electronics

MDPI

Article

Unsupervised and Self-Supervised Tensor Train for Change Detection in Multitemporal Hyperspectral Images

Muhammad Sohail , Haonan Wu , Zhao Chen * and Guohua Liu

School of Computer Science and Technology, Donghua University, Shanghai 201620, China;
416033@mail.dhu.edu.cn (M.S.); 2212539@mail.dhu.edu.cn (H.W.); ghliu@dhu.edu.cn (G.L.)
* Correspondence: chenzhao@dhu.edu.cn

Abstract: Remote sensing change detection (CD) using multitemporal hyperspectral images (HSIs) provides detailed information on spectral–spatial changes and is useful in a variety of applications such as environmental monitoring, urban planning, and disaster detection. However, the high dimensionality and low spatial resolution of HSIs do not only lead to expensive computation but also bring about inter-class homogeneity and inner-class heterogeneity. Meanwhile, labeled samples are difficult to obtain in reality as field investigation is expensive, which limits the application of supervised CD methods. In this paper, two algorithms for CD based on the tensor train (TT) decomposition are proposed and are called the unsupervised tensor train (UTT) and self-supervised tensor train (STT). TT uses a well-balanced matricization strategy to capture global correlations from tensors and can therefore effectively extract low-rank discriminative features, so the curse of the dimensionality and spectral variability of HSIs can be overcome. In addition, the two proposed methods are based on unsupervised and self-supervised learning, where no manual annotations are needed. Meanwhile, the ket-augmentation (KA) scheme is used to transform the low-order tensor into a high-order tensor while keeping the total number of entries the same. Therefore, high-order features with richer texture can be extracted without increasing computational complexity. Experimental results on four benchmark datasets show that the proposed methods outperformed their tensor counterpart, the tucker decomposition (TD), the higher-order singular value decomposition (HOSVD), and some other state-of-the-art approaches. For the Yancheng dataset, OA and KAPPA of UTT reached as high as 98.11% and 0.9536, respectively, while OA and KAPPA of STT were at 98.20% and 0.9561, respectively.

Keywords: multitemporal hyperspectral images; change detection; unsupervised; self-supervised; tensor train; low-rank

check for updates

Citation: Sohail, M.; Wu, H.; Chen, Z.; Liu, G. Unsupervised and Self-Supervised Tensor Train for Change Detection in Multitemporal Hyperspectral Images. *Electronics* **2022**, *11*, 1486. https://doi.org/10.3390/electronics11091486

Academic Editor: Gemma Piella

Received: 1 April 2022
Accepted: 3 May 2022
Published: 6 May 2022

1. Introduction

The ability to comprehend global change in its entirety is essential for giving timely and accurate information about the Earth [1–5]. Multitemporal remote sensing images are currently used in a wide range of change detection applications, including ecological change studies [6], natural disaster investigations [7], and especially in the urban development race [8]. Meanwhile, the high spectral resolution of hyperspectral images (HSIs) has attracted the attention of researchers in the CD community as compared to synthetic aperture radar (SAR) images and multispectral images. It is because subtler variations can be detected via HSIs with the detailed composition of various reflected objects. The abundant spectral information contained in HSIs have been made full use of in a lot of HSI CD methods [9,10]. Moreover, CD can also be regarded as a classification task where different types of changes, including changes in the background, are considered as different classes [11,12]. With the rapid developments in deep learning [13–16], satisfactory results in HSI CD have also been achieved.

However, there still exist some challenging problems in HSI CD. Apart from the useful change information, there are also a lot of redundant information contained in HSIs which may lead to pseudo changes. Meanwhile, the high dimensionality of HSIs increases the computational complexity, and the low spatial resolution can bring about inter-class homogeneity and inner-class heterogeneity. Therefore, some unsupervised methods such as K-Means [17] cannot precisely differentiate between a region that has undergone change and one which has not. Tensor-based methods [18–20] treat HSIs as three-order tensors, with two dimensions for spatial and one for spectral, and make use of tensor algebra [21,22] to extract low-rank discriminative features from the original HSIs. As the density matrices [23] that represent the quantum states are two-order tensors in nature, the effectiveness of tensor algebra in extracting discriminative features can be confirmed by virtue of matricization and quantum information theory [23]. Another challenge in HSI CD comes from the expensive cost of obtaining manual annotations, which not only requires high-level geographical expertise but also takes a lot of time. Therefore, supervised learning methods are not practical in the field of HSI CD, where accurate ground truth is necessary.

To tackle the above-mentioned problems, this study proposes two novel tensor train (TT)-based techniques for multitemporal HSI CD. The first one is an unsupervised TT-based technique called UTT. The second one is a self-supervised TT-based technique named STT. The motivations are as follows. First, tensor algebra is well-suited for HSI processing as HSIs are three-order tensors in nature. TT decomposition is one type of low-rank tensor decomposition [24] that can effectively remove the redundancy and overcome the curse of dimensionality. Second, TT can capture more global information between temporal images than TD as TT is based on a well-balanced matricization scheme [25] (k-modes versus the rest), while TD is based on an unbalanced matricization scheme (one mode versus the rest). Therefore, the features extracted by TT contain more changed information and are thus more discriminative. Third, manual annotations are difficult to obtain as field investigation is expensive and high-level geographical expertise is necessary. The proposed UTT and STT are based on unsupervised and self-supervised learning, respectively, which indicates that laborious manual annotations are not needed. Additionally, the ket-augmentation (KA) [26] scheme can obtain higher-order tensor representations of change features while retaining the total number of entries. This means that high-order rich texture features can be obtained without increasing computational complexity. In addition, TT decomposition is more efficient for the tensor augmented by KA because the local structure of the data can be exploited effectively in terms of computational resources [25].

The proposed UTT and STT work as follows. High-order difference tensors are firstly produced using KA and substraction for both UTT and STT. Then, UTT directly employs low TT-rank optimization to reconstruct the difference tensor, while STT combines clustering and classification in a self-supervised manner. Classification is used as the pretext task in order to make the features learnt friendlier to binary clustering, which is the main task. Clustering centroids and the classification network are optimized jointly in STT. For STT, clustering is used to produce the initial pseudo labels and divide the learnt features into the categories of changed or unchanged, while TT is used to extract low-rank features and perform pseudo classification. Both the UTT and STT approaches are followed by binary clustering in order to categorize the changed and unchanged features.

The novelty and contributions of this study can be summarized as follows:

(1). Inspired by the knowledge from quantum information theory, this work theoretically proves that TT decomposition exhibits greater ability than the traditional TD in capturing global correlations between changed and unchanged tensor entries. Thus, TT is used to extract spectral–spatial low-rank features for multitemporal HSI CD, which decomposes a high-order tensor into a set of low-order tensors by approximating the optimal TT rank.

(2). KA is used to obtain higher-order tensor representations of changed features. This technique leverages the representation of changed features and provides discriminative information for CD while retaining the total number of entries.

(3). Two novel TT models that do not require manual annotations are proposed for CD. In the first model, UTT bypasses SVD, which is a usual but computationally expensive algorithm for optimization, in order to extract changed and unchanged features. In the second one, STT leverages pseudo clustering labels to train an accurate change classifier built on TT. Experimental results show that STT is more accurate than UTT, while UTT is more efficient. Moreover, they both outperform state-of-the-art models upon comparison.

The remainder of this work is structured in the following manner. Closely related works are briefly reviewed in Section 2. The background knowledge is introduced in Section 3. In Section 4, proof of the superiority of TT and details of the algorithm for the suggested approaches as well as their implementation are presented. Experiments on four real-world datasets are conducted in Section 5, and the conclusion is discussed in Section 6.

2. Related Works

This section gives a brief review of some closely related works, including change detection in multitemporal hyperspectral images, self-supervision for image analysis, and tensor analysis.

2.1. Change Detection in Multitemporal Hyperspectral Images

Change detection methods can be mainly divided into four categories according to the ways of extracting the changed information.

Firstly, algebra-based methods [18,27–29] leverage algebra operations to extract the changed information. These methods are based on an assumption that changes can be reflected in the difference between corresponding pixel values. Absolute distance (AD) and Euclidian distance (ED) [27] consider each pixel as a vector and calculate the absolute distance and the Euclidian distance between each pixel as the changed information. For these methods, the detection accuracy highly relies on the accuracy of radiometric and geometric correction results.

Secondly, transformation-based methods [30–32] convert the original remote sensing data into another feature space. Then, the CD result is obtained in the new space. In local subspace-based change detection (LSCD) and adaptive subspace-based change detection (ASCD) [30], background subspace is constructed, and the change detection result can be achieved by calculating the subspace distance. The new feature space is more discriminative and is able to differentiate between changed and unchanged. However, some spectrum information is unavoidably lost during the process of transformation in the methods of the second category.

Thirdly, classification-based methods [11,12] categorize the HSIs obtained at different stages separately, and the change detection results can be achieved by comparing the classification results. In [11], the support vector machine (SVM) was used to perform classification independently, and the final CD map could be achieved via post-classification fusion. Since two classification results were obtained separately, environmental factors during the process of HSI acquisition, such as the atmosphere, could be eliminated. However, CD results rely highly on the classification results in this category.

Fourthly, deep learning-based methods [13–16] have become increasingly prosperous with the development of deep learning. Dual attentive fully convolutional Siamese networks (DASNet) [13] and the change detection generative adversarial network (CDGAN) [14], which use the dual attention mechanism and GAN, respectively, have both achieved superior change detection results. Deep models have strong abilities in feature representation and are therefore suitable for solving complex tasks such as HSI CD.

Other methods [10,33–37] such as spectral unmixing [10,34–37] have also achieved great change detection results in HSI CD. In [34], subpixel information was fully leveraged for HSI CD via multi-level spectral unmixing. However, all the mentioned methods above only leverage the spectral information instead of spectral and spatial information together due to the flattening operation they use. This means that the inherent spatial structure information is inevitably lost.

2.2. Self-Supervision for Image Analysis

Self-supervised learning methods, as a subset of unsupervised learning methods, aim to avoid the extensive costs of annotating large-scale datasets. It includes a variety of pretext tasks such as image classification [38], colorizing grayscale images [39], image inpainting [40], and image jigsaw puzzle [41]. Features can be learnt in pretext tasks, where pseudo labels are generated automatically; thus, the learnt features perform better on the main task where no annotations are available. Additionally, the features learnt can also be evaluated in a variety of downstream tasks, including semantic segmentation [42], object detection [43], and so on.

In view of the difficulty of acquiring a lot of annotated samples and the requirements for more discriminative features, the self-supervised have won researchers' attention in the field of CD. For example, in [31], deep neural networks and unsupervised K-Means were combined to learn Gaussian-distributed difference representations, and then the learnt representations were used to detect multiple types of changes. In [44], a multiscale self-attention deep clustering technique was proposed, which combined the convolutional neural network (CNN) with K-Means for remote sensing images CD. In [38], a self-supervised tensor network SSTN composed of pre-train and fine-tune stages was proposed for HSI CD. In [35], an image differencing algorithm and spectral unmixing manner were combined to generate pseudo training data; consequently, a more accurate HSI change detector could be obtained. These self-supervised learning approaches demonstrate how the data structure is used to give supervisory signals in order to learn changed features from temporal images. The self-supervised methods use temporal prediction to represent change features that are more consistent and discriminative than when temporal differences are directly computed. This results in better changed features, with altered areas considerably enhanced and unchanged areas suppressed [45]. This also makes the analysis for the final change map less complex.

2.3. Tensor Analysis

As is already known, HSIs are three-order tensors in nature, with two orders for spatial and the rest for spectral. Tensor analysis and multilinear algebra [22], which can capture the structural characteristics of high dimensional tensors, are therefore suitable for hyperspectral image processing. Tensor decomposition [21], e.g., Tucker decomposition, tensor train, tensor ring, is one of the most significant methods in tensor analysis and has already found applications in dimension reduction [46], target detection [47], anomaly detection [48], hyperspectral images classification [49], etc.

There have also been attempts to approach CD in multitemporal HSIs based on tensor decomposition [18–20]. In [20], the tensor-based approach using 4-D HOSVD was proposed, in which bitemporal HSIs were stacked to extract changed features using Tucker decomposition (TD). In [18], a three-order Tucker decomposition and reconstruction detector (TDRD) was proposed for HSI CD, where unpurified bitemporal HSIs were reconstructed using TD before the changed features were extracted. Nevertheless, TD based on SVD provides stable (i.e., non-iterative) decomposition and optimal approximations regarding their matrix versions due to the orthonormal and diagonal characteristics of the factor matrices. However, the non-diagonality of the tensor core confining the tucker decomposition to ensure optimality at the tensor level results in a lack of changed feature extraction in higher-order tensors [50]. Similarly, components of the Tucker rank are ranks of matrices generated using an unbalanced matricization approach, which is a conceptual problem (one mode versus the rest), ignores the global correlation between changed/unchanged features in bitemporal images. Furthermore, matrix rank approximation is only effective when the matrix is more balanced [25]. TT is another type of tensor decomposition and can capture the global correlation due to its well-balanced matricization scheme. Therefore, TT is promising for the acquisition of more discriminative low-rank features and thus achieves greater change detection results in HSI CD.

3. Background Knowledge

This section gives some background knowledge necessary to comprehend the following contents, including tensor operations, tensor train (TT) decomposition, and quantum information theory.

3.1. Tensor Operations

Basic notations and operations in tensor algebra that are closely related to our work are introduced here. To better characterize the following framework, graphical representation is accepted, as shown in Figure 1. Basic blocks such as scalar (0-order tensor), vector (1-order tensor), matrix (2-order tensor), and high-order tensors are shown in Figure 1a.

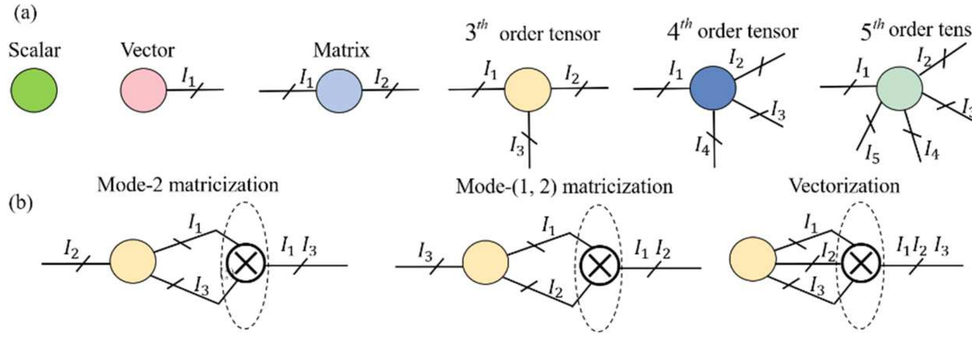

Figure 1. Graphical representation of tensor manipulations. (**a**) Basic building blocks for tensor network diagrams; (**b**) mode-2 matricization (left), mode-(1,2) matricization (middle), and vectorization (right) of a 3-order tensor $\mathbf{G} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$.

Let $\mathbf{G} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an N-order tensor, where N represents the number of dimensions or modes and $I_n (n = 1, 2, \cdots, N)$ represents the dimension of the n-th mode. An entry or element of $\mathbf{G}$ can be denoted by $(\mathbf{G})_{i_1, i_2, \cdots, i_N}$, where $i_n = 1, 2, \cdots, I_n$ and $n = 1, 2, \cdots, N$.

Matricization is also termed as the unfolding or flattening of a tensor. It has two different forms, namely mode-*k* and mode-(1, 2, . . . , *k*) matricization for TD and TT, respectively. Mode-k and mode-(1, 2, . . . , *k*) matricization transforms a tensor $\mathrm{G} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ into $\mathrm{X}_{(k)} \in \mathbb{R}^{m_1 \times n_1}$ and $\mathrm{X}_{[k]} \in \mathbb{R}^{m_2 \times n_2}$, where $m_1 = I_k$, $n_1 = \prod_{j=1, j \neq k}^{N} I_j$, $m_2 = \prod_{j=1}^{k} I_j$, and $n_2 = \prod_{j=k+1}^{N} I_j$. The correspondence of elements between the original tensor and the matrix after matricization can be expressed as follows:

$$\left(\mathbf{X}_{(k)}\right)_{i_k, \overline{i_1 \cdots i_{k-1} i_{k+1} \cdots i_N}} = \left(\mathbf{G}\right)_{i_1, i_2, \cdots, i_N}, \tag{1}$$

$$\left(\mathbf{X}_{[k]}\right)_{\overline{i_1 \cdots i_k}, \overline{i_{k+1} \cdots i_N}} = \left(\mathbf{G}\right)_{i_1, i_2, \cdots, i_N}. \tag{2}$$

A multi-index is an index that combines all values of indices following a specific order and can be calculated as follows:

$$\overline{i_1 \cdots i_N} = i_N + (i_{N-1} - 1) I_N + (i_{N-2} - 1) I_N I_{N-1} + \cdots + (i_1 - 1) \prod_{i=N}^{2} I_i. \tag{3}$$

The vectorization of the tensor **G** is represented by vec(**G**), which is operated on the mode-1 unfolded matrix $\mathbf{X}_{(1)}$ as:

$$\mathbf{b} = \mathrm{vec}\left(\mathbf{G}\right) = \mathrm{vec}\left(\mathbf{X}_{(1)}\right) \in \mathbb{R}^{\Pi_{\mathrm{k}=1}^{\mathrm{N}} I_k}. \tag{4}$$

The graphical representations of mode-k matricization, mode-$(1, 2, \ldots, k)$ matricization, and vectorization can be clearly seen in Figure 1b.

3.2. Tensor Train (TT) Decomposition

A tensor train decomposition represents an N-order tensor $\mathbf{G} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with N 3-order tensors [24], and each element of **G** can be expressed as follows:

$$\mathbf{G}_{i_1,\cdots,i_N} = \left(\mathbf{X}_1\right)_{:,i_1,:} \left(\mathbf{X}_2\right)_{:,i_2,:} \cdots \left(\mathbf{X}_N\right)_{:,i_N,:}, \tag{5}$$

where $\mathbf{X}_k \in \mathbb{R}^{r_{[k-1]} \times I_k \times r_{[k]}}$ is the k-th core tensor with $r_{[0]} = r_{[N]} = 1$. $r_{[0]}, r_{[1]}, \cdots, r_{[N]}$ are called TT ranks.

In index form, Equation (5) can be written as follows:

$$\mathbf{G}_{i_1,\cdots,i_N} = \sum_{\alpha_0,\cdots,\alpha_N} (\mathbf{X}_1)_{\alpha_0,i_1,\alpha_1} (\mathbf{X}_2)_{\alpha_1,i_2,\alpha_2} \cdots (\mathbf{X}_N)_{\alpha_{N-1},i_N,\alpha_N}. \tag{6}$$

Figure 2 shows a typical representation of a 5-order TT decomposition in graphical notations. There exist some applications, such as the low-rank tensor completion (LRTC) [25], which make full use of the ability of low-rank features extraction of TT. Their process of extracting low-rank features can be summarized as a low TT-rank optimization problem. One solution to solve this problem is by using singular value decomposition (SVD) [21], which can be called TT-SVD. Another way that uses the multilinear matrix factorization model [25] to approximate the TT rank of a tensor can bypass the computationally expensive SVD; this is called a TT-noSVD.

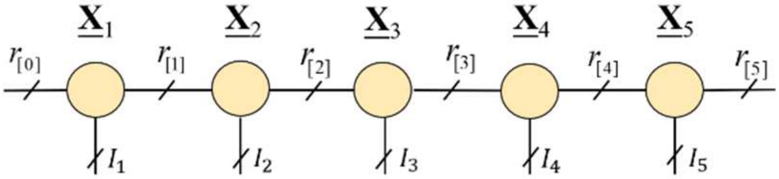

Figure 2. Visualization of a 5-order tensor train decomposition in graphical notations.

3.3. Quantum Information Theory

Quantum states can be represented using density matrices [23], which are naturally 2-order tensors. Every tensor corresponds to a quantum state if the matricization operation is used. Therefore, it is promising to validate the effectiveness of tensor analysis using the knowledge in the field of quantum information theory as it can help to analyze the ability of different tensor methods in capturing useful information.

The information in the quantum field is measured by the von Neumann entropy [23], which is an extension of the Shannon entropy in the classical field. The von Neumann entropy is defined as follows:

$$\mathrm{S}(\boldsymbol{\rho}) = -\mathrm{tr}(\boldsymbol{\rho}\log_2(\boldsymbol{\rho})), \tag{7}$$

where $\boldsymbol{\rho}$ and $\mathrm{tr}(\cdot)$ represent the quantum state and the trace operation, respectively.

The correlation between two subsystems, $\mathbf{H}_A \in \mathbb{R}^m$ and $\mathbf{H}_B \in \mathbb{R}^n$, of the composite system $\mathbf{H}_{AB} \in \mathbb{R}^{m \times n}$ can be quantified by quantum mutual information [23]:

$$\mathrm{I}(A:B) = \mathrm{S}(\boldsymbol{\rho}^A) + \mathrm{S}(\boldsymbol{\rho}^B) - \mathrm{S}(\boldsymbol{\rho}^{AB}), \tag{8}$$

where $\boldsymbol{\rho}^A$, $\boldsymbol{\rho}^B$, and $\boldsymbol{\rho}^{AB}$ represent the quantum states of the corresponding systems $\mathbf{H}_A$, $\mathbf{H}_B$, and $\mathbf{H}_{AB}$. The quantum mutual information can also be considered as a measure of global correlation [51].

4. Methodology

In this section, the proof for the advantages of TT over TD is firstly shown by virtue of quantum information theory. Then, two TT-based methods, UTT and STT are proposed for CD. The first one performs unsupervised CD using multilinear matrix factorization to approximate the TT rank, as shown in Figure 3, and the second one performs self-supervised CD using TT decomposition, as shown in Figure 4. The proposed methods can yield accurate CD results without manual annotations.

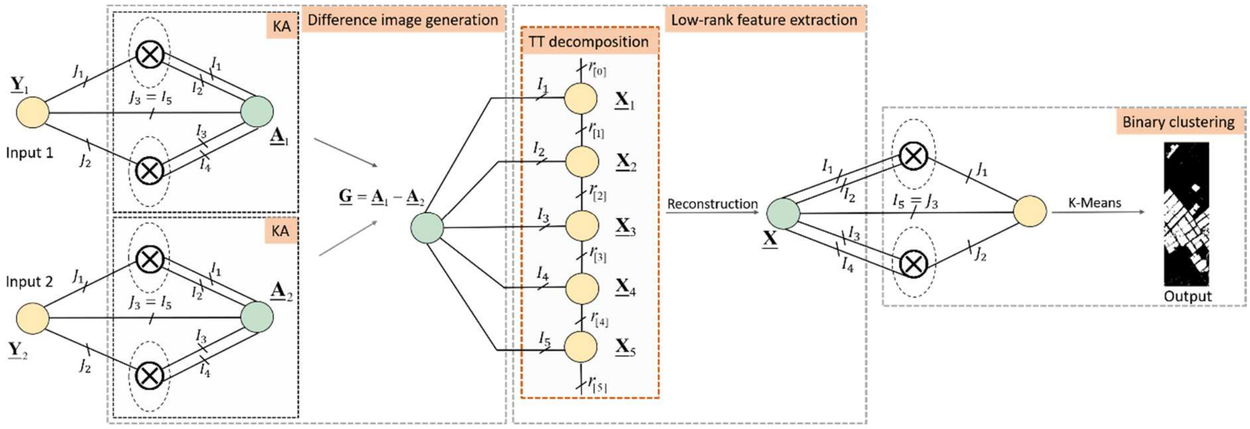

Figure 3. Framework of the proposed UTT for multitemporal HSI change detection.

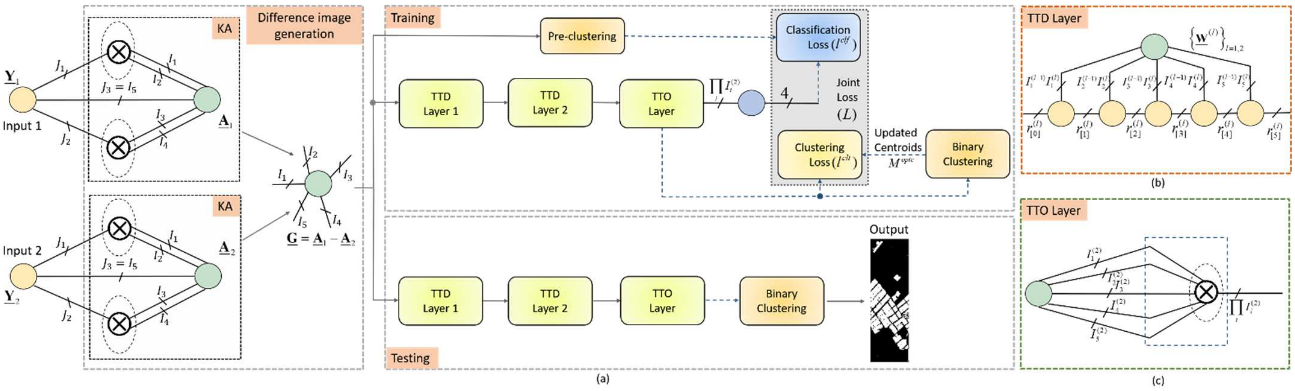

Figure 4. Framework of the proposed STT for multitemporal HSI change detection. (**a**) Overall framework of STT, (**b**) tensor train decomposition (TTD) layer, (**c**) tensor train output (TTO) layer.

4.1. *Ability of Tensor Train in Capturing the Global Correlation*

A core challenge in the field of HSI CD is that the features extracted cannot be discriminative enough. It is because the features may not contain enough changed information. Therefore, it is crucial to ensure the ability of the CD methods in capturing changed information (or correlation in quantum information theory). As density matrices representing the quantum states are 2-order tensors in nature, matricization operation and quantum information theory are combined here to compare the ability of TT and TD in capturing the changed information of the tensors; the complete proof is shown below.

For any given matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, if SVD is applied, then $\mathbf{X} = \mathbf{W}\sum \mathbf{V}^T$ can be produced, where $\mathbf{W} \in \mathbb{R}^{m\times m}$, $\sum \in \mathbb{R}^{m\times n}$, and $\mathbf{V} \in \mathbb{R}^{n\times n}$. Matrices $\mathbf{W}$ and $\mathbf{V}$ consist of eigenvectors of $\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{m\times m}$ and $\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{n\times n}$, respectively. Therefore, $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ contain significant information from the dimensions m and n of $\mathbf{X}$, respectively. In the field of quantum theory, $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ can represent density matrices [23] in two subspaces $\mathbf{H}_A \in \mathbb{R}^m$ and $\mathbf{H}_B \in \mathbb{R}^n$, respectively. The density matrix is one of the forms representing a quantum state. To correspond to the notation in quantum physics, $\boldsymbol{\rho}^A$ and $\boldsymbol{\rho}^B$ are used to replace $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ in the following statements.

According to the Schmidt decomposition [23] and the fact that $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ have the same nonvanishing eigenvalues, there exist orthonormal bases $\{|i_A\rangle\}$ in $\mathbf{H}_A$ and $\{|i_B\rangle\}$ in $\mathbf{H}_B$, such that:

$$\boldsymbol{\rho}^A = \sum_{i=1}^{r_k} \lambda_i^2 |i_A\rangle\langle i_A|, \tag{9}$$

$$\boldsymbol{\rho}^B = \sum_{i=1}^{r_k} \lambda_i^2 |i_B\rangle\langle i_B|, \tag{10}$$

where r_k is the rank of $\mathbf{X}$, $\{\lambda_i\}_{i=1,\cdots,r_k}$ are nonvanishing singular values satisfying $\sum_{i=1}^{r_k} \lambda_i^2 = 1$. Then, the density matrix of a composite system AB in the space $\mathbf{H}_{AB} \in \mathbb{R}^{m\times n}$ can be represented as follows:

$$\boldsymbol{\rho}^{AB} = \sum_{i=1}^{r_k} \lambda_i^2 |i_A\rangle\langle i_A| \otimes |i_B\rangle\langle i_B|, \tag{11}$$

where ⊗ represents the tensor product [23] and $\mathbf{H}_{AB}$ is the tensor product of two subspaces $\mathbf{H}_A \in \mathbb{R}^m$ and $\mathbf{H}_B \in \mathbb{R}^n$.

The global correlation between two subsystems A and B can be studied via quantum mutual information, which is defined in Equation (8). Substituting Equation (9) into Equation (7), the von Neumann entropy of $\boldsymbol{\rho}^A$ can be calculated as follows:

$$\mathrm{S}(\boldsymbol{\rho}^A) = -\mathrm{tr}(\boldsymbol{\rho}^A \log_2(\boldsymbol{\rho}^A)) = -\sum_{i=1}^{r_k} \lambda_i^2 \log_2 \lambda_i^2. \tag{12}$$

Similarly, it can be obtained that:

$$\mathrm{S}(\boldsymbol{\rho}^B) = \mathrm{S}(\boldsymbol{\rho}^{AB}) = -\sum_{i=1}^{r_k} \lambda_i^2 \log_2 \lambda_i^2, \tag{13}$$

Simply, $\mathrm{S}(\boldsymbol{\rho}^A) = \mathrm{S}(\boldsymbol{\rho}^B) = \mathrm{S}(\boldsymbol{\rho}^{AB}) = S$, where:

$$S = -\sum_{i=1}^{r_k} \lambda_i^2 \log_2 \lambda_i^2. \tag{14}$$

With S substituted into Equation (8), it can be achieved that the global correlation between two subsystems $\mathbf{H}_A \in \mathbb{R}^m$ and $\mathbf{H}_B \in \mathbb{R}^n$ is equal to S. According to the quantum relative entropy and Klein inequality [23], it is shown in Equation (15) that S is bounded by $0 \le S \le \log_2 r_k$, where $\mathbf{I}$ represents the identity matrix:

$$\begin{aligned} \mathrm{S}(\boldsymbol{\rho}||\mathbf{I}/r_k) &= \mathrm{tr}(\boldsymbol{\rho}\log_2(\boldsymbol{\rho})) - \mathrm{tr}(\boldsymbol{\rho}\log_2(\mathbf{I}/r_k)) \\ &= -S - \textstyle\sum_{i=1}^{r_k} \lambda_i^2 \log_2(1/r_k) \\ &= -S + \log_2 r_k \ge 0 \end{aligned}. \tag{15}$$

Here, it is assumed that $I_1 = I_2 = \cdots = I_N = I$ for simplicity. For the mode-k matricization of $\mathbf{G} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ from TD, $\mathbf{X}_{(k)} \in \mathbb{R}^{m_1\times n_1}$ can be obtained, where

$m_1 = I_k$ and $n_1 = \prod_{j=1,j\neq k}^{N} I_j$. Therefore, $r_{(k)}$, which represents the rank of $\mathbf{X}_{(k)}$, satisfies the following:

$$r_{(k)} \leq \min(I_k, \prod_{j=1,j\neq k}^{N} I_j) = I. \tag{16}$$

Then, for the mode-$(1, 2, \ldots, k)$ matricization from TT, $\mathbf{X}_{[k]} \in \mathbb{R}^{m_2 \times n_2}$ can be obtained, where $m_2 = \prod_{j=1}^{k} I_j$ and $n_2 = \prod_{j=k+1}^{N} I_j$. Therefore, $r_{[k]}$, which represents the rank of $\mathbf{X}_{[k]}$, satisfies the following equation:

$$r_{[k]} \leq \min(\prod_{j=1}^{k} I_j, \prod_{j=k+1}^{N} I_j) = I^{\min(k,N-k)}. \tag{17}$$

From the analysis above, it can be seen that the upper bound of $r_{[k]}$ in TT is far greater than its counterpart $r_{(k)}$ in TD for $k = 2, 3, \cdots, N-2$ and is equivalent to it only when k is equal to 1 or N-1. Therefore, the conclusion that the upper bound of the global correlation that TT can capture is far greater than its counterpart TD in the vast majority of cases can be drawn from Equation (15). Because TT can capture more changed information which TD cannot, the superiority of TT in HSI CD is confirmed.

4.2. Unsupervised Tensor Train for Change Detection

Due to the effective changed information extraction ability of TT, it is used in the proposed algorithms to extract discriminative features. The first proposed algorithm leverages unsupervised learning to avoid the complex and time-consuming manual annotation process. The proposed UTT is composed of three parts: (1) difference image generation, (2) TT decomposition for spectral–spatial low-rank feature extraction, and (3) binary clustering to distinguish between changed and unchanged regions via the features extracted by the TT decomposition.

Difference image generation consists of two steps, KA and substraction. Let $\underset{1}{\mathbf{Y}} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ and $\underset{2}{\mathbf{Y}} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ be a set of bitemporal HSIs acquired at two different times, t_1 and t_2, respectively, where J_1 is the number of rows, J_2 is the number of columns, and J_3 is the number of bands. They can be represented by a set of orthonormal bases $\left\{ \mathrm{y}_{j_k}^{(k)} \in \mathbb{R}^{J_k} \right\}_{k=1,2,3}$, as follows:

$$\underset{t}{\mathbf{Y}} = \sum_{j_1,j_2,j_3} (\underset{t}{\mathbf{Y}})_{j_1 j_2 j_3} \mathbf{y}_{j_1}^{(1)} \circ \mathbf{y}_{j_2}^{(2)} \circ \mathbf{y}_{j_3}^{(3)}, \tag{18}$$

where $t = 1, 2$ and $\circ$ represents outer product. To obtain high-order rich texture features [26] and make full use of the local structure of the data in terms of computational resources [25], KA is applied to three-dimensional HSIs $\underset{1}{\mathbf{Y}}$ and $\underset{2}{\mathbf{Y}}$. This process does not increase the computational complexity as the total number of elements does not change. Then, two five-dimensional tensors, $\underset{1}{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4 \times I_5}$ and $\underset{2}{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4 \times I_5}$, are obtained:

$$\underset{t}{\mathbf{A}} = \sum_{i_1,i_2,i_3,i_4,i_5} (\underset{t}{\mathbf{A}})_{i_1 i_2 i_3 i_4 i_5} \mathbf{e}_{i_1}^{(1)} \circ \mathbf{e}_{i_2}^{(2)} \circ \mathbf{e}_{i_3}^{(3)} \circ \mathbf{e}_{i_4}^{(4)} \circ \mathbf{e}_{i_5}^{(5)}, \tag{19}$$

where $t = 1, 2$, $I_1 I_2 I_3 I_4 I_5 = J_1 J_2 J_3$, and $\left\{ \mathbf{e}_{i_k}^{(k)} \in \mathbb{R}^{I_k} \right\}_{k=1,2,3,4,5}$ represents a set of orthonormal bases in the corresponding Hilbert space. Then, substraction is conducted between $\underset{1}{\mathbf{A}}$ and $\underset{2}{\mathbf{A}}$ to get the high-order difference tensor:

$$\mathbf{G} = \underset{1}{\mathbf{A}} - \underset{2}{\mathbf{A}} \quad s.t \;\; \mathbf{G}, \underset{1}{\mathbf{A}}, \underset{2}{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4 \times I_5}, \tag{20}$$

where I_k is the corresponding mode-k dimension for $k = 1, \cdots, 5$.

Taking into account the intrinsic complexity and structure of HSI, TT is used for change detection in HSI as an unsupervised technique that extracts changed and unchanged features efficiently. As proved in Section 4.1, more global correlations can be captured with TT as compared to its tensor counterpart TD. This means that TT can extract changed information more effectively. The process of the spectral–spatial low-rank feature extraction of the generated high-order difference tensor **G** can be conducted via the following process of TT rank optimization:

$$\min_{\mathbf{X}_{[k]}} \sum_{k=1}^{N-1} \alpha_k \text{rank}(\mathbf{X}_{[k]}) \quad s.t\ \mathbf{X}_{\Omega} = \mathbf{G}_{\Omega}, \tag{21}$$

where α_k represents the weight that the TT rank of the matrix $\mathbf{X}_{[k]}$ contributes to, with the constraint $\sum_{k=1}^{N-1} \alpha_k = 1$, and Ω represents a given index set. After the process of low TT rank feature extraction, the reconstructed tensor **X** obtained by Equation (21) is sent to K-Means to obtain the final CD results $\mathbf{T} \in \mathbb{R}^{J_1 \times J_2}$. The pseudocode and framework of the proposed UTT are summarized in Algorithm 1 and Figure 3, respectively. Different groups of variables are alternatively optimized using the block coordinate descent (BCD) method [25], and the update of **X** in each loop is given by Equation (22).

Algorithm 1. Pseudocode of the proposed UTT for multitemporal HSI change detection.

```
Input: Observed data $\mathbf{G} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4 \times I_5}$, index set $\Omega$
Parameters: $\alpha_i$, $r_i$, $i = 1, \cdots, N-1$
Initialization: Initialize $\mathbf{U}^0, \mathbf{V}^0, \mathbf{X}^0$, with $\mathbf{X}^0_{\Omega} = \mathbf{G}_{\Omega}, l = 0$
While not converged do:
    for $k$ = 1 to $N$-1 do
        Unfold tensor $\mathbf{G}^l$ to get $\mathbf{X}^l_{[k]}$
        $\mathbf{U}^{l+1}_k = \mathbf{X}^l_{[k]} (\mathbf{V}^l_k)^T (\mathbf{V}^l_k (\mathbf{V}^l_k)^T)^{\dagger}$
        $\mathbf{V}^{l+1}_k = ((\mathbf{U}^{l+1}_k)^T \mathbf{U}^{l+1}_k)^{\dagger} (\mathbf{U}^{l+1}_k)^T \mathbf{X}^l_{[k]}$
        $\mathbf{X}^{l+1}_{[k]} = \mathbf{U}^{l+1}_k \mathbf{V}^{l+1}_k$
    end for
    Update tensor $\mathbf{X}^{l+1}$ using Equation (22)
End while
Apply K-Means to the reconstructed tensor $\mathbf{X}$
Output: Change detection results $\mathbf{T}$
```

$$\mathbf{X}^{l+1}_{i_1 i_2 i_3 i_4 i_5} = \begin{cases} (\sum\limits_{k=1}^{N-1} \alpha_k \text{fold}(\mathbf{X}^{l+1}_{[k]}))_{i_1 i_2 i_3 i_4 i_5}, & (i_1, i_2, i_3, i_4, i_5) \notin \Omega \\ \mathbf{G}_{i_1 i_2 i_3 i_4 i_5}, & (i_1, i_2, i_3, i_4, i_5) \in \Omega \end{cases}, \tag{22}$$

where folding is the inverse process of unfolding, i.e., matricization in Equation (2).

The specific process of optimizing Equation (21) in UTT is described as follows. Factorization model $\mathbf{X}_{[k]} = \mathbf{UV}$ is leveraged to minimize the Frobenius norm, where $\mathbf{X}_{[k]} \in \mathbb{R}^{m \times n}$ is a matrix of rank $r_{[k]}$, $\mathbf{U} \in \mathbb{R}^{m \times r_{[k]}}$ and $\mathbf{V} \in \mathbb{R}^{r_{[k]} \times n}$.

Equation (21) can be re-modeled as follows:

$$\min_{\mathbf{U}_k, \mathbf{V}_k, \mathbf{X}_{[k]}} \sum_{k=1}^{N-1} \frac{\alpha_k}{2} \left\| \mathbf{U}_k \mathbf{V}_k - \mathbf{X}_{[k]} \right\|_F^2 \qquad s.t.\ \mathbf{X}_{\Omega} = \mathbf{G}_{\Omega}, \tag{23}$$

Where $\mathbf{U}_k \in \mathbb{R}^{\prod_{j=1}^{k} I_j \times r_{[k]}}$ and $\mathbf{V}_k \in \mathbb{R}^{r_{[k]} \times \prod_{j=k+1}^{N} I_j}$. This problem is now convex when each variable $\mathbf{U}_k$, $\mathbf{V}_k$, and $\mathbf{X}_{[k]}$ is modified while keeping the other two fixed. Therefore, the variables are updated alternatively following Equations (24)–(26):

$$\mathbf{U}_k^{l+1} = \mathbf{X}_{[k]}^{l} (\mathbf{V}_k^{l})^{T} (\mathbf{V}_k^{l} (\mathbf{V}_k^{l})^{T})^{\dagger}, \tag{24}$$

$$\mathbf{V}_k^{l+1} = ((\mathbf{U}_k^{l+1})^{T} \mathbf{U}_k^{l+1})^{\dagger} (\mathbf{U}_k^{l+1})^{T} \mathbf{X}_{[k]}^{l}, \tag{25}$$

$$\mathbf{X}_{[k]}^{l+1} = \mathbf{U}_k^{l+1} \mathbf{V}_k^{l+1}, \tag{26}$$

where † denotes the Moore–Penrose pseudoinverse. After updating $\mathbf{U}_k$, $\mathbf{V}_k$, and $\mathbf{X}_{[k]}$, elements of the tensor $\mathbf{X}^{l+1}$ can be computed with Equation (22).

The pseudocode is summarized in Algorithm 1. The main benefit of this approach is that it bypasses the SVD, which saves a lot of time in the computation. However, since Equation (21) can also be optimized using SVD [21,25], UTT-SVD and UTT-noSVD are used here to differentiate whether SVD is used in the optimization process.

4.3. *Self-Supervised Tensor Train for Change Detection*

STT uses self-supervised learning to avoid the complex and time-consuming manual annotation process. Pseudo classification is used as the pretext task to learn features that are friendly to binary clustering. K-Means is used both to generate the initial pseudo labels and divide the learnt features into the categories of changed or unchanged, while the classification network is typically composed of two kinds of layers, i.e., the TT decomposition (TTD) layers and the TT output (TTO) layers.

The input of STT is denoted by an N-order difference tensor $\mathbf{G} = \mathbf{Y}^{(0)} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $I_i = I_i^{(0)} (i = 1, 2, \cdots, N)$. It can be obtained by difference image generation defined in UTT. For TTD layers, features extracted from the input samples can be defined as follows:

$$\mathbf{Y}^{(l)}_{j_1,\ldots,j_N} = \sigma\left(\sum_{i_1,\cdots,i_N}^{I_1^{(l-1)},\cdots,I_N^{(l-1)}} \mathbf{W}^{(l)}_{j_1,\cdots,j_N,i_1,\cdots,i_N} \mathbf{Y}^{(l-1)}_{i_1,\cdots,i_N} + \mathbf{B}^{(l)}_{j_1,\cdots,j_N} \right), \tag{27}$$

where $\mathbf{Y}^{(l)} \in \mathbb{R}^{I_1^{(l)} \times I_2^{(l)} \times \cdots \times I_N^{(l)}} (1 = 1, 2, \ldots, L)$ is the output of the l-th TTD layer, and it can be treated as a primary feature tensor, L is the number of TTD layers, σ is the ReLU activation function, and $\theta^{(l)} = \left\{ \mathbf{W}^{(l)}, \mathbf{B}^{(l)} \right\}$ is the parameter set of the l-th TTD layer, where $\mathbf{W}^{(l)} \in \mathbb{R}^{I_1^{(l)} \times \cdots \times I_N^{(l)} \times I_1^{(l-1)} \times \cdots \times I_N^{(l-1)}}$ and $\mathbf{B}^{(l)} \in \mathbb{R}^{I_1^{(l)} \times \cdots \times I_N^{(l)}}$.

The coefficient tensor $\mathbf{W}^{(l)}$ of Equation (27) in the l-th TTD layer is decomposed into N three-order tensors $\mathbf{X}_k^{(l)} \in \mathbb{R}^{r_{[k-1]}^{(l)} \times I_k^{(l-1)} I_k^{(l)} \times r_{[k]}^{(l)}}$ [52] for $k = 1, 2, \cdots, N$, while $\left\{ \mathrm{r}_{[k]}^{(l)} \right\}_{k=0,1,\cdots,N}$ represents the set of TT ranks in the l-th TTD layer with $r_{[0]}^{(l)} = r_{[N]}^{(l)} = 1$, as shown in Equation (5) and Figure 4b. This is so that low-rank features can be efficiently extracted, and the number of parameters can be significantly compressed. Each element in $\mathbf{W}^{(l)}$ can be represented equivalently by the following matrix product:

$$\mathbf{W}^{(l)}{}_{j_1,\ldots,j_N,i_1,\cdots,i_N} = \left(\mathbf{X}_1^{(l)} \right)_{:,\overline{j_1 i_1},:} \left(\mathbf{X}_2^{(l)} \right)_{:,\overline{j_2 i_2},:} \cdots \left(\mathbf{X}_N^{(l)} \right)_{:,\overline{j_N i_N},:}, \tag{28}$$

where $\left(\mathbf{X}_k^{(l)} \right)_{:,\overline{j_k i_k},:} \in \mathbb{R}^{r_{[k-1]}^{(l)} \times r_{[k]}^{(l)}}$ is the lateral slice of the tensor $\mathbf{X}_k^{(l)}$ for $k = 1, 2, \cdots, N$.

Applying the TT format for the weight subtensor in Equation (27), the l-th TTD layer can be redefined as follows:

$$\mathbf{Y}^{(l)}_{j_1,\cdots,j_N} = \sigma\left(\sum_{i_1,\cdots,i_N}^{I_1^{(l-1)},\cdots,I_N^{(l-1)}} \left(\mathbf{X}^{(l)}_{1}\right)_{:,\overline{j_1 i_1},:} \left(\mathbf{X}^{(l)}_{2}\right)_{:,\overline{j_2 i_2},:} \cdots \left(\mathbf{X}^{(l)}_{N}\right)_{:,\overline{j_N i_N},:} \mathbf{Y}^{(l-1)}_{i_1,\cdots,i_N} + \mathbf{B}^{(l)}_{j_1,\cdots,j_N} \right). \tag{29}$$

In our STT model, N = 5 is used so that the coefficient tensors are decomposed into five three-order tensors.

For the TTO layers, the output of the last TTD layer $\mathbf{Y}^{(L)}$ is firstly vectorized according to Equation (4), and the output features are calculated as follows:

$$\mathbf{z} = f_{\vartheta}(\mathbf{G}) = vec(\mathbf{Y}^{(L)}) \in \mathbb{R}^{d}, \tag{30}$$

where $d = I_1^{(L)} I_2^{(L)} \cdots I_N^{(L)}$ is the dimension of the features. This can be seen clearly in Figure 4c. Afterward, a fully connected layer is employed to $f_{\vartheta}(\mathbf{G})$ to compute the posterior output $\mathbf{y} \in \mathbb{R}^{C_1}$ for the classification problem over C_1 pseudo class labels corresponding to the input tensor $\mathbf{G}$ as follows:

$$\mathbf{y} = s(f_{\vartheta}(\mathbf{G}) \times_1 \boldsymbol{\omega}) = s_{\boldsymbol{\omega}}(f_{\vartheta}(\mathbf{G})), \tag{31}$$

where s$(\cdot)$ represents the softmax activation function, and $\boldsymbol{\omega} \in \mathbb{R}^{d \times C_1}$ is the weight of the fully connected layer.

For the training set $\left\{ \underset{i}{\mathbf{G}} \right\}_{i=1,2,\ldots,P}$, which represents the high-order difference tensor generated from each patch, P is the number of the training samples. STT achieves self-learning through the joint training of TTD/TTO layers for pseudo-label classification and clustering layers for binary clustering. Pseudo-labels are generated by implementing an unsupervised method, i.e., K-Means, to the set of high-order difference tensors $\left\{ \underset{i}{\mathbf{G}} \right\}_{i=1,2,\ldots,P}$. The cross-entropy classification loss function is defined with regard to the parameter set $\zeta = \left\{ \theta^{(l)}, \boldsymbol{\omega} \right\}$ as:

$$l^{clf}(\zeta; \left\{ \underset{i}{\mathbf{G}} \right\}) = -\frac{1}{P} \sum_{i=1}^{P} \hat{\mathbf{y}}_i^{clf} \log s_{\boldsymbol{\omega}}(f_{\vartheta}(\underset{i}{\mathbf{G}})), \tag{32}$$

where $\left\{ \hat{\mathbf{y}}_{\mathrm{i}}^{\mathrm{clf}} \in \mathbb{R}^{C_1} \right\}_{i=1,2,\ldots,P}$ represents the reference pseudo labels.

Then, with respect to the part of clustering, the features computed by Equation (30) in the TTO layer are clustered by the binary clustering module to update the original centroid matrix $\mathbf{M}^{ptc} \in \mathbb{R}^{d \times C_2}$ for the changed cluster and the unchanged one, where $C_2 = 2$, as illustrated in Figure 4. The difference between the output features of STT in Equation (30) and the clustering results is used as the other part of the self-learning loss for the optimization of STT, which is computed as follows:

$$\begin{aligned} l^{clt} &= \tfrac{1}{P} \sum_{i=1}^{P} \left\| f_{\vartheta}(\underset{i}{\mathbf{G}}) - \mathbf{M}^{optc} \hat{\mathbf{y}}_i^{opt} \right\|_F^2 \\ s.t \;\; & \hat{\mathbf{y}}_{j,i}{}^{opt} \in \{0,1\}, \mathbf{1}^T \hat{\mathbf{y}}_i{}^{opt} = 1, \forall i, j, \end{aligned} \tag{33}$$

where $\mathbf{M}^{optc} \in \mathbb{R}^{d \times C_2}$ is an updated centroid matrix with initial values $\mathbf{M}^{optc} = \mathbf{M}^{ptc}$ and $\hat{\mathbf{y}}^{opt} = \hat{\mathbf{y}}^{pt}$. Afterwards, the new centroid can be updated as [17]:

$$\mathbf{M}_k^{optc} = \mathbf{M}_k^{optc} - \left(\frac{\mathbf{M}_k^{optc} - f_{\vartheta}(\underset{i}{\mathbf{G}})}{n_{k,i}} \right) \hat{\mathbf{y}}_{k,i}^{opt} \;\; s.t \; k = 1, 2, \cdots C_2, \tag{34}$$

where $n_{k,i}$ is the number of times the algorithm allocated a sample to cluster k before handling the incoming sample $\mathbf{G}_{i}$, and $1/n_{k,i}$ can be used to regulate the learning rate as a gradient step size.

In the STT model, the pseudo classification and the binary clustering process are performed alternatively. The classification loss and cluster loss are combined as the total network loss:

$$L(\zeta;\left\{\mathbf{G}_{i}\right\}) = l^{clt} + \mu l^{clf}, \tag{35}$$

where μ is a hyper-parameter to constrain the balance between classification and clustering. The STT learns low-rank changed features in a self-supervised manner using the pre-clustering pseudo labels and the binary clustering layers throughout the training process.

For STT training, the Adam optimizer [53] is used to minimize the loss function with regard to the parameter set $\zeta = \left\{\mathbf{W}^{(l)}, \mathbf{B}^{(l)}, \boldsymbol{\omega}\right\}$. The STT parameters are learnt in a better way by computing the gradient of loss function $L(\zeta;\left\{\mathbf{G}_{i}\right\})$ directly, with reference to the cores of the TT-representation of $\left\{\mathbf{W}^{(l)}\right\}_{l=1,2,\ldots,L}$ [52].

The backpropagation (BP) method is used to update the whole parameter set $\zeta = \left\{\theta^{(l)}, \boldsymbol{\omega}\right\}$ during the training process. Firstly, BP starts to calculate the gradients of the loss function with reference to the parameters in the output layer, which is $\boldsymbol{\omega}$ in our case. Then, the BP method is applied to calculate the gradients of the parameter set from the TTO layer to the first TTD layer using the chain rule. After the gradients are calculated, the Adam optimizer is used to update them at the same time.

In the testing phase of the STT model, the optimized and updated parameters $\zeta = \left\{\mathbf{W}^{(l)}, \mathbf{B}^{(l)}, \boldsymbol{\omega}\right\}$ are used on the samples generated from every pixel to obtain the binary change detection map $\mathbf{T} \in \mathbb{R}^{J_1 \times J_2}$. The optimization algorithm for STT is summarized in Algorithm 2. The maximum number of epochs is specified so that iterations can be halted for practical reasons. The framework of the proposed STT is summarized in Figure 4.

TT decomposition overcomes the curse of dimensionality by reducing the number of network parameters significantly as compared to the tucker decomposition. Let $I = \max\left\{I_1^{(l)}, I_2^{(l)}, \cdots, I_N^{(l)}\right\}$ be the maximum dimension and $r = \max\left\{r_{[0]}^{(l)}, r_{[1]}^{(l)}, \cdots, r_{[N]}^{(l)}\right\}$ be the maximum TT rank, then the storage complexity of each weight subtensor is $O(r^N + NIr)$ in the Tucker format, and the storage complexity is $O(NIr^2)$ if each subtensor is stored in the TT format. This reveals that the number of parameters in the TT format scales linearly in N, whereas it scales exponentially in the Tucker format. Another advantage of the TT decomposition is its simple practical implementations. TT decomposition achieves the optimal low TT-ranks due to its balanced matricization scheme using current algorithms such as the SVD-based or the alternative low-rank matrix approximation, whereas tucker decomposition is uncertain to achieve the optimal ranks because of its unbalanced matricization scheme.

5. Experiments

To demonstrate the efficacy and efficiency of the proposed model, experiments on four hyperspectral datasets are conducted. The detailed analysis is as follows.

5.1. Datasets

The proposed algorithms are tested on four real-world bitemporal HSI datasets. Each dataset consists of three images, two sets of HSIs, and a ground-truth map. The first dataset, "Yancheng", is shown in Figure 5a,e (bands 20, 100, and 10 as RGB). It has 450 × 140 pixels and depicts the countryside in Yancheng in Jiangsu Province, China [30]. The two HSIs were acquired on 3 May 2006 and 23 April 2007. After noise removal, 155 bands were chosen for the change detection. The second dataset, "Bay Area", is shown in Figure 5b,f (bands 50, 60, and 70 as RGB), with the AVIRIS sensor surrounding the city of Patterson (California), whose spatial dimensions are 600 × 500 pixels and includes 224 spectral bands [54]. The third dataset, "River", is shown in Figure 5c,g (bands 20, 40, and 60 as RGB). The two River HSI datasets were acquired in Jiangsu Province, China, on 3 May 2013 and 31 December 2013. After removing the noisy bands, this dataset has a size of 463 × 241 pixels and 198 bands [15]. The fourth dataset, "Hermiston city", is shown in Figure 5d,h (bands 20, 40, and 60 as RGB). In the years 2004 and 2007, the two HSIs were taken with the HYPERION sensor over the Hermiston city area (Oregon), whose spatial dimensions are 390 × 200 pixels and includes 242 spectral bands [32].

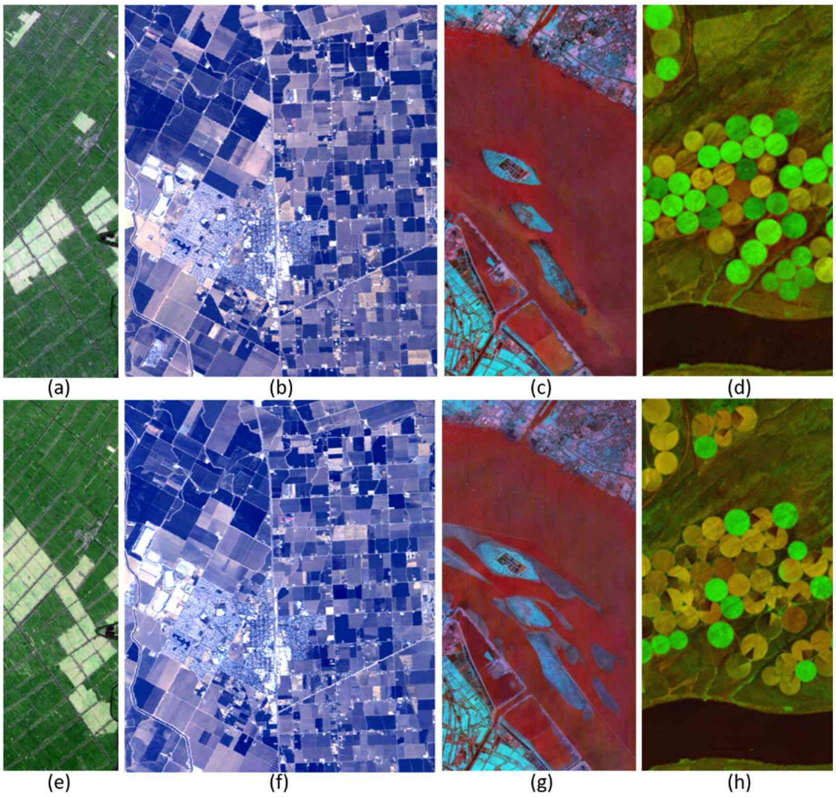

Figure 5. False-color composition of (**a**,**e**) Yancheng, (**b**,**f**) Bay Area, (**c**,**g**) River, and (**d**,**h**) Hermiston city.

Algorithm 2. Pseudocode of the proposed STT for multitemporal HSI change detection.

```
Input: High-order difference tensors $\left\{\mathbf{G}_i \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4 \times I_5}\right\}_{i=1,2,\ldots,P}$, pseudo-labels $\hat{\mathbf{y}}^{clf}$,
maximum epochs
Parameters: $\zeta = \left\{\mathbf{W}^{(l)}, \mathbf{B}^{(l)}, \boldsymbol{\omega}\right\}, \mathbf{M}^{optc}, \hat{\mathbf{y}}^{opt}, C_1, C_2$

Initialization: Initialize $\zeta, \mathbf{M}^{ptc}, \hat{\mathbf{y}}^{pt}, C_1, C_2, \hat{\mathbf{y}}^{clf}$. Maximum epochs
While not converged or maximum epochs not reached, do:
        Compute $l^{clf}$ and $l^{clt}$ with Equations (32) and (33)
        Update network loss L with Equation (35)
        Update parameter set $\zeta$ using Adam optimizer
        Update centroids $M^{optc}$ with Equation (34)
End while
    Obtain features $\mathbf{z}$ with Equation (30)
Perform binary clustering on $\mathbf{z}$
Output: Change detection results $\mathbf{T}$
```

5.2. Setup

For UTT, KA is applied to the HSI tensors $\mathbf{Y}_1$ and $\mathbf{Y}_2$ to represent low-order tensors with higher-order tensors $\mathbf{A}_1$ and $\mathbf{A}_2$. Therefore, the dimensions of high-order tensors after KA should be firstly determined, with $\{I_i\}_{i=1,2,\cdots,5}$ in our case. Different combinations satisfying $\prod_{i=1}^{5} I_i = \prod_{j=1}^{3} J_j$ are attempted to achieve the best CD results, and the settings are different for different datasets, i.e., for the Yancheng dataset, the original tensor size of 450 × 140 × 155 becomes 15 × 30 × 7 × 20 × 155 after the KA scheme. The TT ranks for the UTT model were initialized as $(r_{[i]}, i = 1, 2, \cdots, N-1)$ by the matrix product state (MPS) [55], and the weight parameter α_k was initialized as follows:

$$\alpha_k = \frac{\beta_k}{\sum_{k=1}^{N-1} \beta_k}, \tag{36}$$

where $\beta_k = \min(\prod_{i=1}^{k} I_i, \prod_{i=k+1}^{N} I_i)$ and $k = 1, 2, \cdots, N-1$. Hence, larger weights can be applied to the more balanced matrices in this fashion. To get the initial TT ranks for UTT, each rank $r_{[i]}$ is constrained by keeping only the singular values that satisfy the following:

$$\frac{\lambda_j^{[i]}}{\lambda_1^{[i]}} > th, \tag{37}$$

where $j = 1, 2, \cdots, r_{[i]}$ and $\left\{\lambda_j^{[i]}\right\}$ is arranged in descending order. The threshold th is selected empirically. This criterion is used so that more singular values will be truncated for low-rank matricization.

For STT, the backbone comprises two TTD layers and one TTO layer. Each TTD layer is followed by rectified linear units (ReLU) for nonlinear mapping. The properties of the TTD and TTO layers are investigated, and different methods are compared to determine their parameters, such as the number of TTD/TTO layers, dimensions of the tensors after KA, and the TT-ranks of the compressed weight matrix $\mathbf{W}^{(l)}$. The dimensions of the tensors after the KA scheme and TT ranks $(r_{[i]}^{(l)}, i = 1, 2, 3, 4, 5, l = 1, 2)$ are configured separately for each dataset, i.e., for the Yancheng dataset, a patch size of 30 × 30 × 9 becomes 5 × 6 × 5 × 6 × 9 after the KA scheme, and their ranks for the corresponding dimensions are set as $(r_{[i]}^{(l)} = 3, i = 1, 2, 3, 4, 5, l = 1, 2)$ to control the compression factor. A similar procedure is followed for the remaining datasets with different KA and rank settings in order to achieve the best change detection results.

To initialize the pseudo labels of the classification network in STT, the difference of the bitemporal images is computed after KA, K-Means is used as a classical unsupervised method on the extracted difference image, and C_1 is set as 4. The best parameter settings for μ in Equation (35), which balances the classification loss and clustering loss to achieve the best results in the four datasets mentioned above, are $\mu = [0.35, 0.25, 0.015, 0.34]$, and C_2 is set to 2. The batch size is set to 20 for all the datasets. The default parameter settings may not give the best performance all the time, but they generally give satisfactory results. In the training phase of the proposed self-supervised method, 80% of the samples are chosen randomly for each bitemporal HSI, and the remaining 20% of the random samples are used for validation. During the testing phase, all samples of each bitemporal HSI dataset are used to generate binary change maps.

The proposed model is compared with two classical methods, LSCD and ASCD [30], and six state-of-the-art techniques, HOSVD [20], TDRD [18], PCANet [12], DSFANet [28], HI-DRL [31], and the tucker decomposition-based self-supervised tensor network technique SSTN [38]. For all unsupervised methods, K-Means is used to segment output features into changed areas and unchanged areas. For visual evaluation, white pixels represent the changed areas, and the black pixels represent the unchanged areas.

For numerical evaluation, overall accuracy (OA), class accuracy of unchanged area (CA_UN), class accuracy of changed area (CA_CH), KAPPA coefficient (KAPPA), and Area Under Receiver Operating Characteristic (ROC) Curve (AUC) are used, whereas OA, CA_UN, and CA_CH are presented in percentages, and KAPPA and AUC range from 0 to 1. Each of the indexes indicates better results when it shows higher values. All of the algorithms were developed using TensorFlow in a Python environment, on an NVIDIA GeForce GTX 1080 Ti with 256 GB memory.

5.3. Results

5.3.1. Efficacy

As shown in Figure 6, the visual CD maps produced for the Yancheng dataset by the UTT and STT methods are better than those produced by the classical methods and the counterpart tensor techniques. This viewpoint is supported by the numerical data in the first row of Table 1. It can be seen from Table 1 that although the CA_UN of the proposed method, STT, is not the best, its OA and KAPPA are both the best, which confirms that TT can capture more global correlations. It also proves that TT can detect changed and unchanged regions in a more balanced way (CA_UN and CA_CH of STT are both relatively higher compared to other methods). Though the CA_UN of ASCD is the best, its OA and KAPPA are nearly the worst, which shows that ASCD cannot capture enough global correlations to detect changed regions in a balanced way. The same conclusion can be drawn from the visual CD maps of ASCD and STT in Figure 6c,l respectively. ASCD smooths out the changed features, and the white (changed) regions in the CD map of ASCD in Figure 6c are significantly fewer than STT in Figure 6l, which results in a slightly better CA_UN but a much worse OA. It can also be seen from the white speckles on the top-right and bottom-left corner of Figure 6d,g that results produced by HOSVD and DSFANet have more false alarm rates. As the feature extraction process of HOSVD is based on TD, it can be concluded that the ability of TD in capturing changed information is worse than TT.

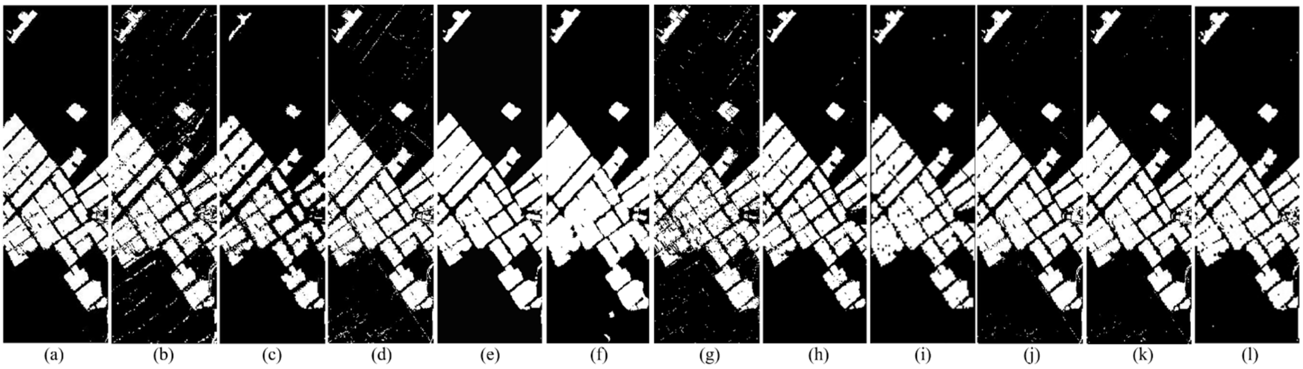

Figure 6. Binary detection results in the Yancheng dataset: (**a**) ground truth, (**b**) LSCD, (**c**) ASCD, (**d**) HOSVD, I TDRD, (**f**) PCANet, (**g**) DSFANet, (**h**) HI-DRL, (**i**) SSTN, (**j**) UTT-SVD, (**k**) UTT-noSVD, and (**l**) STT.

Similar conclusions can be drawn for the River dataset, as shown in the visual CD maps in Figure 7 and the numerical data in the second row of Table 1. It can be seen from Table 1 that though CA_UN and CA_CH of STT are not the best, its OA and KAPPA turn out to be the best. On the contrary, although TDRD and HI-DRL have advantages in detecting changed and unchanged regions, respectively, they cannot capture enough global correlations such that they cannot detect changed and unchanged regions in a balanced way. That is why they have the best CA_CH and CA_UN, respectively, but their OA and KAPPA are relatively lower than STT. The visual CD maps in Figure 7 can further validate the advantages of the proposed TT-based CD methods. Figure 7b,d,e show that the bottom-left corner has visible white speckles as compared to UTT and STT. PCANet detects more white regions (pseudo changes), while HI-DRL suppresses the changed features excessively so that it ignores real changes, as shown in Figure 7f,h, respectively. ASCD in Figure 7c smooths out the changed features and results in the lower prediction of changed and unchanged features, which greatly affect the accuracy. The visual results demonstrate that the proposed methods do not only reduce the white noise, but they also discriminate the changed and unchanged features effectively.

Table 1. Evaluation of change detection results in different datasets.

Data Set	**Metric**	**LS CD**	**AS CD**	**HO SVD**	**TD RD**	**PCA-Net**	**DSFA-Net**	**HI-DRL**	**SSTN**	**UTT-SVD**	**UTT-noSVD**	**STT**
Yancheng	CA_UN(%)	94.33	**99.94**	97.79	95.67	94.11	97.01	99.10	98.19	98.17	98.15	97.79
	CA_CH(%)	85.13	75.89	**98.69**	95.09	98.35	92.40	92.00	96.25	97.78	98.03	**98.69**
	OA(%)	91.73	93.13	98.04	97.31	95.31	95.71	97.09	97.64	98.07	98.11	**98.20**
	KAPPA	0.7959	0.8176	0.9524	0.9345	0.8890	0.8942	0.9270	0.9420	0.9528	0.9536	**0.9561**
River	CA_UN(%)	92.71	97.92	92.37	92.03	83.17	97.06	**98.76**	97.34	92.16	93.71	98.44
	CA_CH(%)	44.13	40.32	90.72	**94.03**	68.55	66.21	58.48	72.84	91.01	85.95	69.24
	OA(%)	88.48	92.93	92.23	92.21	81.52	94.25	94.23	94.58	92.30	93.10	**95.90**
	KAPPA	0.3364	0.4768	0.6283	0.6365	0.3590	0.6768	0.6640	0.7210	0.6285	0.6462	**0.7237**
Bay Area	CA_UN(%)	86.90	**99.54**	87.30	85.96	89.36	82.19	97.59	93.17	90.72	91.81	94.29
	CA_CH(%)	31.44	11.02	42.96	45.28	38.50	**48.64**	25.02	36.60	42.73	39.54	38.72
	OA(%)	73.32	77.86	76.44	75.37	78.23	73.97	**81.71**	80.79	78.97	78.98	81.02
	KAPPA	0.2028	0.1500	0.3220	0.3006	0.3040	0.3046	0.2970	0.3460	0.3708	0.3652	**0.3841**
Hermiston	CA_UN(%)	94.77	99.74	98.14	76.82	85.48	98.76	**99.76**	99.33	98.20	98.48	99.02
	CA_CH(%)	80.88	70.13	93.36	55.55	67.78	91.35	83.86	93.88	93.29	92.95	**95.02**
	OA(%)	92.99	95.95	97.53	74.09	83.19	97.81	97.72	98.63	97.57	97.76	**98.83**
	KAPPA	0.7068	0.7938	0.8922	0.2181	0.4140	0.9018	0.8910	0.9380	0.8936	0.9010	**0.9384**

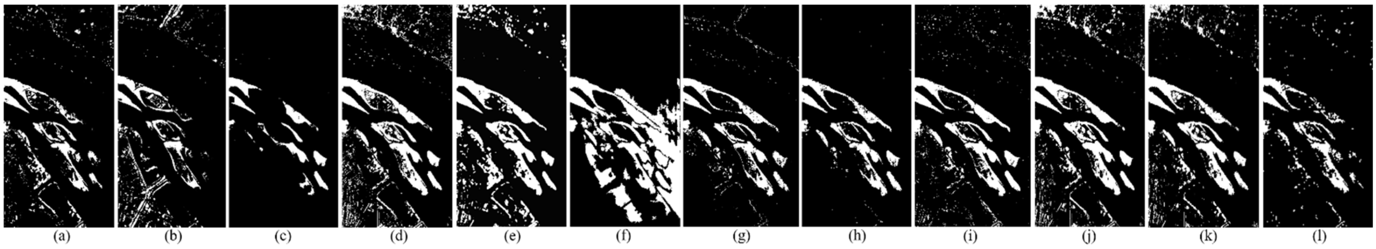

Figure 7. Binary detection results in the River dataset: (**a**) ground truth, (**b**) LSCD, (**c**) ASCD, (**d**) HOSVD, (**e**) TDRD, (**f**) PCANet, (**g**) DSFANet, (**h**) HI-DRL, (**i**) SSTN, (**j**) UTT-SVD, (**k**) UTT-noSVD, and (**l**) STT.

The visual CD results and numerical data of the Bay Area and Hermiston datasets are shown in Figure 8 and the third row of Table 1, and in Figure 9 and the fourth row of Table 1, respectively. For the Bay Area dataset, although the OA of HI-DRL is the best, its CA_CH is nearly the worst, which limits the practical application of this method in HSI CD. Similarly, ASCD and DSFANet cannot detect changed and unchanged regions in a balanced way despite the highest CA_UN and CA_CH that they respectively have, which brings out relatively lower OA and KAPPA compared to STT. As for the Hermiston dataset, although the CA_UN of STT is slightly lower than HI-DRL, its CA_CH is significantly better than HI-DRL and all other methods, which makes STT obtain the best OA and KAPPA compared to all other classical and state-of-the-art methods. The visual results of CD maps in Figures 8 and 9 can further validate the mentioned conclusions.

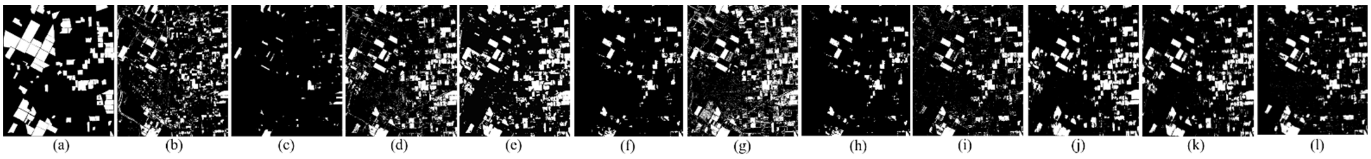

Figure 8. Binary detection results in the Bay Area dataset: (**a**) ground truth, (**b**) LSCD, (**c**) ASCD, (**d**) HOSVD, (**e**) TDRD, (**f**) PCANet, (**g**) DSFANet, (**h**) HI-DRL, (**i**) SSTN, (**j**) UTT-SVD, (**k**) UTT-noSVD, and (**l**) STT.

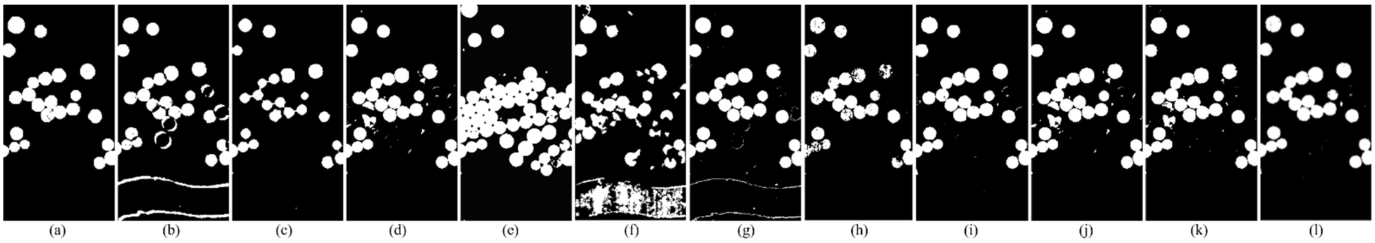

Figure 9. Binary detection results in the Hermiston dataset: (**a**) ground truth, (**b**) LSCD, (**c**) ASCD, (**d**) HOSVD, (**e**) TDRD, (**f**) PCANet, (**g**) DSFANet, (**h**) HI-DRL, (**i**) SSTN, (**j**) UTT-SVD, (**k**) UTT-noSVD, and (**l**) STT.

A comparison of AUC values between different methods is also made, and the results are shown in Figure 10 and Table 2, respectively. It can be seen from Figure 10a,b that the ROC curve of STT in the Yancheng and Hermiston datasets is almost at the upper left corner as compared to other methods. In Figure 10c of the River dataset, UTT without SVD outperforms the other methods, which shows that TT-based methods outperform other methods not only in OA and KAPPA, but also in AUC. The numerical results in Table 2 also validate our conclusion. In the Bay Area dataset shown in Figure 10d, although all methods have a lower AUC compared to other datasets, STT still obtained the highest AUC. However, as an evaluation index, the AUC of ROC has its own limits. For example, different ROC curves may have similar AUC. This can be observed in Table 2 and Figure 10. For the Bay Area dataset, the AUC of ASCD is 0.6927 and the AUC of UTT-noSVD is 0.6907, which are very close. Meanwhile, the indexes such as OA and KAPPA in Table 1 show that UTT-noSVD largely outperforms ASCD. Therefore, an evaluation of the CD methods by different indexes is recommended.

Table 2. Evaluation of AUC values for different methods.

Methods	Yancheng	Hermiston	River	Bay Area
LSCD	0.9400	0.9224	0.8629	0.5968
ASCD	0.9796	0.9420	0.8082	0.6927
HOSVD	0.9960	0.9908	0.9719	0.6685
TDRD	0.9968	0.8022	0.9807	0.6699
DSFANet	0.9887	0.9841	0.9273	0.6595
UTT-SVD	0.9964	0.9910	0.9813	0.6863
UTT-noSVD	0.9967	0.9919	**0.9817**	0.6907
STT	**0.9970**	**0.9951**	0.9531	**0.6978**

The above analysis shows that the proposed technique surpasses the current clustering unsupervised algorithms in terms of OA, CA_UN, CA_CH, KAPPA, and AUC accuracy. This is because the characteristics learned using these techniques are not discriminative enough for clustering. Unlike unsupervised techniques such as subspace projection, HOSVD, and SSTN, our TT-based clustering methodology uses a balanced matricization scheme to capture the global correlation of tensor elements for change detection and can thus suppress the unchanged features and enhance changed features at the same time. In Figure 11, the discriminative efficacy of STT is further highlighted. Because of the overlapping samples from multiple classes, the initial high-order difference tensors are obviously unsuitable for clustering, as shown in Figure 11a,c for the Yancheng and Hermiston datasets, respectively. Meanwhile, as shown in Figure 11b,d, the features of different categories learned by STT are considerably separated. The accuracy of clustering and change detection increases due to the discriminative features having high intra-class similarity and inter-class dissimilarity.

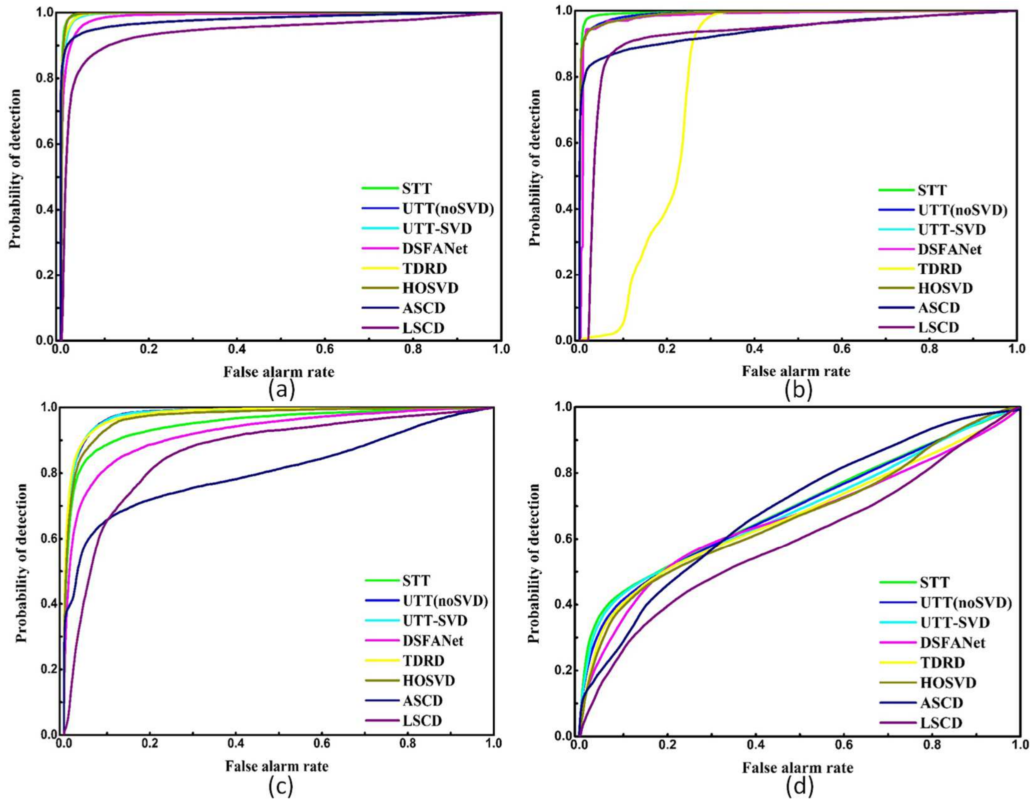

Figure 10. ROCs of different methods for the (**a**) Yancheng dataset (**b**) Hermiston dataset (**c**) River dataset (**d**) Bay Area dataset.

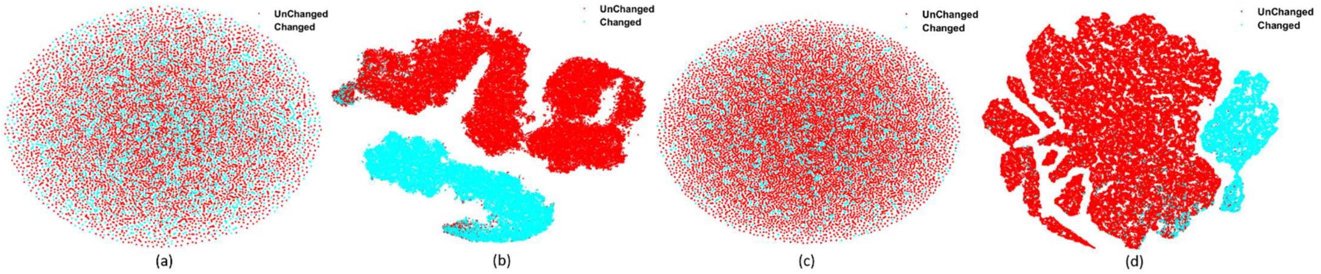

Figure 11. Scatter plots produced by t-SNE of all samples from the Yancheng dataset (**a**,**b**) and Hermiston dataset (**c**,**d**). Red circles represent unchanged samples, and blue circles represent changed samples. Yancheng (**a**) original difference tensor, (**b**) features learnt by STT, Hermiston (**c**) original difference tensor, (**d**) features learnt by STT.

5.3.2. Ablation Study

To validate the effectiveness of the proposed TT methods for CD, an ablation study is conducted for unsupervised TT CD. The experiments are carried out using UTT with SVD, named UTT-SVD, and are compared with the proposed UTT without SVD, named UTT-noSVD. As shown in Figure 6j,k, the proposed UTT-noSVD provides better visual results. Moreover, the numerical evaluation listed in Table 1 validates our point of view for the Yancheng dataset. The ablation experiments for the other datasets for UTT-SVD and the UTT-noSVD are shown in Figure 7j,k for the River dataset, Figure 8j,k for the Bay Area dataset, and in Figure 9j,k for the Hermiston dataset. All the results show that UTT-noSVD is more efficient and provide better results as compared to the UTT-SVD. Their corresponding OA and KAAPA accuracy values, listed in Table 1, also support our point of view.

Similarly, for STT, an ablation study is conducted by lower-dimensional input patch size without using the KA scheme. Lower three-dimensional input patches are compared with the proposed higher five-dimensional input patch sizes. In Figure 12, the visual change detection maps of the River and Bay Area datasets clearly show that the proposed STT model performs better on a higher dimension (5-dimensional input patches), as seen in Figure 12a,c, as compared to a lower dimension (3-dimensional input patches), as seen in Figure 12b,d. This is because high-order rich texture features can be extracted after using KA. Table 3 demonstrates OA and KAPPA accuracy by using different input dimensions for the River and Bay Area datasets. The visual results and numerical evaluation show that STT using a higher dimension performs better than that with low-dimension input patches.

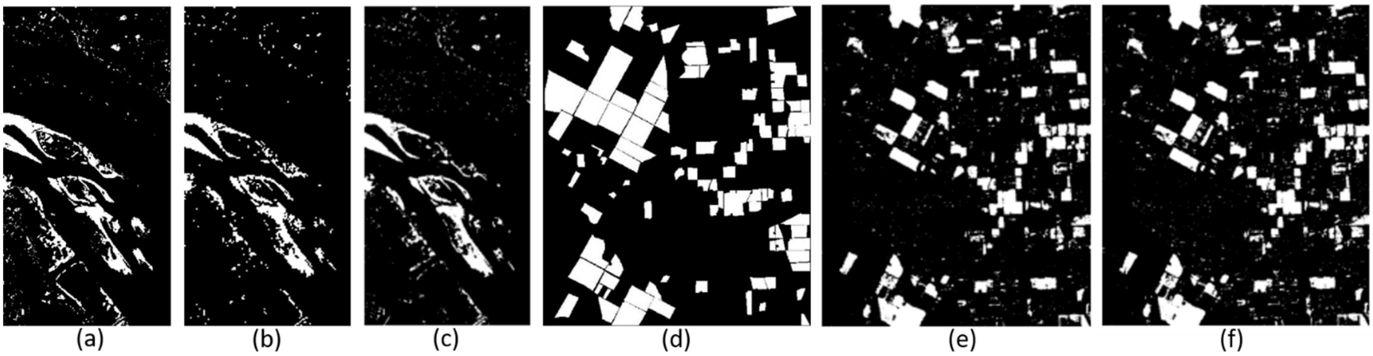

Figure 12. Ablation study of STT. River dataset (**a**) ground truth, (**b**) with 5-dimensional input patches, (**c**) with 3-dimensional input patches; Bay Area dataset (**d**) ground truth, (**e**) with 5-dimensional input patches, and (**f**) with 3-dimensional input patches.

Table 3. Ablation study: STT with 5-dimensional vs. 3-dimensional input patches.

Methods	**River**		**Bay Area**	
	OA (%)	**KAPPA**	**OA (%)**	**KAPPA**
STT(3-Dimensional)	95.62	0.7080	80.36	0.3820
STT(5-Dimensional)	**95.90**	**0.7237**	**81.02**	**0.3841**

5.3.3. Efficiency

Table 4 lists the time cost per iteration and the total time cost of convergence of comparable techniques. Each time cost is the average value observed from all four HSI datasets. While the same strategy and the same computation platform is adopted, LSCD and ASCD are three times slower than UTT-noSVD and HOSVD, and four times slower than UTT-noSVD for the Yancheng dataset. For the Bay Area dataset, LSCD is 40 times slower than the proposed UTT-noSVD, ASCD is 25 times slower, and HOSVD is 8 times slower than UTT-noSVD. For the River dataset, LSCD is 38 times slower, ASCD is 19 times, and HSOVD is 11 times slower than the proposed UTT-noSVD. Similarly, for the Hermiston city dataset, UTT-noSVD outperforms the LSCD, ASCD, and HOSVD methods with an 18, 7, and 8-fold difference, respectively. The efficiency of UTT-noSVD is benefited from bypassing SVD to compute the TT-ranks and applying KA to the hyperspectral data.

Table 4. Execution time of various methods using different experimental datasets (seconds).

Methods	Yancheng	Bay Area	River	Hermiston
LSCD	3.945	217.662	38.804	18.838
ASCD	3.866	129.609	20.487	7.421
HOSVD	4.908	43.480	11.830	9.316
UTT-SVD	8.992	262.747	5.709	21.018
UTT-noSVD	**1.476**	**5.441**	**0.934**	**0.678**

The convergence curve of STT for the Yancheng dataset is shown in Figure 13a. It shows that network loss (combined loss) converges more smoothly and reduces the error efficiently as compared to the use of individual classification and cluster loss. For the Hermiston dataset, the STT convergence curves for the cluster, classification, and network losses are shown in Figure 13b. It shows that the combination of cluster loss with classification loss produces better convergence. The convergence graphs in Figure 13c,d also validate the efficiency of the proposed self-supervised TT technique for the River and Bay Area dataset, respectively.

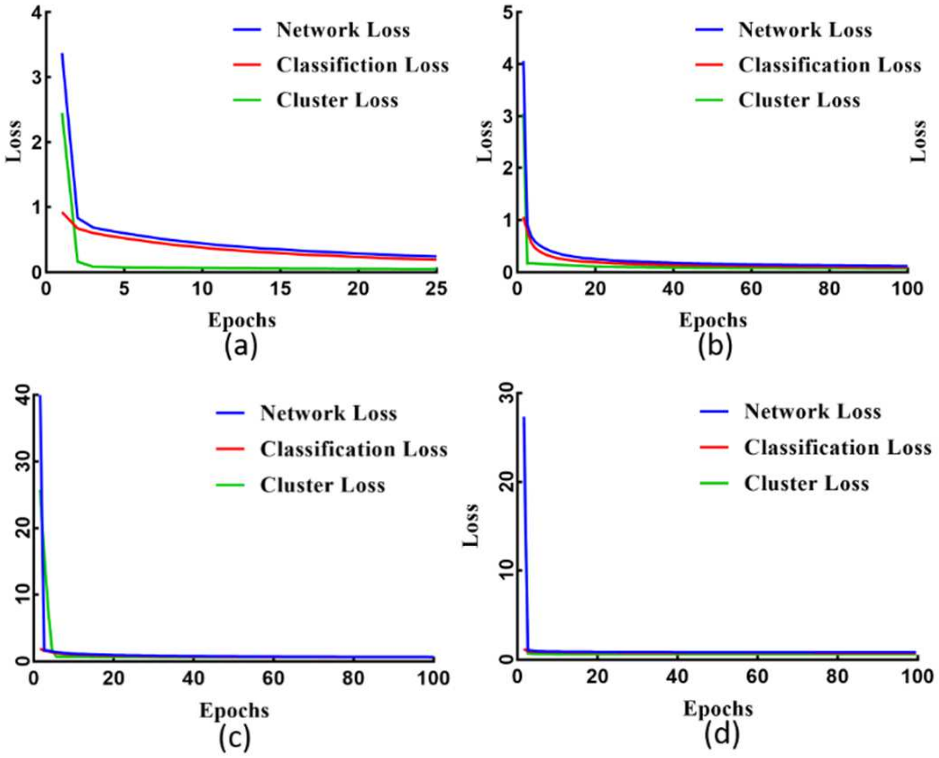

Figure 13. Convergence curve of STT for different datasets. (**a**) Yancheng dataset, (**b**) Hermiston dataset, (**c**) River dataset, (**d**) Bay Area dataset.

5.3.4. Discussions

The experiments above demonstrate the capability of TT-based CD in multitemporal HSIs. The strength of UTT and STT can be summarized as follows:

(1). The inter-class homogeneity and inner-class heterogeneity of HSIs are addressed by UTT and STT effectively by exploiting the ability of TT in capturing global correlations. To be specific, UTT and STT can detect changed and unchanged regions in a more balanced way due to the correlations that TT captures between the changed and unchanged information contained in the original HSIs. This can be validated by the better OA, KAPPA, and AUC values of UTT and STT as compared to the TD-based methods HOSVD [20], TDRD [18], and SSTN [38], whose low-rank features are extracted in an unbalanced way. The T-SNE results in Figure 11 also indicate that the features extracted by the TT-based methods are discriminative enough to differentiate between the changed and unchanged regions in HSI CD.

(2). Both UTT and STT successfully handle the high dimensionality of HSIs by TT decomposition, which decomposes N-order weight tensors into small three-order tensor cores by approximating the low-order optimal TT ranks. Hence, the dimensionality can be reduced and the redundant information can be removed. At the same time, the execution time of UTT-noSVD is obviously lower than other existing unsupervised HSI CD methods such as LSCD [30]. Costly manual annotations can also be removed as unsupervised learning and self-learning are introduced into UTT and STT, respectively.

(3). Tensor augmentation is achieved through the KA scheme, which involves replacing a low-order tensor with a higher-order tensor without changing the number of tensor entries. Therefore, a high-order tensor with richer texture features can be achieved without increasing computation complexity. It can be seen in Figure 12 and Table 3 that KA indeed works in our proposed methods.

Despite all of the benefits of the TT-based change detection methods UTT/STT, there is still potential for improvement. Here are few points that could be further studied. First, UTT/STT both use K-Means for binary classification as well as during the pre-clustering stage to cluster the changed and unchanged features. The cluster number specification in the advance non-handling of outliers and their dependency on convex data cause the K-Means to be inadequate for complicated datasets. Similarly, during pre-clustering, pseudo-labels can be generated by more advanced methods to improve the accuracy of STT. Second, although STT and UTT are used for change and background classification for bitemporal images, they could be further applied to multiple change detection with multitemporal images.

6. Conclusions

Inspired by knowledge from the quantum information theory, this paper proves that TT decomposition is capable of capturing more global correlations between the changed/unchanged information than TD. Based on this, two novel approaches are proposed for HSI CD. The unsupervised UTT technique is based on multilinear matrix factorization and is more efficient since it bypasses the expensive SVD. STT, the second one, is a self-supervised technique. STT performs much better than the direct unsupervised technique, UTT. TT-based CD methods use a well-balanced matricization strategy so that they can detect changed and unchanged regions in a more balanced way. Tensors are augmented from low-order to higher-order through a KA scheme while not changing the total number of entries in order to extract changed features efficiently without increasing computational complexity. Then, the proposed TT-based CD methods are tested on four real-world bitemporal HSI datasets. Experiments have demonstrated that the proposed TT-based CD methods outperform their tensor counterparts TDRD, HOSVD, SSTN, and other state-of-the-art techniques in OA, CA_UN, CA_CH, KAPPA, and AUC. With such high accuracy and no need for manual annotations, the proposed methods exhibit the potential of applications to deal with emergencies such as landslide detection and earthquake damage estimation. For future work, more real-world multitemporal HSI datasets will be acquired to further investigate the proposed methods and update the UTT/STT so that it could be adapted for change detection in multitemporal HSIs.

Author Contributions: Conceptualization, Z.C.; methodology, Z.C. (all), M.S. (part of Sections 4.2 and 4.3), and H.W (part of Section 4.1); validation, M.S.; writing—original draft preparation, M.S.; writing—review and editing, H.W.; visualization, H.W.; supervision, Z.C. and G.L.; funding acquisition, Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded in part by the "Chenguang Program", supported by the Shanghai Education Development Foundation and the Shanghai Municipal Education Commission under

Grant 18CG38, the National Natural Science Foundation of China under Grant 61702094, and the Science and Technology Commission of Shanghai Municipal under Grant 17YF1427400.

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

References

1. Bovolo, F.; Marchesi, S.; Bruzzone, L. A framework for automatic and unsupervised detection of multiple changes in multitemporal images. *IEEE Trans. Geosci. Remote Sens.* **2011**, *50*, 2196–2212. [CrossRef]
2. Liu, S.; Bruzzone, L.; Bovolo, F.; Du, P. Hierarchical unsupervised change detection in multitemporal hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 244–260.
3. Deng, J.S.; Wang, K.; Deng, Y.H.; Qi, G.J. PCA based land use change detection and analysis using multitemporal and multisensor satellite data. *Int. J. Remote Sens.* **2008**, *29*, 4823–4838. [CrossRef]
4. Lu, D.; Mausel, P.; Brondizio, E.; Moran, E. Change detection techniques. *Int. J. Remote Sens.* **2004**, *25*, 2365–2401. [CrossRef]
5. Kennedy, R.E.; Townsend, P.A.; Gross, J.E.; Cohen, W.B.; Bolstad, P.; Wang, Y.Q.; Adams, P. Remote sensing change detection tools for natural resource managers: Understanding concepts and tradeoffs in the design of landscape monitoring projects. *Remote Sens. Environ.* **2009**, *113*, 1382–1396. [CrossRef]
6. Coppin, P.R.; Bauer, M.E. Digital change detection in forest ecosystems with remote sensing imagery. *Remote Sens. Rev.* **1996**, *13*, 207–234. [CrossRef]
7. Yang, X.; Chen, L. Using multi-temporal remote sensor imagery to detect earthquake-triggered landslides. *Int. J. Appl. Earth Obs. Geoinf.* **2010**, *12*, 487–495. [CrossRef]
8. Ma, L.; Li, M.; Blaschke, T.; Ma, X.; Tiede, D.; Cheng, L.; Chen, Z.; Chen, D. Object-based change detection in urban areas: The effects of segmentation strategy, scale, and feature space on unsupervised methods. *Remote Sens.* **2016**, *8*, 761. [CrossRef]
9. Hasanlou, M.; Seydi, S.T. Hyperspectral change detection: An experimental comparative study. *Int. J. Remote Sens.* **2018**, *39*, 7029–7083. [CrossRef]
10. Guo, Q.; Zhang, J.; Zhong, C.; Zhang, Y. Change Detection for Hyperspectral Images via Convolutional Sparse Analysis and Temporal Spectral Unmixing. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4417–4426. [CrossRef]
11. Wu, C.; Zhang, L.; Du, B. Kernel slow feature analysis for scene change detection. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2367–2384. [CrossRef]
12. Gao, F.; Dong, J.; Li, B.; Xu, Q. Automatic change detection in synthetic aperture radar images based on PCANet. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1792–1796. [CrossRef]
13. 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 of high resolution satellite images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *14*, 1194–1206. [CrossRef]
14. Hou, B.; Liu, Q.; Wang, H.; Wang, Y. From W-Net to CDGAN: Bitemporal change detection via deep learning techniques. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 1790–1802. [CrossRef]
15. Wang, Q.; Yuan, Z.; Du, Q.; Li, X. GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 3–13. [CrossRef]
16. 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]
17. Yang, B.; Fu, X.; Sidiropoulos, N.D.; Hong, M. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 3861–3870.
18. Hou, Z.; Li, W.; Tao, R.; Du, Q. Three-Order Tucker Decomposition and Reconstruction Detector for Unsupervised Hyperspectral Change Detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 6194–6205. [CrossRef]
19. Huang, F.; Yu, Y.; Feng, T. Hyperspectral remote sensing image change detection based on tensor and deep learning. *J. Vis. Commun. Image Represent.* **2019**, *58*, 233–244. [CrossRef]
20. Chen, Z.; Wang, B.; Niu, Y.; Xia, W.; Zhang, J.Q.; Hu, B. Change detection for hyperspectral images based on tensor analysis. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Milan, Italy, 26–31 July 2015; pp. 1662–1665.
21. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [CrossRef]
22. Cichocki, A. Era of big data processing: A new approach via tensor networks and tensor decompositions. *arXiv* **2014**, arXiv:1403.2048.
23. Nielsen, M.A.; Chuang, I. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2002.
24. Oseledets, I.V. Tensor-train decomposition. *SIAM J. Sci. Comput.* **2011**, *33*, 2295–2317. [CrossRef]
25. Bengua, J.A.; Phien, H.N.; Tuan, H.D.; Do, M.N. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Trans. Image Process.* **2017**, *26*, 2466–2479. [CrossRef]
26. Latorre, J.I. Image compression and entanglement. *arXiv* **2005**, arXiv:quant-ph/0510031.
27. Du, P.; Liu, S.; Gamba, P.; Tan, K.; Xia, J. Fusion of difference images for change detection over urban areas. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *5*, 1076–1086. [CrossRef]
28. Du, B.; Ru, L.; Wu, C.; Zhang, L. Unsupervised deep slow feature analysis for change detection in multi-temporal remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9976–9992. [CrossRef]

29. Li, Q.; Gong, H.; Dai, H.; Li, C.; He, Z.; Wang, W.; Mu, T. Unsupervised Hyperspectral Image Change Detection via Deep Learning Self-Generated Credible Labels. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 9012–9024. [CrossRef]
30. Wu, C.; Du, B.; Zhang, L. A subspace-based change detection method for hyperspectral images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 815–830. [CrossRef]
31. Zhang, P.; Gong, M.; Zhang, H.; Liu, J.; Ban, Y. Unsupervised difference representation learning for detecting multiple types of changes in multitemporal remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 2277–2289. [CrossRef]
32. López-Fandiño, J.; Garea, A.S.; Heras, D.B.; Arguello, F. Stacked autoencoders for multiclass change detection in hyperspectral images. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 1906–1909.
33. Hou, Z.; Li, W.; Li, L.; Tao, R.; Du, Q. Hyperspectral change detection based on multiple morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5507312. [CrossRef]
34. Seydi, S.T.; Shah-Hosseini, R.; Hasanlou, M. New framework for hyperspectral change detection based on multi-level spectral unmixing. *Appl. Geomat.* **2021**, *13*, 763–780. [CrossRef]
35. Seydi, S.T.; Hasanlou, M. A New Structure for Binary and Multiple Hyperspectral Change Detection Based on Spectral Unmixing and Convolutional Neural Network. *Measurement* **2021**, *186*, 110137. [CrossRef]
36. Hasanlou, M.; Seydi, S.T.; Shah-Hosseini, R. A sub-pixel multiple change detection approach for hyperspectral imagery. *Can. J. Remote Sens.* **2018**, *44*, 601–615. [CrossRef]
37. Jafarzadeh, H.; Hasanlou, M. An unsupervised binary and multiple change detection approach for hyperspectral imagery based on spectral unmixing. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 4888–4906. [CrossRef]
38. Zhou, F.; Chen, Z. Hyperspectral Image Change Detection by Self-Supervised Tensor Network. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 19–24 July 2020; pp. 2527–2530.
39. Zhang, R.; Isola, P.; Efros, A.A. Colorful image colorization. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 649–666.
40. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544.
41. Noroozi, M.; Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 69–84.
42. 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 June 2015; pp. 3431–3440.
43. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
44. Dong, H.; Ma, W.; Jiao, L.; Liu, F.; Li, L. A Multiscale Self-Attention Deep Clustering for Change Detection in SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–16. [CrossRef]
45. Chen, Y.; Bruzzone, L. Self-supervised Change Detection in Multi-view Remote Sensing Images. *arXiv* **2021**, arXiv:2103.05969.
46. Deng, Y.J.; Li, H.C.; Fu, K.; Du, Q.; Emery, W.J. Tensor low-rank discriminant embedding for hyperspectral image dimensionality reduction. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 7183–7194. [CrossRef]
47. Zhao, M.; Li, W.; Li, L.; Ma, P.; Tao, R. Three-order tensor creation and tucker decomposition for infrared small-target detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *99*, 1–16. [CrossRef]
48. Li, L.; Li, W.; Qu, Y.; Zhao, C.; Tao, R.; Du, Q. Prior-based tensor approximation for anomaly detection in hyperspectral imagery. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 1037–1050. [CrossRef]
49. Velasco-Forero, S.; Angulo, J. Classification of hyperspectral images by tensor modeling and additive morphological decomposition. *Pattern Recognit.* **2013**, *46*, 566–577. [CrossRef]
50. Zniyed, Y. Breaking the Curse of Dimensionality Based on Tensor Train: Models and Algorithms. Ph.D. Thesis, Paris-Saclay, Paris, France, 2019.
51. Henderson, L.; Vedral, V. Classical, quantum and total correlations. *J. Phys. A. Math. Gen.* **2001**, *34*, 6899–6905. [CrossRef]
52. Novikov, A.; Podoprikhin, D.; Osokin, A.; Vetrov, D. Tensorizing neural network. *arXiv* **2015**, arXiv:1509.06569.
53. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
54. López-Fandiño, J.; B Heras, D.; Arguello, F.; Dalla, M.M. GPU framework for change detection in multitemporal hyperspectral images. *Int. J. Parallel Program.* **2019**, *47*, 272–292. [CrossRef]
55. Perez-Garcia, D.; Verstraete, F.; Wolf, M.M.; Cirac, J.I. Matrix product state representations. *arXiv* **2006**, arXiv:quant-ph/0608197. [CrossRef]