



Article

PolSAR Scene Classification via Low-Rank Constrained Multimodal Tensor Representation

Bo Ren ^{1,*}, Mengqian Chen ¹, Biao Hou ¹, Danfeng Hong ², Shibin Ma ¹, Jocelyn Chanussot ³ and Licheng Jiao ¹

¹ School of Artificial Intelligence, Xidian University, No. 2 South Taibai Road, Yanta District, Xi'an 710071, China; chenmq@bjxx.tech (M.C.); houbiao@mail.xidian.edu.cn (B.H.); sbma@stu.xidian.edu.cn (S.M.); lchjiao@mail.xidian.edu.cn (L.J.)

² Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; hongdf@aircas.ac.cn

³ CNRS, Grenoble INP, GIPSA-Lab, Université Grenoble Alpes, 38000 Grenoble, France; jocelyn.chanussot@gipsala.grenobleinp.fr

* Correspondence: boren@xidian.edu.cn

Abstract: Polarimetric synthetic aperture radar (PolSAR) data can be acquired at all times and are not impacted by weather conditions. They can efficiently capture geometrical and geographical structures on the ground. However, due to the complexity of the data and the difficulty of data availability, PolSAR image scene classification remains a challenging task. To this end, in this paper, a low-rank constrained multimodal tensor representation method (LR-MTR) is proposed to integrate PolSAR data in multimodal representations. To preserve the multimodal polarimetric information simultaneously, the target decompositions in a scene from multiple spaces (e.g., Freeman, H/A/ α , Pauli, etc.) are exploited to provide multiple pseudo-color images. Furthermore, a representation tensor is constructed via the representation matrices and constrained by the low-rank norm to keep the cross-information from multiple spaces. A projection matrix is also calculated by minimizing the differences between the whole cascaded data set and the features in the corresponding space. It also reduces the redundancy of those multiple spaces and solves the out-of-sample problem in the large-scale data set. To support the experiments, two new PolSAR image data sets are built via ALOS-2 full polarization data, covering the areas of Shanghai, China, and Tokyo, Japan. Compared with state-of-the-art (SOTA) dimension reduction algorithms, the proposed method achieves the best quantitative performance and demonstrates superiority in fusing multimodal PolSAR features for image scene classification.

Keywords: PolSAR; scene classification; multimodal features; low-rank; tensorial representations



Citation: Ren, B.; Chen, M.; Hou, B.; Hong, D.; Ma, S.; Chanussot, J.; Jiao, L. PolSAR Scene Classification via Low Rank Constrained Multimodal Tensor Representation. *Remote Sens.* **2022**, *14*, 3117. <https://doi.org/10.3390/rs14133117>

Academic Editors: Xiangtao Zheng, Fulin Luo and Qi Wang

Received: 12 May 2022

Accepted: 27 June 2022

Published: 28 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

Polarimetric synthetic aperture radar (PolSAR), a kind of multichannel synthetic aperture radar (SAR), has four channels, HH, HV, VH, and VV, which are formed according to the vertical and horizontal transmission and reception of electromagnetic waves. The polarization of back-scattering waves changes after being reflected by the targets; then, the polarimetric sensors provide more information to describe the land surface structure, land covers, geometrical structures, etc. The targets on the ground can be described using a so-called scattering coefficient that can represent the interaction of an electromagnetic wave with the targets [1].

Compared with optical radar, PolSAR data can actively obtain more information on land cover by alternately transmitting and receiving electromagnetic waves. After obtaining more comprehensive backscattering information, this multichannel sensor can describe land covers from many perspectives, such as color and texture in pseudo-color images, components in coherent or incoherent target decomposition algorithms and parameters, and coefficients in back-scattering matrices [2–4]. Recently, lots of researchers have devoted

themselves to PolSAR land cover classification, segmentation, clustering, or other pixel-level tasks [5–7].

With the development of radar imaging techniques, more sensors with high precision and polarimetric channels (such as ALOS-2, GF3, and TerraSAR-X) have been launched. The continual launching of remote sensing satellites produces a monotonically increasing amount of data. These satellites provide large amounts of data for learning and produce more high-resolution images to extend the research field from pixel-level to target-level and scene-level tasks [8]. Limited by the difficulty of accessing the full polarimetric data, they lag behind the development of PolSAR image scene classification. However, there is still a huge potential for PolSAR data to deal with scene-level data understanding [9].

PolSAR data have multimodal features in different representation spaces to create a comprehensive description of the targets. The eigenvalue decomposition $H/A/\alpha$ with Wishart-based distance was adopted for unsupervised classification in the early days [10]. Furthermore, target decomposition theorems (including model-, eigenvector-, or eigenvalue-based incoherent and coherent decompositions) are exploited to provide physical interpretations of the targets, in which some feature extraction methods are proposed to describe the PolSAR samples as much as possible [1,11–16]. In the literature [2], texture is first investigated as a valuable resource for PolSAR classification, and some image feature extraction techniques (such as the gray-level co-occurrence matrix [17,18], Gabor wavelets [19,20], fractal features [21,22], etc.) are analyzed to represent the pixels in the image.

Overall, with those multimodal features, one still needs to analyze the properties of those features and make a comprehensive fusion technique for the corresponding tasks. Current existing multi-view space clustering (MSC) methods open a new avenue for combining those features, which maintains consistency among different feature spaces in a clustering structure [23]. Some low-rank or sparse constraints are exploited to construct the affinity matrix and encourage the sparsity and low-rankness of the solution [24]. Especially for image-level tasks, an image can be represented by a tensor; then, the tensor-based MSC method is designed, which may also be constrained by a variety of regularization terms [25]. PolSAR data are represented by features in multiple spaces, and PolSAR images can be formulated as tensors. Therefore, determining how to make full use of those PolSAR tensorial forms and exploit the underlying properties in a multimodal features space is very significant for PolSAR data interpretation.

Inspired by multi-view space clustering algorithms, a novel low-rank constrained multimodal tensor representation method (LR-MTR) is proposed to combine multiple features in PolSAR data for scene classification. Different from the most prevalent existing tensor-based PolSAR or hyperspectral land cover classification methods regarding each sample as a tensor [26–30], the proposed method constructs all the representation matrices in different spaces as a tensor, as shown in Figure 1. Furthermore, the high-order representation tensor is constrained by a low-rank norm to model the cross-information from multiple different subspaces. It reduces the unnecessary information of subspace representations in calculations and captures the complementary information among those multiple features. A projection matrix is also calculated by minimizing the difference between all the cascaded features and the features in the corresponding space, which will reduce the difference between the global and single feature spaces. The learned projection matrix can effectively solve the out-of-sample problem in a large-scale data set. Two typical matrices are alternately updated in the optimization: the affinity matrices learned from the multimodal PolSAR data samples and the projection matrix calculated by the representation tensor. Finally, to produce the final results, K-nearest neighbor (KNN) and support vector machine (SVM) classifiers are applied. To solve the complex minimization problem, an augmented Lagrangian alternating direction minimization (AL-ADM) [31,32] algorithm has been applied.

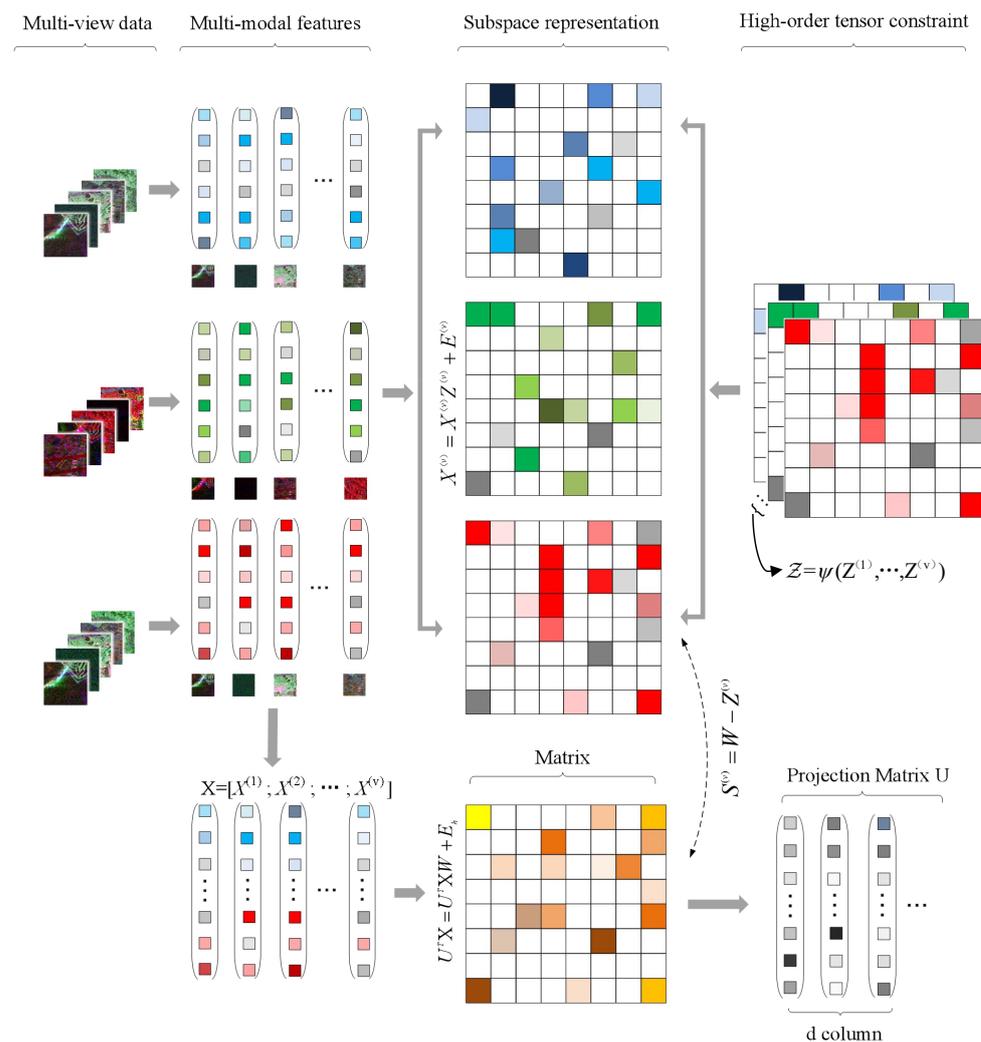


Figure 1. Overview of LR-MTR.

This study has four contributions, as follows:

(1) A new dimension reduction algorithm is introduced. This new algorithm integrates the original multimodal features and extracts an eigenvalue decomposition-based projection matrix. We develop a new framework for image feature extraction. Our method can search the optimal projection matrix and optimal subspace simultaneously.

(2) PolSAR coherency/covariance/scattering matrices and various polarimetric target decomposition can be utilized to describe PolSAR data in multimodal feature spaces. To derive a comprehensive representation, the pseudo-color images from Freeman, Cloude, and Pauli are provided to represent the data. This paper describes PolSAR data via multimodal images and tests the visual information via some efficiency texture extraction methods.

(3) To obtain the correlation among the underlying multimodal data and to congregate those features, LR-MTR regards all the subspace features as high-order, so it can be seen as a tensor. With the constraint of a low-rank term, the tensor models the cross-information and obtains representation matrices in the multimodal subspace. LR-MTR reduces the redundancy of the learned subspace representation. Meanwhile, the LR-MTR algorithm also performs with better accuracy on PolSAR scene classification.

(4) Due to the difficulty of accessing PolSAR data, studies in PolSAR image scene classification are relatively rare. To effectively process and investigate this field, we use the semantic information to label images according to the actual land cover situation. Two new full-polarization PolSAR data sets covering Shanghai and Tokyo, produced by ALOS-2, are exploited to build scene classification data.

On the algorithm level, this method introduces a novel multimodal tensor representation method. It does not directly represent samples in the data domain, but constructs a representation tensor under a new tensor low-rank constraint. It learns a representative projection matrix to reduce the redundant information and projects all the features from different modalities into the representation space. Furthermore, the fused features are obtained for the subsequent PolSAR scene classification.

We organize the remaining parts of this study as follows. Section 2 introduces the related works. In Section 3, we introduce the motivation of our algorithm, the theory and methodology of the proposed algorithm, and the optimal solution. In Section 4, experiments are undertaken to show the efficiency of our algorithm. The experimental analysis and visualization results are used to present our data sets. In Section 5, the conclusion is provided.

2. Related Work

2.1. Multi-View Learning

In the real world, a sample can be described by the features of a multi-view space, the same way a person can be recognized by their face, fingerprints, iris, and gait. Many kinds of multi-view methods have been used in different fields. In general, there are three main kinds of multi-view methods: co-training, co-regularizations, and margin-consistency style algorithms [33,34].

Co-regularized approaches design specific regularization terms for objective functions to explore the complementary information by jointly regularizing the hypotheses within two or more distinct views [35]. The co-training approach is one of the earliest algorithms to train alternately to maximize the manual information and solve the problem of semi-supervised learning [24,36]. The margin-consistency-style algorithm was recently proposed to regularize the margins from multi-view data consistently in large-margin classifiers [37,38]. Apart from those above-mentioned styles, there are different multi-view learning algorithms in some specific tasks, which have significant relationships with those three main categories and provide some fine research directions.

Co-regularization algorithms, as a significant branch of multi-view learning, can be divided into a few parts according to the criterion of designing a constraint for regularization in learning a latent subspace or applying labeled or unlabeled data to maintain consistency in a multi-view space. The first series are canonical correlation analysis (CCA)-based algorithms [39], which can exploit the underlying co-relationships within the multi-view space. Specifically, kernel CCA is put forward to extend the original CCA to a kernel form [40], and tensor CCA is a high-order form to handle data with any number of views [41]. Unlike CCA, Fisher's multi-view discriminant analysis, as an extension of linear discriminant analysis (LDA), makes use of labeled data to learn an informative projection matrix to maximize the intra-class distance and minimize the intra-class distance [42]. Based on those works, some graph-based methods refer to the fusion of multiple graphs for multi-view features into a common graph before clustering [43]. They all attempt to construct an embedding projection or affinity matrix to represent the graph or similarity among different samples.

2.2. Multi-View Subspace Clustering

Unlike previous multi-view learning algorithms, multi-view subspace clustering (MSC) first calculates the affinity matrix in each view and then constructs the representation tensor. It can capture the high-order correlations of different views instead of independently constructing them. Inspired by the information correlation in MSC, we extend MSC to a linear dimension reduction technique [44], namely LR-MTR. LR-MTR mainly searches for an informative projection matrix for all the viewed data; subspace representation in a multi-view space is achieved as well.

In this section, the MSC model will be introduced. The algorithm constructs the affinity matrix by reconstructing the coefficients. Through multi-view data, which features columns, we have a D -dimensional vector: $X = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{D \times N}$; $Z =$

$[z_1, z_2, \dots, z_N] \in \mathbb{R}^{N \times N}$ is learned by the MSC model, where each of its columns z_i is a subspace representation of the sample vector x_i . A multi-view subspace representation matrix will be computed through the following equations:

$$\begin{aligned} \min_{Z^{(v)}, E^{(v)}} \sum_{v=1}^V \left(R(Z^{(v)}) + \lambda_v \|E^{(v)}\|_{2,1} \right) \\ \text{s.t. } X^{(v)} = X^{(v)} Z^{(v)} + E^{(v)}, v = 1, 2, \dots, V, \end{aligned} \quad (1)$$

where each view's value of the loss penalty is controlled by the parameter λ_v , $R(\cdot)$, which is the regularizer. $Z^{(v)}$, $X^{(v)}$, and $E^{(v)}$ represent the subspace, the data matrix, and the error matrix in the v^{th} view, respectively. $\|E\|_{2,1} = \sum_{j=1}^n \left(\sum_{i=1}^m ([E]_{i,j})^2 \right)^{\frac{1}{2}}$ is the $\ell_{2,1}$ -norm, which is used in Equation (1) for the loss between different data cases; it also promotes the columns to become zero. We use V as the number of views. The objective is to find the series of affinity matrices representing the similarities of the samples in each view.

2.3. Low-Rank Representation

To find the correlations among multi-view spaces, low-rank regularization can be added to the objective function. As a subspace clustering method, to capture the LRR in the given data set, the following problem is to be solved:

$$Z^* = \arg \min_Z \text{rank}(Z), \text{s.t. } X = XZ \quad (2)$$

As we all know, to introduce low-rank regularization to subspace clustering, we usually apply a nuclear norm of a matrix:

$$Z^* = \arg \min_Z \|Z\|_*, \text{s.t. } X = XZ, \quad (3)$$

in which $\|\cdot\|_*$ stands for the matrix's nuclear norm. The LRR method is effective for solving robust clustering problems.

However, we cannot use it on the linear dimension reduction problem. We use the smallest number of rank-1 tensors to define the rank of a tensor \mathcal{X} that represents this tensor \mathcal{X} as their sum [45]. It seems that the definition of the matrix X 's rank is the smallest number of rank-1 matrices that can generate the matrix X . Given the NP-hard problem of searching for the tensor rank, its definition is not unique. Therefore, some low-rank tensor decomposition approaches to simulate higher-order tensors have been proposed [46].

There is no straightforward method to determine a specific tensor. Thus, given a range of approximate approaches to CANDECOMP/PARAFAC (CP) decomposition, Tucker-decomposition-based methods are expanding [47,48]. Additionally, a tensor rank approximation problem can be formulated as a convex optimization problem to form norm regularizations [49].

3. LR-MTR

In this section, we introduce the motivation of LR-MTR, the specific algorithm, and the optimization. $X^{(1)} \dots X^{(V)}$ represent multi-view features data collection. LR-MTR calculates the projection matrix U by minimizing the difference between the whole cascaded data set and the features in the corresponding space and combines the subspace representations features into a low-rank tensor Z . The projection matrix is used for linear dimension reduction. LR-MTR utilizes the tensor to model the high-order correlations from multimodal data and obtains a series of representation matrices in the multimodal subspace. The overview of LR-MTR is shown in Figure 1.

3.1. The Motivation of the Method

Most of the existing representative linear dimension reduction methods mainly use geometric, sample metric, or class-specific structures to calculate the similarity of samples and learn a projection matrix for feature extraction. For example, the well-known locality preserving projection (LPP) constructs a similarity graph and retains the local relationships among the data, neighbor preserving embedding (NPE) uses the same thought, and linear discriminant analysis (LDA) employs label information to search for a linear combination of features from two or more known categories. These approaches all process the cascaded features without considering the consistency and independence of the multi-view features from one sample. In the real world, one modal feature is not guaranteed to effectively describe a sample, especially for remote sensing data. The PolSAR is a multi-channel radar sensor and can be represented by some typical modalities, such as color and textures in pseudo-images, coefficients in incoherent/coherent target decompositions, and free parameters in scattering matrices. PolSAR scene classification, taking each image as one sample, can make full use of the information from the PolSAR image, including not only the spatial knowledge, but also the redundant features provided by PolSAR multi-channel descriptors.

Thus, it is urgent to simultaneously exploit the multimodal PolSAR features for data processing. In different representation spaces, PolSAR data have multimodal features to create a comprehensive description for the targets, which can be directly constructed into a high-order tensor. It will seek the spatial information from multi-view data and correlate multimodal features. These high-order tensor samples can extend the linear dimension reduction algorithms to tensor forms, such as tensor locality preserving projection (TLPP) or tensor neighbor preserving embedding (TNPE), but this also results in the huge computation of calculating the tensor similarities and constructing tensor graphs. We try to use representation matrices from multi-view subspaces to construct a tensor that can effectively improve the extraction of correlations among multi-views. Compared with these tensor dimension reduction techniques, LR-MTR constructs a representation tensor instead of a feature tensor, effectively reducing the computation complexity in optimization.

Therefore, the robustness of LRR to noise, corrosion, and occlusion can be fully equipped when learning linear dimension reduction in a robust multi-view subspace. Given that LRR can extract robust features matrix Z between data sets, the low-rank representation relation is explored.

3.2. The Main Algorithm

To achieve linear dimension reduction in PolSAR data with sample-specific outliers and corruptions, we assume that the PolSAR data can be approximately reconstructed by the orthogonal projection matrix U . Thus, the following objective function can be designed based on Equation (1) in a multi-view subspace:

$$\begin{aligned}
 & \min_{Z^{(v)}, E^{(v)}} \sum_{v=1}^V \left(\|Z^{(v)}\|_* + \lambda_v \|E^{(v)}\|_{2,1} \right) + \\
 & \lambda \|U^T X - U^T X W\|_{2,1} + \sum_{v=1}^V \|S^{(v)}\|_F^2, \\
 & \text{s.t. } X^{(v)} = X^{(v)} Z^{(v)} + E^{(v)}, v = 1, 2, \dots, V, \\
 & \quad U^T U = I, \\
 & \quad S^{(v)} = W - Z^{(v)}, \\
 & \quad X = [X^{(1)}; X^{(2)}; \dots; X^{(V)}]
 \end{aligned} \tag{4}$$

This objective function was built to calculate the optimal projection and representation matrix. The term $\|U^T X - U^T X W\|_{2,1}$ is used to measure the reconstructive error, and the

term $\left\|S^{(v)}\right\|_F^2$ is used to minimize the subspace representation matrix $Z^{(v)}$ and the overall self-representing matrix W . The rank minimization problem has the NP-hard property. To solve this problem, the rank function $R(\cdot)$ with the convex lower bound $\|\cdot\|_*$ was replaced.

Common approaches to solve the multi-view subspace clustering problem deal with each view feature separately. Considering the correlations of different views, we introduce a tensor nuclear norm to capture hidden related information from different views and learn their subspace representations. The tensor nuclear norm can be written as:

$$\|\mathcal{Z}\|_* = \sum_{m=1}^M \xi_m \|Z_{(m)}\|_*, \quad (5)$$

where ξ_m 's are constant, satisfying $\xi_i > 0$, and $\sum_{m=1}^M \xi_m = 1$. $\mathcal{Z} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ is an M -order tensor. Unfolding the tensor \mathcal{Z} along the m^{th} mode, the matrix $Z_{(m)} \in \mathbb{R}^{I_m \times (I_1 \times \dots \times I_{m-1} \times I_{m+1} \times \dots \times I_M)}$ is formed. Through the nuclear norm $\|\cdot\|_*$, we can restrict the tensor to a low rank. The nuclear norm is an approximation of the rank of matrices because it is a compact convex envelope of the matrix rank. The low-rank tensor representation's objective function can be written as follows:

$$\begin{aligned} & \min_{Z^{(v)}, E^{(v)}} \|\mathcal{Z}\|_* + \lambda \|E\|_{2,1}, \\ & \text{s.t. } E = [E^{(1)}; E^{(2)}; \dots; E^{(V)}], \\ & \mathcal{Z} = \psi(Z^{(1)}, Z^{(2)}, \dots, Z^{(V)}), \\ & X^{(v)} = X^{(v)} Z^{(v)} + E^{(v)}, v = 1, 2, \dots, V, \end{aligned} \quad (6)$$

where the function $\psi(\cdot)$ is used to fuse different matrices $Z^{(v)}$ to a third-order tensor \mathcal{Z} . E is formed by concatenating the column of errors corresponding to each view. The columns were encouraged to be zero by $\ell_{2,1}$ -norm. This corruption is sample-specific, which is an underlying assumption. This means that some instances may have been corrupted. In the integrated approach, $E^{(v)}$ will be constrained by a consistent and common quantity value. To reduce the variation of the error magnitude between multi-views, the data matrices are normalized so that the errors of different views have the same scale. We normalized x_i with $x_i \leftarrow \frac{x_i}{\|x_i\|_2}$. The lowest-rank self-representation matrix comes from the objective function.

Based on Equations (1) and (5), an objective function is designed to calculate the projection matrix U by minimizing the difference between the whole cascaded data set and the features in the corresponding subspace. We combine this method of calculating the projection matrix with the previously proposed method of constructing higher-order tensors to achieve higher-order correlation and linear dimension reduction among multi-view data. To adopt the alternating-direction minimizing strategy and make the equation separable, S Auxiliary G_m is used to replace $Z_{(m)}$ in the tensor nuclear norm. The function is as follows:

$$\begin{aligned} & \min_{U, W} \lambda_1 \|U^T X - U^T X W\|_{2,1} + \lambda_2 \|E\|_{2,1} + \\ & \sum_{m=1}^M \gamma_m \|G_m\|_* + \sum_{v=1}^V \left\|S^{(v)}\right\|_F^2, \\ & \text{s.t. } P_m z = g_m, m = 1, 2, \dots, M, \\ & E = [E^{(1)}; E^{(2)}; \dots; E^{(V)}], \\ & \mathcal{Z} = \psi(Z^{(1)}, Z^{(2)}, \dots, Z^{(V)}), \\ & X^{(v)} = X^{(v)} Z^{(v)} + E^{(v)}, v = 1, 2, \dots, V, \end{aligned}$$

$$\begin{aligned}
U^T U &= I, \\
S^{(v)} &= W - Z^{(v)}, \\
X &= [X^{(1)}; X^{(2)}; \dots; X^{(V)}]
\end{aligned} \tag{7}$$

where γ_m is the strength of the nuclear norm, z is the vectorization of Z , and g_m is the matrix of G_m s. P_m is the alignment matrix corresponding to the mode- k unfolding, which is used to align the corresponding elements between $Z_{(m)}$ and G_m . The feature matrix $X = [X^{(1)}; X^{(2)}; \dots; X^{(V)}]$ is arranged in rows by the matrices $X^{(v)}$ that form different modalities. As G_m is low-rank, the solution Z is ensured to be low-rank in the first constraint. Meanwhile, the features that form multi-views to the representing samples are constrained by the representation matrix in the low-rank tensor norm. The projection matrix U is also calculated to minimize the difference between the whole cascaded data set and the features in the corresponding subspace.

3.3. The Optimal Solution

We introduce our method to the optimization problem in this part. For the two variables in the model, there is no method to attain the optimal solutions synchronously, so a different method is proposed. There are two steps of the iterative algorithms: fix variable W to calculate the optimal U , and then fix U to calculate W .

Step 1: Fix U to calculate W

In Step 1, the AL-ADM method will be used for the optimization problem Equation (7) to minimize the Lagrangian function below:

$$\begin{aligned}
&\mathcal{L}_{\mu > 0}(Z^{(1)}, \dots, Z^{(V)}; E^{(1)}, \dots, E^{(V)}; S^{(1)}, \dots, S^{(V)}; G_1, \dots, G_M; E_h) \\
&= \lambda_1 \|E_h\|_{2,1} + \lambda_2 \|E\|_{2,1} \\
&+ \sum_{m=1}^M (\gamma_m \|G_m\|_* + \mu \Phi(\alpha_m, p_m z - g_m)) \\
&+ \sum_{v=1}^V \left(\|S^{(v)}\|_F^2 + \mu \Phi(Y_v^T, X^{(v)} - X^{(v)} Z^{(v)} - E^{(v)}) \right. \\
&\left. + \mu \Phi(R_v^T, S^{(v)} - W + Z^{(v)}) \right) + \mu \Phi(\beta^T, U^T X - U^T X W - E_h)
\end{aligned} \tag{8}$$

in which $\mu > 0$ is a penalty parameter, $\|\cdot\|_F^2$ stands for the Frobenius norm of a matrix, and Y_v and α_m are the Lagrangian multipliers. The definition is given as $\phi(Y, C) = \langle Y, C \rangle + \frac{1}{2} \|C\|_F^2$, where $\langle \cdot, \cdot \rangle$ represents the matrix's inner product. The above problem is unconstrained, and by fixing the other variables we can minimize in turn with respect to the variables $Z^{(v)}$, $S^{(v)}$, G_m , E , and E_h , and the Lagrangian multipliers Y_v and α_m will be updated. For each iteration, we update each variable through the following equations:

1. $Z^{(v)}$ subproblem:

The subspace representation $Z^{(v)}$ can be updated through the following subproblem:

$$\begin{aligned}
Z^{(v)*} &= \arg \min_{Z^{(v)}} \sum_{m=1}^M \mu \Phi(\alpha_m, p_m z - g_m) \\
&+ \mu \Phi(Y_v^T, X^{(v)} - X^{(v)} Z^{(v)} - E^{(v)}) \\
&+ \mu \Phi(R_v^T, S^{(v)} - W + Z^{(v)}) \\
&= \arg \min_{Z^{(v)}} \sum_{m=1}^M \mu \Phi(\Omega^v(\alpha_m^T), \Omega^k(p_m z - g_m)) \\
&+ \mu \Phi(Y_v^T, X^{(v)} - X^{(v)} Z^{(v)} - E^{(v)}) \\
&+ \mu \Phi(R_v^T, S^{(v)} - W + Z^{(v)})
\end{aligned} \tag{9}$$

where $\Omega^v(\cdot)$ is the operation that selects the elements and reshapes them into a matrix corresponding to the v -th view. Thus, we can find $Z^{(v)}$ through the following:

$$\begin{aligned}
Z^{(v)*} &= \left(\sum_{m=1}^M B_m^{(v)} - \sum_{m=1}^M A_m^{(v)} + X^{(v)T} X^{(v)} - X^{(v)T} E^{(v)} \right. \\
&\left. + X^{(v)T} Y_v - S^{(v)} + W - R_v \right) \left[(M+1)I + X^{(v)T} X^{(v)} \right]^{-1}
\end{aligned} \tag{10}$$

with

$$\begin{cases} A_m^{(v)} = \Omega^v(\alpha_m) \\ B_m^{(v)} = \Omega^v(g_m) \end{cases} \tag{11}$$

An M -way tensor has M unfolding ways. For all these modes in our experiments, $N \times N$ elements corresponding to the v -th views were chosen by the $\Omega^v(\cdot)$. After that, it is reshaped to the $N \times N$ dimensional matrices $A_m^{(v)}$ and $B_m^{(v)}$ corresponding to $Z^{(v)}$.

2. z subproblem:

We update z directly by $z^* \leftarrow Z^{(v)}$. As in each iteration, $E^{(v)}$ and Y_v have no relation to other views, z is updated independently among multiple views. When all $Z^{(v)}$ s are obtained, z can be updated.

3. E subproblem:

The error matrix E can be updated by the following:

$$\begin{aligned}
E^* &= \arg \min_E \lambda_2 \|E\|_{2,1} \\
&+ \sum_{v=1}^V \mu \Phi(Y_v^T, X^{(v)} - X^{(v)} Z^{(v)} - E^{(v)}) \\
&= \arg \min_E \frac{\lambda_2}{\mu} \|E\|_{2,1} + \frac{1}{2} \|E - F\|_F^2 \\
&= \arg \min_E \frac{\lambda_2}{\mu} \|E\|_{2,1} + \frac{1}{2} \left\| E - (X^{(v)} - X^{(v)} Z^{(v)} + Y_v) \right\|_F^2
\end{aligned} \tag{12}$$

where we vertically concatenate the matrices $X^{(v)} - X^{(v)} Z^{(v)} + Y_v$ with the column to form F . The above subproblem can be solved through Lemma 3.2 in [50].

4. Y_v subproblem:

In each iteration, the multiplier Y_v can be updated by the following:

$$Y_v^* = Y_v + (X^{(v)} - X^{(v)} Z^{(v)} - E^{(v)}) \tag{13}$$

5. W subproblem:

The optimizer W can be updated through the following subproblem:

$$\begin{aligned} W^* &= \arg \min_W \mu \Phi(\beta^T, U^T X - U^T XW - E_h) \\ &+ \mu \Phi(R_v^T, S^{(v)} - W + Z^{(v)}) \end{aligned} \quad (14)$$

After taking the derivative of the above formula, the solution of W is obtained by the following:

$$\begin{aligned} W^* &= (X^T U U^T X + X^T U \beta - X^T U E_h + S^{(v)} + Z^{(v)}) \\ &+ R_v \left[I + X^T U U^T X \right]^{-1} \end{aligned} \quad (15)$$

6. R_v subproblem:

In each iteration, the multiplier R_v is updated by the following:

$$R_v^* = R_v + (S^{(v)} - W + Z^{(v)}) \quad (16)$$

7. $S^{(v)}$ subproblem:

The optimizer $S^{(v)}$ can be updated through the following subproblem:

$$\begin{aligned} S^{(v)*} &= \arg \min_S \sum_{v=1}^V \|S^{(v)}\|_F^2 \\ &+ \sum_{v=1}^V \mu \Phi(R_v^T, S^{(v)} - W + Z^{(v)}) \\ &= \arg \min_S \frac{1}{\mu} \sum_{v=1}^V \|S^{(v)}\|_F^2 + \frac{1}{2} \|S^{(v)} - (W - Z^{(v)} - R_v)\|_F^2 \end{aligned} \quad (17)$$

The solution of $S^{(v)}$ is obtained by the following:

$$\begin{aligned} S^{(v)*} &= \mu (W - Z^{(v)} - R_v) [2I + \mu I]^{-1} \\ &= \frac{\mu}{2 + \mu} (W - Z^{(v)} - R_v) \end{aligned} \quad (18)$$

8. E_h subproblem:

The error matrix E_h can be optimized by the following equation:

$$\begin{aligned} E_h^* &= \arg \min_{E_h} \lambda_1 \|E_h\|_{2,1} \\ &+ \sum_{v=1}^V \mu \Phi(\beta^T, U^T X - U^T XW - E_h) \\ &= \arg \min_{E_h} \frac{\lambda_1}{\mu} \|E_h\|_{2,1} \\ &+ \frac{1}{2} \|E_h - (U^T X - U^T XW + \beta)\|_F^2 \end{aligned} \quad (19)$$

Similarly, to update E , the solution of E_h can be obtained when we vertically concatenate the matrices $(U^T X - U^T XW + \beta)$.

9. β subproblem:

The multiplier β can be updated by the following:

$$\beta^* = \beta + (U^T X - U^T XW - E_h) \quad (20)$$

10. G_m^* subproblem:

G_m can be updated by the following:

$$\begin{aligned} G_m^* &= \arg \min_{G_m} \|G_m\|_* + \mu \Phi(\alpha_m, P_m z - g_m) \\ &= \text{prox}_{\beta_m}^{\text{tr}}(\Omega_m(P_m z + \alpha_m)) \end{aligned} \quad (21)$$

where we give a definition $\Omega_m(P_m z + \alpha_m)$, which reshapes the vector $P_m z + \alpha_m$ into a matrix corresponding to the m^{th} mode unfolding. $\text{prox}_{\beta_m}^{\text{tr}}(L) = U \max(S - \beta_m, 0) V^T$ is the threshold of the spectral soft threshold operation, where $\beta_m = \frac{\gamma_m}{\mu}$ denotes the threshold and $L = USV^T$ is the singular value decomposition (SVD) of the matrix L .

11. g_m subproblem:

We replace the corresponding elements to update g_m :

$$g_m^* \leftarrow G_m \quad (22)$$

12. α_m subproblem:

The variable α_m can be updated by the following:

$$\alpha_m^* = \alpha_m + (P_m z - g_m) \quad (23)$$

Step 2: Fix W to calculate U

When W is given, the problem is transformed into the following:

$$\begin{aligned} U^* &= \arg \min_U \|U^T X - U^T X W\|_{2,1} \\ \text{s.t. } &U^T U = I \end{aligned} \quad (24)$$

We can use the proposed algorithm, $L_{2,1}$ norm minimization, to solve this equation. The method includes two steps. First, we can calculate a diagonal matrix H :

$$H_{ii} = \frac{1}{2 \|(U^T X - U^T X W)^i\|_2} \quad (25)$$

in which the i^{th} column of the matrix is represented by $(\cdot)^i$. After that, we can transform Equation (25) into the equivalent trace minimization equation:

$$\begin{aligned} U^* &= \arg \min_U \left(U^T X (I - W) H (I - W)^T X U \right) \\ \text{s.t. } &U^T U = I \end{aligned} \quad (26)$$

The optimal solution of Equation (26) comes from the eigenfunction problem:

$$X(I - W)H(I - W)^T X u = \alpha u \quad (27)$$

in which α is the eigenvalue and u is the corresponding eigenvector. The optimal solution U^* has eigenvectors corresponding to the smaller non-zero eigenvalues.

Algorithm 1 gives the steps of the optimal solution. To facilitate better understanding, the block diagram of the LR-MTR algorithm is shown in Figure 2.

Algorithm 1 Algorithm LR-MTR.

Input: Multimodal feature matrices: $X^{(1)}, X^{(2)}, \dots, X^{(V)}$, the numbers of iterations T and the cluster number K and the parameter λ_1 and λ_2 .

Output: the optimal projection U

initialize: Set $G = I$ and U as the matrix with orthogonal column vectors.

for $t = 1, 2, \dots, T$ **do**

Initialize: $E_h = 0; W = 0; \beta = 0; R_1 = \dots = R_V = 0;$

$Z^{(1)} \dots = Z^{(V)} = 0; S^{(1)} = 0, \dots, S^{(V)} = 0;$

$E^{(1)} = 0, \dots, E^{(V)} = 0; Y_1 = \dots = Y_V = 0;$

$\alpha_1 = \alpha_2 = \dots = \alpha_M = 0; G_1 = 0, \dots, G_M = 0;$

$\mu = 10^{-6}; \rho = 1.1; \epsilon = 10^{-8}; \max_{\mu} = \infty$

while not converged **do**

for each of V views **do**

Update $Z^{(v)}, E^{(v)}, Y_v, W, R_v, S^{(v)}$ according to Equations (10), (12), (13), (15), (16) and (18);

end for

Update z, E_h, β by Equations (19) and (20);

for each of M modes **do**

Update G_m, g_m and α_m according to Equations (21), (22) and (23);

end for

Update μ by $\mu = \min(\rho\mu; \max_{\mu})$;

check the convergence conditions:

$\|X^{(v)} - X^{(v)}Z^{(v)} - E^{(v)}\|_{\infty} < \epsilon$ and

$\|S^{(w)} - W + Z^{(v)}\|_{\infty} < \epsilon$

$\|U^T X - U^T X W - E\|_{\infty} < \epsilon$

$\|P_m z - g_m\|_{\infty} < \epsilon;$

end while

Update H using Equation (25).

Solve the eigenfunction Equation (27) to obtain the optimal U

end for

Project the samples onto the low-dimensional subspace to obtain low dimensional $\tilde{X} = U^T X$ for classification.

3.4. Complexity Analysis

In the optimization procedure, two-level iteration is used to calculate the results. The first loop includes the projection matrix U and affinity matrix W calculations. When calculating U , a sub-loop is also needed to iteratively update every parameter, including $Z^{(v)}, S^{(v)}, G_m, E, E_h$, and Lagrangian multipliers Y_v . First, to calculate U , the ALM optimization involves some sub-optimization problems. However, the main computation complexity $O(nmn)$ comes from multiplying the matrix in updating $Z^{(v)}, W$, and $S^{(v)}$. Second, to calculate U , the computational complexity is from the eigen-decomposition of Equation (27) covering $O(n^3)$. If the LR-MTR converges in T steps, the computational complexity is, at most, $O(Tn^3 + Ttnmn)$, where t is the loop number of the inner iteration of ALM. The outer iteration is fast (three iterations at most), so the computational burden is not too high.

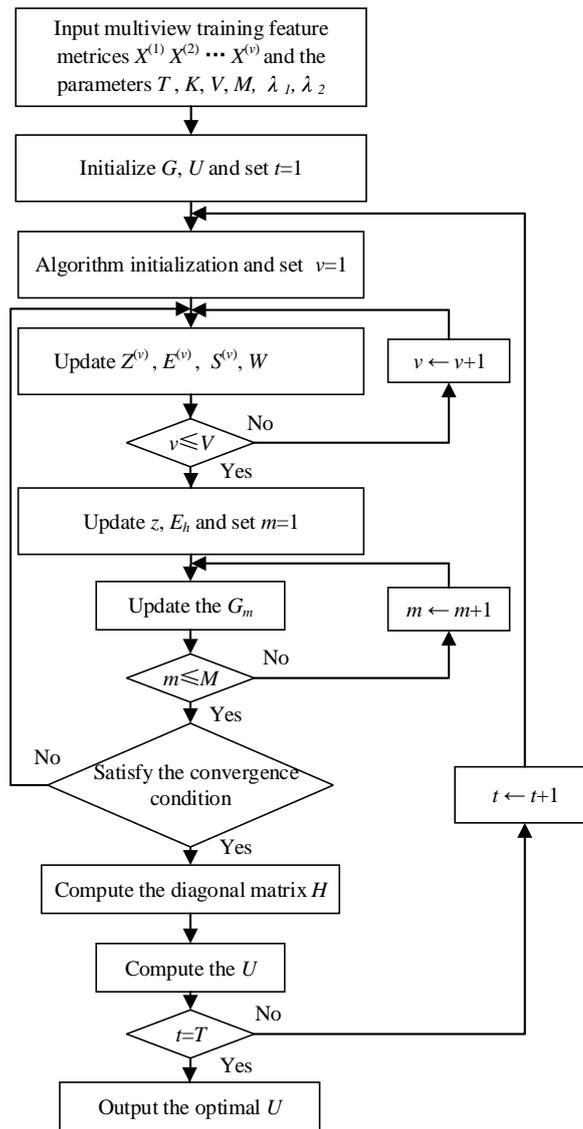


Figure 2. LR-MTR algorithm flow chart.

3.5. Convergence Analysis

The objective function of LR-MTR is solved through alternative iterative optimization. We divide this optimization into two parts, one for updating W and another for U . The suboptimization of W is an AL-ADM process, and U can be updated via an eigenvalue decomposition [51,52].

We find a function $J(W, U)$ from the objective function of Equation (7). Then the theorem can be given.

Theorem 1. *The sequence $J(W_t, U_t)$ monotonically decreases in the iterative scheme.*

Proof. Suppose that, in the t -th iteration of the outer loop, we have the result from Equation (8) as follows:

$$J(W_t, U_t) \triangleq J(W_t, U_t; E, G_m|_{m=1}^M, S^{(v)}|_{v=1}^V) \tag{28}$$

That means W_t updates in the inner loop have the parameters $E, G_m|_{m=1}^M$, and $S^{(v)}|_{v=1}^V$ according to the AL-ADM method, which reduce the value of $J(W_t, U_t)$ in both loops, and an optimized W is obtained. The convergence properties of the inner loop can also be proven as in the literature [53]. Then it has

$$\begin{aligned} J(W_t, U_t) &\stackrel{\Delta}{=} J(W_t, U_t; E, G_m|_{m=1}^M, S^{(v)}|_{v=1}^V) \\ &\geq J(W_{t+1}, U_t; E, G_m|_{m=1}^M, S^{(v)}|_{v=1}^V) \\ &\stackrel{\Delta}{=} J(W_{t+1}, U_t) \end{aligned} \quad (29)$$

We can have $J(W_t, U_t) \geq J(W_{t+1}, U_t)$. After updating W , we should furthermore update U via Equation (24). A diagonal matrix H should be constructed in Equation (25), and matrix U can be obtained by a standard eigenfunction problem in Equation (26). It has $J(W_{t+1}, U_t) \stackrel{\Delta}{=} J(W_{t+1}, U_t, H_{t+1})$. That will further reduce the objective value as follows:

$$\begin{aligned} J(W_{t+1}, U_t, H_{t+1}) &\geq J(W_{t+1}, U_{t+1}, H_{t+1}) \\ &\geq J(W_{t+1}, U_{t+1}) \end{aligned} \quad (30)$$

It can be concluded from the above formulations that $J(W_{t+1}, U_t) \geq J(W_{t+1}, U_{t+1})$ and $J(W_t, U_t) \geq J(W_{t+1}, U_{t+1})$, which means it is monotonically decreasing. As the equation has a positive lower bound, and due to the monotonic decrease in *Theorem 1*, it is simple to know that the sequence $J(W_t, U_t)$ will converge to a local optimal solution. \square

4. Experiments

To test the effectiveness of the LR-MTR algorithm, we run several experiments using the proposed method, which we compare with the previous algorithms. Two PolSAR data sets are used to demonstrate the LR-MTR algorithm. SVM and KNN classifiers are used for the final classification. In our experiment, terrain classification is performed by labeling each tile in a large PolSAR image with some predefined terrain category labels. In high-resolution and large-scale PolSAR image scenes, patch-level classification is more effective than pixel-level classification, as the patches contain richer terrain details which cannot be achieved by a single pixel.

4.1. Data Sets

The LR-MTR method was evaluated on two data sets. One data set covers the region of Tokyo, Japan, in January 2015. The geographical location of the Tokyo data set and the scene categories are presented in Figure 3. The Tokyo data set comes from ALOS-2 fully polarimetric SAR data, whose spatial resolution is six meters. The size of the PolSAR scene in the experiments is $14,186 \times 16,514$.

For the typical land covers, water and smooth surfaces correspond to surface scattering, rough covers to diffuse scattering, thick tree trunks and buildings to dihedral scattering, and areas of foliage to volume scattering. The patches covering all those typical land covers or jointly constructed by them are selected as the terrain categories. Then, five main terrain categories are given in the Tokyo data set: building, water, woodland, coast, and farmland. Moreover, a category which is not associated with any of the given categories is counted as the last uncertain category. The Tokyo data set is labeled with the reference of categories above. After being labeled, these data need to be preprocessed to construct the PolSAR scene data set. First, geographic calibration of the raw data is performed in the PolSARpro software. Second, Lee filtering is performed on the data. Finally, we cut the large image into small scene patches (samples). There are 256×256 pixels in each sample. Then, the cut small samples are labeled. We have access to 2000 samples in the Tokyo data set. There are 855, 697, 205, 187, and 56 scenes for the building, water, woodland, coast, and farmland categories, respectively.

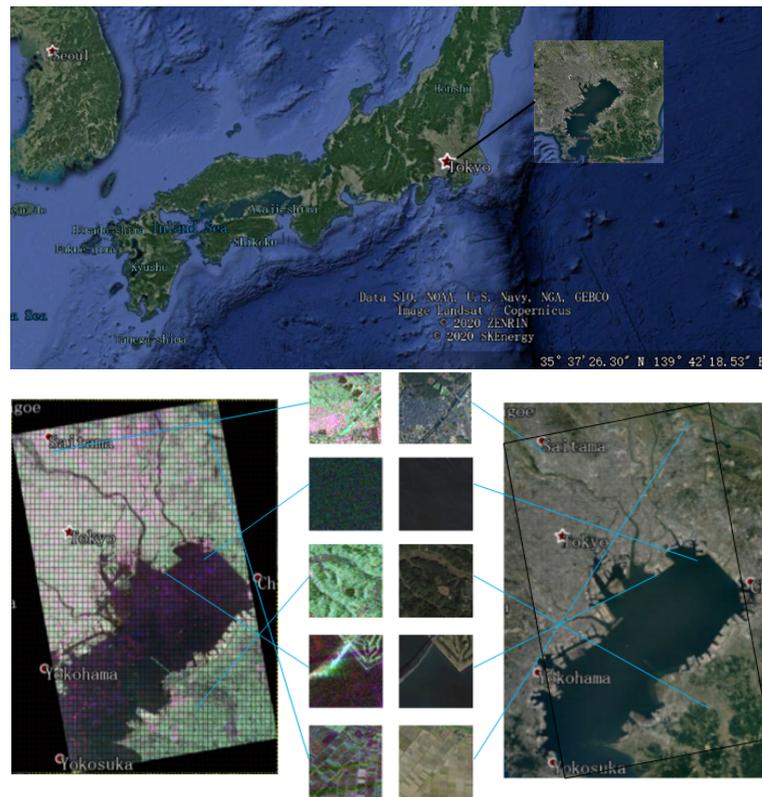


Figure 3. Geographical location of the Tokyo data set and its scene categories.

To extract the multimodal features, we apply the parameters from polarimetric decomposition. Further, to extract features hidden in the polarimetric phases, we use three polarimetric parameter groups: Pauli decomposition, H/alpha decomposition, and Freeman decomposition. Pauli decomposition is a typical coherency decomposition. The scattering S matrix was expressed as the complex sum of the Pauli matrices.

Figure 4 shows four images from different categories. The first column shows the Pauli image of each category, where red comes from $|S_{HH} - S_{VV}|$, green from $|S_{HV} + S_{VH}|$, and blue from $|S_{HH} + S_{VV}|$. The second column is the Cloude decomposition pseudo-color image of each category, where red represents T_{22} , green T_{33} , and blue T_{11} , as described in Figure 6.10 in [1]. The third column is the Freeman decomposition pseudo-color image of each category, in which red is Odd, green is Dbl, and blue is Vol. The fourth column's pictures present the category in Google Earth.

The second data set covers the region of Shanghai, China, in March 2015, as seen in Figure 5. The geographical location of the Shanghai data set and the scene categories are presented in Figure 3. The Shanghai data set also comes from ALOS-2. The resolution is the same as that of the Tokyo data set. The size of the labeled PolSAR scene is $8300 \times 11,960$, and the data set has five main categories: urban areas, suburban areas, farmland, water, and coast. Each sample has 256×256 pixels, and there are 1600 samples. There are 262, 346, 378, 417, and 197 scenes for the urban areas, suburban areas, farmland, water, and coast categories, respectively. In the Shanghai data set, we also use polarimetric decomposition to extract more multimodal features, and these four images also come from different categories in the Shanghai data set, as shown in Figure 6, similar to the Tokyo data set.

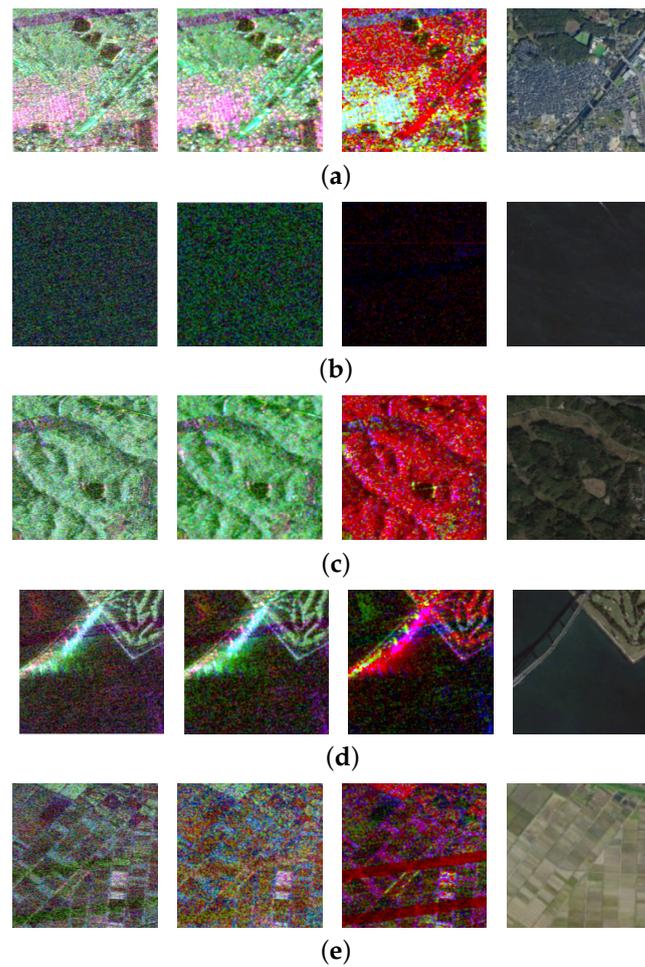


Figure 4. Visualization examples from the Tokyo data set. (a) Building. (b) Water. (c) Woodland. (d) Coast. (e) Farmland.

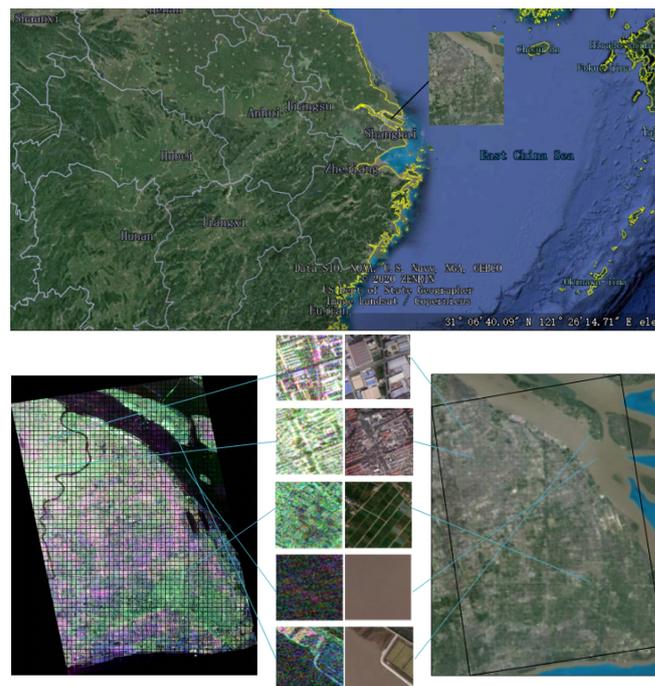


Figure 5. Geographical location of the Shanghai data set and its scene categories.

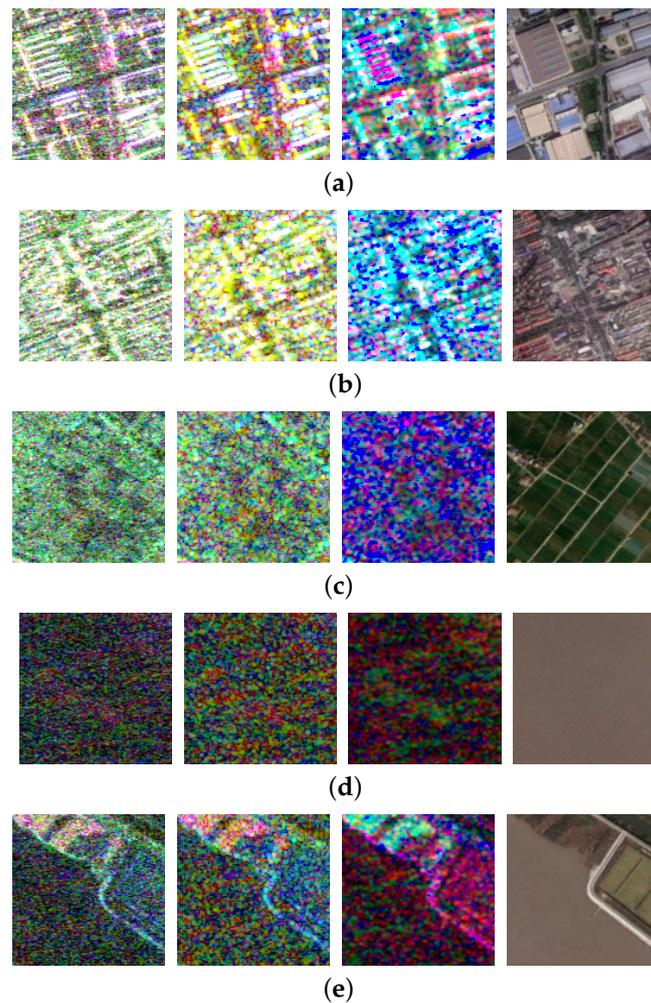


Figure 6. Visualization examples from the Shanghai data set. (a) Urban areas. (b) Suburban areas. (c) Farmland. (d) Water. (e) Coast.

4.2. Feature Descriptors

To evaluate the classification performance, we use two feature descriptors for comparison. Different features were extracted from the following algorithms: local binary pattern (LBP) and gray-level co-occurrence matrix (GLCM). The dimensions of GLCM and LBP are eight and fifteen, respectively. In a 3×3 window, the other eight adjacent pixels are compared with the center one on the gray-scale value level, and we use the value from the center pixel as the threshold; this is where the original LBP operator comes from. If the surrounding pixels have greater values than the threshold, the position where this pixel comes from is set as 1; if not, it is set as 0. By doing so, eight pixels in the 3×3 window can be calculated to generate an 8-bit binary number. The center pixel's LBP value can be obtained, which can show the texture information of the region of interest.

4.3. Experiment Setup

To evaluate the LR-MTR algorithm, we select some different SOTA methods to compare it with in the experiments. There are linear dimension reduction algorithms (multiple dimensional scaling (MDS), independent component analysis (ICA), and principal component analysis (PCA)), linearization of manifold learning methods (such as locality preserving projection (LPP), neighbor preserving embedding (NPE), and locally linear embedding (LLE)), and nonlinear dimension reduction methods (such as kernel principal component analysis (KPCA) and stochastic neighbor embedding (SNE)). We set 15 as the number of neighbors in graph embedding algorithms. To ensure the fairness of the

experiment, the features of these algorithms are reduced to a fixed dimension. Before the experiments, we need to ascertain the influence of the parameters in the experiments. The feature dimensions of the data sets also influence the algorithm performance. During data processing, training samples were selected from the whole group of samples randomly. Furthermore, in learning projection matrices, the number of training samples is important and also needs to be considered. We selected L images from each individual for the training process, and the other images in the data set were used as the test set. We set different L by the scale of each individual/object in the two data sets, i.e., L = 10, 20, 30, 40, 50, 60, and 70 for the Tokyo and Shanghai databases. In both data sets, for the LR-MTR algorithm, the M parameter was set as the same value, i.e., $\gamma_1 = \gamma_2 = \dots = \gamma_M = \gamma$, and the parameter γ was tuned. Each task was conducted five times, and then the mean accuracy was recorded. For each comparative experiment, we tuned the parameters to their best performance. Overall accuracy (OA), average accuracy (AA), and Kappa coefficients are used to show the performance.

4.4. Experimental Result

As discussed in the last subsection, we used seven dimension reduction methods to process 24 and 45 high-dimensional features for comparison. They are composed of three modes of GLCM and LBP features. The high-dimensional features were reduced to the same dimension. The final results are recorded in two PolSAR data sets that have different typical terrain categories. The classification results and maps are shown in Tables 1–4, Figures 7 and 8. The confusion matrix in Figure 7 shows that our method has good classification ability on different categories of both data sets. The classification map in Figure 8 is randomly selected from one of the repeated experiments. Fifty samples were randomly selected to train the projection matrix U for the linear dimension reduction algorithm, and 50 samples were used to train the classifiers (KNN and SVM). The parameter k , which represents the nearest neighbor number, is set to 21 in the KNN classifier. The SVM classifier uses cross-validation to seek the optimal model parameters. In the following experimental analysis, a KNN classification map was used to show the visualization results.

Table 1. GLCM Tokyo.

Class	PCA		LPP		NPE		ICA		KPCA		LLE		SNE		MDS		LR-MTR	
	KNN	SVM																
1	94.97	97.66	87.49	92.98	49.24	60.82	25.50	39.65	94.74	98.25	90.18	79.30	96.02	83.63	84.91	58.48	97.78	95.79
2	51.79	64.29	82.14	96.43	62.50	46.43	83.93	76.79	53.57	50.00	50.00	64.29	53.57	87.50	62.50	48.21	73.93	67.14
3	100.00	100.00	89.81	91.10	51.08	16.93	91.39	93.40	99.86	100.00	99.71	81.21	100.00	27.40	99.86	59.68	97.13	98.85
4	42.93	40.98	77.07	60.49	58.05	23.90	86.34	84.39	41.46	35.61	30.24	62.93	47.32	0.00	29.27	76.10	56.10	79.81
5	78.07	72.19	97.86	75.94	64.71	67.38	92.51	67.91	88.24	83.96	80.75	71.12	86.63	77.01	43.32	52.94	82.89	84.92
OA	88.60	89.35	88.05	87.50	52.60	41.95	62.60	66.65	89.30	89.75	85.35	77.10	90.35	54.95	80.00	59.90	92.28	93.40
AA	73.55	75.02	86.88	83.39	57.11	43.09	75.93	72.43	75.57	73.56	70.18	71.77	76.71	55.11	63.97	59.08	82.12	83.36
Kappa	0.8319	0.8420	0.8305	0.8204	0.3879	0.2666	0.5302	0.5670	0.8422	0.8476	0.7855	0.6790	0.8572	0.4158	0.7109	0.4750	0.9986	0.9204

Table 2. GLCM Shanghai.

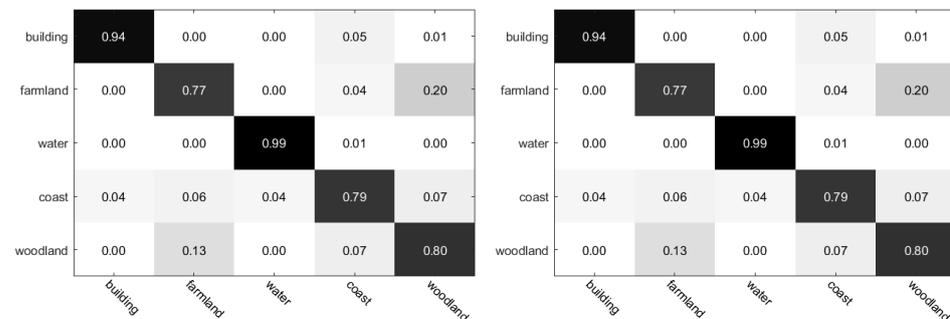
Class	PCA		LPP		NPE		ICA		KPCA		LLE		SNE		MDS		LR-MTR	
	KNN	SVM																
1	83.21	69.85	79.77	66.79	33.97	32.82	74.81	83.59	79.77	54.96	79.01	55.34	82.06	72.90	62.21	46.56	66.85	70.77
2	67.34	79.48	82.37	81.21	49.42	60.40	71.10	78.90	71.10	71.68	58.09	78.03	65.32	78.32	62.43	69.94	77.31	73.72
3	89.68	84.13	84.13	86.24	53.97	41.53	81.48	87.83	83.33	83.86	89.68	75.66	87.57	79.63	85.71	67.72	81.38	82.96
4	98.32	81.29	83.21	73.86	76.74	43.41	77.94	76.02	98.08	63.79	98.08	70.26	96.88	79.38	98.32	54.92	84.21	76.04
5	45.69	64.97	56.85	77.66	32.99	57.36	64.97	78.68	43.65	76.14	27.41	54.31	41.62	54.82	23.86	50.25	65.60	76.80
OA	80.07	77.69	79.44	77.69	57.67	46.63	56.88	81.00	80.25	70.31	75.18	68.81	78.63	75.13	72.64	59.25	82.88	81.29
AA	76.85	75.94	77.27	77.16	49.42	47.11	74.06	81.00	75.19	70.09	70.46	66.72	74.69	73.01	66.51	57.88	75.07	76.06
Kappa	0.7523	0.7185	0.7396	0.7206	0.4016	0.3406	0.6857	0.7609	0.7331	0.6306	0.6874	0.6094	0.7269	0.6862	0.6463	0.4951	0.9981	0.6583

Table 3. LBP Tokyo.

Class	PCA		LPP		NPE		ICA		KPCA		LLE		SNE		MDS		LR-MTR	
	KNN	SVM	KNN	SVM	KNN	SVM	KNN	SVM	KNN	SVM	KNN	SVM	KNN	SVM	KNN	SVM	KNN	SVM
1	67.60	88.07	82.92	85.96	13.92	23.51	69.24	60.00	65.61	87.84	64.33	77.08	68.89	83.51	63.04	72.05	71.93	82.34
2	78.57	80.36	89.29	94.64	37.50	28.57	48.21	62.50	73.21	78.57	69.64	71.43	66.07	76.79	64.29	78.57	83.93	87.50
3	97.27	60.69	97.85	88.52	26.97	14.49	53.66	22.24	95.70	60.55	94.98	77.47	94.55	88.09	98.13	93.97	85.65	70.45
4	46.83	12.20	52.20	40.49	12.68	22.93	39.02	65.85	49.76	9.76	40.49	13.17	43.41	22.93	51.71	15.12	64.39	74.63
5	87.70	63.10	94.12	83.96	28.34	6.95	60.43	74.87	65.78	44.39	51.87	40.64	69.52	53.48	63.10	37.43	86.63	92.51
OA	78.00	68.20	86.20	82.25	30.45	18.90	59.30	48.90	74.70	66.00	71.55	67.10	75.20	75.90	74.15	70.80	77.65	78.50
AA	75.60	60.88	83.27	78.72	23.88	19.29	54.11	57.09	70.01	56.22	64.26	55.96	68.49	64.96	68.05	59.43	78.51	81.49
Kappa	0.6926	0.5641	0.8017	0.7469	0.0243	-0.0610	0.4422	0.3374	0.6493	0.5350	0.6063	0.5469	0.6541	0.6586	0.6411	0.5922	0.9985	0.9986

Table 4. LBP Shanghai.

Class	PCA		LPP		NPE		ICA		KPCA		LLE		SNE		MDS		LR-MTR	
	KNN	SVM																
1	60.31	46.95	71.37	83.21	85.50	15.65	66.79	73.28	59.54	54.58	45.04	41.60	55.73	66.41	53.44	57.63	69.47	80.15
2	80.06	70.23	80.92	75.14	65.32	24.28	75.14	76.01	81.79	69.94	71.97	66.76	81.79	68.21	78.03	72.25	77.75	79.48
3	83.60	77.51	81.22	88.36	30.69	29.89	79.37	87.57	65.34	79.63	66.14	79.37	60.58	75.93	50.26	69.84	87.57	87.83
4	98.08	77.22	98.08	86.81	88.73	17.75	87.29	82.25	90.89	77.46	99.28	76.50	99.52	79.86	84.65	79.38	87.05	74.82
5	67.01	71.57	62.94	75.13	22.34	59.39	61.42	76.65	60.91	80.71	54.82	77.16	44.67	76.14	47.72	67.01	78.68	91.37
OA	80.75	70.13	81.69	82.63	61.25	26.81	76.25	80.00	74.06	73.00	71.19	69.44	72.56	73.75	65.44	70.50	82.25	82.81
AA	77.81	68.70	78.91	81.73	58.51	29.39	74.00	79.15	71.70	72.46	67.45	68.28	68.46	73.31	62.82	69.22	80.10	82.73
Kappa	0.7550	0.6255	0.7667	0.7810	0.5077	0.0988	0.6976	0.7479	0.6711	0.6613	0.6332	0.6169	0.6485	0.6699	0.5633	0.6279	0.9976	0.9975



(a) Tokyo data results.

(b) Shanghai data results.

Figure 7. Confusion matrix for the Tokyo data set (a) and the Shanghai data set (b).

From the classification results and classification map, it is not difficult to see that the LR-MTR algorithm has better classification performance in terms of all indices than other SOTA algorithms. For OA, the LR-MTR algorithm is approximately 2%, 2%, and 1% better than the second best algorithms. This reveals that the LR-MTR algorithm is an effective algorithm for PolSAR scene classification. In Tables 1–4, we assign different terrain categories with the corresponding numbers. In Tables 1 and 2, we use the numbers 1–5 to represent different categories: building, farmland, water, coast, and woodland, respectively. In Tables 3 and 4, we use the numbers 1–5 to represent urban areas, suburban areas, water, farmland, and coast, respectively.

4.5. Experiment Analysis

(1) Dimension: Fifty training samples were randomly selected from each category and used to learn the projection matrix U . As the dimension changes from 1, 3, 5, ..., 15, the KNN results are also affected, as shown in Figure 8. This also shows that the LR-MTR method has a better classification accuracy than other dimension reduction algorithms when the number of dimensions is greater than seven. When it is greater than three, all algorithms except the NPE algorithm have better classification performance. Once the algorithm performance remains stable, all the algorithms will achieve better results.

(2) Parameter Sensitivity: Figure 9 shows the parameter tuning examples. In the LR-MTR algorithm, the balance parameter γ is the most important parameter in the model. From Figure 9, it is not difficult to find that the parameter γ has some effectiveness on the two data sets. The value of γ is bigger in the Shanghai data set than in the Tokyo data set

for promising results. As shown in the figure, the results are different with the increase in parameters. It is not difficult to find that the parameters have certain influence on the two different data sets, and there will be an optimal parameter, but the value of the optimal parameter is different in different data sets.

(3) Convergence: To demonstrate the convergence of the LR-MTR algorithm, Figure 10 shows the objective values versus the iteration steps on the two representative data sets. The convergent properties of LR-MTR on the Tokyo data set and the Shanghai data set are shown in Figure 10a,b where the errors 1–4 correspond to the computed results of the four convergence conditions in Algorithm 1, respectively. This shows that the four parameters of convergence in our method reduce as the number of iterations increases. Generally, the LR-MTR method is able to converge within 40 to 60 iterations. In the experiment, the outer loop iteration is set as two or three.

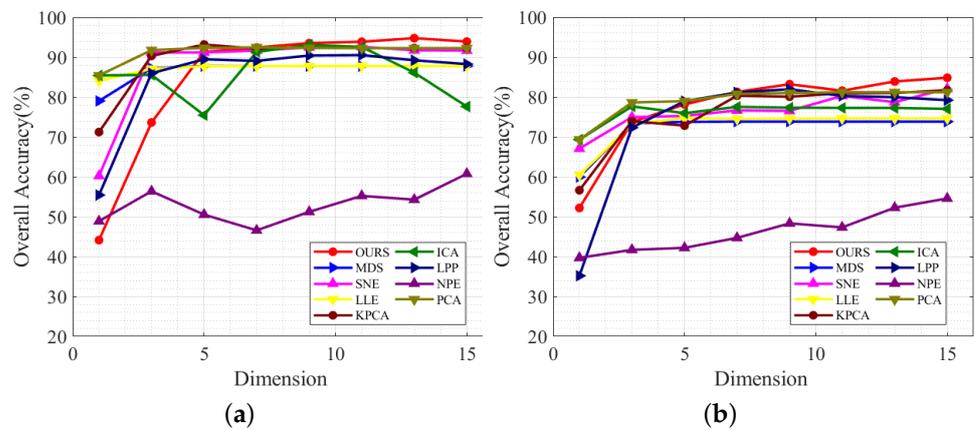


Figure 8. OA results of KNN with the dimension changes. (a) Tokyo data results. (b) Shanghai data results.

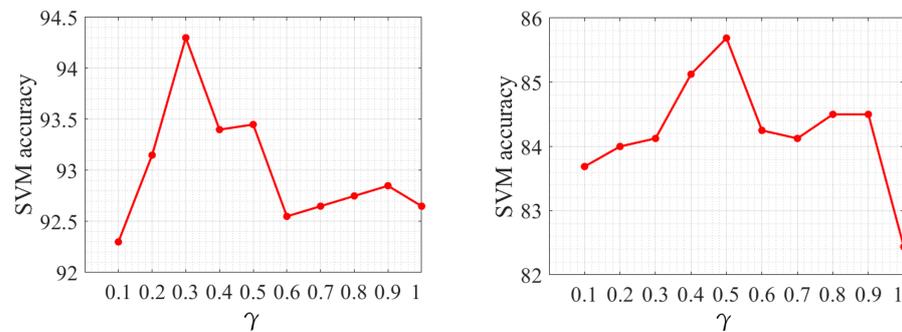


Figure 9. Adjusted parameters according to the accuracy of the Tokyo data set (left) and the Shanghai data set (right). As we set $\gamma = \gamma_2 = \dots = \gamma_M = \gamma$ in our experiments, γ is the only parameter that needs to be tuned.

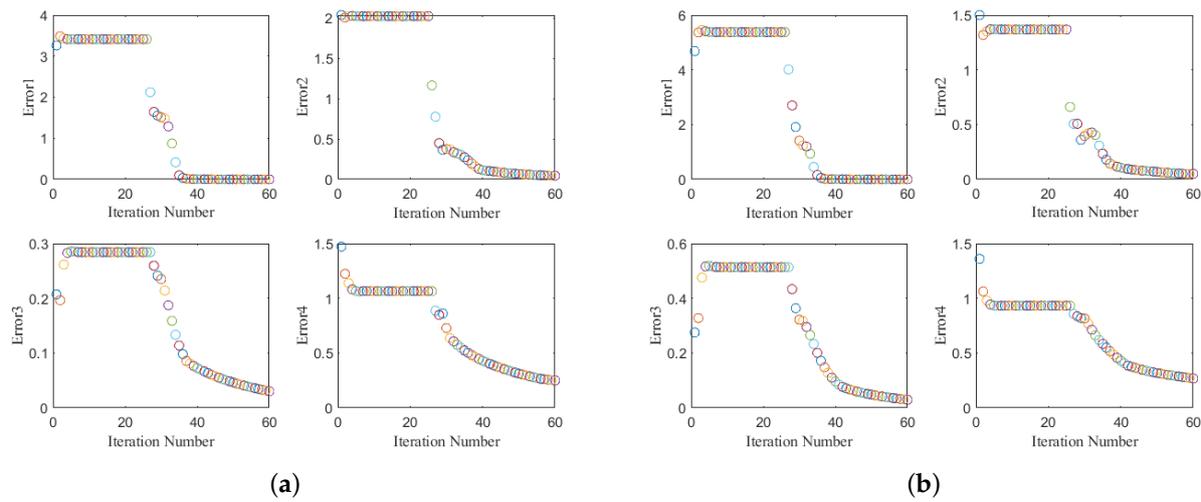


Figure 10. The convergence property of the proposed algorithm. (a) Tokyo data results. (b) Shanghai data results.

(4) **Running Time:** LR-MTR and the contrasting algorithms with the same number of samples (50 samples per category) were run, and the time was recorded, as shown in Table 5. Compared with other algorithms, our algorithm cannot achieve a good performance time, as it takes more time to improve the feature representative. This is because the LR-MTR algorithm is based on an optimization algorithm with two loops, which takes more time during the convergence process. The running time can be further reduced accordingly to improve the optimal solutions.

Table 5. Running time.

Algorithm	PCA	LPP	NPE	FastICA	KernelPCA	LLE	SNE	MDS	LR-MTR
time(s)	0.64	1.23	0.5	1.49	0.62	1.29	2.72	0.42	6.32

(5) **Generalization:** In order to test the generalization of the proposed method, we used the learned projected matrix on another data set for classification. For a fair comparison, those DR techniques, including PCA, LPP, NPE, ICA, KPCA, LEE, SNE, and MDS, all used the same 50 samples from each category for learning the projection matrices. The OA and specific accuracy are given. Table 6 shows the results of projection matrices learning from the Tokyo data set and being used on the Shanghai data set. Meanwhile, Table 7 shows the results of projection matrices learning from the Shanghai data set and being used on the Tokyo data set. In the test data sets, we chose 50 samples from one data set as training samples and used the other samples for testing. From these experiments, our algorithm is able to achieve the best classification OA 79.81 (Tokyo learned U used on Shanghai data set) and 87.35 (Shanghai learned U used on Tokyo data set). This is due to the ability to capture the relevance between features, which proves that LR-MTR also has better generalization than other DR techniques.

Table 6. Results of Tokyo Learned U for Shanghai data set.

	Buildings	Farmland	Water	Woodland	Coast	OA
PCA	84.35	56.94	82.54	82.01	18.27	69.25
LPP	41.6	77.75	67.99	83.45	69.04	69.94
NPE	75.57	66.18	73.54	75.78	38.07	68.5
ICA	87.02	45.09	80.42	64.03	12.18	61.19
KPCA	86.64	44.22	85.98	93.05	21.32	70.94
LLE	80.92	58.96	82.28	95.68	12.18	71.88
SNE	83.59	62.72	89.68	93.76	40.1	77.81
MDS	81.68	43.06	85.98	93.05	19.29	69.63
LT-MTR	85.11	62.71	91.79	89.44	59.39	79.81

Table 7. Results of Shanghai Learned U for Tokyo data set.

	Urban	Suburban	Farmland	Water	Coast	OA
PCA	92.98	60.71	98.28	18.05	79.14	84.95
LPP	79.42	64.29	81.21	24.39	78.61	73.9
NPE	36.61	26.79	59.25	29.76	59.89	45.7
ICA	92.98	60.71	98.28	18.05	79.14	84.95
KPCA	37.43	41.07	96.27	42.44	50.8	59.8
LLE	52.16	57.14	97.27	40.98	83.42	69.8
SNE	95.56	55.36	92.68	22.44	81.28	84.6
MDS	96.84	44.64	52.51	44.88	68.45	71.95
LT-MTR	97.31	53.57	94.98	27.32	89.31	87.35

5. Conclusions

The LT-MTR method is proposed to coordinate PolSAR data in multi-view representation and solve the PolSAR scene classification problem. This method has shown a way to keep the spatial and polarimetric information simultaneously by exploiting the complementary information of multi-view data jointly. As for the high-order matrix, we use a type of high-order matrix tensor to find the relation information of underlying multi-view data. This is also constrained by the low-rank norm to model the cross-information from multiple spaces. To solve the out-of-sample problem in the large-scale data set, the projection matrix is calculated. The experimental results show that LR-MTR algorithm has advantages on the two ALOS-2 PolSAR data sets. The low-rank representation has strong robustness, which ensures that the proposed method can be used to acquire high-quality classification results. Compared with the current popular vector dimensionality reduction methods (PCA, LLE, SNE, MDS, etc.), LR-MTR takes into account the association of data from different perspectives, which can better capture the features that are beneficial to scene classification and achieve better classification results. In our future research, we will focus on the feature extraction techniques for data, apply LR-MTR to different tasks, and optimize the time complexity and computational complexity.

Author Contributions: Conceptualization, B.R. and M.C.; Funding acquisition, B.H. and L.J.; Project administration, B.H.; Software, M.C. and S.M.; Writing—original draft, B.R., M.C. and S.M.; Writing—review and editing, D.H. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China: 62001355, 61671350, 61771379, 61836009; Fundamental Research Funds for the Central Universities: No. XJS211901; the Foundation for Innovative Research Groups of the National Natural Science Foundation of China: 61621005; the Key Research and Development Program in Shaanxi Province of China: 2022GY-067, 2019ZDLGY03-05; China Postdoctoral Science Foundation: No. 2018M633468.

Data Availability Statement: Not applicable.

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

References

1. Lee, J.S.; Pottier, E. *Polarimetric Radar Imaging: From Basics to Applications*; CRC Press: Boca Raton, FL, USA, 2009.
2. Uhlmann, S.; Kiranyaz, S. Integrating color features in polarimetric SAR image classification. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 2197–2216. [[CrossRef](#)]
3. Ren, B.; Hou, B.; Wen, Z.; Xie, W.; Jiao, L. PolSAR image classification via multimodal sparse representation-based feature fusion. *Int. J. Remote Sens.* **2018**, *39*, 7861–7880. [[CrossRef](#)]
4. Akbarizadeh, G.; Rahmani, M. Efficient combination of texture and color features in a new spectral clustering method for PolSAR image segmentation. *Natl. Acad. Sci. Lett.* **2017**, *40*, 117–120. [[CrossRef](#)]
5. Ren, B.; Hou, B.; Zhao, J.; Jiao, L. Unsupervised classification of polarimetric SAR image via improved manifold regularized low-rank representation with multiple features. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *10*, 580–595. [[CrossRef](#)]
6. Wu, W.; Li, H.; Li, X.; Guo, H.; Zhang, L. PolSAR image semantic segmentation based on deep transfer learning-Realizing smooth classification with small training sets. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 977–981. [[CrossRef](#)]
7. Liu, F.; Jiao, L.; Tang, X. Task-oriented GAN for PolSAR image classification and clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2707–2719. [[CrossRef](#)]
8. Cheng, G.; Xie, X.; Han, J.; Guo, L.; Xia, G.S. Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 3735–3756. [[CrossRef](#)]
9. Wu, W.; Li, H.; Zhang, L.; Li, X.; Guo, H. High-resolution PolSAR scene classification with pretrained deep convnets and manifold polarimetric parameters. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6159–6168. [[CrossRef](#)]
10. Cloude, S.R.; Pottier, E. An entropy based classification scheme for land applications of polarimetric SAR. *IEEE Trans. Geosci. Remote Sens.* **1997**, *35*, 68–78. [[CrossRef](#)]
11. Yang, J.; Peng, Y.N.; Yamaguchi, Y.; Yamada, H. On Huynen's decomposition of a Kennaugh matrix. *IEEE Geosci. Remote Sens. Lett.* **2006**, *3*, 369–372. [[CrossRef](#)]
12. Cloude, S.R.; Pottier, E. A review of target decomposition theorems in radar polarimetry. *IEEE Trans. Geosci. Remote Sens.* **1996**, *34*, 498–518. [[CrossRef](#)]
13. Freeman, A.; Durden, S.L. A three-component scattering model for polarimetric SAR data. *IEEE Trans. Geosci. Remote Sens.* **1998**, *36*, 963–973. [[CrossRef](#)]
14. Van Zyl, J.J. Unsupervised classification of scattering behavior using radar polarimetry data. *IEEE Trans. Geosci. Remote Sens.* **1989**, *27*, 36–45. [[CrossRef](#)]
15. Yamaguchi, Y.; Moriyama, T.; Ishido, M.; Yamada, H. Four-component scattering model for polarimetric SAR image decomposition. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 1699–1706. [[CrossRef](#)]
16. Touzi, R.; Charbonneau, F. Characterization of target symmetric scattering using polarimetric SARs. *IEEE Trans. Geosci. Remote Sens.* **2002**, *40*, 2507–2516. [[CrossRef](#)]
17. Dell'Acqua, F.; Gamba, P. Texture-based characterization of urban environments on satellite SAR images. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 153–159. [[CrossRef](#)]
18. Kandaswamy, U.; Adjeroh, D.A.; Lee, M.C. Efficient texture analysis of SAR imagery. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 2075–2083. [[CrossRef](#)]
19. Clausi, D.A. Comparison and fusion of co-occurrence, Gabor and MRF texture features for classification of SAR sea-ice imagery. *Atmos. Ocean* **2001**, *39*, 183–194. [[CrossRef](#)]
20. Li, H.C.; Celik, T.; Longbotham, N.; Emery, W.J. Gabor feature based unsupervised change detection of multitemporal SAR images based on two-level clustering. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2458–2462.
21. Kaplan, L.M. Improved SAR target detection via extended fractal features. *IEEE Trans. Aerosp. Electron. Syst.* **2001**, *37*, 436–451. [[CrossRef](#)]
22. Solberg, A.S.; Jain, A.K. Texture fusion and feature selection applied to SAR imagery. *IEEE Trans. Geosci. Remote Sens.* **1997**, *35*, 475–479. [[CrossRef](#)]
23. Gao, H.; Nie, F.; Li, X.; Huang, H. Multi-view subspace clustering. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4238–4246.
24. Brbić, M.; Kopriva, I. Multi-view low-rank sparse subspace clustering. *Pattern Recognit.* **2018**, *73*, 247–258. [[CrossRef](#)]
25. Zhang, C.; Fu, H.; Liu, S.; Liu, G.; Cao, X. Low-rank tensor constrained multiview subspace clustering. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1582–1590.
26. Ren, B.; Hou, B.; Chanussot, J.; Jiao, L. PolSAR Feature Extraction Via Tensor Embedding Framework for Land Cover Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 2337–2351. [[CrossRef](#)]
27. Ren, B.; Hou, B.; Chanussot, J.; Jiao, L. Modified Tensor Distance-Based Multiview Spectral Embedding for PolSAR Land Cover Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 2095–2099. [[CrossRef](#)]
28. Hong, D.; Gao, L.; Yokoya, N.; Yao, J.; Chanussot, J.; Qian, D.; Zhang, B. More Diverse Means Better: Multimodal Deep Learning Meets Remote-Sensing Imagery Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 4340–4354. [[CrossRef](#)]
29. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Antonio, P.; Chanussot, J. Graph Convolutional Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *7*, 5966–5978. [[CrossRef](#)]

30. Hong, D.; Yokoya, N.; Chanussot, J.; Xu, J.; Zhu, X.X. Joint and Progressive Subspace Analysis (JPSA) with Spatial-Spectral Manifold Alignment for Semi-Supervised Hyperspectral Dimensionality Reduction. *IEEE Trans. Cybern.* **2020**, *51*, 3602–3615. [[CrossRef](#)]
31. Hong, D.; Yokoya, N.; Chanussot, J.; Zhu, X. An augmented linear mixing model to address spectral variability for hyperspectral unmixing. *IEEE Trans. Image Process.* **2019**, *28*, 1923–1938. [[CrossRef](#)]
32. Hong, D.; Yokoya, N.; Ge, N.; Chanussot, J.; Zhu, X. Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification. *ISPRS J. Photogramm. Remote Sens.* **2019**, *147*, 193–205. [[CrossRef](#)]
33. Zhao, J.; Xie, X.; Xu, X.; Sun, S. Multi-view learning overview: Recent progress and new challenges. *Inf. Fusion* **2017**, *38*, 43–54. [[CrossRef](#)]
34. Gao, L.; Han, Z.; Hong, D.; Zhang, B.; Chanussot, J. CyCU-Net: Cycle-Consistency Unmixing Network by Learning Cascaded Autoencoders. *IEEE Trans. Geosci. Remote Sens.* **2020**, *60*, 1–14. [[CrossRef](#)]
35. Rai, A.K.; Daumé, H. Co-Regularized Multi-View Spectral Clustering. In *NIPS'11: Proceedings of the 24th International Conference on Neural Information Processing Systems, Granada, Spain, 12–15 December 2011*; Curran Associates Inc.: Red Hook, NY, USA, 2011.
36. Evans, X.Z.; Dugelay, J. A subspace co-training framework for multi-view clustering. *Pattern Recognit. Lett.* **2014**, *41*, 73–82.
37. Cao, X.; Zhang, C.; Fu, H.; Liu, S.; Zhang, H. Diversity-induced multi-view subspace clustering. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015*; pp. 586–594.
38. Chao, G.; Sun, S. Consensus and complementarity based maximum entropy discrimination for multi-view classification. *Inf. Sci.* **2016**, *367*, 296–310. [[CrossRef](#)]
39. Hardoon, D.R.; Szedmak, S.; Shawe-Taylor, J. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.* **2004**, *16*, 2639–2664. [[CrossRef](#)]
40. Bach, F.R.; Jordan, M.I. Kernel independent component analysis. *J. Mach. Learn. Res.* **2002**, *3*, 1–48.
41. Luo, Y.; Tao, D.; Ramamohanarao, K.; Xu, C.; Wen, Y. Tensor canonical correlation analysis for multi-view dimension reduction. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 3111–3124. [[CrossRef](#)]
42. Kan, M.; Shan, S.; Zhang, H.; Lao, S.; Chen, X. Multi-view discriminant analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 188–194. [[CrossRef](#)]
43. Ma, H.T.H.L.Z.W. Graph based multi-modality learning. In *Proceedings of the 13th Annual ACM International Conference on Multimedia, Singapore, 6–11 November 2005*; pp. 6–11.
44. Hong, D.; He, W.; Yokoya, N.; Yao, J.; Gao, L.; Zhang, L.; Chanussot, J.; Zhu, X.X. Interpretable Hyperspectral Artificial Intelligence: When Non-Convex Modeling meets Hyperspectral Remote Sensing. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 52–87. [[CrossRef](#)]
45. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [[CrossRef](#)]
46. Grasedyck, L.; Kressner, D.; Tobler, C. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitt.* **2013**, *36*, 53–78. [[CrossRef](#)]
47. Comon, P. Tensors: A brief introduction. *IEEE Signal Process Mag.* **2014**, *31*, 44–53. [[CrossRef](#)]
48. Liu, Y.; Long, Z.; Huang, H.; Zhu, C. Low CP rank and tucker rank tensor completion for estimating missing components in image data. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 944–954. [[CrossRef](#)]
49. Liu, J.; Musialski, P.; Wonka, P.; Ye, J. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 208–220. [[CrossRef](#)] [[PubMed](#)]
50. Guangcan, L.; Zhouchen, L.; Shuicheng, Y.; Ju, S.; Yong, Y.; Yi, M. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 171–184.
51. 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. Geos. Remote Sens.* **2018**, *56*, 7183–7194. [[CrossRef](#)]
52. Wong, W.K.; Lai, Z.; Wen, J.; Fang, X.; Lu, Y. Low-rank embedding for robust image feature extraction. *IEEE Trans. Image Process.* **2017**, *26*, 2905–2917. [[CrossRef](#)]
53. Lin, Z.; Chen, M.; Ma, Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv* **2010**, arXiv:1009.5055.