Learning a Robust Local Manifold Representation for Hyperspectral Dimensionality Reduction

Danfeng Hong, *Student Member, IEEE*, Naoto Yokoya, *Member, IEEE*, and Xiao Xiang Zhu, *Senior Member, IEEE*

***Abstract*—Local manifold learning has been successfully applied to hyperspectral dimensionality reduction in order to embed nonlinear and nonconvex manifolds in the data. Local manifold learning is mainly characterized by affinity matrix construction, which is composed of two steps: neighbor selection and computation of affinity weights. There is a challenge in each step: First, the neighbor selection is sensitive to complex spectral variability due to nonuniform data distribution, illumination variations, and sensor noise; second, the computation of affinity weights is challenging due to highly correlated spectral signatures in the neighborhood. To address the two issues, in this paper, a novel manifold learning methodology based on locally linear embedding is proposed through learning a robust local manifold representation. More specifically, a hierarchical neighbor selection is designed to progressively eliminate the effects of complex spectral variability using joint normalization and to robustly compute affinity (or reconstruction) weights reducing multicollinearity via the refined neighbor selection. Additionally, an idea that combines spatial–spectral information is introduced into the proposed manifold learning methodology to further improve the robustness of affinity calculations. Classification is explored as a potential application for validating the proposed algorithm. The classification accuracy in the use of different dimensionality reduction methods is evaluated and compared, while two kinds of strategies are applied in selecting the training and test samples: random sampling and region-based sampling. Experimental results show the classification accuracy obtained by the proposed method is superior to those state-of-the-art dimensionality reduction methods.**

***Index Terms*—Dimensionality reduction (DR), hyperspectral image, local manifold learning (LML), multicollinearity, nonuniform data distribution.**

I. Introduction

HYPERSPECTRAL data are characterized by very rich spectral information, which enables us to detect targets of interest and analyze data attributes more easily, but also introduces drawbacks caused by its high dimensionality. As a result, the dimensionality reduction (DR) is a necessary and essential ingredient to address the aforementioned issue. A large number of DR techniques have been developed for a wide range of applications, including image segmentation [1], biometric [2], large-scale data classification [3], image/video analysis [4], and visualization [5]. Generally, these DR approaches can be categorized into linear and nonlinear methods.

Classical linear methods, such as principal component analysis (PCA) [6], easily fail to excavate the underlying data structure that lies in the complex real world. Comparatively, many nonlinear techniques, such as manifold learning (Isomap [7], locally linear embedding (LLE) [8], Laplacian eigenmaps (LE) [9], and local tangent space alignment (LTSA) [10]), exhibit unique advantages in DR and obtain state-of-the-art results in many fields. These examples of successful use of manifold learning mentioned above have widely attracted the attention of researchers working in the field of hyperspectral data analysis. Owing to merits of manifold learning, which can effectively map nonlinear and nonconvex manifolds in low-dimensional space, massive related approaches are introduced into hyperspectral image processing and successfully applied to various tasks, e.g., feature extraction [11], [12], classification [13]–[16], detection [17], [18], and multitemporal analysis [19]. In addition, it has been proven in [3] that the algorithm performance with global manifold methods is inferior to that with local manifold methods. As a typical and benchmark local manifold learning (LML) method, LLE explores locally linear and globally nonlinear assumptions to effectively capture the underlying intrinsic structure of data. LLE has been successfully applied to hyperspectral classification. Ma *et al.* [13] integrated LML with improved k-nearest neighbor for hyperspectral classification tasks. In [14], Ma *et al.* extended their work and proposed a kind of semisupervised hyperspectral image classification method based on LML. Tang *et al.* [16] proposed manifold based on sparse representation for hyperspectral classification, and they embedded the local geometric property using the local manifold representation into classification framework based on sparse representation in order to enforcedly keep consistent from sparse code to local manifold representation.

Current research on manifold learning methods in hyperspectral data processing mostly focuses on their potential for classification or detection tasks and frequently neglects the representation capability of the manifold structure, leading to difficulty in improving the classification accuracy. In other words,

Manuscript received September 30, 2016; revised December 11, 2016 and January 26, 2017; accepted February 25, 2017. Date of publication July 18, 2017; date of current version July 17, 2017. This work was supported in part by the European Research Council (ERC) under the European Unions Horizon 2020 Research and Innovation Programme under Grant Agreement ERC-2016-StG-714087 (Acronym: So2Sat), and in part by the Helmholtz Association under the framework of the Young Investigators Group "SiPEO" (VH-NG-1018, www.sipeo.bgu.tum.de). *(Corresponding author: Xiao Xiang Zhu.)*

D. Hong and X. X. Zhu are with the Remote Sensing Technology Institute (IMF), German Aerospace Center, Weßling 82234, Germany, and also with the Signal Processing in Earth Observation, Technical University of Munich, Munich 80333, Germany (e-mail: danfeng.hong@dlr.de; xiao.zhu@dlr.de).

N. Yokoya is with the Remote Sensing Technology Institute (IMF), German Aerospace Center, Weßling 82234, Germany, with the Signal Processing in Earth Observation, Technical University of Munich, Munich 80333, Germany, and also with the Department of Advanced Interdisciplinary Studies, University of Tokyo, Tokyo 153-8904, Japan (e-mail: yokoya@sal.rcast.u-tokyo.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2017.2682189

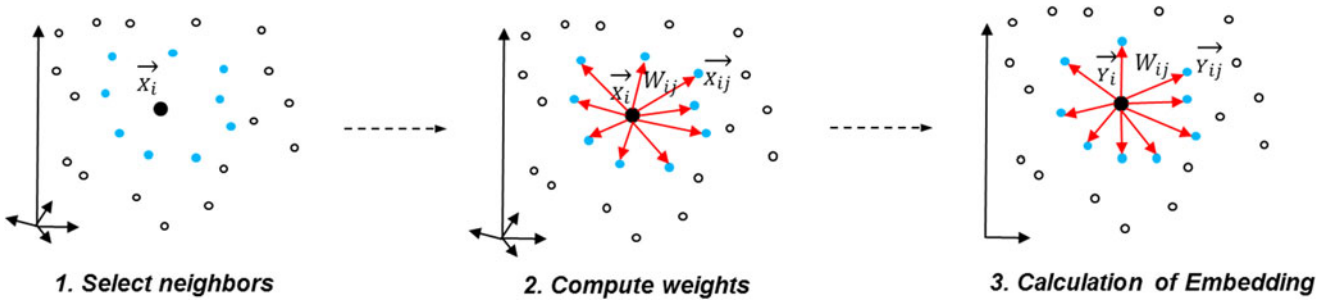


Fig. 1. Unified framework of the LML algorithm.

considerable attention has been paid to feature fusion and classifier design on manifold-based hyperspectral data processing; however, studies on manifold representation are still lacking. Consequently, the classification accuracy can be limited by bottlenecks in manifold learning, where a breakthrough in the level of the classifier is hardly made. To this end, a better manifold representation can break the stalemate.

In general, LML can be regarded as local graph embedding [20], while the most important part of the graph-embedding framework is the calculation of affinities (or similarities) of vertex pairs in a graph, i.e., the affinity matrix. The construction of the affinity matrix comprises two steps: neighbor selection (NS) and computation of affinity weights. There is a challenge in each step: 1) The NS is sensitive to the complex spectral variability due to environmental conditions (e.g., illumination and atmospheric conditions) and instrumental configurations (e.g., sensor noise) as well as data inherent structure (e.g., data distribution); 2) the computation of affinity weights is challenging due to highly correlated spectral signatures in the neighborhood. The latter issue is called *multicollinearity* when multiple regression analysis is used to obtain affinity weights. More specifically, multicollinearity refers to a phenomenon where multiple explanatory variables (spectral signatures in our case) are highly correlated in a linear regression model. This phenomenon in LML easily results in an inaccurate estimation of the affinity matrix.

To tackle these challenges, it is important to develop a robust and effective local manifold representation approach. In this paper, we mainly focus on improving LLE, which is one of the benchmark LML methods in many fields. A novel LML methodology on the basis of LLE is proposed, which aims at learning a robust local manifold representation (RLMR). Two main contributions of this paper are as follows: First, the hierarchical NS (HNS), which comprises joint normalization (JN) and refined NS (RNS), has been embedded into the original LLE framework to robustly select neighbors and mitigate multicollinearity in calculating affinity weights at the same time; Second, inspired by successful applications of spatial information in the hyperspectral classification, we model the spatial information into the proposed DR methodology in order to further improve the robustness of affinity calculations.

The remainder of this paper is described as follows: In Section II, we begin with a brief review of LML with three representative LML methods and provide comparative analysis. Section III introduces our methodology. Experimental results on classification are presented in Section IV. Finally, we provide conclusions and future outlook in Section V.

II. Local Manifold Learning

In this section, three representative LML methods, i.e., LE, LLE, and LTSA, are introduced in the graph-embedding framework, focusing on their advantages and disadvantages.

Generally, LML methods attempt to capture the underlying local manifold structure of the original data and preserve it in a low-dimensional space, which enables nonlinear DR. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ denotes N data samples that have D-dimensional features and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in \mathbb{R}^{d\times N}$ denotes their low-dimensional representations, where $d \ll D$. LML comprised mainly three steps:

1) neighbor selection;
2) computation of affinity weights; and
3) calculation of embedding.

The above-mentioned steps are illustrated in Fig. 1. Pairwise similarity measurements are performed to selected k neighbors for each data sample. Euclidean distance is commonly used for similarity measurement. Let $\mathbf{W} \in \mathbb{R}^{N\times N}$ be a sparse affinity matrix with the (i, j)th entry of the matrix representing the affinity weight from the ith sample and jth sample, where $j \in \phi_i$ and ϕ_i is a set of neighbors of the ith sample. The calculation of embedding coordinates is generally formulated as [20]

$$\begin{aligned}\hat{\mathbf{Y}} &= \arg\min_{\mathbf{Y}} \left\{ \sum_{i=1}^{N} \sum_{j\in\phi_i} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \mathbf{W}_{ij} \right\}, \text{s.t. } \mathbf{Y}\mathbf{B}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \\ &= \arg\min_{\mathbf{Y}} \left\{ \mathrm{tr}\left(\mathbf{Y}\mathbf{L}\mathbf{Y}^{\mathrm{T}}\right) \right\}, \text{ s.t. } \mathbf{Y}\mathbf{B}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \end{aligned} \quad (1)$$

where $\mathbf{L} \in \mathbb{R}^{N\times N}$ is the Laplacian matrix defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and $\mathbf{D}$ is a diagonal matrix defined by $\forall i\ \mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. $\mathbf{B}$ is a constant matrix defined by the formulation of each manifold learning method. LML methods can be mainly characterized by the construction of the affinity matrix $\mathbf{W}$, as described below.

In the following, three popular LML methods—namely LE, LLE, and LTSA—are introduced in details according to the aforementioned unified framework of the LML algorithm.

LE: The basic principle is to compute the affinity matrix for each data point in the original high-dimensional space using the

Gaussian function as [9]

$$\mathbf{W}_{ij}^{\mathrm{LE}} = \begin{cases} \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \big/ 2\sigma^2\right) & \text{if } j \in \phi_i \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The constant matrix $\mathbf{B}$ is defined as $\mathbf{B} = \mathbf{D}$. The low-dimensional representations can be obtained by solving the optimization equation (1).

LE is a very typical graph-based embedding method, which has been proven in [9] to be simple to implement and robust against outliers and noise. However, its limitation is also obvious [21], namely a local manifold structure is artificially designed by exploiting approximately pairwise distances with heat kernel, which brings relatively weak representation of local manifold without considering the property of local neighbors.

LLE: It represents the underlying local manifold structure by exploiting the local symmetries of linear reconstructions [5] between each data point and its neighbors in the high-dimensional space and then computes the low-dimensional embedding coordinates that preserve the reconstruction coefficients. The reconstruction coefficients, denoted as $\mathbf{A} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$, are obtained by the minimization

$$\hat{\mathbf{A}} = \arg\min_{\mathbf{A}} \left\{ \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j\in\phi_i} \mathbf{A}_{ij}\mathbf{x}_j \right\|_2^2 \right\} \quad \text{s.t.} \sum_{j\in\phi_i} \mathbf{A}_{ij} = 1 \tag{3}$$

where $\mathbf{A}_{ij}$ denotes the reconstruction weight between $\mathbf{x}_i$ and $\mathbf{x}_j$, if the *j*th data point is not one of the *k* neighbors of the *i*th data point ($j \in \phi_j$); otherwise $\mathbf{A}_{ij} = 0$. The reconstruction weights obey an important symmetry of being invariant to rotations, rescalings, and translations of any target data point and its neighbors [5]. The low-dimensional coordinates are obtained by minimizing the embedding cost function as

$$\hat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} \left\{ \sum_{i=1}^{N} \left\| \mathbf{y}_i - \sum_{j\in\phi_i} \mathbf{A}_{ij}\mathbf{y}_j \right\|_2^2 \right\} \quad \text{s.t.} \sum_{i=1}^{N} \mathbf{y}_i = 0, \ \frac{1}{N}\sum_{i=1}^{N} \mathbf{y}_i\mathbf{y}_i^{\mathrm{T}} = \mathbf{I}. \tag{4}$$

From the viewpoint of the graph-embedding framework, LLE can also be induced as the graph-embedding problem; therefore, (4) can be rewritten in the form of (1) as

$$\begin{aligned} \hat{\mathbf{Y}} &= \arg\min_{\mathbf{Y}} \left\{ \sum_{i=1}^{N} \left\| \mathbf{y}_i - \sum_{j\in\phi_i} \mathbf{A}_{ij}\mathbf{y}_j \right\|_2^2 \right\}, \text{s.t.} \mathbf{Y}\mathbf{B}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \\ &= \arg\min_{\mathbf{Y}} \left\{ \sum_{i=1}^{N} \sum_{j\in\phi_i} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \, \mathbf{W}_{ij}^{\mathrm{LLE}} \right\}, \text{s.t.} \mathbf{Y}\mathbf{B}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \\ &= \arg\min_{\mathbf{Y}} \left\{ \mathrm{tr}\left(\mathbf{Y}\mathbf{L}^{\mathrm{LLE}}\mathbf{Y}^{\mathrm{T}}\right) \right\}, \text{s.t.} \mathbf{Y}\mathbf{B}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \end{aligned} \tag{5}$$

where the affinity matrix ($\mathbf{W}^{\mathrm{LLE}}$) can be computed by the following equation [20]:

$$\mathbf{W}_{ij}^{\mathrm{LLE}} = \begin{cases} \mathbf{A}_{ij} + \mathbf{A}_{ji} - \mathbf{A}_{ij}\mathbf{A}_{ji} & \text{if } j \in \phi_i \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

and Laplacian matrix of LLE can be given by $\mathbf{L}^{\mathrm{LLE}} = \mathbf{D} - \mathbf{W}^{\mathrm{LLE}} = (\mathbf{I} - \mathbf{A})^{\mathrm{T}}(\mathbf{I} - \mathbf{A})$ [5].$\mathbf{B}$ is defined as $\mathbf{B} = \mathbf{I}$.

With a local regression technique [22], the property of local data is fully taken into consideration in LLE, which means that a local manifold structure can be effectively learned from local data. It is natural that it is able to improve the representation ability of the local manifold. That is not to say, however, that the RLMR can be obtained using LLE, since LLE is very sensitive to data distribution [23], variability [24], as well as multicollinearity.

LTSA: Similar to LLE, LTSA attempts to mine the underlying local manifold structure assuming local linearity. The core idea of LTSA is to utilize a local tangent space to represent a local manifold structure via a linear mapping, such as PCA. Therefore, it can be solved naturally as a graph-embedding problem, and the affinity matrix can be defined as $\mathbf{W}^{\mathrm{LTSA}} = \mathbf{D} - \mathbf{L}^{\mathrm{LTSA}}$, more specifically formulated as follows [14]:

$$\mathbf{W}_{ij}^{\mathrm{LTSA}} = \begin{cases} \frac{1}{k} + \frac{1}{k-1}\theta_i^{\mathrm{T}}\mathbf{\Lambda}^{-1}\theta_j & \text{if } j \in \phi_i \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where θ_i and θ_j are the local tangent coordinates of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, and $\mathbf{\Lambda}$ stands for the leading d eigenvalues of the covariance matrix of ϕ_i, and k is the number of neighbors for $\mathbf{x}_i$. The low-dimensional embedding is calculated by the following minimization:

$$\begin{aligned} \hat{\mathbf{Y}} &= \arg\min_{\mathbf{Y}} \left\{ \mathrm{tr}\left(\mathbf{Y}\mathbf{L}^{\mathrm{LTSA}}\mathbf{Y}^{\mathrm{T}}\right) \right\}, \text{s.t.} \mathbf{Y}\mathbf{B}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \\ &= \arg\min_{\mathbf{Y}} \left\{ \sum_{i=1}^{N} \sum_{j\in\phi_i} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \, \mathbf{W}_{ij}^{\mathrm{LTSA}} \right\}, \text{s.t.} \mathbf{Y}\mathbf{B}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \\ &= \arg\min_{\mathbf{Y}} \left\{ \sum_{i=1}^{N} \|\mathbf{y}_i\mathbf{H} - \mathbf{T}_i\theta_i\|_2^2 \right\}, \text{s.t.} \mathbf{Y}\mathbf{B}\mathbf{Y}^{\mathrm{T}} = \mathbf{I} \end{aligned} \tag{8}$$

where $\mathbf{H} = \mathbf{I} - \mathbf{e}\mathbf{e}^{\mathrm{T}}/k$ is the centering matrix, and $\mathbf{e}$ is a uniform vector with the size of $k \times 1$. $\mathbf{T}_i$ is a local transformation matrix with linearity, and $\mathbf{B}$ is defined as $\mathbf{B} = \mathbf{I}$.

Typically, a concept of local tangent space is proposed in LTSA to linearly and approximately estimate the local manifold structure, which is able to better capture the intrinsic structure of the underlying manifold [10]. However, such approximated estimation of the local manifold structure is possibly inaccurate, particularly in nonuniform distributed data [25], due to those data in the local manifold space without lying in, or closing to, a linear subspace. Also, although the performance of LTSA can improve the local manifold representation compared to LLE to some extent, it still fails when taking the data variability (e.g., noise) into consideration [26]. Furthermore, unlike LLE, LTSA explores a linear mapping (e.g., PCA) to find the principle information to depict the local manifold structure, accordingly

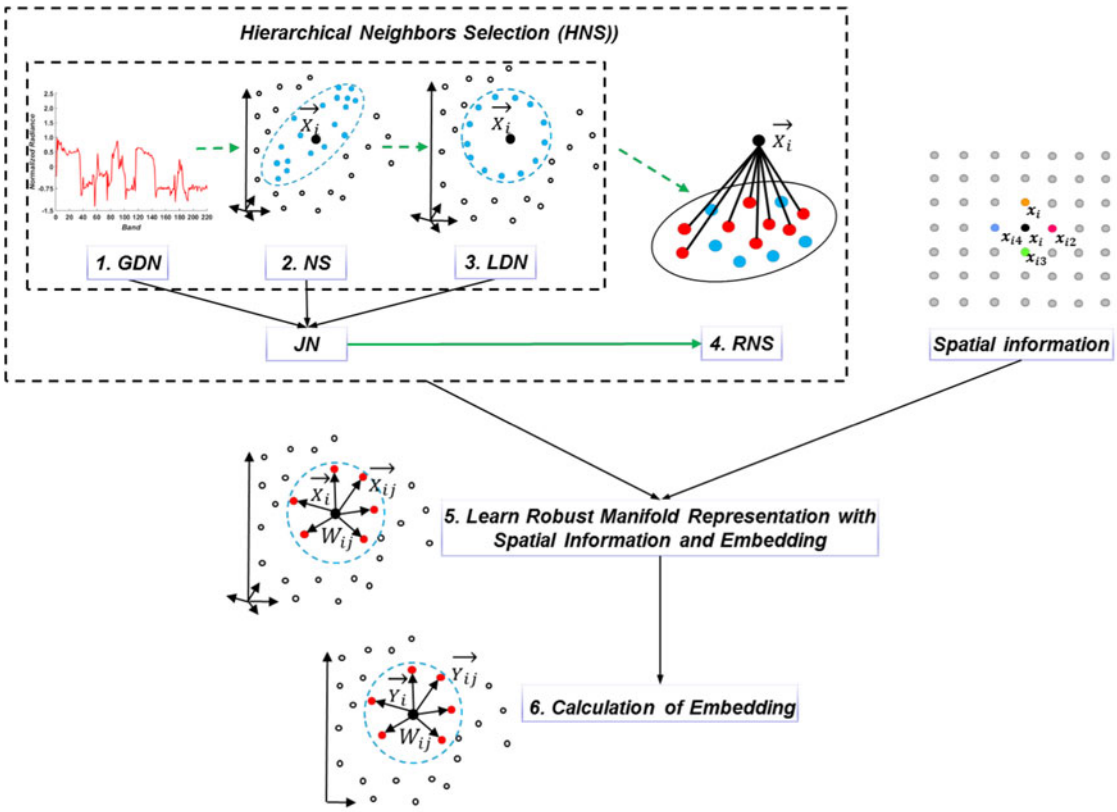


Fig. 2. Holistic diagram of the proposed method.

resulting in inevitable loss of useful information (e.g., geometric structure and local minutiae).

In summary, among the three LML methods, one advantage of LLE and LTSA over LE is that by using LLE or LTSA we can obtain a potentially better performance in DR due to their reasonably linear representation in the local manifold space. But the drawback of LLE and LTSA is that neither is highly robust against complex data variability, e.g., caused by noise, illumination, or nonuniform data distribution. Therefore, how to robustly learn the local manifold representation is an unsolved problem in LML. As a promising LML framework, LLE has been successfully applied in many fields and has obtained some amazing experimental results due to effectively and reasonably local linear assumption, for example, in hyperspectral data processing [3], [13], [14], [16], [17], [22]. However, sensitivity to variability and multicollinearity when calculating the local linear representation are hindering the advancement of LLE toward robustness and high performance. Therefore, in the next section, we emphatically introduce the proposed novel methodology based on LLE in an attempt to address the two issues mentioned above.

III. Robust Local Manifold Representation

In this section, a novel LML methodology is introduced in detail in order to learn an RLMR, mainly including the design of HNS and the integration of spatial contextual information. Fig. 2 shows the holistic diagram of the proposed methodology that mainly comprises the six steps given below, where the first four correspond to HNS and the fifth is the integration of spatial information.

Step 1. *Global data normalization (GDN)* is performed to deal with the spectral variability modeled by scaling and shifting.

Step 2. *NS* coarsely selects local neighbors of the target pixel.

Step 3. *Local data normalization (LDN)* is applied to make local data distribution more uniform and isotropic and further eliminate locally spectral variability.

Step 4. *RNS* aims at mitigating multicollinearity in the local manifold space, making it possible to obtain a relatively accurate and intrinsic structure of underlying manifold.

Step 5. *Computation of reconstruction weights with contextual information* jointly embeds spectral and spatial information for a robust calculation of the reconstruction weights.

Step 6. *Calculation of embedding* obtains the low-dimensional feature representation by embedding robust local manifold properties into the low-dimensional space.

A. Hierarchical Neighbors Selection

Fig. 3 shows the detailed diagram of HNS, which is composed of JN and RNS.

1) Joint Normalization: Data normalization is widely used in data preprocessing procedure, including hyperspectral data analysis [27], [28]. It aims at reducing the effect of numerous variations and improving the performance of subsequent algorithms. Generally, data normalization includes GDN and

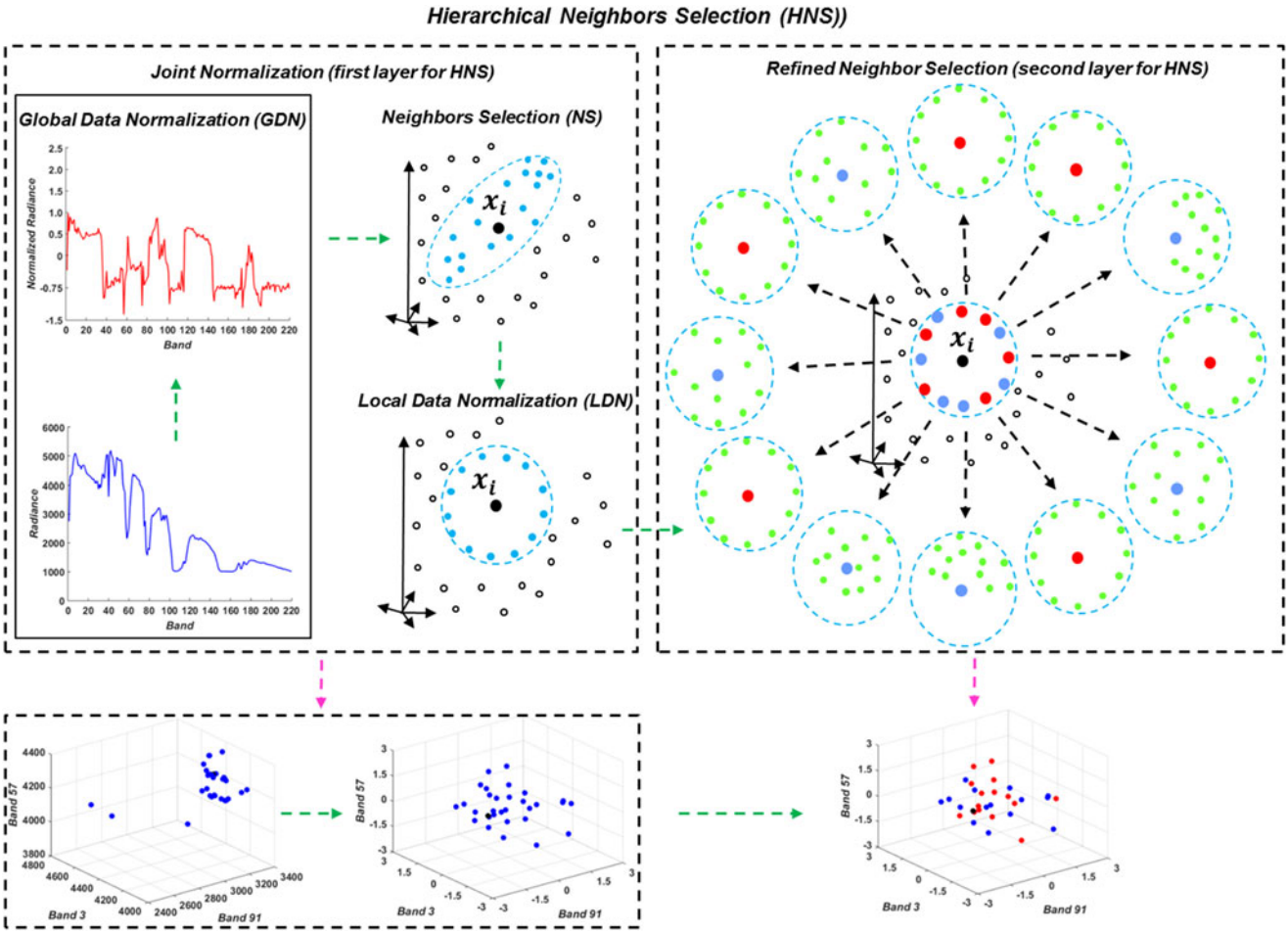


Fig. 3. Detailed diagram of HNS.

LDN [29]. The purpose of GDN is to mitigate illumination variations and modify the global data distribution so that it is more uniform and isotropic [30], [31], enabling them to be measured in the same, or similar, level or unit. Therefore, GND should be performed on the whole hyperspectral image. Unlike GDN, LDN tends to uniformize the mean and variance of the local neighborhood, which is particularly useful for nonuniform distributed data [32], [33]. Owing to the merits of GDN and LDN, JN is an appropriate approach to effectively address the issues of spectral variability and nonuniform data distribution, which can be implemented step-by-step via the following formulations

1) Global data normalization: It performs the following computations:

$$\mathbf{x}_i^{ns} = \frac{\mathbf{x}_i^o - c_i^o}{s_i^o} \tag{9}$$

$$\mathbf{x}_i^g = (\mathbf{x}_i^{ns} - \mathbf{c}^{ns})./\mathbf{s}^{ns} \tag{10}$$

where "./" means the elementwise division, $\mathbf{x}_i^o \in \mathbb{R}^{D\times 1}$ is the ith original spectral signature, and c_i^o and s_i^o are the mean value and variance corresponding to $\mathbf{x}_i^o$, respectively. $\mathbf{x}_i^{ns} \in \mathbb{R}^{D\times 1}$ stands for the normalized spectral signature. $\mathbf{X}^{ns} \in \mathbb{R}^{D\times N}$ represents all normalized spectral signatures made up of $\mathbf{x}_i^{ns}$, and $\mathbf{c}^{ns} \in \mathbb{R}^{D\times 1}$ and $\mathbf{s}^{ns} \in \mathbb{R}^{D\times 1}$ correspond to the mean value and variance of $\mathbf{X}^{ns}$, respectively. $\mathbf{x}_i^g \in \mathbb{R}^{D\times 1}$ stands for the normalized spectral signature of GDN. The normalization obtained by performing (9) can mitigate the effects of spectral variability that can be explained by scaling and shifting, whereas (10) makes the global data distribution more uniform and isotropic and puts the same weight on all the spectral bands, as shown in Fig. 3(Top-left).

2) Local data normalization: After selecting coarse neighbors for each data point using the Euclidean distance, LDN is exploited to make data distribution more uniform and isotropic in the local manifold space, which can be formulated as

$$\mathbf{x}_{ij}^l = \begin{cases} (\mathbf{x}_i^g - \mathbf{c}_i^g)./\mathbf{s}_i^g & j = 0 \\ (\mathbf{x}_{ij}^g - \mathbf{c}_i^g)./\mathbf{s}_i^g & j = 1, 2, \ldots, K \end{cases} \tag{11}$$

where "./" means the elementwise division, $\mathbf{X}_i^g = [\mathbf{x}_i^g, \mathbf{x}_{i1}^g, ..., \mathbf{x}_{ij}^g, ..., \mathbf{x}_{iK}^g] \in \mathbb{R}^{D\times(K+1)}$ consists of the globally normalized spectral features of ith data point and its K neighbors. $\mathbf{c}_i^g \in \mathbb{R}^{D\times 1}$ and $\mathbf{s}_i^g \in \mathbb{R}^{D\times 1}$ represent the mean value and variance of $\mathbf{X}_i^g$, respectively. $\mathbf{X}_i^l = [\mathbf{x}_i^l, \mathbf{x}_{i1}^l, ..., \mathbf{x}_{ij}^l, ..., \mathbf{x}_{iK}^l] \in \mathbb{R}^{D\times(K+1)}$ represents the final normalized spectral features for ith data point and its neighbors by JN. An example of local data distribution is shown in Fig. 3(Bottom-left). We can see that the data distribution becomes more uniform and isotropic

by means of LDN reducing the effects of nonuniform data distribution.

2) Refined Neighbor Selection: After JN, we obtain the rough results of NS where the influence of spectral variability has been mitigated, but multicollinearity still exists among neighbors. Multicollinearity leads to an inaccurate estimation of the affinity matrix, thereby degrade the quality of the local manifold structure. To address this issue, RNS is performed as the second layer of HNS. RNS, which is inspired by the local manifold alignment, is proposed to reduce the information redundancy [34] in the coarse neighborhood, as illustrated in Fig. 3(Right). RNS can mitigate the effects of multicollinearity in the next step, i.e., the calculation of reconstruction weights, while preserving local manifold properties. In detail, LFS is divided into two parts.

First, inspired by [35] and [36], we construct the local structure feature $\mathbf{F}_p^{\text{local}}$ for the data point p in the feature space using its neighbor's information $\mathbf{X}_p^l = [\mathbf{x}_{p1}^l, ..., \mathbf{x}_{pj}^l, ..., \mathbf{x}_{pK}^l] \in \mathbb{R}^{D\times K}$. $\mathbf{F}_p^{\text{local}}$ can be formed by the distance property between the feature of p with those of its neighbors using a Gaussian function:

$$F_{pj}^{\text{local}} = \exp\left(-\left\|\mathbf{x}_p^l - \mathbf{x}_{pj}^l\right\|_2^2\right) \tag{12}$$

$$\mathbf{F}_p^{\text{local}} = \left[F_{p1}^{\text{local}}, \ldots, F_{pj}^{\text{local}}, \ldots, F_{pK}^{\text{local}}\right]. \tag{13}$$

The second part is to screen out new local neighbors that have similar data distribution using the Kullback–Leibler divergence (KLD). The KLD has been justified to effectively measure the similarity of hyperspectral data distribution [37]. The difference of local features $\mathbf{d}^f = [d_1^f, ..., d_q^f, ..., d_K^f] \in \mathbb{R}^{1\times K}$ between the point p and its neighbor q can be measured as

$$d_q^f = \text{KLD}\left(\mathbf{F}_p^{local}||\mathbf{F}_q^{\text{local}}\right) + \alpha\text{KLD}\left(\mathbf{F}_q^{\text{local}}||\mathbf{F}_p^{\text{local}}\right) \tag{14}$$

$$\text{KLD}\left(\mathbf{F}_p^{\text{local}}||\mathbf{F}_q^{\text{local}}\right) = \sum_{j=1}^{K} F_{pj}^{\text{local}} \times \log_2\left(\frac{F_{pj}^{\text{local}}}{F_{qj}^{\text{local}}}\right) \tag{15}$$

$$\text{KLD}\left(\mathbf{F}_q^{\text{local}}||\mathbf{F}_p^{\text{local}}\right) = \sum_{j=1}^{K} F_{qj}^{\text{local}} \times \log_2\left(\frac{F_{qj}^{\text{local}}}{F_{pj}^{\text{local}}}\right) \tag{16}$$

where $\mathbf{F}_p^{\text{local}} \in \mathbb{R}^{1\times K}$ and $\mathbf{F}_q^{\text{local}} \in \mathbb{R}^{1\times K}$ stand for the local structure features of p and q in the spectral domain, respectively, and α is a penalty parameter balancing the two terms described in (15) and (16). Neighbors with the k smallest $\mathbf{d}^f$ value are chosen from the coarse neighbors as the new neighbors of the data point p, namely $\mathbf{X}_p^{nl} = [\mathbf{x}_{p1}^{nl}, ..., \mathbf{x}_{pj}^{nl}, ..., \mathbf{x}_{pk}^{nl}] \in \mathbb{R}^{D\times k}$. k is the final number of neighbors for each point, and we make the value of K equal to twofold k.

An example showing the effect of RNS is given in Fig. 4, where correlations between the target pixel and its neighbors are shown with and without using RNS. To be specific, given any target pixel, k neighbors were selected without RNS, whereas for RNS, $2k$ were selected at first and then k neighbors are refined from $2k$ neighbors. Therefore, the same number of neighbors k was obtained without RNS and with RNS. Fig. 4(Left) shows spectral signatures of neighbors from two different strategies (without RNS and with RNS). Although it is not so obvious, it still

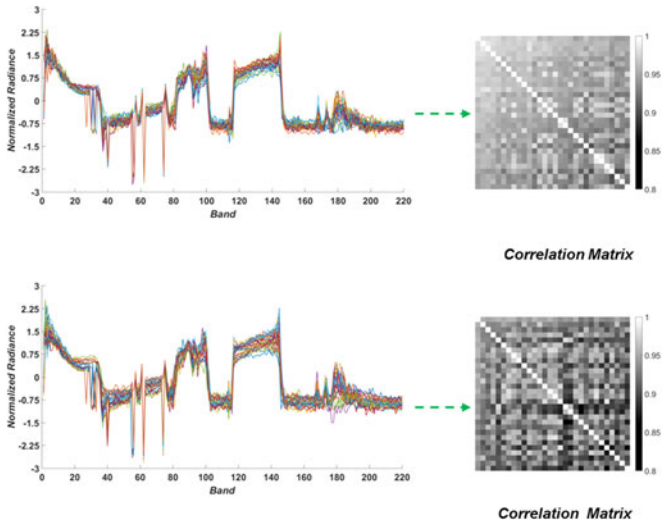


Fig. 4. (Left) Spectral signatures of local neighbors for an exemplar data point and (right) their correlations (top) without RNS and (bottom) with RNS.

emerges the slight difference that spectral signatures without RNS are more intensive than those with RNS, which means that those without RNS are likely to generate multicollinearity when computing the affine matrix (weight matrix). Fig. 4(Right) shows relatively obvious results regarding the reduction of multicollinearity. We can see that the values of correlation matrix with RNS are lower than those without RNS, which demonstrates that the linear correlations observed in the correlation matrix are effectively reduced after using RNS.

B. Local Manifold Representation With Spatial Contextual Information

To further improve the robustness of the calculation of reconstruction weights, the spatial information is incorporated into linear reconstructions. We assume that spatially neighboring spectral pixels can be explained by the same or similar reconstruction weights [38], if spatially neighboring pixels include similar spectral components. The calculation of reconstruction weights with spatial contextual information can be formulated based on (1) by adding the constraint that the reconstruction weights of the target pixel are approximately equal to the average of those of its neighboring pixels, as shown in the following:

$$\begin{aligned}\mathbf{a}_i^0 = \arg\min_{\mathbf{w}_i^0} &\left\{\sum_{s=0}^{4}\left\|\mathbf{x}_{is}^{nl} - \mathbf{X}_i^{nl}\mathbf{a}_i^s\right\|_2^2\right\}, \\ \text{s.t.} &\left\|\mathbf{X}_i^{nl}\left(4\mathbf{a}_i^0 - \sum_{s=1}^{4}\mathbf{a}_i^s\right)\right\|_2^2 \le \eta,\ (\mathbf{a}_i^s)^{\mathrm{T}}\mathbf{a}_i^s = 1, \\ &s = 0, 1, \ldots, 4\end{aligned} \tag{17}$$

where $\mathbf{X}_i^{nl} = [\mathbf{x}_{i1}^{nl}, ..., \mathbf{x}_{ij}^{nl}, ..., \mathbf{x}_{ik}^{nl}] \in \mathbb{R}^{D\times k}$ is the k-nearest neighbors selected by HNS. $\mathbf{x}_{is}^{nl}$, $s = 0, 1, ..., 4$ are the target spectral pixel and its four spatial neighbors, respectively, as an example shown in Fig. 5. Correspondingly, $\mathbf{a}_i^s \in \mathbb{R}^{k\times 1}$, $s = 0, 1, ..., 4$ are their reconstruction weights. η is a tiny real number

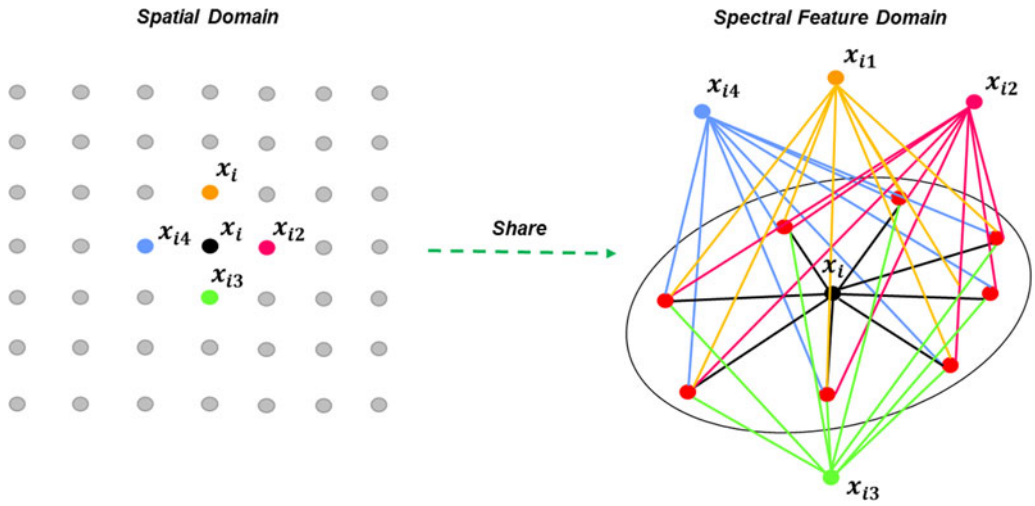


Fig. 5. Diagram for spatial–spectral combination in hyperspectral DR.

(here $\eta = 10^{-3}$) that represents the limit of error. Note that LDN should be conducted on this dataset composed of target spectral pixel and its spatial and spectral neighbors before calculating reconstruction weights.

We can regard (17) as a joint optimization problem. In this case, the objective function of (17) can be rewritten as

$$\mathbf{a}_i^0 = \arg\min_{\mathbf{a}_i^0} \left\{ \left\| \hat{\mathbf{X}}_i^{nl} - \mathbf{L}\hat{\mathbf{A}}_i \right\|_F^2 \right\}, \text{s.t.} \mathbf{C}\hat{\mathbf{A}}_i = [1\ 1\ 1\ 1\ 1]^{\mathrm{T}}$$

$$\mathbf{L} = \begin{bmatrix} 4\beta\mathbf{X}_i^{nl} & -\beta\mathbf{X}_i^{nl} & -\beta\mathbf{X}_i^{nl} & -\beta\mathbf{X}_i^{nl} & -\beta\mathbf{X}_i^{nl} \\ \mathbf{X}_i^{nl} & & & & \\ & \mathbf{X}_i^{nl} & & & \\ & & \mathbf{X}_i^{nl} & & \\ & & & \mathbf{X}_i^{nl} & \\ & & & & \mathbf{X}_i^{nl} \end{bmatrix},$$

$$\hat{\mathbf{A}}_i = \begin{bmatrix} \mathbf{a}_i^0 \\ \mathbf{a}_i^1 \\ \mathbf{a}_i^2 \\ \mathbf{a}_i^3 \\ \mathbf{a}_i^4 \end{bmatrix}, \quad \hat{\mathbf{X}}_i^{nl} = \begin{bmatrix} \mathbf{0} \\ \mathbf{x}_{i0}^{nl} \\ \mathbf{x}_{i1}^{nl} \\ \mathbf{x}_{i2}^{nl} \\ \mathbf{x}_{i3}^{nl} \\ \mathbf{x}_{i4}^{nl} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{e} & & & & \\ & \mathbf{e} & & & \\ & & \mathbf{e} & & \\ & & & \mathbf{e} & \\ & & & & \mathbf{e} \end{bmatrix} \tag{18}$$

where the sizes of $\mathbf{L}$, $\hat{\mathbf{X}}_i^{nl}$, $\hat{\mathbf{A}}_i$,$\mathbf{C}$ are $6D \times 5k$, $6D \times 1$, $5k \times 1$ and $5 \times 5k$, respectively. And $\mathbf{e} \in \mathbb{R}^{1\times k}$ is the unit vector with a size of $1 \times k$, and β is a penalty parameter to balance the importance between error item and constraint item in (18).

In order to solve (18), it can be further relaxed by means of Lagrange multipliers as represented by

$$\mathbf{a}_i^0 = \arg\min_{\mathbf{a}_i^0} \left\{ \left\| \hat{\mathbf{X}}_i^{nl} - \mathbf{L}\hat{\mathbf{A}}_i \right\|_F^2 + \lambda \left\| \mathbf{C}\hat{\mathbf{A}}_i - \hat{\mathbf{e}} \right\|_2^2 \right\} \tag{19}$$

where λ is also a penalty parameter, and here let it be 1 for simplicity as well as $\hat{\mathbf{e}} = [1\ 1\ 1\ 1\ 1]^{\mathrm{T}} \in \mathbb{R}^{5\times 1}$. The solution in (19) can be analytically derived [39] by matrix derivation operation as

$$\mathbf{a}_i^0 = \left(\mathbf{L}^{\mathrm{T}}\mathbf{L} + \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\right)^{-1}\left(\mathbf{L}^{\mathrm{T}}\hat{\mathbf{X}}_i^{nl} + \lambda\mathbf{C}^{\mathrm{T}}\hat{\mathbf{e}}\right). \tag{20}$$

Therefore, $\mathbf{a}_i^0$ is the weight vector for *i*th pixel by using RLMR. Following the framework shown in Fig. 2, the result of DR can be obtained by calculating the embedding using (1).

IV. Experiment

In this section, we explore the classification as a potential application and quantitatively evaluate the performance of DR algorithms using overall classification accuracy. The main focus of this paper is to learn a more robust and discriminative feature representation, rather than how to develop a more advanced classifier. Therefore, we use two common classifiers, namely the nearest neighbor (NN) algorithm based on the Euclidean distance and linear support vector machines (SVMs).

A. Hyperspectral Datasets

The experiments are carried out using two benchmark hyperspectral datasets.

1) Indian Pines AVIRIS Image: The first dataset was acquired by NASA's AVIRIS sensor over the Indian Pines test site in Northwest Indiana with the size of $145 \times 145 \times 220$ and 10 nm spectral resolutions over the range of 400–2500 nm, mainly including several kinds of vegetation. More specific classes and the number of samples can be found in Table I.
2) 2013 IEEE GRSS Data Fusion Contest (DFC) image: The second dataset was provided for the 2013 IEEE GRSS DFC acquired by the ITRES-CASI 1500 sensor with the size of $349 \times 1905 \times 144$ in the range of 380–1050 nm, which includes more varied categories.

B. Results of Indian Pines AVIRIS Data

For the first dataset, we adopted two sampling strategies to select training samples and test samples: random sampling and region-based sampling. Random sampling is a common way

TABLE I
NUMBER OF TRAINING SAMPLES AND TEST SAMPLES FOR EACH CLASS

No.	Class Name	Total	Cross Validation	Training	Testing
1	Corn-Notill	1434	50	50	1334
2	Corn-Mintill	834	50	50	734
3	Corn	234	50	50	134
4	Grass-Pasture	497	50	50	397
5	Grass–Trees	747	50	50	647
6	Hay-Windrowed	489	50	50	389
7	Soybean-Notill	968	50	50	868
8	Soybean-Mintill	2468	50	50	2368
9	Soybean-Clean	614	50	50	514
10	Wheat	212	50	50	112
11	Woods	1294	50	50	1194
12	Bldg-Gra-Tr-Driv	380	50	50	280
13	Stone–Stel–Tower	95	15	15	65
14	Alfalfa	54	10	10	34
15	Grass-Past-Mowed	26	5	5	16
16	Oats	20	5	5	10

for the validation of the hyperspectral classification. In contrast, classification using region-based sampling is more practical and challenging due to high correlation and limited variability of training samples, and thus an effective way to investigate the performance of the proposed method. We randomly assigned around 5% of total samples as cross-validation samples and then divided the rest into two parts: training samples (5% of total samples), by random sampling or region-based sampling, and test samples (90% of total samples). Moreover, ten replications were performed for selecting training and test samples based on the two aforementioned sampling strategies. The specific number of cross validation, training, and test samples is listed in Table I [40]. We compare the classification results on dimensionality-reduced data using the proposed method with those using some benchmark DR methods (PCA, KPCA [41], LLE, LE, and LTSA) and original spectral features (OSF). Three step-by-step methods, i.e., JN, HNS, and RLMR, are used for the proposed methods to investigate the effects of JN, LFS, and the integration of spatial information.

1) Performance Comparison and Analysis Between RLMR and Classical DR Methods: Initially, we conducted a fivefold cross validation on training samples in order to select the optimal parameter combination. Table II gives the classification accuracies obtained by using the nine methods with optimal parameters (*d, k*). It should be noted that two kinds of classification accuracy are applied here, including overall accuracy (total classification accuracy of all classes) and average accuracy (the average of the classification accuracy of each class), to evaluate the performance of the listed methods.

The proposed methods outperform the other methods both with random sampling and region-based sampling. Compared to OSF, JN, HNS, and RLMR increase the overall accuracy by 8.25%, 12.71%, and 21.1%, respectively, with random sampling, and 7.42%, 8.83%, and 10.46%, respectively, with region-based sampling. For the average accuracy, on the other hand, the corresponding increases are, respectively, 10.2%, 12.89%, 18.11% with random sampling, and 9.68%, 10.95%, 11.54% with region-based sampling.

The classification maps are shown in Figs. 6 and 7. It can be seen that the classification maps of JN, HNS, and RLMR include less salt-and-pepper errors. In particular, those of RLMR are smoother in the local spatial region, resulting from the embedding of spatial information. These results demonstrate the effectiveness of all three technical components of the RLMR, i.e., JN, RNS, and the integration of spatial information, and imply that they successfully contribute to extracting robust and discriminative low-dimensional feature representations. In contrast, the classification accuracies of the classical LML methods (e.g., LLE, LTSA) are holistically higher than those obtained by using OSF and PCA, and yet lower than the results of our proposed methods due to the sensitivity of variability with respect to LLE and the unavoidable loss of information with respect to LTSA. As for the performance of LE, it is even inferior to the performances of OSF and PCA, and considerably lower than LLE and LTSA, as discussed in Section II. This indicates that the performance of these methods is unstable in DR due to challenges involved in NS and affinity calculations.

To effectively support the conclusion obtained by the NN classifier, an advanced and common classifier—SVM [44] is also applied for classification under the same condition. In this paper, a linear version of SVM is selected for the classifier rather than nonlinear versions to investigate the capability of handing nonlinear structure in the data for all DR methods under comparison. Classification accuracies obtained via SVM and corresponding optimal parameters for nine methods are listed in Table III. Figs. 8 and 9 show classification maps for the different methods using the random sampling and region-based sampling strategies, respectively.

In addition, we can observe from Tables II and III that the performance of JN, HNS, and RLMR is progressively increased, which can be contributed by the used of normalization, RNS, and spatial information, respectively. To investigate the effectiveness of RNS, we compare the performance with RNS and without RNS via the NN classifier, listed in Table IV. We can clearly see that the classification accuracies of those methods with RNS are stably higher than those without RNS while the proposed method JN+RNS (HNS) shows the best performance.

2) Sensitivity Analysis of Parameters and Robustness Against Noise

a) Sensitivity analysis of parameters: The sensitivity of parameters is examined by varying the number of neighbors (k) and the size of reduced dimensionality (d) for LML methods, and the variance (v) of kernel for KPCA. As shown in Figs. 10 and 11, the performance of the LML methods is less sensitive to the parameters. In general, as observed from the data dimensionality point of view, the classification accuracy increases with increasing dimensionality, to a certain extent, and then holds steady. When the reduced dimensionality d reaches approximately 50, the results are basically stable for those ML-based methods, while the number of neighbors k is around 60 when accuracy reaches the nearly optimum level. As the number of neighbors gradually increases, the corresponding classification accuracy progressively increases to a peak (e.g., k is equal to around 50) and then dramatically drops. A large number of neighbors may obscure the local structure, whereas a small number of

TABLE II
CLASSIFICATION ACCURACIES USING OPTIMAL PARAMETERS VIA NN FOR DIFFERENT DR METHODS IN INDIAN PINE DATASET

Method	Optimal Parameters	Classification Accuracy			
		Random Sampling		Region-based sampling	
		Overall Accuracy	Average Accuracy	Overall Accuracy	Average Accuracy
OSF	/	64.74%	72.72%	44.78%	56.67%
PCA	$d = 50$	64.62%	72.66%	44.74%	56.64%
KPCA	$d = 50, v = 10$	66.95%	76.03%	48.79%	61.25%
LLE	$d = 60, k = 40$	68.49%	75.51%	47.45%	59.55%
LE	$d = 60, k = 7$	59.57%	68.19%	40.92%	52.73%
LTSA	$d = 60, k = 70$	71.22%	81.12%	51.63%	66.09%
JN	$d = 70, k = 40$	72.99%	82.92%	52.20%	66.35%
HNS	$d = 70, k = 40$	77.45%	85.61%	53.61%	67.62%
RLMR	$d = 50, k = 80$	**85.84%**	**90.83%**	**55.24%**	**68.21%**

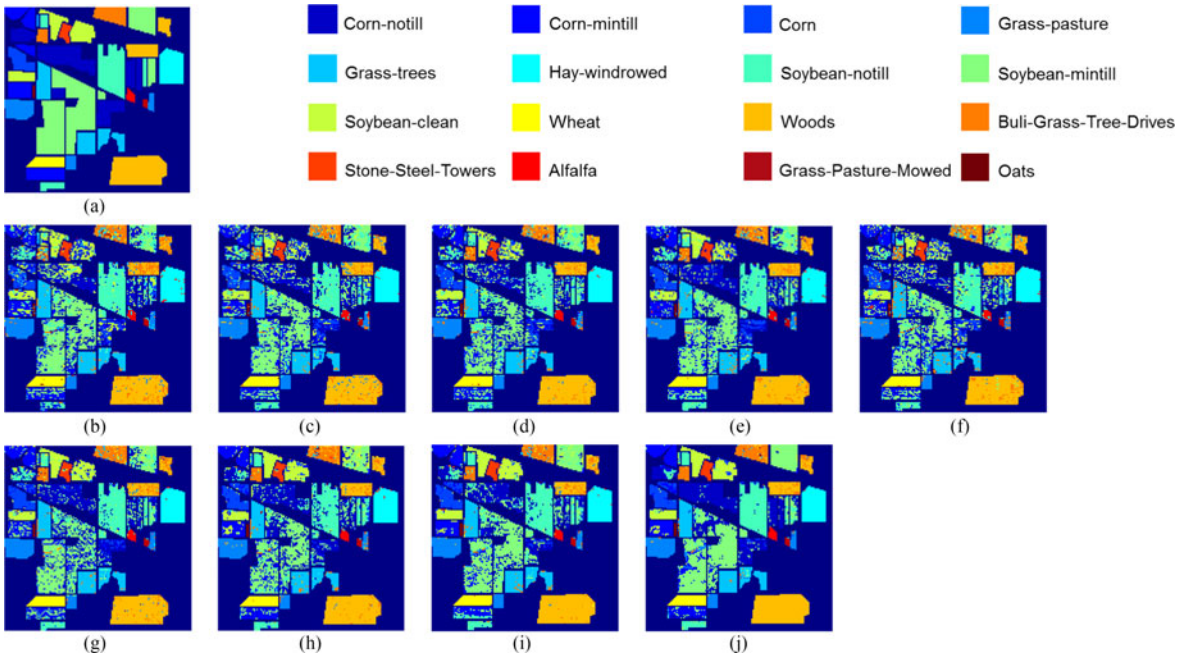


Fig. 6. NN classification maps for the Indian Pines dataset using all DR methods under comparison with the optimal parameters in Table II based on random sampling. (a) Ground truth and (b)–(j) results for OSF, PCA, KPCA, LLE, LE, LTSA, JN, HNS, and RLMR, respectively.

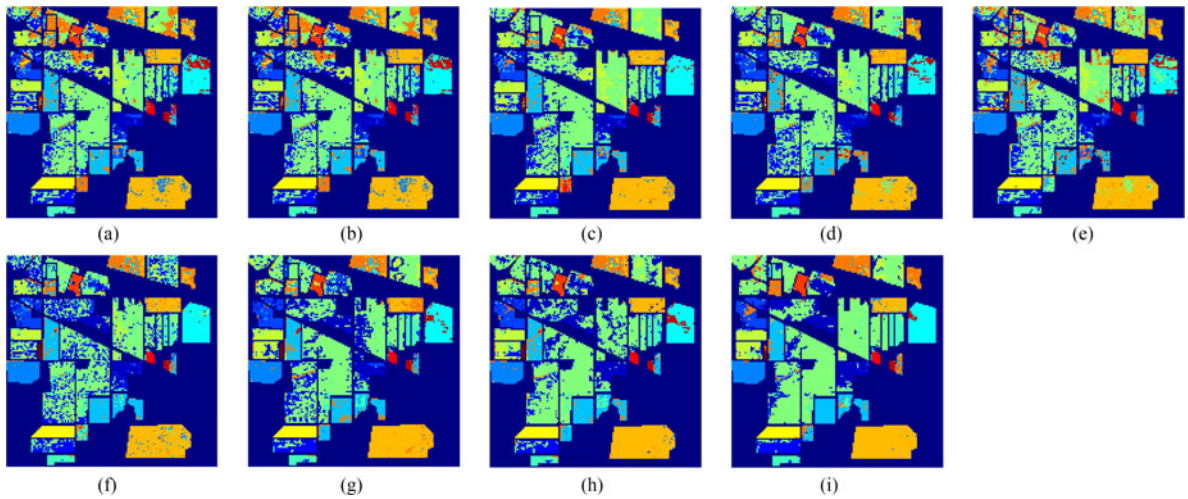


Fig. 7. NN classification maps for the Indian Pines dataset using all DR methods under comparison with the optimal parameters in Table II based on region-based sampling. (a)–(i) Results for OSF, PCA, KPCA, LLE, LE, LTSA, JN, HNS, and RLMR, respectively.

TABLE III
CLASSIFICATION ACCURACIES USING OPTIMAL PARAMETERS VIA SVM FOR DIFFERENT DR METHODS IN INDIAN PINES DATASET

Method	Optimal Parameters	Classification Accuracy			
		Random Sampling		Region-Based Sampling	
		Overall Accuracy	Average Accuracy	Overall Accuracy	Average Accuracy
OSF	/	73.86%	76.04%	47.39%	61.87%
PCA	$d = 30$	70.60%	79.50%	47.82%	58.38%
KPCA	$d = 60$, $v = 10$	72.16%	80.88%	50.36%	63.52%
LLE	$d = 40$, $k = 50$	71.47%	72.51%	47.23%	62.49%
LE	$d = 80$, $k = 3$	56.93%	65.06%	36.59%	52.85%
LTSA	$d = 40$, $k = 70$	75.49%	84.93%	52.79%	64.51%
JN	$d = 90$, $k = 60$	76.52%	83.03%	52.83%	66.95%
HNS	$d = 100$, $k = 50$	78.75%	85.04%	54.73%	68.03%
RLMR	$d = 40$, $k = 90$	**87.06%**	**90.93%**	**56.92%**	**69.24%**

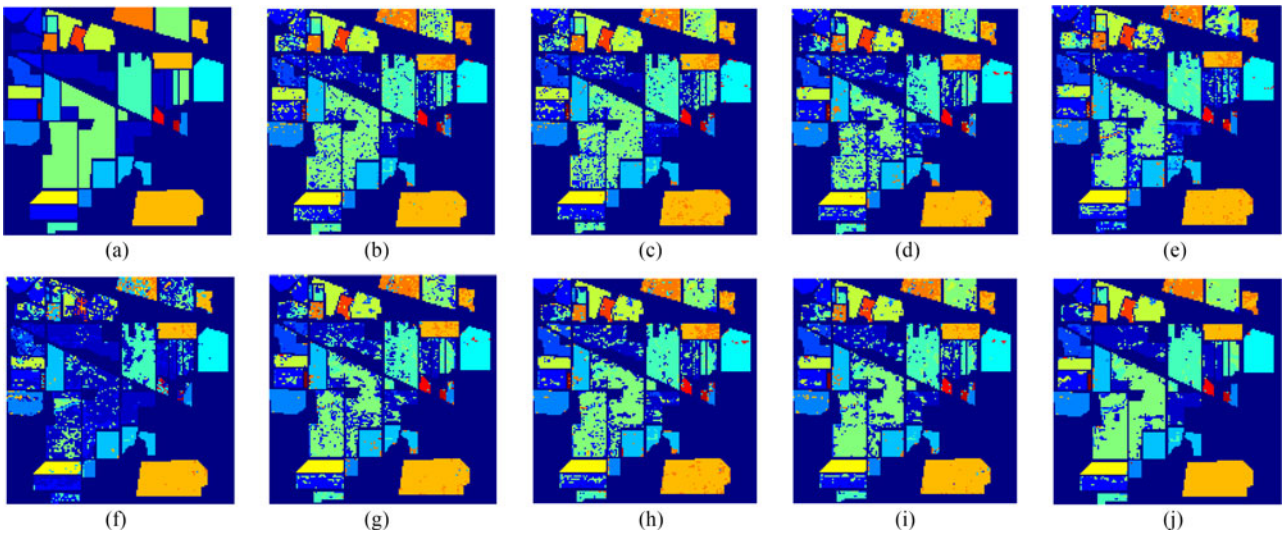


Fig. 8. SVM classification maps for the Indian Pines dataset using all DR methods under comparison with the optimal parameters in Table III based on random sampling. (a) Ground truth and (b)–(j) results for OSF, PCA, KPCA, LLE, LE, LTSA, JN, HNS, and RLMR, respectively.

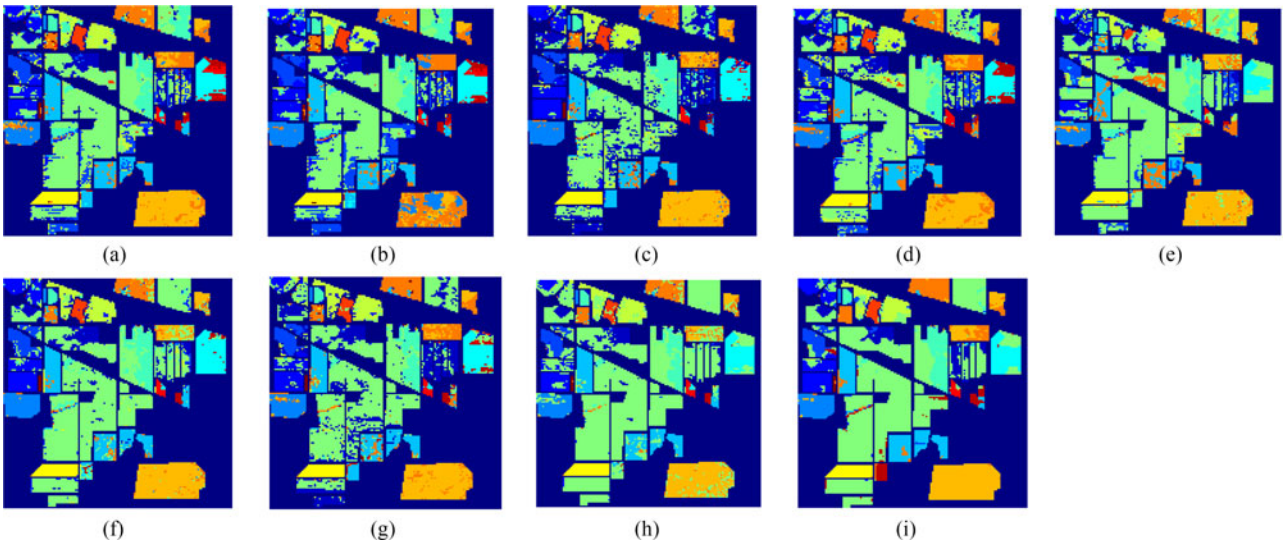


Fig. 9. SVM classification maps for the Indian Pines dataset using all DR methods under comparison with the optimal parameters in Table III based on region-based sampling. (a)–(i) Results for OSF, PCA, KPCA, LLE, LE, LTSA, JN, HNS, and RLMR, respectively.

TABLE IV
CLASSIFICATION ACCURACIES OBTAINED WITH NN CLASSIFIER FOR THE INDIAN PINES DATASET USING LLE WITH DIFFERENT NS METHODS

NS Method	Optimal Parameters	Classification Accuracy	
		Random Sampling	Region-Based Sampling
Euclidean	$d = 60, k = 40$	68.49%	47.45%
Euclidean+RNS	$d = 90, k = 50$	70.24%	48.85%
SAM	$d = 60, k = 80$	70.85%	48.97%
SAM+RNS	$d = 70, k = 50$	72.67%	49.50%
JN	$d = 70, k = 40$	72.99%	52.20%
JN+RNS (HNS)	$d = 70, k = 40$	**77.45%**	**53.61%**

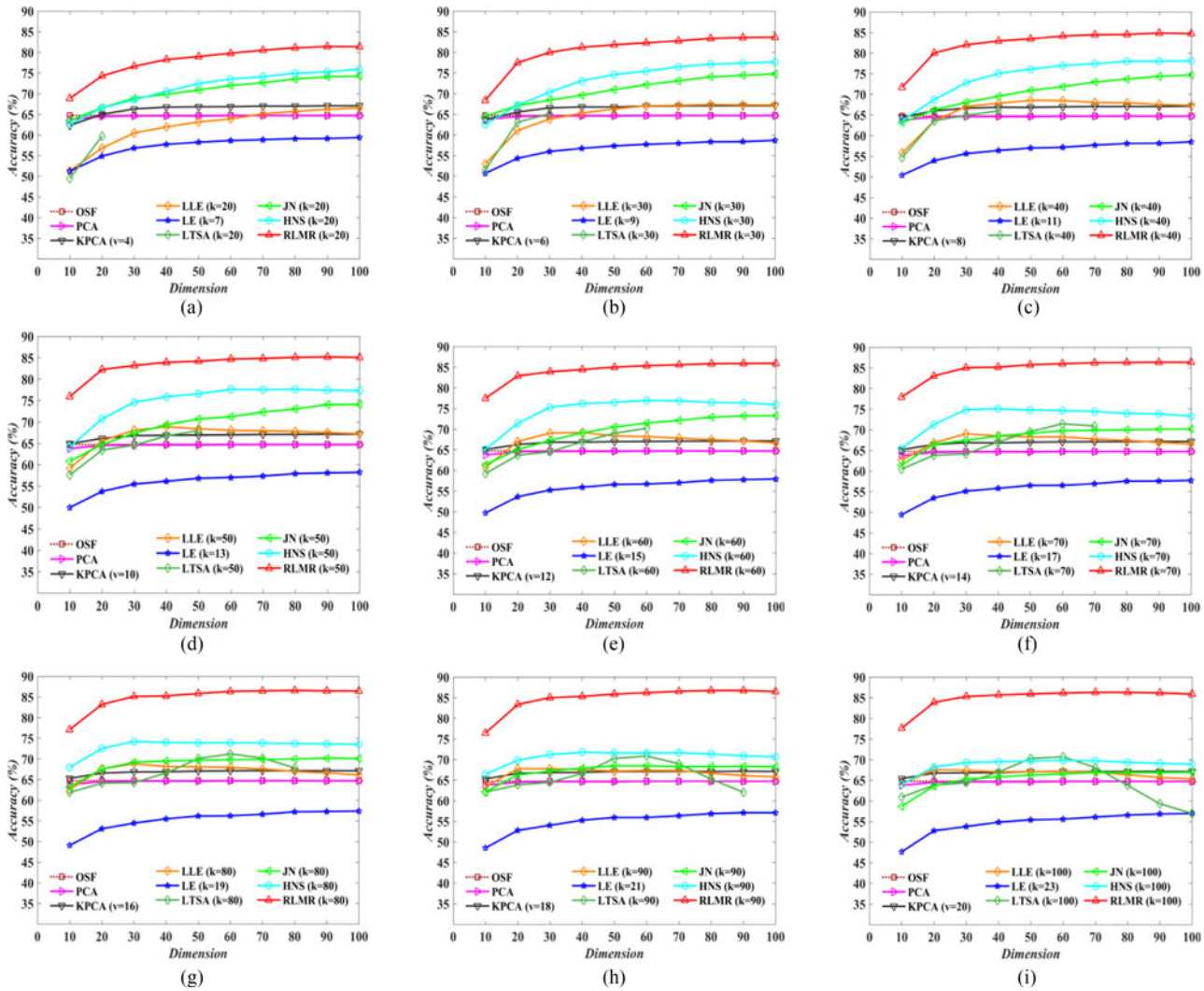


Fig. 10. Performance comparison: Classification accuracy as a function of data dimension using random sampling for the Indian Pines dataset. (a)–(i) Results using different numbers of neighbors, respectively.

neighbors may not sufficiently represent the local structure, causing the degradation of the DR performance. Proper parameters are determined from Figs. 10 and 11, which are basically consistent with parameter selection defined via cross validation given in Table II, where the LML methods are used for classification. However, it is worth noting that due to robustness of our proposed method (RLMR), its results remain stable with the increase in the number of neighbors k and reduced dimensionality d. Conversely, the performances of JN and HNS are progressively degrading with the change of parameters; particularly in a situation with a large k, the classification accuracies even degrade to a level similar to classical LML methods.

Unlike manifold learning methods, the size of reduced dimensionality (d) is the only parameter for PCA, and a limited number of d, around 30, is sufficient to obtain the best classification accuracy. Compared to PCA, KPCA shows a better

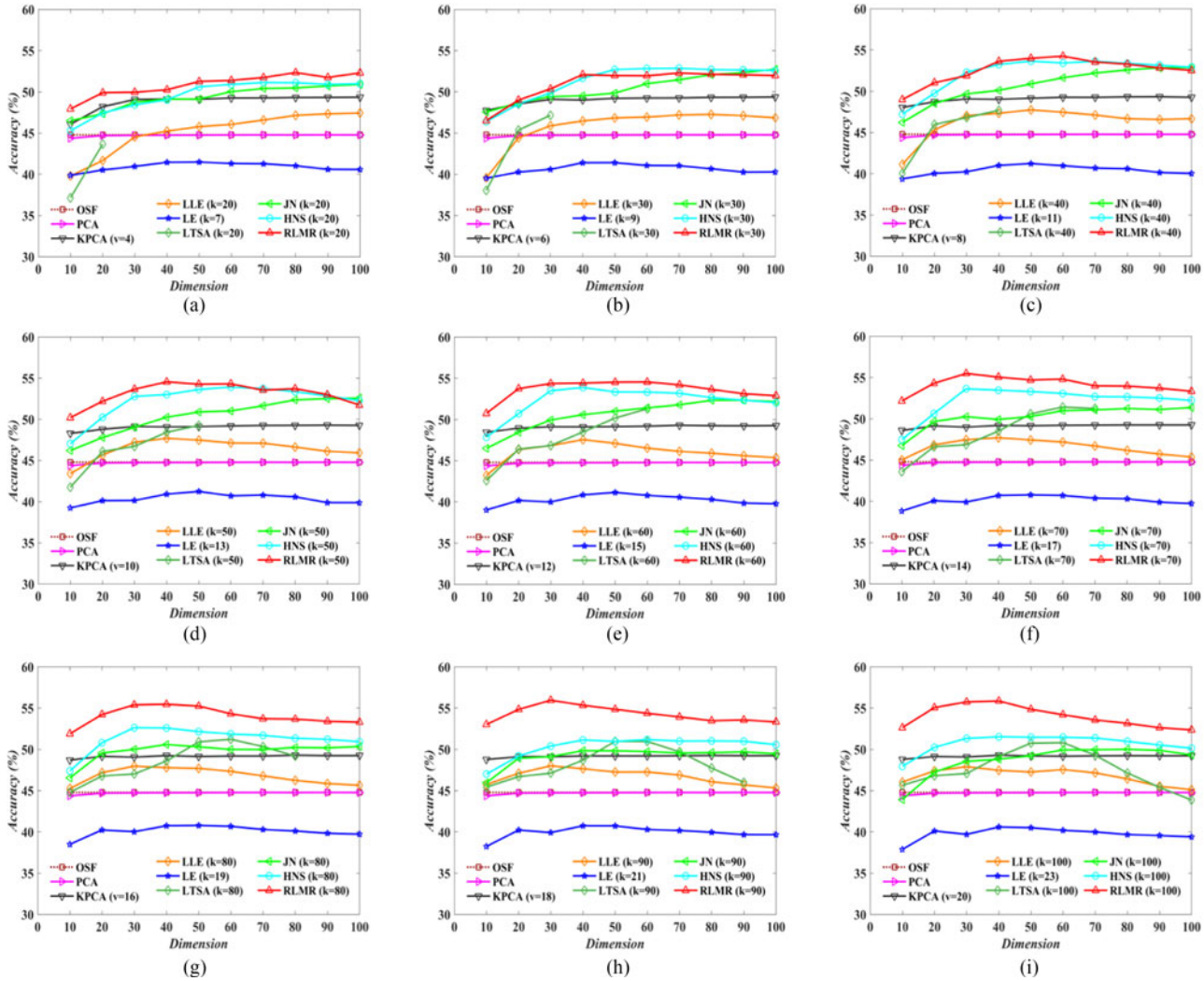


Fig. 11. Performance comparison: Classification accuracy as a function of data dimension using region-based sampling for the Indian Pines dataset. (a)–(i) Results using different numbers of neighbors, respectively.

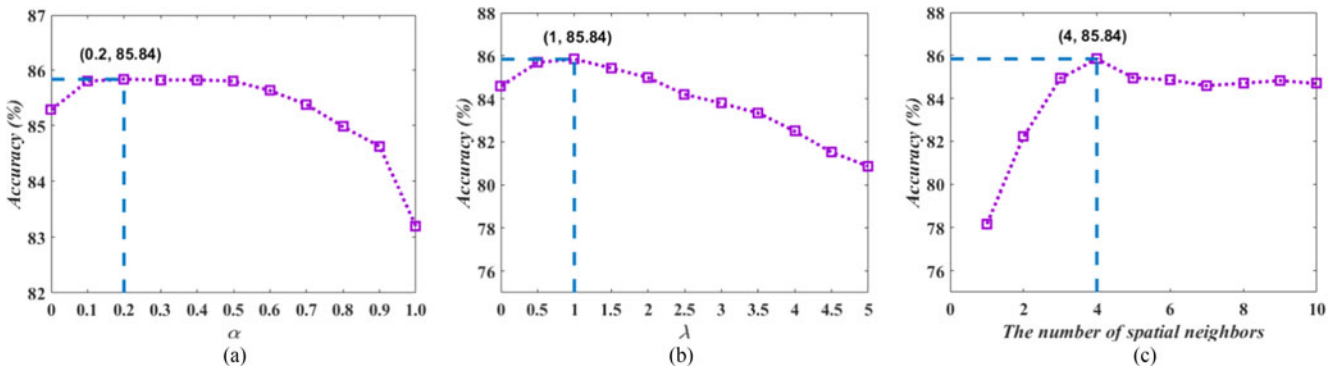


Fig. 12. Sensitivity of the proposed method to the change of three parameters: (a) $\boldsymbol{\alpha}$, (b) $\boldsymbol{\lambda}$, and (c) number of spatial neighbors.

performance owing to its advantage to capture nonlinear properties of the data; however, the parameter selection of kernel is important.

Except for the two parameters, the number of neighbors (k) and the size of reduced dimensionality (d), there are still several parameters in the proposed method, including α in RNS (14), the penalty parameter λ (19), and the number of spatial neighbors (17). With the change of these parameters, the best classification accuracies can be found on the Indian Pines dataset via the NN classifier, and the optimal parameters can be obtained accordingly, as shown in Fig. 12. More specifically, the parameter α in (14) balances similarities generated by KLD between

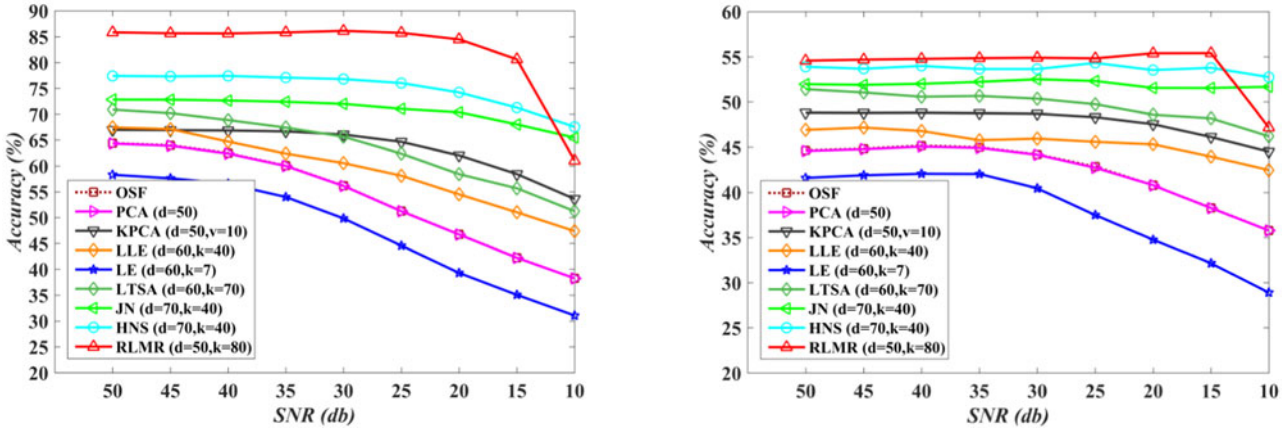


Fig. 13. Classification accuracy for the Indian Pines dataset with different SNRs using all DR methods under comparison based on (left) random sampling and (right) region-based sampling.

the target point and its neighbors. Equation (14) consists of two parts: One is the similarity of data distribution from the target point to its neighbors, and the other is the similarity of data distribution from neighbors to the target point. Obviously, the similarity of the former should be more important than that of latter, which means the parameter α should be less than 1. The optimal value of α is 0.2 corresponding to the best classification accuracy. Regarding the parameter λ, it is used to strike a balance between the error and constraint terms in (19). A proper value ($\lambda = 1$) is set according to experimental results shown in Fig. 12(b). If the number of spatial neighbors is too large or too small, spatial information can be overused or underused, as indicated by degraded classification performance in Fig. 12(c). The value of this parameter should be selected eclectically, and it is set as 4 in terms of the best classification accuracy observed in Fig. 12(c).

b) Robustness analysis: In order to validate the robustness of RLMR, a further experiment is performed, which adds noise with a different signal-to-noise ratio (SNR) into the AVIRIS Indian Pines image. The Gaussian noises are added to the image band by band with the same SNR. Classification was performed with various SNRs to investigate the robustness of the DR algorithms against noise. Fig. 13 shows the classification accuracies under the two sampling strategies. As the SNR decreases, the performance of JN, HNS, and RLMR are comparatively stable and superior compared to those of classical ML methods, PCA, KPCA, and OSF. This demonstrates the robustness of the proposed method against noise and implies its effectiveness for low SRN hyperspectral images.

TABLE V
CLASSIFICATION ACCURACIES FOR THE DFC DATASET USING NN AND DIFFERENT DR METHODS WITH OPTIMAL PARAMETERS

Method	Optimal Parameters	Classification Accuracy	
		Overall Accuracy	Average Accuracy
OSF	/	72.83%	76.16%
PCA	$d = 50$	72.85%	76.19%
KPCA	$d = 50, v = 10$	73.80%	77.79%
LLE	$d = 40, k = 50$	74.23%	77.49%
LE	$d = 60, k = 20$	66.70%	70.66%
LTSA	$d = 40, k = 50$	75.40%	78.75%
JN	$d = 60, k = 50$	77.45%	80.69%
HNS	$d = 80, k = 70$	78.52%	81.75%
RLMR	$d = 70, k = 50$	**80.87%**	**82.77%**

TABLE VI
CLASSIFICATION ACCURACIES FOR THE DFC DATASET USING SVM AND DIFFERENT DR METHODS WITH OPTIMAL PARAMETERS

Method	Optimal Parameters	Classification Accuracy	
		Overall Accuracy	Average Accuracy
OSF	/	74.68%	77.84%
PCA	$d = 30$	74.78%	77.79%
KPCA	$d = 30, v = 10$	75.12%	78,14%
LLE	$d = 60, k = 40$	75.33%	78.03%
LE	$d = 20, k = 30$	70.71%	72.98%
LTSA	$d = 30, k = 50$	76.04%	79.18%
JN	$d = 70, k = 60$	77.86%	80.12%
HNS	$d = 90, k = 60$	78.98%	82.01%
RLMR	$d = 90, k = 100$	**81.13%**	**82.79%**

C. Results of 2013 IEEE GRSS DFC Data

Similarly, we obtained the classification accuracies for the nine methods under the optimal parameters tuned by fivefold cross validation via NN and SVM classifiers using the given training samples in DFC, as listed in Tables V and VI. As can be seen in Tables V and VI, RLMR outperforms the other methods in DFC dataset. This demonstrates that the proposed novel ML method can indeed obtain the good feature representation, thereby further improving the classification accuracy. To be specific, similar results from the different classifiers listed in Tables V and VII also demonstrate the effectiveness and stability of the proposed method.

For simplicity, a general framework for the out-of-samples extension of ML proposed by Bengio [42], [43] is used in this paper in order to obtain the full classification map. The out-of-samples extension can be separated into two parts: first, an appropriate kernel function should be constructed (here, a Gaussian kernel is chosen); next, the Nystrom formulation should be applied for the generalization of a new data point. Classification

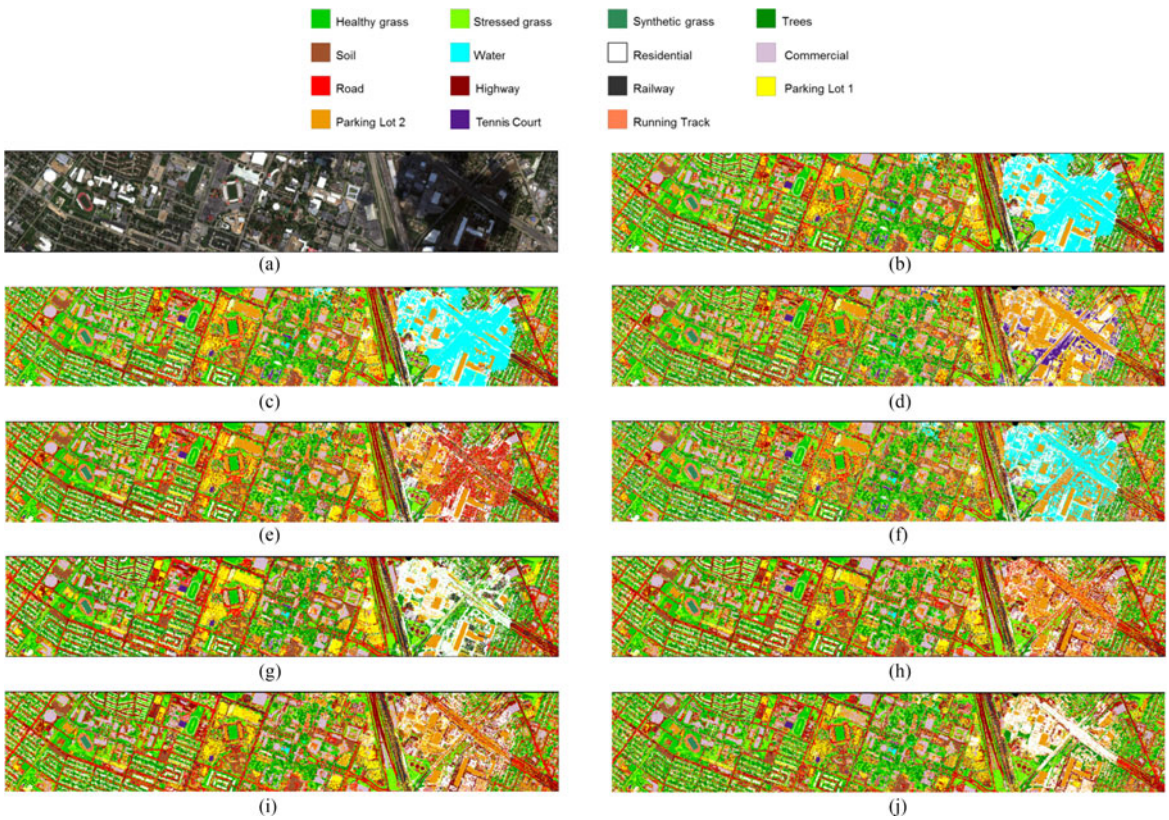


Fig. 14. NN classification maps of the DFC dataset using all DR methods under comparison with optimal parameters in Table V. (a) RGB image from the original hyperspectral image. (b)–(j) Results using OSF, PCA, KPCA, LLE, LE, LTSA, JN, HNS, and RLMR, respectively.

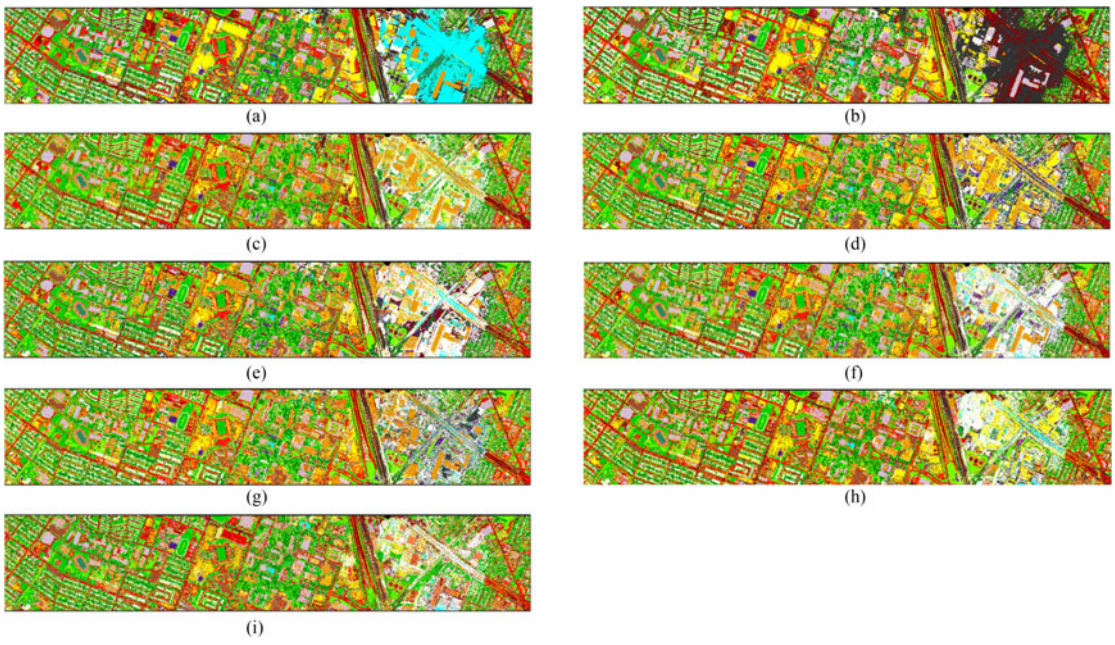


Fig. 15. SVM classification maps of the DFC dataset using all DR methods under comparison with optimal parameters in Table VI. (a)–(i) Results using OSF, PCA, KPCA, LLE, LE, LTSA, JN, HNS, and RLMR, respectively.

maps for different DR methods using the aforementioned optimal parameters are given in Figs. 14 and 15, respectively, corresponding to NN and SVM classifiers. As shown in Fig. 14(a), the east side of the scene is covered with shadows of clouds, resulting in the performance degradation of those previous DR methods—such as in Fig. 14(b)–(g) and Fig. 15(a)–(f)—while our proposed methods are rather robust against this variability observed in Figs. 14(h)–(j) and Fig. 15(g)–(j).

V. Conclusion

In this paper, a novel LML methodology—RLMR—is developed for hyperspectral DR in order to tackle two challenges of LML, involving: 1) NS due to complex spectral variability (e.g., noise, illumination, nonuniform data distribution), and 2) the computation of affinity weights due to multicollinearity. The proposed method is based on JN, RNS, and the integration of spatial information. It was validated via the classification using two benchmark hyperspectral datasets. Compared to other state-of-the-art methods, the proposed method achieves better performance in terms of the classification accuracy. RLMR has a more robust and stable performance than the other methods due to JN, RNS, and the embedding of spatial information, as shown in a series of experiments. In the future, we will further focus on how to more effectively embed the spatial information into DR framework. Additionally, the application of manifold learning methods to large-scale data should be given more attention in the future.

References

[1] Q. Zhang, R. Souvenir, and R. Pless, "On manifold structure of cardiac MRI data: Application to segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1092–1098.

[2] J. Yang, D. Zhang, J. Yang, and B. Niu, "Globally maximizing, locally minimization: Unsupervised discriminant projection with application to face and palm biometrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 650–664, Apr. 2007.

[3] D. Lunga, S. Prasad, M. M. Crawford, and O. Ersoy, "Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning, " *IEEE Signal Process.*, vol. 31, no. 1, pp. 55–66, Jan. 2014.

[4] D. Tosato, M. Farenzena, M. Spera, V. Murino, and M. Cristani, "Multi-class classification on Riemannian manifolds for video surveillance," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 378–391.

[5] L. K. Saul and S. T. Roweis, "Thing globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, Jun. 2003.

[6] I. T. Jolliffe, *Principle Component Analysis*. New York, NY, USA: Springer-Verleg, 1986.

[7] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, Dec. 2000.

[8] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5550, pp. 2323–2326, 2000.

[9] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Mar. 2003.

[10] Z. Y. Zhang and H. Y. L. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, Dec. 2004.

[11] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Exploiting manifold geometry in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 441–454, Mar. 2005.

[12] J. He, L. Zhang, Q. Wang, and Z. Li, "Using diffusion geometric coordinates for hyperspectral imagery representation, " *IEEE Trans. Geosci. Remote Sens.*, vol. 6, no. 4, pp. 767–771, Jan. 2009.

[13] L. Ma, M. M. Crawford, and J. W. Tian, "Local manifold learning-based-k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[14] L. Ma, M. M. Crawfor, X. Yang, and Y. Guo, "Local-manifold-learning-based graph construction for semisupervised hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2832–2844, May. 2015.

[15] H. Huang, H. Huo, and T. Fang, "Hierarchical manifold learning with application to supervised classification for high-resolution remotely sensed images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 3, pp. 1677–1692, Mar. 2013.

[16] Y. Tan, H. Yuan, and L. Li, "Manifold-based sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7606–7618, Dec. 2014.

[17] L. Ma, M. M. Crawford, and J. W. Tian, "Anomaly detection for hyperspectral images based on robust locally linear embedding, "*J. Infrared Millim. THz Waves*, vol. 31, no. 6, pp. 753–763, 2010.

[18] L. Zhang, L. Zhang, D. Tao, and X. Huang, "Sparse transfer manifold embedding for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 2, pp. 1030–1042, Feb. 2014.

[19] H. L. Yang and M. M. Crawford, "Spectral and spatial proximity-based manifold alignment for multitemporal hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 51–64, Jan. 2016.

[20] S. Yan, X. Dong, H. J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 40–51, Jan. 2007.

[21] S. Gerver, T. Tasdizen, and R. Whitaker, "Robust non-linear dimensionality reduction using successive 1-dimensional Laplacian eigenmaps," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 281–288.

[22] H. Chang and D. Y. Yeung, "Robust locally linear embedding," *Pattern Recognit.*, vol. 39, no. 6, pp. 1053–1065, Jun. 2006.

[23] Y. Goldberg and Y. A. Ritov, "LDR-LLE: LLE with low-dimensional neighborhood representation," in *Advances in Visual Computing*, vol. 5359. Berlin, Germany: Springer, Dec. 2008, pp. 43–54.

[24] J. A. Lee and M. Verleysen, "*Nonlinear Dimensionality Reduction*. New York, NY, USA: Springer, Oct. 2007.

[25] P. Zhang, H. Qiao, and B. Zhang, "An improved local tangent space alignment method for manifold learning," *Pattern Lett.*, vol. 32, no. 2, pp. 181–189, Jan. 2011.

[26] Y. Zhan and J. Yin, "Robust local tangent space alignment," *Neural Inf. Process.*, vol. 5863. Berlin, Germany: Springer, Dec. 2009, pp. 293–301.

[27] S. T. Monteiro, K. Uto, Y. Kosugi, K. Oda, Y. Lino, and G. Saito, "Hyperspectral image classification of grass species in northeast Japan," in *Proc. IEEE Int. Symp. Geosci. Remote Sens.*, Jul. 2008, pp. IV-399–IV-402.

[28] A. A. Nielsen, "The regularized iteratively reweighted MAD method for change detection in multi- and hyperspectral data," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 463–478, Jan. 2007.

[29] S. Lyu and E. P. Simoncelli, "Nonlinear image representation using divisive normalization, " in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[30] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2007, pp. 85–86.

[31] S. O. Los, P. R. J. North, W. M. F. Gery, and M. J. Barnsley, "A method to convert AVHRR normalized difference vegetation index time series to a standard viewing and illumination geometry," *Remote Sens. Environ.*, vol. 99, no. 4, pp. 400–411, Dec. 2005.

[32] D. Sage, "Local normalization filter to reduce the effect of non-uniform illumination," 2011. [Online]. Available: http://bigwww.epfl.ch/sage/soft/localnormalization/

[33] S. Azadi, J. Maitin-Shepard, and P. Abbeel, "Optimization-based artifact correction for electron microscopy image stacks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 219–235.

[34] D. Wang, F. Nie, and H. Huang, "Feature selection via global redundancy minimization," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2743–2755, Sep. 2015.

[35] Y. Pei, F. Huang, F. Shi, and H. Zhi, "Unsupervised image matching based on manifold alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1658–1664, Jun. 2012.

[36] C. Wang, J. Lai, and J. Zhu, "Graph-based multiprototype competitive learning and its applications," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev*., vol. 42, no. 6, pp. 934–946, Dec. 2012.

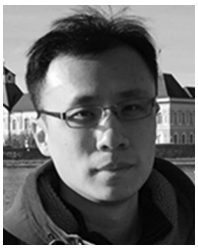
[37] M. Jia, M. Gong, E. Zhang, Y. Li, and L. Jiao, "Hyperspectral image classification based on nonlocal means with a novel class-relativity measurement," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 7, pp. 1300–1304, Jul. 2014.

[38] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation, " *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

[39] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 471–478, Nov. 2011.

[40] P. Ghamisi, J. A. Benediktsson, and M. O. Ulfarsson, "Spectral-spatial classification of hyperspectral images based on hidden Markov random fields, " *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2565–2574, Feb. 2014.

[41] B. Scholkopf, A. J. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural. Comput.*, vol. 10, no. 5, pp. 583–588, 1998.
[42] Y. Bengio, O. Delalleau, and N. Le Roux, "Learning eigenfunctions links spectral embedding and kernel PCA," *Neural Comput.*, vol. 16, no. 10, pp. 2197–2219, 2004.
[43] Y. Bengio, J. F. Paiement, and P. Vincent., "Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, Cambridge, MA, USA, 2003, pp. 177–184.
[44] J. A. Benediktsson and P. Ghamisi, *Spectral-Spatial Classification of Hyperspectral Remote Sensing Images*. Boston, MA, USA: Artech House.

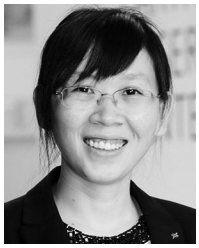


Danfeng Hong (S'16) received the B.Sc. degree in computer science and technology from the Neusoft College of Information, Northeastern University, Dalian, China, in 2012, the M. Sc. degree in computer vision from Qingdao University, Qingdao, China, in 2015. He has been working toward the Ph.D. degree in the hyperspectral data analysis including dimensionality reduction and nonlinear spectral unmixing, Technical University of Munich, Munich, Germany, and the Remote Sensing Technology Institute, German Aerospace Center (DLR), Wessling, Germany, since September 2015.

His research interests include image processing, pattern recognition, and machine learning and their applications in hyperspectral data analysis.

Naoto Yokoya (S'10–M'13) received the M.Sc. and Ph.D. degrees in aerospace engineering from the University of Tokyo, Tokyo, Japan, in 2010 and 2013, respectively.

From 2012 to 2013, he was a Research Fellow with the Japan Society for the Promotion of Science, Tokyo, Japan. Since 2013, he has been an Assistant Professor with the University of Tokyo. Since 2015, he has also been an Alexander von Humboldt Research Fellow with the German Aerospace Center (DLR), Oberpfaffenhofen, Germany, and Technical University of Munich (TUM), Munich, Germany. His research interests include image analysis and data fusion in remote sensing.



Xiao Xiang Zhu (S'10–M'12–SM'14) received the Bachelor's degree in space engineering from the National University of Defense Technology, Changsha, China, in 2006. She received the M.Sc., Dr.-Ing., and Habilitation degrees in the field of signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

Since 2011, she has been a Scientist with the Remote Sensing Technology Institute, German Aerospace Center (DLR), Oberpfaffenhofen, Germany, where she is also the Head of the Team Signal Analysis. Since 2013, she has also been a Helmholtz Young Investigator Group Leader and appointed as a TUM Junior Fellow. In 2015, she was appointed as a Professor in the Signal Processing in Earth Observation, TUM. She was a Guest Scientist or Visiting Professor at the Italian National Research Council (CNR-IREA), Naples, Italy, Fudan University, Shanghai, China, the University of Tokyo, Tokyo, Japan, and the University of California, Los Angeles, Los Angeles, CA, USA, in 2009, 2014, 2015, and 2016, respectively. Her main research interests include advanced InSAR techniques, such as high-dimensional tomographic SAR imaging and SqueeSAR; computer vision in remote sensing including object reconstruction and multidimensional data visualization; big data analysis in remote sensing; and modern signal processing, including innovative algorithms, such as sparse reconstruction, nonlocal means filter, robust estimation, and deep learning, with applications in the field of remote sensing, such as multi/hyperspectral image analysis.

Dr. Zhu is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.