feature-differencing-based framework called Barlow Twins for self-supervised pre-training and fine-tuning in CD (BTCD). The proposed approach employs absolute feature differences to directly learn unique representations associated with regions that have changed from unlabeled bi-temporal remote sensing images in a self-supervised manner. Moreover, we introduce invariant prediction loss and change consistency regularization loss to enhance image alignment between bi-temporal images in both the decision and feature space during network training, thereby mitigating the impact of variation in radiation conditions, noise, and imaging viewpoints. We select the improved UNet++ model for fine-tuning self-supervised pre-training models and conduct experiments using two publicly available LULC CD datasets. The experimental results demonstrate that our proposed approach outperforms existing SOTA methods in terms of competitive quantitative and qualitative performance metrics.

Keywords: Barlow Twins loss function; land use/land cover change detection; pre-training; self-supervised contrastive learning

check for updates

Citation: Feng, W.; Guan, F.; Sun, C.; Xu, W. Feature-Differencing-Based Self-Supervised Pre-Training for Land-Use/Land-Cover Change Detection in High-Resolution Remote Sensing Images. *Land* **2024**, *13*, 927. https://doi.org/10.3390/land13070927

Academic Editor: Jūratė Sužiedelytė-Visockienė

Received: 13 May 2024
Revised: 16 June 2024
Accepted: 24 June 2024
Published: 26 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Land-use and land-cover (LULC) change detection (CD) is applied to identify surface-related changes using bi-temporal remote sensing images, making it crucial for earth observation. The generated information can aid in the monitoring of urban development, natural resources, minerals, and assessments of military damage [1–4]. To date, many theoretical models and technical methods for diverse LULC CD applications have been proposed, including traditional algebraic comparison, change vector analysis, post-classification comparison, object-oriented image analysis, time series image analysis, and machine-learning-based methods [2]. Similar to other remote sensing image interpretation

tasks, LULC CD involves dealing with multi-scene remote sensing images that cover an area at different times. Data sources can include both homologous and heterologous situations. Data processing involves radiometric and geometric preprocessing, CD algorithms, threshold segmentation, and accuracy evaluation [1,2].

In recent years, the emergence of remote sensing big data and the rapid development of artificial intelligence have led to a new academic trend in LULC CD research [5–12]. Deep learning has revolutionized traditional methods, as it adopts an end-to-end learning mode that directly extracts change area information from bi-temporal remote sensing images, thus avoiding the dependence on difference images. Furthermore, the features extracted by deep networks have strong noise robustness, making them suitable for handling bi-temporal images from the same sensor or different sources. A representative method is the fully convolutional Siamese network [10]. The key to its effectiveness lies in a large number of manually labelled variation samples, which are time-consuming to collect and annotate from bi-temporal remote sensing images and require professional domain knowledge. To overcome this challenge, many researchers have attempted transfer learning strategies that leverage the knowledge learned from a large-scale natural image dataset, ImageNet [13]. However, while pre-trained ImageNet models achieve better results than those trained from scratch, they still face significant limitations due to the domain gap between natural images and remote sensing images. To tackle these issues, the idea of utilizing a small set of manually labelled samples or even completely unlabeled samples has gained increasing popularity in LULC CD applications, and there are two main research directions. The first involves the construction of massive remote sensing image datasets (such as Million AID [14], fMoW [15], and BigEarthNet [16]), which usually come with sparsely noisy or clean labelled data. These datasets are leveraged for supervised remote sensing pre-training to acquire transferable image representations, which can then be tailored to downstream LULC CD tasks with only a few labelled samples. The other main direction involves utilizing unsupervised contrastive learning to pre-train an encoder on unlabeled massive remote sensing image datasets (such as SeCo [17] and SSL4EO-S12 [18]) to obtain well-initialized parameters for downstream CD tasks. As a unique form of unsupervised learning, self-supervised contrastive learning (SSCL) trains models through self-designed proxy tasks and sometimes produces better CD results than related supervised methods. However, gathering remote sensing imagery data on a scale of millions is costly, and carrying out unsupervised contrastive learning on such a dataset demands substantial GPU computing resources.

Recently, SSCL methods have made significant improvements in the field of computer vision, allowing for the learning of useful representations from unlabeled data [19–25]. This approach has become increasingly popular, particularly in situations where labelling is costly, such as in medical and satellite imaging. Self-supervised contrastive pre-training can learn meaningful feature representations by utilizing a large amount of unlabeled remote sensing image data. These meaningful feature representations can improve the performance of various downstream CD tasks, which has drawn the attention of many researchers [26,27]. Although self-supervised contrastive pre-training holds great potential, many advanced SSCL methods are currently difficult to implement in practice. They require a large number of data samples and computational resources for effective application, limiting their practicality. For instance, while SimCLR [21] and its improved versions achieve state-of-the-art (SOTA) performance on ImageNet classification tasks, they necessitate a batch size of 4096. Similarly, BYOL [24] achieves cutting-edge performance through positive contrastive learning, symmetric networks, and stop-gradient methods, but demands 64 graphical processing units (GPUs). MoCo [22] tackles negative samples by introducing momentum encoders and queue sampling strategies, yet its computation speed is hindered by the necessity of sampling negative samples across the entire queue. SimSiam [20], foregoing the need for negative samples, employs asymmetric network structures and cross-gradient updates to counter trivial solutions. However, it requires more computational resources and exhibits sensitivity to hyperparameters. SwAV [23], leveraging online

clustering, circumvents the need for negative samples via multi-view prediction encoding, presenting advantages in high-label-cost scenarios. Nevertheless, it imposes high computational resource requirements and a relatively complex training process due to its utilization of online clustering algorithms. In contrast to the aforementioned methods, Barlow Twin (BT) [25] introduces innovative thinking into SSCL. Unconstrained by batch sample size limitations and devoid of the need for negative samples, BT focuses on the embedding itself to avoid asymmetric structural design. By computing the cross-correlation matrix of augmented samples and utilizing a loss function to reduce redundancy, BT ensures that the cross-correlation matrix closely approximates the identity matrix. This ensures that different augmented versions of the same sample possess similar feature vectors and minimize redundancy across different dimensional components, thereby enhancing the efficiency of feature representation.

Due to the advantages of BT, we adopted it as an SSCL objective in our proposed LULC CD framework, as it does not require large negative samples to avoid trivial solutions [25]. Here, we introduce the feature-differencing-based BT self-supervised pre-training and fine-tuning CD (BTCD) framework in this paper. Our proposed method utilizes feature difference to learn discriminatory representations that correspond to areas of change, which is highly beneficial for CD tasks. In addition, we introduce the invariant prediction (IP) loss and change consistency regularization (CCR) loss to enhance image alignment between bi-temporal images in the decision and feature space during network training, thereby reducing the effects of variation in radiation conditions, noise, and imaging viewpoints. In the proposed framework, an SSCL method is used to pre-train the encoder of the improved UNet++ model [4]. This encoder has excellent parameter initialization and can effectively solve downstream CD tasks. The SSCL algorithm does not require CD label images. Instead, bi-temporal images are used to construct sample pairs for comparison, and a powerful encoder can be pre-trained for downstream CD tasks. Our contributions can be summarized as follows:

(1) We propose a novel framework called BTCD, which consists of two cascaded stages: self-supervised contrastive pre-training and fine-tuning. In the first stage, the algorithm leverages an absolute feature difference self-supervised pre-training method to learn task-specific change representation for CD. In the second stage, the pre-trained encoder undergoes fine-tuning, which effectively enhances the downstream CD network by taking advantage of favorable parameter initialization.
(2) To mitigate the impact of radiation differences, noise variation, and imaging perspective differences caused by bi-temporal images, we incorporate an IP loss and a CCR loss into the self-supervised contrastive pre-training of the original BT loss function. This approach improves the performance of the pre-training process.
(3) We evaluated the proposed approach on two publicly available LULC CD datasets and demonstrated that fine-tuning using only the pre-trained encoder surpasses ImageNet supervised pre-training methods and several recently proposed SSCL pre-training methods without requiring additional data.
(4) We verified the effectiveness of self-supervised pre-training under insufficiently labelled sample data. When labelled data were insufficient, the proposed pre-training method significantly improved CD model performance.

In the following sections, we present our proposed BTCD method and experimental details (Section 2), present comparison results (Section 3), discuss our findings (Section 4), and conclude our work (Section 5).

2. Materials and Methods

2.1. BTCD Framework

Figure 1 provides an overview of our proposed BTCD framework. We assume that X^1_{train} and X^2_{train} are bi-temporal images of the same geographic location captured at times t^1 and t^2, respectively. We pre-train a model on the unlabeled training set $\left\{U = \left(X^1_{\text{train}}, X^2_{\text{train}}\right)_i\right\}_{i=1}^{N}$ using a self-supervised algorithm, so that the pre-trained en-

coder can be easily fine-tuned on the labelled training set $\{L = (X^1_{\text{train}}, X^2_{\text{train}}, Y_{\text{train}})_i\}_{i=1}^{N}$. The proposed framework consists of two distinct stages. During the SSCL stage, only bi-temporal images X^1_{train} and X^2_{train} are used for training. In the subsequent CD stage, the framework is trained using both bi-temporal images X^1_{train} and X^2_{train}, as well as labelled images Y_{train} from the training set. The overall framework is enhanced in two key steps: the first CNN is trained on a pretext task, the encoder component that provides the best features for CD is selected, and the selected encoder is utilized by the second CNN to perform CD.

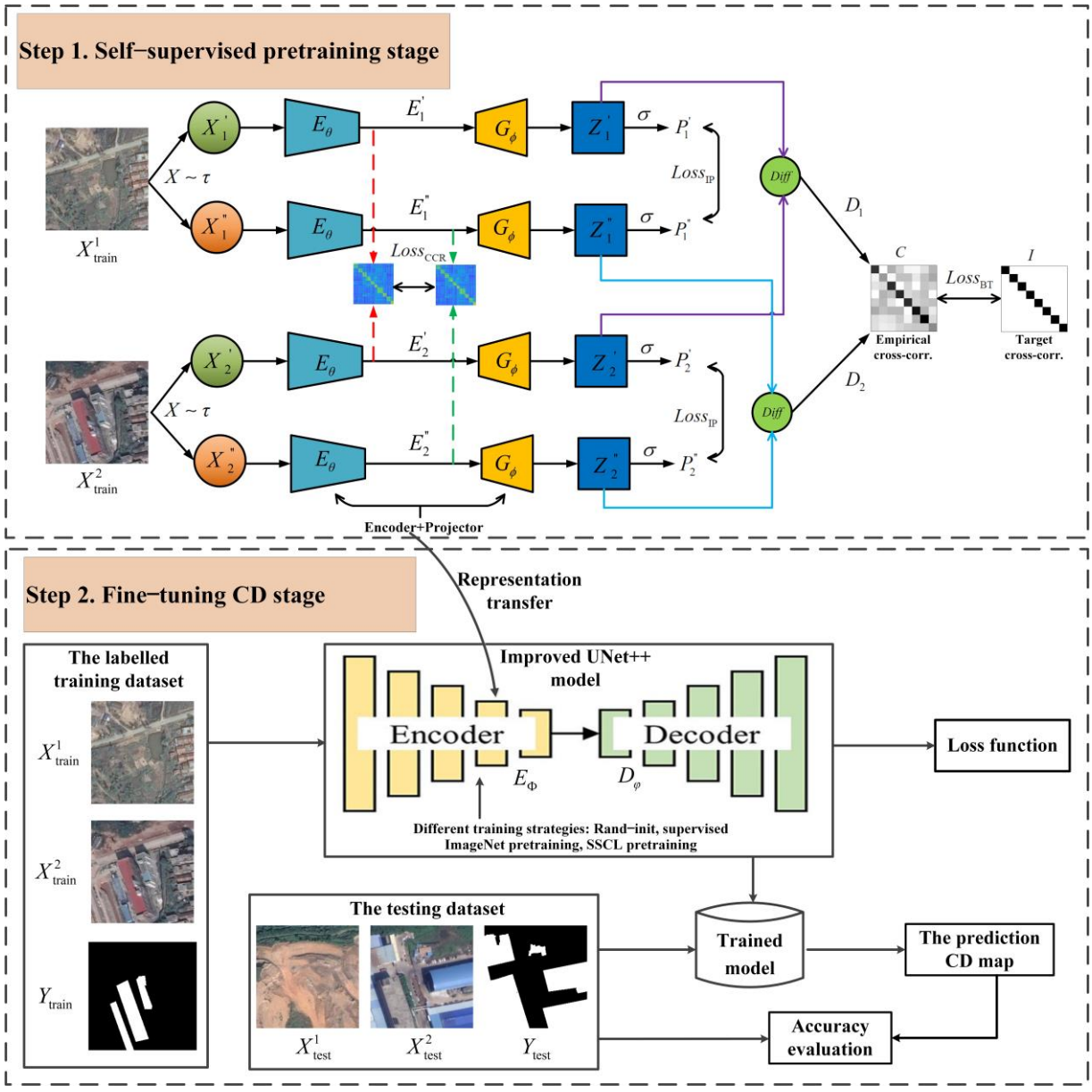


Figure 1. Pipeline of the proposed LULC CD Framework.

(1) Self-supervised Pre-training Stage. In this stage, we initialize our proposed self-supervised pre-training models with the supervised ImageNet weights instead of random initialization. An encoder E_θ is trained, where θ is a learnable parameter used to solve a pretext task. The BTCD algorithm uses three objective functions during the training process: BT loss, IP loss and CCR loss. Self-supervised by the hybrid loss

function *Loss* of a pretext task "contrasting", the parameters θ^i of the encoder E_θ for the i-th iteration are updated as follows:

$$\theta^{i+1} = \theta^i - \kappa \frac{\partial Loss}{\partial \theta^i} \tag{1}$$

where κ represents the learning rate. Ultimately, we obtain a pre-training encoder E_Φ with well-initialized parameters.

(2) Fine-tuning CD Stage. In this stage, we utilize the pre-trained encoder E_Φ to train the improved UNet++ model end-to-end. The improved UNet++ model extracts features from the given bi-temporal images X^1_{train} and X^2_{train} and classifies them as either "changed" or "unchanged" to predict the change map $\hat{Y}_{\text{train}}$.

$$\hat{Y}_{\text{train}} = D_\varphi(E_\Phi(X^1_{\text{train}}), E_\Phi(X^2_{\text{train}})) \tag{2}$$

where D_φ represents the decoder and the CD head, and φ is a learnable parameter. With parameters Φ and φ, the improved UNet++ model is updated as follows:

$$\Phi^{i+1} = \Phi^i - \kappa \frac{\partial Loss_{\text{CD}}(\hat{Y}_{\text{train}}, Y_{\text{train}})}{\partial \Phi^i} \tag{3}$$

$$\varphi^{i+1} = \varphi^i - \kappa \frac{\partial Loss_{\text{CD}}(\hat{Y}_{\text{train}}, Y_{\text{train}})}{\partial \varphi^i} \tag{4}$$

where $Loss_{\text{CD}}$ represents the loss function and Y_{train} represents the ground truth.

2.2. BTCD Pretraining Method

2.2.1. BTCD

The primary objective of CD is to detect and identify changes in surface objects between bi-temporal images. Therefore, the radiometric space of the bi-temporal images must be aligned with the low-level features of the CD network. This alignment poses a significant challenge because regions of change are highly susceptible to variation in seasons and noise. To facilitate low-level feature alignment between bi-temporal images, we employed a self-supervised pre-training algorithm as the baseline model utilizing the BT loss function [25]. By maximizing the cross-correlation between unchanged regions, this function implicitly minimizes the differences between the bi-temporal images in the feature space. The BT loss function was selected because it intersects with other competitive self-supervised contrastive learning algorithms while requiring only a few negative samples to learn robust representations. To strengthen the correlation between bi-temporal images, we developed a self-supervised pre-training algorithm that uses the absolute feature difference to learn essential representations for CD. Figure 2 provides an overview of the BTCD algorithm.

The developed BTCD algorithm takes bi-temporal images (X_1, X_2) as inputs and applies color distortion and Gaussian blur transformation to obtain augmented image pairs $(X_1 \to X_1', X_1'', X_2 \to X_2', X_2'')$. Random clipping was excluded from preprocessing of the training set, as areas of change are significantly smaller than unchanged areas. The BTCD algorithm has a Siamese architecture that uses ResNet50 (excluding the final classification layer) as a feature extractor, followed by a projector network. The input image, sized at $256 \times 256 \times 3$, undergoes several stages of processing through ResNet50. Initially, it passes through multiple layers of convolution and pooling operations. Additionally, dilated convolutions are applied to enhance the receptive field without increasing the number of parameters significantly. The stride of 16 is then used to reduce the dimensionality of the input image, resulting in a more compact representation. This series of operations ultimately produces a feature vector with dimensions of $16 \times 16 \times 2048$, effectively capturing essential features while maintaining computational efficiency and a wide context range. Then, we applied 2D adaptive average pooling to the feature vector and passed it to the projection head. The projector network comprises two linear layers, each with a hidden layer size of

512 output units. Due to high computational requirements, the output of the projection network was modified to achieve an embedding size of 256, while the BT network had an embedding size of 8192 [25]. The first layer of the projector was followed by a batch normalization layer and rectified linear unit. Afterward, we applied the absolute difference to the output embeddings to obtain the difference embeddings. We used the BT loss function on the difference embeddings of size 1×256 to generate a cross-correlation matrix of size 256×256. The Siamese encoder (E_θ), which has shared parameters θ, was employed to extract the feature vectors E'_1 and E'_2 from bi-temporal images. Next, nonlinear projection G_ϕ was applied to the encoded feature vectors E'_1 and E'_2 to produce the representations Z'_1 and Z'_2. Then, the absolute feature difference was applied to the projection outputs to learn representations of changed features between input images and is described as follows:

$$\begin{aligned} D_1 &= \left|Z'_1 - Z'_2\right| = \left|G\left(E\left(X'_1\right)\right) - G\left(E\left(X'_2\right)\right)\right| \\ D_2 &= \left|Z''_1 - Z''_2\right| = \left|G\left(E\left(X''_1\right)\right) - G\left(E\left(X''_2\right)\right)\right| \end{aligned} \tag{5}$$

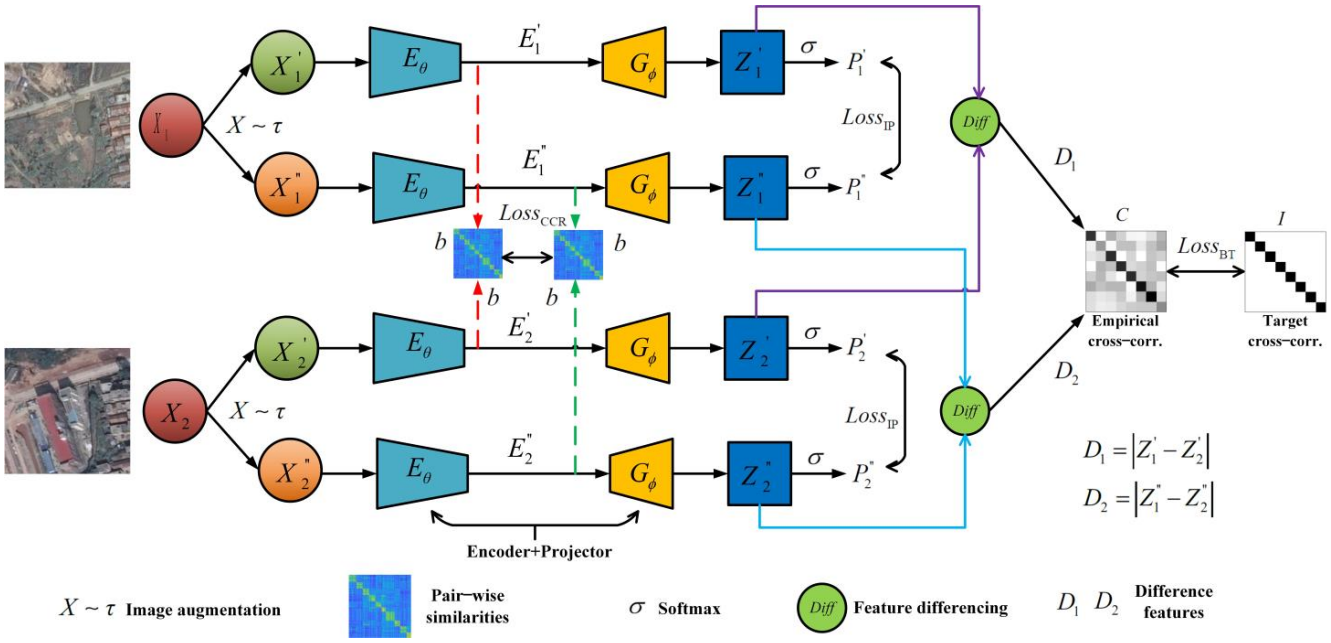


Figure 2. Flow chart of the BTCD algorithm.

We assumed that the semantic changes between the bi-temporal image features (X'_1, X'_2) and (X''_1, X''_2) should remain consistent. Therefore, we constrained our model to ensure $D_1 \simeq D_2$, regardless of the image enhancement method applied. To achieve this, we applied the BT objective function (Section 2.2.2) on the difference feature representations D_1 and D_2 to maximize the cross-correlation of the change features. This encouraged the model to learn specific information about the changes that occur between the bi-temporal images. After the self-supervised pre-training stage was complete, the encoder (E_Φ) parameters were passed on to the downstream CD task.

2.2.2. Loss Function

BT Loss. The algorithm uses the BT loss function proposed by Jure et al. [25] for self-supervised training. However, unlike those authors, who maximized the cross-correlation of the augmented views of the same input image to be closer to the identity matrix, we maximized the information of the difference representations (D_1, D_2) between the corresponding bitemporal images to be similar in the feature space. Afterwards, the model

was trained in a self-supervised manner using the modified BT loss function $Loss_{\mathrm{BT}}$, which is defined as follows:

$$Loss_{\mathrm{BT}} \triangleq \underbrace{\sum_{i}(1-C_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_{i}\sum_{j\neq i} C_{ij}^2}_{\text{redundancy reduction term}} \tag{6}$$

$$C_{ij} \triangleq \frac{\sum_b (D_1)_{b,i}(D_2)_{b,j}}{\sqrt{\sum_b \left((D_1)_{b,i}\right)^2}\sqrt{\sum_b \left((D_2)_{b,j}\right)^2}} \tag{7}$$

where λ is the trade-off constant, b indexes batch samples, i, j index the vector dimensions of the network outputs, and C is the cross-correlation matrix calculated between the difference representations D_1 and D_2.

IP Loss. When performing CD using bi-temporal imagery, we ensured that samples with semantic similarity produced similar predictions, regardless of different radiation conditions. To achieve IP, we enhanced the features between the augmented views (X_1', X_1'') and (X_2', X_2'') of the bi-temporal imagery. To accomplish this, we used Jensen-Shannon divergence D_{JS} as the objective to achieve IP [28].

$$D_{\mathrm{JS}}^{X_1} = \frac{1}{2} \times \left(D_{\mathrm{KL}}(\sigma(Z_1'))||AP_1 + D_{\mathrm{KL}}(\sigma(Z_1''))||AP_1\right) = \frac{1}{2} \times \left(D_{\mathrm{KL}}(P_1')||AP_1 + D_{\mathrm{KL}}(P_1'')||AP_1\right) \tag{8}$$

$$D_{\mathrm{JS}}^{X_2} = \frac{1}{2} \times \left(D_{\mathrm{KL}}(\sigma(Z_2'))||AP_2 + D_{\mathrm{KL}}(\sigma(Z_2''))||AP_2\right) = \frac{1}{2} \times \left(D_{\mathrm{KL}}(P_2')||AP_2 + D_{\mathrm{KL}}(P_2'')||AP_2\right) \tag{9}$$

$$AP_1 = \frac{P_1' + P_1''}{2} \tag{10}$$

$$AP_2 = \frac{P_2' + P_2''}{2} \tag{11}$$

$$Loss_{\mathrm{IP}} = D_{\mathrm{JS}}^{X_1} + D_{\mathrm{JS}}^{X_2} \tag{12}$$

where σ is the softmax function, D_{KL} is the Kullback–Leibler divergence [29], and AP_1 and AP_2 are the average probabilities belonging to the corresponding features. We utilized the loss function $Loss_{\mathrm{IP}}$ during network training to compel the model to learn features that are resilient to radiometric differences between bi-temporal images.

CCR Loss. CD is challenging because bi-temporal images can be heavily influenced by noise, seasonal changes, and differences in imaging angles. Therefore, it is critical to strengthen the similarity between images in the feature space, because the probability of change between images is typically lower than the probability of no change. To strengthen the similarity between images, we developed a CCR that preserved the semantic similarity between the temporal image pairs in the feature space. Specifically, the network encoder in the BTCD algorithm was coerced into producing similar activations for the bi-temporal images, thus incorporating temporal invariance into the network model. This, in turn, implicitly enhanced the model's robustness to noise variation during the pre-training stage. The BTCD network encoder generated activation maps from input images X_1 that underwent augmented transformations. These activation maps are represented by $E_1' \in \mathbb{R}^{b\times c\times h\times w}$ and $E_1'' \in \mathbb{R}^{b\times c\times h\times w}$, where b denotes the batch size, c denotes the number of output channels, and h and w denote the spatial dimensions. Similarly, the network encoder generated activation maps for transformed X_2 images, represented by E_2' and E_2''. To maintain temporal consistency between the input images X_1 and X_2 in the feature space, we constructed a pairwise similarity matrix $G' = \langle E_1', E_2' \rangle$ between the feature maps E_1'

and E_2'. Similarly, we construct a pairwise similarity matrix $G'' = \langle E_1'', E_2'' \rangle$ between the feature maps E_1'' and E_2''. We defined the CCR loss function as follows:

$$Loss_{\text{CCR}} = \frac{\|G' - G''\|_F^2}{b^2} \tag{13}$$

where $\|.\|_F$ is the Frobenius norm and $\langle ., . \rangle$ is the dot product [30].

Overall Loss. As shown in Figure 2, the output of the projector was utilized to calculate the IP loss before the differences are computed, while the CCR loss was applied to the output of the feature extractor. We developed a weighted combination of three loss functions (BT, IP, and CCR) to derive the final network loss function:

$$Loss_{\text{Total}} = Loss_{\text{BT}} + \alpha Loss_{\text{IP}} + \beta Loss_{\text{CCR}} \tag{14}$$

where α and β are the loss balancing weights.

2.3. CD Algorithm

We evaluated the performance of a self-supervised pre-trained model through fine-tuning using an existing CNN-based CD algorithm. We selected the improved UNet++ model [4] due to its ability to achieve SOTA results with CD datasets. The improved UNet++ model can capture multi-scale features using skip connections and integrate them via feature superposition. It utilizes an encoder–decoder architecture that includes a spatial and channel squeeze and excitation (scSE) module [31]. To train the model, the bands of the bi-temporal images were superimposed to create a new image with twice the number of bands, which was then fed into the improved UNet++ model. This approach has the advantage of allowing the model to learn both low- and high-level features of the input images, which is not possible with Siamese networks. It also simplifies the CD problem into a semantic segmentation task.

2.4. Experimental Datasets

To assess the BTCD algorithm, we utilized two publicly available datasets, LEVIR-CD [12] and SYSU-CD [32]. LEVIR-CD consists of 637 pairs of highly detailed very high-resolution (0.5 m/pixel) Google Earth image patches, each measuring 1024 × 1024 pixels. These bi-temporal images were collected from 20 distinct areas across Texas, USA, spanning the years 2002 to 2018. Images from different regions may have been captured at different times, ranging from 5 to 14 years apart, and exhibit notable changes in land use, particularly in building structures. LEVIR-CD encompasses various types of constructions, including villa residences, high-rise apartments, small-scale garages, and large warehouses, with a specific focus on changes related to buildings, such as growth and demolition [12]. To prepare for training, we randomly partitioned the images into three non-overlapping sets: a training set of 7120 image pairs, a validation set of 1024 image pairs, and a testing set of 2048 image pairs, all of which were further divided into patches of size 256 × 256. Examples of the LEVIR-CD dataset are shown in Figure 3. The SYSU-CD dataset comprises 20,000 pairs of aerial images, each measuring 256 × 256 pixels, captured in Hong Kong between 2007 and 2014, with a resolution of 0.5 m per pixel. The primary types of changes documented in the dataset include (a) newly erected urban structures; (b) the expansion of suburban areas; (c) preliminary groundwork prior to construction; (d) alterations in vegetation cover; (e) widening of roads; and (f) construction activities in marine environments [32]. We used 12,000 image pairs for training, 4000 for validation, and 4000 for testing. Examples of the SYSU-CD dataset are shown in Figure 4.

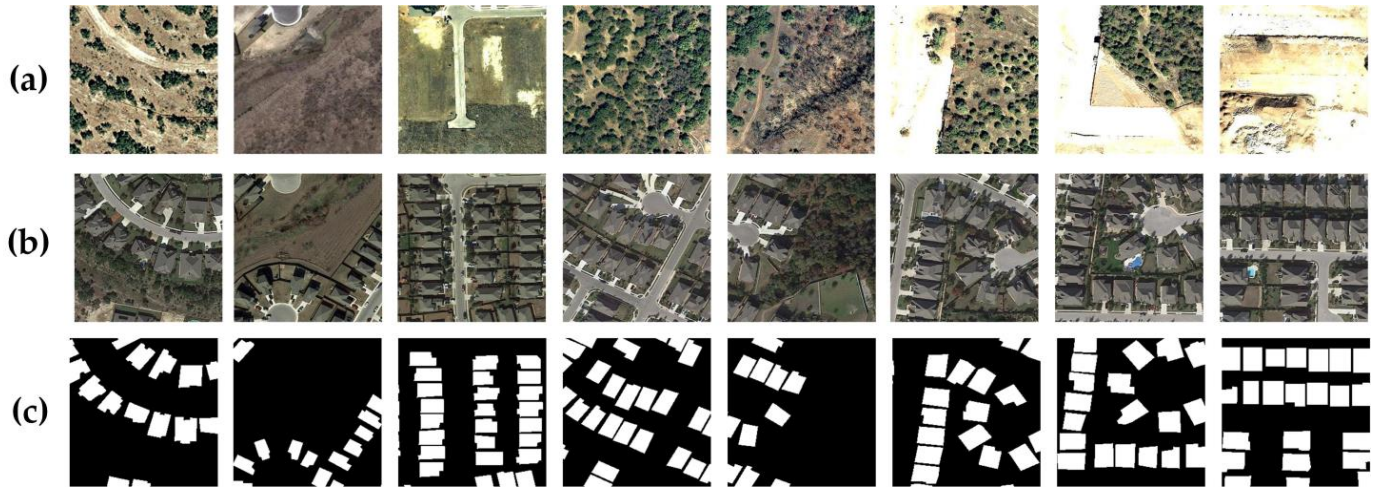


Figure 3. Examples of the LEVIR-CD dataset. (**a**) Remote sensing images at time T1. (**b**) Remote sensing images at time T2. (**c**) Ground truth.



Figure 4. Examples of the SYSU-CD dataset. (**a**) Remote sensing images at time T1. (**b**) Remote sensing images at time T2. (**c**) Ground truth.

2.5. Implementation Details and Evaluation Metrics

We utilized the PyTorch framework to implement the BTCD algorithm and conducted experiments on a workstation featuring a 12th Gen Intel Core i9–12900K @ 3.19 GHz processor, 64 GB RAM, and a NVIDIA GeForce RTX 3090 (24 GB) graphics card. To optimize the model, we followed the BT protocol of two previous studies [25,33]. During the SSCL stage, we trained the ResNet50 model using image pairs from the training and validation sets without labels. We set the batch size to 40 and used the LARS optimizer to train for 400 epochs. The initial learning rate was 0.003, which was adjusted through multiplication with the batch size and division by 256. We incorporated a learning rate warm-up period of 10 epochs, followed by a reduction in the learning rate by a factor of 1000 using a cosine decay schedule following Ramkumar et al. [33]. The trade-off parameter λ of the loss function was set to $\lambda = 5 \times 10^{-3}$, and we used a weight decay parameter value of 1.5×10^{-6}. We set the hyperparameters α and β to 100 and 2000, respectively. The sensitivity analysis results for these hyperparameters are presented in the Section 4. Further experimental details can be found in Section 4.2.

We assessed the performance of the BTCD algorithm on downstream CD tasks through fine-tuning. For consistency, we used ResNet50 as the feature extractor and fine-tuned with

the SOTA UNet++ model. We applied data augmentation techniques, including random flipping, clipping, and Gaussian blurring. We employed the AdamW optimizer with an initial learning rate of 0.003, batch size of 30, and 150 epochs. We employed the hybrid loss function developed in a previous study [34], which combines weighted cross-entropy and dice loss with equal weights. In the CD stage, we also applied the cosine decay schedule. We compared the performance of our algorithm to SOTA methods using four metrics: precision, recall, F1 score, and intersection over union (IoU). These metrics are defined as follows:

$$\text{precision} = \frac{TP}{TP + FP} \tag{15}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{16}$$

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{17}$$

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{18}$$

where TP denotes true-positive values, TN denotes true-negative values, FP denotes false-positive values, and FN denotes false-negative values.

2.6. Comparison to SOTA Approaches

We compared our proposed method for CD with several SOTA approaches, including FC-EF [10], FC-Siam-Diff [10], FC-Siam-Conc [10], DASNet [11], STANet [12], SNUNet [34], DSIFN [35], DSAMNet [32], SRCDNet [36], BiDateNet [37], and the latest transformer-based methods BIT [38], ChangeFormer [39], MSTDSNet-CD [40], and Hybrid-TransCD [41]. We fine-tuned the improved UNet++ model on the LEVIR-CD and SYSU-CD datasets using three strategies: random initialization (Rand-init), supervised ImageNet pre-training (ImageNet-sup), and pre-training with BTCD. To evaluate the effectiveness of the proposed pre-training method, we conducted a detailed comparative analysis with several other self-supervised pre-training approaches, namely CMC [19], MoCo v2 [22], SimCLR [21], and BT [25].

3. Experimental Results

3.1. Performance Comparison for the LEVIR-CD Dataset

Our method achieved the highest F1 score (91.30%) and IoU (83.81%) among all tested methods (Table 1). Hybrid-TransCD and ChangeFormer had the second- and third-highest F1 score (90.06% and 90.04%, respectively) and the third- and second-highest IoU (81.92% and 82.48%, respectively). Both methods use self-attention mechanisms, which can effectively capture global information and map it to multiple spaces using Multi-Head, thereby enhancing the models' expressive power. Thus, Hybrid-TransCD and ChangeFormer achieved better CD results than CNN-based methods.

Our developed method does not employ transformer-based global self-attention mechanisms but instead uses the improved UNet++ model framework to address the problem of varying change area scales. In the encoder, we used an efficient ResNet 50, while the decoder embedded the scSE module. The dense skip connections in the improved UNet++ model effectively captured multi-scale and multi-level feature information, ensuring the efficient utilization of fine-grained local features. This enhanced the model's ability to perceive change areas. This approach was more suitable for handling local variations in images compared to the global self-attention mechanism of transformers, which may fall short in capturing subtle local features. Furthermore, the scSE module applied attention weighting to the spatial and channel dimensions of the feature map, highlighting important features while suppressing irrelevant information. This module enhanced the utilization of more useful and discriminative features, enabling the generation of more accurate feature maps of the change area. This demonstrates that the combination of dense skip connections

in the improved UNet++ and the scSE module can effectively improve the accuracy of CD results. For the LEVIR-CD dataset, we found that ImageNet pre-training did not significantly improve the accuracy of CD results compared to random initialization. By contrast, our SSCL method, which involved BTCD, outperformed ImageNet pre-training.

Table 1. Performance comparison for the LEVIR-CD dataset.

Methods	LEVIR-CD			
	Precision	**Recall**	**F1**	**IoU**
FC-EF	86.91	80.17	83.40	71.53
FC-Siam-Diff	89.53	83.31	86.31	75.92
FC-Siam-Conc	91.99	76.77	83.69	71.96
DASNet	80.76	79.53	79.91	74.65
STANet	83.81	**91.00**	87.26	77.40
DSIFN	**94.02**	82.93	88.13	78.77
SNUNet	89.18	87.17	88.16	78.83
BiDateNet	85.65	89.98	87.76	78.19
BIT	89.24	89.37	89.31	80.68
ChangeFormer	**92.05**	88.80	**90.04**	**82.48**
MSTDSNet-CD	85.52	**90.84**	88.10	78.73
Hybrid-TransCD	91.45	88.72	**90.06**	**81.92**
Ours (UNet++ with Rand-init)	88.89	86.74	87.80	78.25
Ours (UNet++ with ImageNet-sup)	89.99	85.92	87.91	78.43
Ours (UNet++ with BTCD)	**92.03**	**90.59**	**91.30**	**83.81**

Developed as part of the present study. All values are percentages. Bold red text indicates highest, bold blue text indicates second-highest, and bold black text indicates third-highest performances.

To evaluate the effectiveness of different pre-training methods, we fine-tuned downstream CD network (the improved UNet++) using the pre-trained models CMC, MoCo v2, SimCLR, and BT. To ensure a fair comparison, we only used the pre-trained model parameters to initialize the backbone part of the improved UNet++ model, i.e., the ResNet 50 encoder's parameters. Furthermore, we compared our BTCD method with several mainstream self-supervised pre-training methods, including random initialization, and ImageNet pre-training. These mainstream self-supervised pre-training methods were pre-trained using the ResNet-50 model on the ImageNet dataset and achieved good parameter initialization. We fine-tuned the encoder part of the ResNet-50 model obtained by these methods and used it for downstream CD tasks. Our pre-training method achieved the highest F1 score (91.30%) and IoU (83.81%) (Table 2). MoCo v2 and BT had the second- and third-highest F1 scores (89.19% and 89.06%, respectively) and IoU (80.49% and 80.28%, respectively) (Table 2). Visual comparison results for the LEVIR-CD dataset are shown in Figure 5. These scenes reflect the growth of buildings, mainly including changes from soil, grassland, and paved ground to new building areas. Because CMC, MoCo v2, SimCLR, and BT are all self-supervised pre-training methods based on the ImageNet dataset, they lack prior knowledge about CD tasks, which, to some extent, limits the performance of CD models. Considering the domain gap between ImageNet and remote sensing images, our BTCD method directly trained on the LEVIR-CD training dataset for self-supervised contrastive pre-training to learn representations within the domain. The results show that this pre-training strategy is more suitable for downstream CD tasks.

Table 2. Performance of the improved UNet++ model on the LEVIR-CD dataset using different pre-training methods.

Methods			LEVIR-CD			
			Precision	**Recall**	**F1**	**IoU**
Backbone ResNet50	Reliant on the ImageNet dataset	CMC	**89.92**	86.32	88.08	78.71
		MoCo v2	**90.32**	88.09	**89.19**	**80.49**
		SimCLR	89.23	**88.49**	88.86	79.96
		BT	89.86	**88.28**	**89.06**	**80.28**
	Not reliant on the ImageNet dataset	BTCD	**92.03**	**90.59**	**91.30**	**83.81**

All values are percentages. Bold red text indicates highest, bold blue text indicates second-highest, and bold black text indicates third-highest performances.

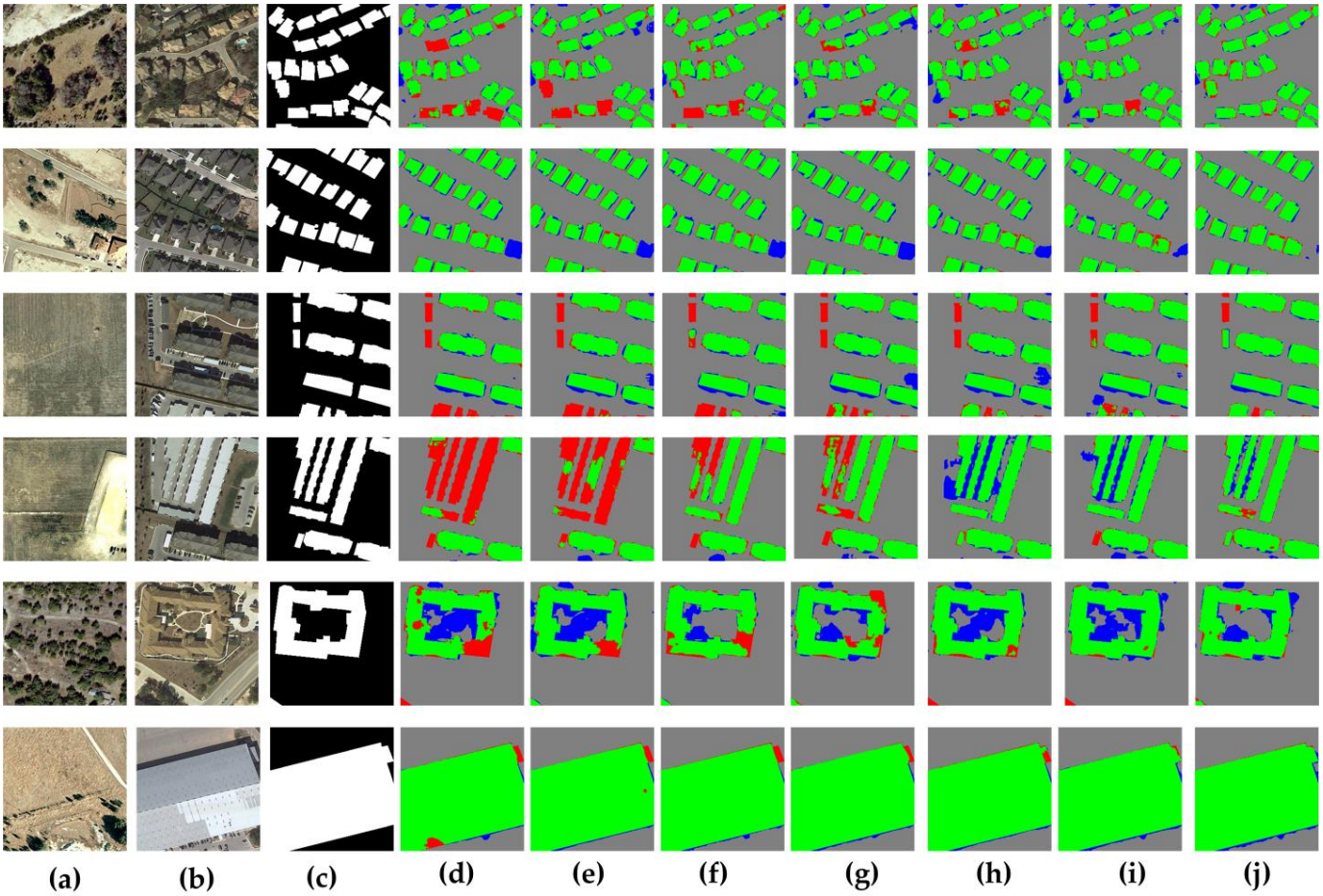


Figure 5. Visual comparisons of the UNet++ model applied to the LEVIR-CD dataset using different pre-training methods. (**a**) Images at time T1, (**b**) images at time T2; (**c**) Ground Truth; (**d**) Rand-init; (**e**) ImageNet-sup; (**f**) CMC; (**g**) SimCLR; (**h**) BT; (**i**) MoCo v2; and (**j**) BTCD. Gray: *TN* pixels; Green: *TP* pixels; Blue: *FP* pixels; Red: *FN* pixels.

3.2. Performance Comparison for the SYSU-CD Dataset

Our method achieved the highest F1 score (81.69%) and IoU value (69.04%). MSTDSNet-CD ranked second with an F1 score of 80.33% and an IoU of 67.13%, while Hybrid-TransCD ranked third with an F1 score of 80.13% and an IoU of 66.84%. MSTDSNet-CD combines a multi-scale Swin Transformer and a deep supervision network for CD tasks, while the Hybrid-TransCD method adopts a hybrid multi-scale transformer module that simulates the multi-scale mixed attention representations of bi-temporal images using fine-grained

self-attention mechanisms. Both methods can enhance the models' expressive ability, thus achieving better CD results compared to CNN-based methods.

For the SYSU-CD dataset, the F1 score of the ImageNet pre-training method was slightly higher than that of the random initialization method, but the difference was not significant. Due to the significant distribution difference between the ImageNet dataset and the SYSU-CD dataset, domain shift occurs. Therefore, training a universal model in an unsupervised manner is more effective and robust for downstream CD tasks. Obtaining an ImageNet pre-trained model requires millions of labelled samples for supervised learning. The results in Table 3 further demonstrate that our BTCD method achieved the best detection performance and higher accuracy compared to other methods.

Table 3. Performance comparison for the SYSU-CD dataset.

Methods	SYSU-CD			
	Precision	**Recall**	**F1**	**IoU**
FC-EF	74.32	75.84	75.07	60.09
FC-Siam-Diff	**89.13**	61.21	72.57	56.96
FC-Siam-Conc	82.54	71.03	76.35	61.75
DASNet	68.14	70.01	69.14	60.65
STANet	70.76	**85.33**	77.37	63.09
DSAMNet	74.81	**81.86**	78.18	64.18
SRCDNet	75.54	**81.06**	78.20	64.21
BiDateNet	81.84	72.60	76.94	62.52
BIT	82.18	74.49	78.15	64.13
MSTDSNet-CD	79.91	80.76	**80.33**	**67.13**
Hybrid-TransCD	**83.05**	77.40	**80.13**	**66.84**
Ours (UNet++ with Rand-init)	80.67	75.82	78.17	64.16
Ours (UNet++ with ImageNet-sup)	80.92	77.94	79.40	65.84
Ours (UNet++ with BTCD)	**85.80**	77.95	**81.69**	**69.04**

Developed as part of the present study. All values are percentages. Bold red text indicates highest, bold blue text indicates second-highest, and bold black text indicates third-highest performances.

For the SYSU-CD dataset, we fine-tuned downstream CD networks (the improved UNet++) using the pre-trained models CMC, MoCo v2, SimCLR, and BT. Our BTCD pre-training method achieved the highest F1 score (81.69%) and IoU (69.04%) (Table 4). MoCo v2 method ranked second with an F1 score of 81.31% and an IoU of 68.50%, while BT ranked third with an F1 score of 80.35%) and an IoU of 67.19%. The visual comparison results for the SYSU-CD dataset are shown in Figure 6. These scenes depict various types of changes, including newly constructed urban buildings, preliminary groundwork before construction, and offshore construction activities. The CD results obtained using methods based on random initialization, ImageNet-sup, and CMC had more omissions (columns 4–6 in Figure 6). By contrast, MoCo v2 and BTCD performed better and provided more complete detection results. Rows 1–3 in Figure 6 demonstrate that for complex CD scenarios, our BTCD method was better able to detect the contour information than the other tested methods. Overall, our BTCD method had fewer errors and greater robustness than all other tested methods, providing superior performance for CD tasks.

Table 4. Performance of the improved UNet++ model on the SYSU-CD dataset using different pre-training methods.

Methods			SYSU-CD			
			Precision	**Recall**	**F1**	**IoU**
Backbone ResNet50	Reliant on the ImageNet dataset	CMC	**83.69**	76.26	79.80	66.39
		MoCo v2	**82.45**	**80.19**	**81.31**	**68.50**
		SimCLR	82.33	76.84	79.49	65.96
		BT	82.03	**78.71**	**80.35**	**67.19**
	Not reliant on the ImageNet dataset	BTCD	**85.80**	**77.95**	**81.69**	**69.04**

All values are percentages. Bold red text indicates highest, bold blue text indicates second-highest, and bold black text indicates third-highest performances.

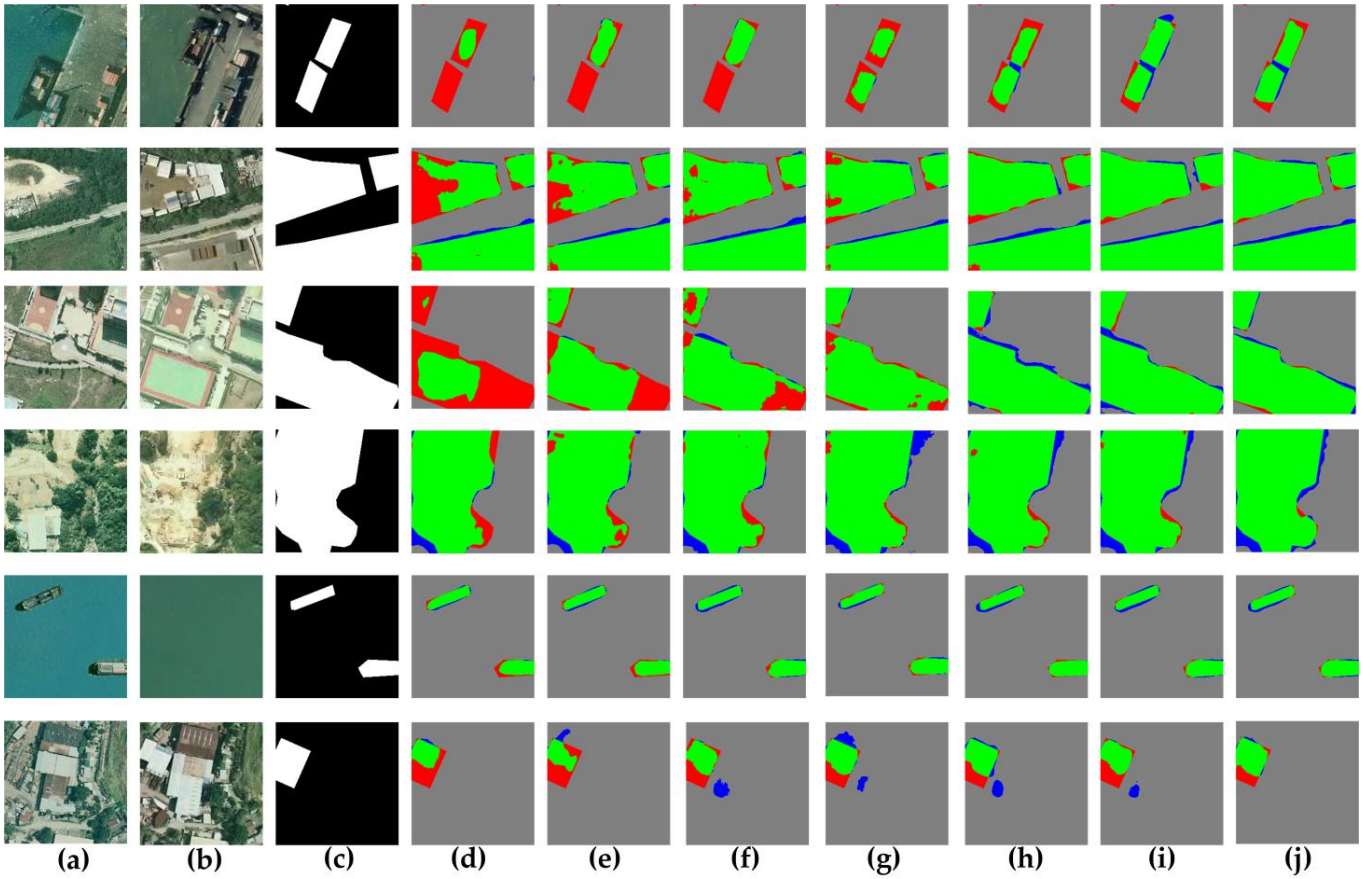


Figure 6. Visual comparisons of the UNet++ model applied to the SYSU-CD dataset using different pre-training methods. (**a**) Images at time T1, (**b**) Images at time T2; (**c**) ground truth; (**d**) Rand-init; (**e**) ImageNet-sup; (**f**) CMC; (**g**) SimCLR; (**h**) BT; (**i**) MoCo v2; and (**j**) BTCD. Gray: *TN* pixels; green: *TP* pixels; blue: *FP* pixels; red: *FN* pixels.

4. Discussion

4.1. Ablation Experiment

We fine-tuned the UNet++ model on the LEVIR-CD and SYSU-CD datasets through three strategies: Rand-init, ImageNet-sup, and pretraining with BTCD. To gauge the efficacy of our proposed pretraining method, we conducted a comprehensive comparative analysis against several other self-supervised pretraining approaches, including CMC, MoCo v2, SimCLR, and BT. According to the results in Table 5, our proposed pretraining method (BTCD) yields the best performance. Compared to Rand-init, applying our BTCD pretraining process increases the F1 score of LEVIR-CD by nearly 3.50%, and the IoU by

approximately 5.56%; for SYSU-CD, the F1 score rises by around 3.52%, with IoU increasing by approximately 4.88%. On the ImageNet dataset, utilizing CMC, MoCo v2, SimCLR, and BT for self-supervised contrastive pretraining outperforms ImageNet-sup when transferred to LEVIR-CD and SYSU-CD datasets in terms of F1 score and IoU. Specifically, on the LEVIR-CD dataset, our proposed pretraining method surpasses the best self-supervised contrastive pretraining method, MoCo v2, by 2.11% in performance and increases IoU by nearly 3.32%. For the SYSU-CD dataset, our proposed pretraining method is roughly equivalent to MoCo v2 in terms of F1 score and IoU, with slight improvement. In summary, ablation experiments demonstrate that our proposed self-supervised contrastive pretraining method for unlabeled images can achieve or even surpass widely used pretraining methods based on the ImageNet dataset. Furthermore, our method mitigates the domain shift problem caused by transferring knowledge from ImageNet weights obtained through pretraining with datasets vastly different from CD datasets. Qualitative and quantitative comparisons indicate that our proposed BTCD exhibits significant effectiveness and superiority for CD tasks.

Table 5. Results of ablation experiments on LEVIR-CD and SYSU-CD datasets.

Methods		Pretraining	ImageNet Dataset	LEVIR-CD		SYSU-CD	
				F1	IoU	F1	IoU
Backbone ResNet50	Rand-init	×	×	87.80	78.25	78.17	64.16
	ImageNet-sup	✓	✓	87.91	78.43	79.40	65.84
	CMC	✓	✓	88.08	78.71	79.80	66.39
	MoCo v2	✓	✓	89.19	80.49	81.31	68.50
	SimCLR	✓	✓	88.86	79.96	79.49	65.96
	BT	✓	✓	89.06	80.28	80.35	67.19
	BTCD	✓	×	91.30	83.81	81.69	69.04

All values are percentages. × signifies steps excluded during the training process, while ✓ denotes their inclusion.

4.2. Hyperparameter Sensitivity Analysis

Inspired by the observation that the predictions of semantically similar remote sensing images should be similar irrespective of different lighting conditions, to achieve invariant prediction, we utilized the IP loss function to compel the BTCD model to learn features unaffected by imaging lighting conditions. Additionally, in order to mitigate the impact of lighting, noise, seasonal variations, and differences in remote sensing imaging angles on CD results, we employed the CCR Loss function. This dual approach not only enhances the alignment of bi-temporal images in both decision and feature spaces but also diminishes the influence of these confounding factors on CD results. To further analyze the impact of different loss balance hyperparameter settings on the final results, akin to the approach by Tung and Mori [30], we determined the CD accuracy results of different hyperparameter settings for the SYSU-CD dataset (Table 6). To evaluate the performance of the IP loss, we kept the coefficient β of the CCR loss constant and varied the α value. Increasing the α value had a limited impact on the downstream CD task performance of the BTCD method (Table 6). Consequently, we fixed the α value at 100 and further analyzed the final impact on CD task accuracy by changing the coefficient of the CCR loss, i.e., by modifying β. Because the scale of the CCR loss is smaller compared to the BT loss and IP loss, we increased β in increments of 1000 to balance losses on the same scale. Increasing β initially slightly improved the performance (Table 6). When $\alpha = 100$ and $\beta = 2000$, the BTCD method achieved the highest F1 score and IoU for the SYSU-CD dataset. After further increasing β, the effect reached saturation. Therefore, the optimal settings for α and β for the SYSU-CD dataset were 100 and 2000, respectively. Subsequently, we also applied the same loss function hyperparameter values to the LEVIR-CD dataset and obtained good CD results. This suggests that the performance of the BTCD algorithm developed in the present paper is not too sensitive to the hyperparameters of the final loss function. Furthermore,

the parameters can be further optimized by cross-validating the validation sets of the two CD datasets.

Table 6. Table of sensitivity analysis for different loss balance hyperparameter settings for the SYSU-CD dataset.

Method	α	β	F1 (%)	IoU (%)	Method	α	β	F1 (%)	IoU (%)
BTCD	100	0	80.28	67.06	BTCD	100	1000	79.42	65.86
	500	0	79.95	66.60		100	2000	81.69	69.04
	1000	0	79.68	66.23		100	3000	80.35	67.19

4.3. Efficiency under Limited Labels

Manually marking areas of change is an expensive process that requires a significant amount of manpower and resources. Therefore, the lack of access to large-scale annotated data remains a major challenge in using bi-temporal remote sensing images for CD. We tested multiple proportions of training data (20%, 40%, 60%, and 100%) and attempted various pre-training methods for the LEVIR-CD and SYSU-CD datasets. Compared to the pre-training methods Rand-init, ImageNet-sup, CMC, MoCO v2, SimCLR, and BT, our BTCD pre-training method performed well on both datasets, even with limited annotated data (Figure 7). The F1 score gradually improved as the percentage of training data increased. When using 60% of the training data, the BTCD method achieved F1 scores of 0.8973 and 0.788 when applied to the LEVIR-CD and SYSU-CD datasets, respectively, both of which were higher than the F1 scores of CNN-based methods using all training data. When the percentage of training data was further reduced to 40%, BTCD achieved an F1 score of 0.8943 when applied to the LEVIR-CD dataset, which was only slightly lower than those achieved by ChangeFormer and Hybrid-TransCD, and an F1 score of 0.7787 when applied to the SYSU-CD dataset, which was comparable to the performance metrics of DSAMNet, SRCDNet, and BIT. When the percentage of training data was reduced to 20%, BTCD achieved F1 scores of 0.8486 and 0.7504 when applied to the LEVIR-CD and SYSU-CD datasets, respectively, which were slightly higher than F1 scores of FC-EF, FC-Siam-Diff, FC-Siam-Conc, and DASNet. We demonstrate that our developed BTCD pre-training method can significantly improve the generalization ability and data efficiency of downstream CD models. It is worth noting that we achieved competitive results using only 20% of the training samples for model training. In summary, when annotated data for CD are limited, our pre-training method can significantly enhance the robustness and general performance of CD models.

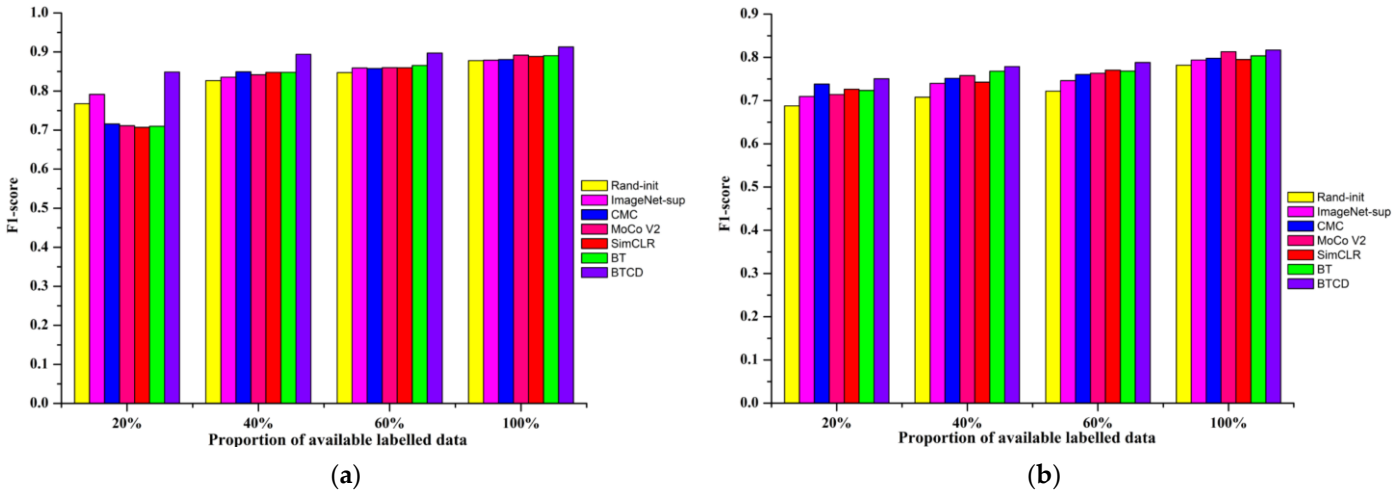


Figure 7. Performance (F1-score) of different pre-training methods evaluated using the improved UNet++ model under varying label availability. (**a**) LEVIR-CD. (**b**) SYSU-CD.

4.4. Evaluation of Model Generalization Performance

The LULC CD mandates that models exhibit genuine generalization capabilities. To validate the generalization performance of the proposed BTCD framework, we conducted experiments on the extensive WHU Building CD (WHU-BCD) dataset [42]. This dataset encompasses an area impacted by a 6.3-magnitude earthquake in February 2011, which was subsequently reconstructed over the following years. Our research focuses on Christchurch, New Zealand, a city that has seen significant new construction since the earthquake. The dataset includes aerial images captured in April 2012, covering 20.5 km^2 and featuring 12,796 buildings (16,077 buildings in the 2016 dataset within the same area). The images have a ground sampling distance of 0.2 m/pixel. Additionally, the dataset provides reference change masks and a pair of co-registered aerial images with a combined size of 15,354 × 32,507 × 3 pixels. The 2012 and 2016 remote sensing images are depicted in Figure 8a and 8b, respectively, while the ground truth for building changes is shown in Figure 8c. In this figure, black areas indicate unchanged regions, and white areas denote changed building regions. We cropped the bi-temporal images into smaller patches of 256 × 256 × 3 pixels, resizing edge patches to 256 × 256 × 3 pixels, resulting in a total of 7620 pairs of image patches.

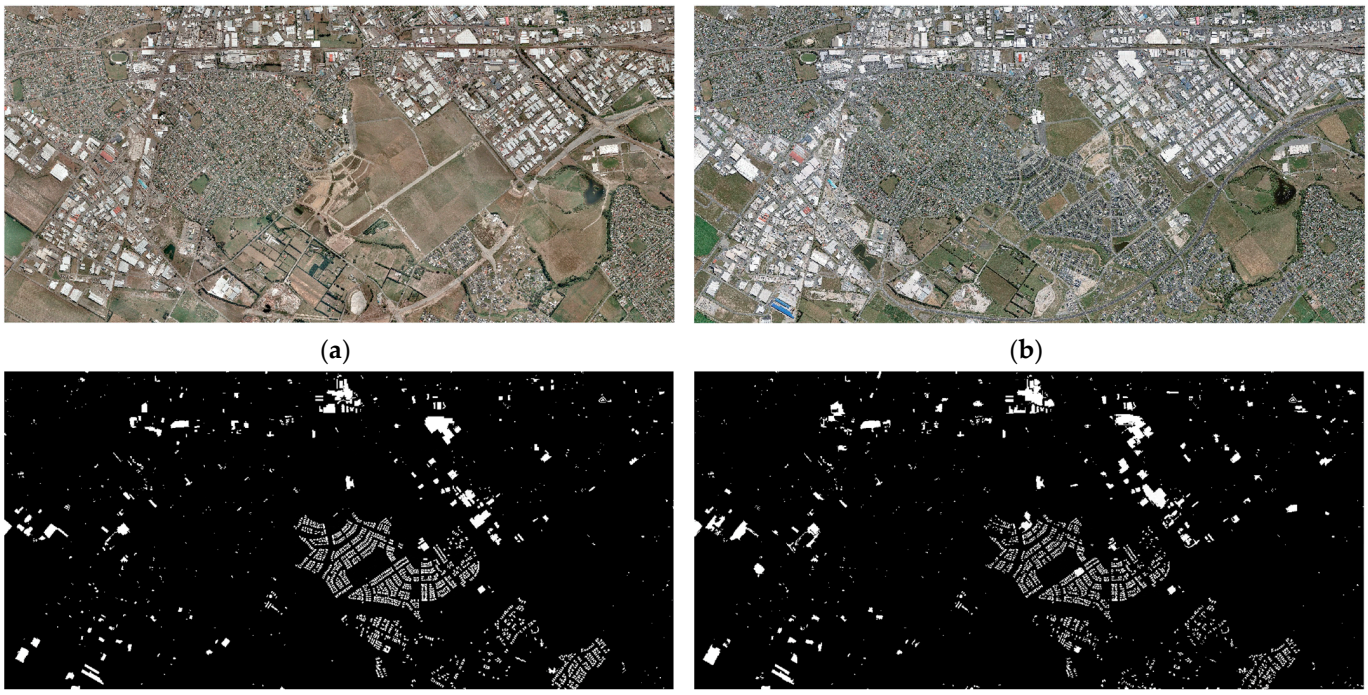


Figure 8. The WHU-BCD dataset. (**a**) An old temporal image from 2012; (**b**) a new temporal image from 2016; (**c**) ground truth; (**d**) the prediction CD map.

During the self-supervised pre-training stage, we utilized these 7620 pairs of unlabeled image patches for SSCL. SSCL involves designing an instance discrimination task to identify the most similar images to the input features within the dataset. It maximizes the similarity between two positive instances while increasing the distance between positive and negative instances. In the fine-tuning CD stage, the trained encoder, equipped with robust feature representations, is transferred to the downstream CD task. We conducted experimental analysis on the WHU-BCD dataset using the improved UNet++ model. The results were stitched together based on the relative positions of the cropped image patches, and the edge patches were resized back to their original size. The final experimental results on the WHU-BCD dataset are shown in Figure 8d. We evaluated the accuracy of the results, yielding a precision of 0.7674, a recall of 0.7771, an F1 score of 0.7722, and an IoU of 0.6289.

Both quantitative and qualitative experimental outcomes demonstrate that the proposed BTCD framework possesses strong generalization performance in practical applications.

Furthermore, it is noteworthy that our method was pre-trained on RGB images due to the nature of the two public datasets employed, which only include RGB imagery. Consequently, we had to exclude information beyond the visible light spectrum. However, in practical scenarios, most remote sensing images encompass not only RGB bands but also information beyond the visible light spectrum. This limitation affects the application of our method to multispectral satellite image data such as that from Sentinel-2 and Landsat. In future work, we plan to extend the proposed BTCD algorithm and investigate various strategies for adapting from RGB images to multispectral images, thereby enhancing the method's broad applicability.

5. Conclusions

When performing LULC CD in bi-temporal remote sensing images, deep learning models based on supervised paradigms require a large amount of annotated data. Unfortunately, collecting and annotating samples that contain the desired areas of change is both time-consuming and labor-intensive. Transfer learning of pre-trained models is an effective method that can alleviate this problem. In the present study, we developed the BTCD framework, which is composed of a self-supervised contrastive pre-training stage and a fine-tuning stage and can learn the inherent differential features of original bi-temporal remote sensing images in a self-supervised manner. Our method can be easily integrated with existing SOTA CD methods. We conducted extensive analyses using two public remote sensing CD datasets and demonstrated the superiority of BTCD over other commonly used ImageNet pre-training and self-supervised pre-training methods. In addition, BTCD does not require additional image data. Our proposed self-supervised pre-training strategy is effective even when using a limited sample size of labelled data, which is particularly valuable for CD applications for which obtaining labelled data for changed regions is difficult and expensive. In the future, we plan to replace the original ResNet50 encoder component of BTCD with a vision transformer to further improve the accuracy of the CD results.

Author Contributions: Conceptualization, W.F. and W.X.; methodology, W.F.; validation, F.G. and C.S.; writing—original draft preparation, W.F. and W.X.; writing—review and editing, F.G. and C.S.; supervision, F.G. and W.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant No. 42101358.

Data Availability Statement: Data associated with this research are available online. The LEVIR-CD dataset is available for download at https://justchenhao.github.io/LEVIR/, accessed on 22 May 2020. The SYSU-CD dataset is available for download at https://github.com/liumency/DSAMNet/, accessed on 4 October 2022. The WHU-BCD dataset is available for download at http://gpcv.whu.edu.cn/data/building_dataset.html, accessed on 14 April 2023.

Acknowledgments: We sincerely thank the anonymous reviewers for their critical comments and suggestions for improving the manuscript.

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

References

1. Zhang, L.; Wu, C. Advance and future development of change detection for multi-temporal remote sensing imagery. *Acta Geod. Cartogr. Sin.* **2017**, *46*, 1447–1459.
2. Sui, H.; Feng, W.; Li, W.; Sun, K.; Xu, C. Review of change detection technology for multi-temporal remote sensing imagery. *Geomat. Inf. Sci. Wuhan Univ.* **2018**, *43*, 1885–1891.
3. 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]

4. 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]
5. Geng, J.; Wang, H.; Fan, J.; Ma, X. Change detection of SAR images based on supervised contractive auto-encoders and fuzzy clustering. In Proceedings of the IEEE International Workshop on Remote Sensing with Intelligent Processing (RSIP), Shanghai, China, 19–21 May 2017.
6. Su, L.; Gong, M.; Zhang, P.; Zhang, M.; Liu, J.; Yang, H. Deep learning and mapping based ternary change detection for information unbalanced images. *Pattern Recognit.* **2017**, *66*, 213–228. [CrossRef]
7. Gong, M.; Zhang, P.; Zhan, T.; Miao, Q. Superpixel-based difference representation learning for change detection in multispectral remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2658–2673. [CrossRef]
8. Feng, G.; Dong, J.; Bo, L.; Xu, Q. Automatic Change Detection in Synthetic Aperture Radar Images Based on PCANet. *IEEE Geoence Remote Sens. Lett.* **2017**, *13*, 1792–1796.
9. Chen, H.; Wu, C.; Du, B.; Zhang, L.; Wang, L. Change detection in multisource VHR images via deep siamese convolutional multiple-layers recurrent neural network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 2848–2864. [CrossRef]
10. Daudt, R.C.; Saux, B.L.; Boulch, A. Fully convolutional Siamese networks for change detection. In Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 4063–4067.
11. Chen, J.; Yuan, Z.; Peng, J.; Chen, L.; Huang, H.; Zhu, J.; Liu, Y.; Li, H. DASNet: Dual attentive fully convolutional Siamese networks for change detection in high-resolution satellite images. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2020**, *14*, 1194–1206. [CrossRef]
12. 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]
13. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
14. Long, Y.; Xia, G.; Li, S.; Yang, W.; Yang, M.Y.; Zhu, X.; Zhang, L.; Li, D. On Creating Benchmark Dataset for Aerial Image Interpretation: Reviews, Guidances, and Million-AID. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4205–4230. [CrossRef]
15. Minetto, R.; Segundo, M.P.; Sarkar, S. Hydra: An Ensemble of Convolutional Neural Networks for Geospatial Land Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6530–6541. [CrossRef]
16. Sumbul, G.; Charfuelan, M.; Demir, B.; Markl, V. BigEarthNet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 5901–5904.
17. Manas, O.; Lacoste, A.; Giro-i-Nieto, X.; Vazquez, D.; Rodriguez, P. Seasonal contrast: Unsupervised pretraining from uncurated remote sensing data. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Los Alamitos, CA, USA, 10–17 October 2021; pp. 9394–9403.
18. Wang, Y.; Albrecht, C.M.; Braham, N.A.A.; Mou, L.; Zhu, X. Self-Supervised Learning in Remote Sensing: A review. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 213–247. [CrossRef]
19. Tian, Y.; Krishnan, D.; Isola, P. Contrastive Multiview Coding. *arXiv* **2019**, arXiv:1906.05849.
20. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15750–15758.
21. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv* **2020**, arXiv:2002.05709.
22. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv* **2019**, arXiv:1911.05722.
23. Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; Joulin, A. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. *arXiv* **2021**, arXiv:2006.09882.
24. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Pires, B.A.; Guo, Z.; Azar, M.G.; et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv* **2020**, arXiv:2006.07733.
25. Jure, Z.; Li, J.; Ishan, M.; Yann, L.C.; Stéphane, D. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv* **2021**, arXiv:2103.03230.
26. Jiang, F.; Gong, M.; Zheng, H.; Liu, T.; Zhang, M.; Liu, J. Self-Supervised Global–Local Contrastive Learning for Fine-Grained Change Detection in VHR Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4400613. [CrossRef]
27. Wang, J.; Zhong, Y.; Zhang, L. Change Detection Based on Supervised Contrastive Learning for High-Resolution Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5601816. [CrossRef]
28. Englesson, E.; Azizpour, H. Generalized jensen-shannon divergence loss for learning with noisy labels. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 30284–30297.
29. Kim, T.; Oh, J.; Kim, N.Y.; Cho, S.W.; Yun, S.Y. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *arXiv* **2021**, arXiv:2105.08919.
30. Tung, F.; Mori, G. Similarity-preserving knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1365–1374.

31. Abhijit, G.R.; Nassir, N.; Christian, W. Concurrent spatial and channel squeeze & excitation in fully convolutional networks. *arXiv* **2018**, arXiv:1803.02579.
32. 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.* **2022**, *60*, 1–16. [CrossRef]
33. Ramkumar, V.R.T.; Arani, E.; Zonooz, B. Differencing based self-supervised pretraining for scene change detection. *arXiv* **2022**, arXiv:2208.05838.
34. 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.* **2022**, *19*, 8007805. [CrossRef]
35. 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]
36. 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]
37. Papadomanolaki, M.; Verma, S.; Vakalopoulou, M.; Gupta, S.; Karantzalos, K. Detecting urban changes with recurrent neural networks from multitemporal Sentinel-2 data. In Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 214–217.
38. Chen, H.; Qi, Z.; Shi, Z. Remote Sensing Image Change Detection with Transformers. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5920416. [CrossRef]
39. Bandara, W.G.C.; Patel, V.M. A transformer-based siamese network for change detection. *arXiv* **2022**, arXiv:2201.01293.
40. Song, F.; Zhang, S.; Lei, T.; Song, Y.; Peng, Z. MSTDSNet-CD: Multiscale swin transformer and deeply supervised network for change detection of the fast-growing urban regions. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6508505. [CrossRef]
41. Ke, Q.; Zhang, P. Hybrid-transCD: A hybrid transformer remote sensing image change detection network via token aggregation. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 263. [CrossRef]
42. Ji, S.; Shen, Y.; Lu, M.; Zhang, Y. Building Instance Change Detection from Large-Scale Aerial Images using Convolutional Neural Networks and Simulated Samples. *Remote Sens.* **2019**, *11*, 1343. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.