



Article

Local Matrix Feature-Based Kernel Joint Sparse Representation for Hyperspectral Image Classification

Xiang Chen ¹, Na Chen ², Jiangtao Peng ^{2,*} and Weiwei Sun ³¹ School of Mathematics and Statistics, Hubei University of Science and Technology, Xianning 437099, China² Hubei Key Laboratory of Applied Mathematics, Faculty of Mathematics and Statistics, Hubei University, Wuhan 430062, China³ Department of Geography and Spatial Information Techniques, Ningbo University, Ningbo 315211, China

* Correspondence: pengjt1982@hubu.edu.cn; Tel.: +86-27-8866-2127

Abstract: Hyperspectral image (HSI) classification is one of the hot research topics in the field of remote sensing. The performance of HSI classification greatly depends on the effectiveness of feature learning or feature design. Traditional vector-based spectral–spatial features have shown good performance in HSI classification. However, when the number of labeled samples is limited, the performance of these vector-based features is degraded. To fully mine the discriminative features in small-sample case, a novel local matrix feature (LMF) was designed to reflect both the correlation between spectral pixels and the spectral bands in a local spatial neighborhood. In particular, the LMF is a linear combination of a local covariance matrix feature and a local correntropy matrix feature, where the former describes the correlation between spectral pixels and the latter measures the similarity between spectral bands. Based on the constructed LMFs, a simple Log-Euclidean distance-based linear kernel is introduced to measure the similarity between them, and an LMF-based kernel joint sparse representation (LMFKJSR) model is proposed for HSI classification. Due to the superior performance of region covariance and correntropy descriptors, the proposed LMFKJSR shows better results than existing vector-feature-based and matrix-feature-based support vector machine (SVM) and JSR methods on three well-known HSI data sets in the case of limited labeled samples.

Keywords: hyperspectral image classification; joint sparse representation; covariance; correntropy



Citation: Chen, X.; Chen, N.; Peng, J.; Sun, W. Local Matrix Feature-Based Kernel Joint Sparse Representation for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 4363. <https://doi.org/10.3390/rs14174363>

Academic Editors: Zhiqiang Zhou, Chunlei Huo, Lurui Xia and Yi Liu

Received: 13 August 2022

Accepted: 31 August 2022

Published: 2 September 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

Hyperspectral images (HSIs) contain hundreds of continuous spectral bands, which can provide a large amount of information for different types of applications, such as military target detection, mineral identification, fine agriculture, natural resource surveys, and so on [1]. In these applications, classification is usually needed. HSI classification is to assign a land cover label to each pixel in an HSI based on the model built on available training samples, which has been one of the hot research topics in the field of remote sensing [2,3].

The performance of HSI classification greatly depends on the effectiveness of feature learning or feature design [4–9]. Because HSI contains spectral and spatial information, different types of spectral features, spatial features, and spectral–spatial joint features have been designed. The spectral characteristics that provide reflections of materials at specific spectral bands can be directly used for classification [10,11]. Early HSI classification methods usually use the spectral features or dimension-reduced spectral features [10,11]. Due to the spatial local homogeneous property, different spatial features have been designed to describe spatial textural or structural information [5]. The typical spatial features include morphological profiles [4], Gabor features [12], and local binary pattern (LBP) features [13]. Integrating the rich spectral and spatial information of HSI, many spatial–spectral joint feature extraction and classification methods have been proposed [2,14], such as composite kernels-based methods [15–18], joint representation-based methods [19–22], and deep

learning methods [23,24]. Typical deep learning methods are convolutional neural network (CNN)-based methods [25], such as 3-D CNN [25], attention-based adaptive spectral-spatial kernel improved residual network (A2S2K-ResNet) [24], and CNN with a noise-inclined module and denoise framework (NoiseCNN) [26].

These vector-based spectral–spatial features have shown good performance in HSI classification. However, when the number of available labeled samples is limited, these vector-based features are usually no longer effective. There is an urgent need to develop advanced feature-extraction methods that can fully exploit the correlation and similarity in spectral and spatial domains. To describe the spectral correlation in a spatial local neighborhood, local covariance matrix features have been designed [7,27]. Local covariance descriptor computes the covariance matrix of samples in a local region and uses the matrix as feature to reflect the correlation of samples in the region [28,29]. It is clear that the covariance feature is a matrix feature whose size is only related to the dimensionality of the data. Therefore, it can be performed on regions with different sizes or shapes [29]. Fang et al. proposed a local covariance matrix representation (LCMR) method for spatial–spectral feature extraction of HSIs [7], where a covariance matrix of neighboring pixels in a refined local spatial neighborhood is computed as features for the SVM classification with the use of Log-Euclidean kernels. In [7], the original HSI data are first preprocessed by the maximum noise fraction (MNF) method to reduce the dimensionality. Rather than using MNF, Yang et al. first performed extended multi-attribute profile (EMAP) transformations to reduce the dimensionality of the original HSI and then extracted the covariance matrix features on the EMAP dimension’s reduced data for the kernel-based joint sparse representation (KJSR) classification [27]. Considering the nonlinearity between HSI pixels, Zhang et al. proposed a local correntropy matrix representation method (LCEM) for HSI classification [30]. Correntropy is a robust similarity metric and can be used to handle nonlinear and non-Gaussian data [31]. Peng et al. used the correntropy metric to replace the least-squares metric in the JSR model, and the resulting model is robust to both band noise and inhomogeneous pixels [20]. In [30], the correntropy matrix feature can represent the correlation between spectral bands in a spatial local region and has shown excellent classification performance.

In recent decades, the JSR-based classification method has attracted much attention due to its simplicity and effectiveness [19–21,32]. Exploiting the similarity of neighboring pixels, JSR uses a common training dictionary to sparsely and linearly represent all neighboring pixels simultaneously and computes the class reconstruction residual for classification [19]. To cope with nonlinear problems, KJSR methods perform JSR in a kernel-based feature space [33]. Traditional KJSR performs on vector-based features. To exploit region features, Yang et al. used the region covariance matrix feature to replace vector feature and proposed a Log-Euclidean kernel-based KJSR (LogEKJSR) method [27]. Although different linear and nonlinear Log-Euclidean kernels are considered, the nonlinear representation ability of covariance matrix feature itself is still insufficient [30].

In previous works [7,30], although both the covariance and correntropy matrices have been introduced to reflect the relations of features in a local region, they are different features. The covariance matrix mainly measures the correlation between neighboring pixels, while the correntropy matrix measures the nonlinear similarity (low-level and high-level similarities) between spectral variables. To make full use of local features from the pixel and variable aspects, we combine the local covariance and correntropy matrix features and design a novel local matrix feature (LMF) for a region in this paper. In the construction of LMF, a new logarithmic transformed feature of covariance or correntropy matrix is first defined to transfer the matrix distance computation in the Riemannian manifold to general distance in the Euclidean space. Then, the LMF is designed as a linear combination of local covariance matrix feature and local correntropy matrix feature, where the former describes the correlation between spectral pixels and the latter measures the similarity between spectral bands. Based on the constructed LMFs, a simple Log-Euclidean distance-based linear kernel is introduced to measure the similarity between them, and an LMF-based

kernel joint sparse representation (LMFKJSR) model is proposed for HSI classification. Due to the superior performance of region covariance and correntropy descriptors, the proposed LMFKJSR shows better results than existing vector-feature-based and matrix-feature-based support vector machine (SVM) and JSR methods on three well-known HSI data sets in the case of limited labeled samples.

2. The Proposed Method

Figure 1 shows the framework of the proposed local matrix feature-based kernel joint sparse representation (LMFKJSR) method. The maximum noise fraction (MNF) method is first used to reduce the the dimensionality of the original HSI data. Then, spatial local neighborhoods are constructed and local matrix features (LMFs) are extracted. By computing kernels for LMFs and performing the KJSR, HSI pixels can be classified.

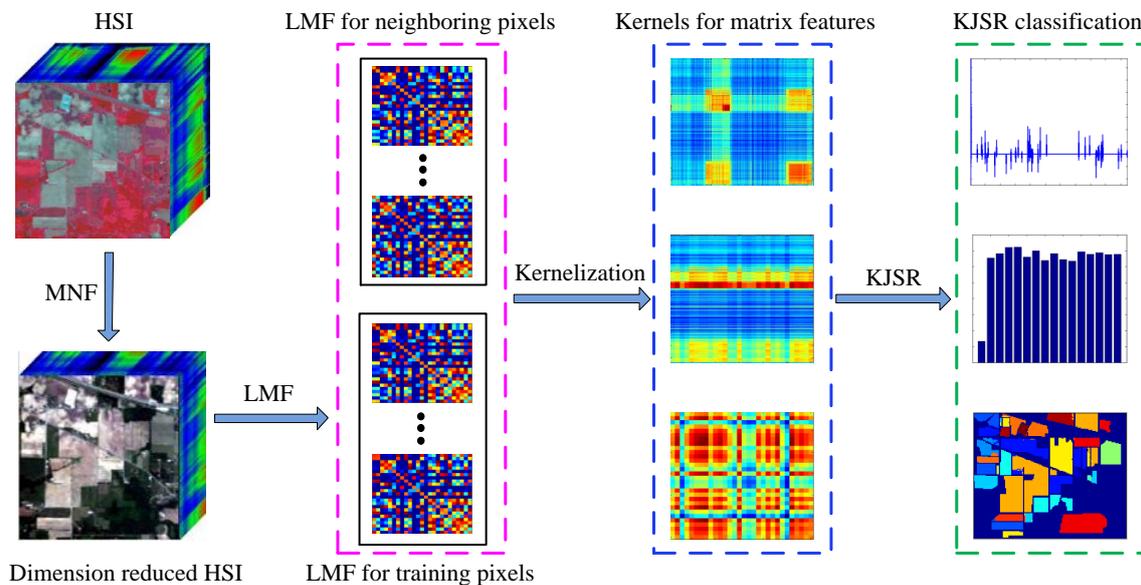


Figure 1. The framework of the proposed LMFKJSR algorithm, which mainly consists of three steps, i.e., maximum noise fraction-based dimensionality reduction, local matrix feature extraction, and matrix kernel-based JSR classification.

2.1. Maximum Noise Fraction

Considering that the original HSI has high dimensionality and meanwhile may contain different types of noise, the maximum noise fraction (MNF) method is used to reduce the dimensionality of and denoise HSIs [7,30,34]. It finds a linear transformation matrix \mathbf{A} to reduce the dimensionality of original data such that the signal-to-noise ratio of the lower-dimensional data is maximized.

Given an observed data $\mathbf{X} \in \mathcal{R}^{N \times D}$ with N observations and D variables, assume that \mathbf{X} can be represented as the summation of an idea datum \mathbf{X}_0 and a noise \mathbf{E} : $\mathbf{X} = \mathbf{X}_0 + \mathbf{E}$, and the idea datum \mathbf{X}_0 and noise \mathbf{E} are unrelated; then, the covariance of \mathbf{X} is:

$$Cov(\mathbf{X}) \triangleq \Sigma_{\mathbf{X}} = \Sigma_{\mathbf{X}_0} + \Sigma_{\mathbf{E}}. \tag{1}$$

The linear transformation matrix $\mathbf{A} \in \mathcal{R}^{D \times d}$ is obtained by solving the following problem:

$$\arg \max_{\mathbf{A}} \left\{ \frac{\mathbf{A}^T \Sigma_{\mathbf{X}_0} \mathbf{A}}{\mathbf{A}^T \Sigma_{\mathbf{E}} \mathbf{A}} = \frac{\mathbf{A}^T \Sigma_{\mathbf{X}} \mathbf{A}}{\mathbf{A}^T \Sigma_{\mathbf{E}} \mathbf{A}} - 1 \right\}. \tag{2}$$

Equation (2) can be transferred to a generalized eigenvector and eigenvalue problem, and its solution consists of the eigenvectors corresponding to the first d largest eigenvalues of $(\Sigma_E)^{-1}\Sigma_X$. The dimension-reduced data are

$$\mathbf{Z} = \mathbf{X}\mathbf{A}. \quad (3)$$

In the experiments, the noise covariance Σ_E is first estimated based on the minimum/maximum autocorrelation factor method [7,34], and the dimension d is set to 25.

2.2. Local Neighborhood Construction

Previous works have demonstrated that inhomogeneous or interfering pixels in a spatial neighborhood will affect the performance of spatial-spectral-based classifiers [7,9,21,27,30,32,35,36]. To reduce the effect of spatial inhomogeneous pixels, researchers have proposed different methods to construct spatial adaptive neighborhoods, such as superpixel or image-segmentation-based methods [9,36], neighboring-pixel-weighting methods [21,32,35], and neighboring-pixel-selection methods [7,27,30].

As the focus of this manuscript is the local matrix feature representations, we use a simple way to eliminate the effect of spatial inhomogeneous pixels by directly deleting several inhomogeneous neighboring pixels [27]. Given a pixel \mathbf{z} , a $w_1 \times w_1$ spatial window centered at \mathbf{z} is first determined. Then, the distance between each spatial neighboring pixel and the center pixel \mathbf{z} is computed. By sorting the distance in ascending order, the first m_1 pixels with the smallest distances are retained to construct a new local neighborhood (i.e., deleting $w_1^2 - m_1$ inhomogeneous pixels).

2.3. Local Matrix Representation

2.3.1. Local Covariance Matrix Representation

A local covariance descriptor is originally proposed to extract second-order features in local image patches [37,38]. The covariance descriptor reflects the correlation of local features and shows good discriminative performance for image classification [7,30,37].

For a local region \mathcal{R} that contains m pixels, i.e., $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m$ with $\mathbf{z}_i \in \mathbb{R}^d$, we can represent it as a matrix: $\mathbf{Z}_{\mathcal{R}} = [\mathbf{z}_1, \dots, \mathbf{z}_m]^T \in \mathbb{R}^{m \times d}$. The covariance matrix of size $d \times d$ between these pixels is computed as:

$$\Sigma_{\mathbf{Z}_{\mathcal{R}}} = \frac{1}{m-1} \sum_{i=1}^m (\mathbf{z}_i - \bar{\mathbf{z}})(\mathbf{z}_i - \bar{\mathbf{z}})^T, \quad (4)$$

where $\bar{\mathbf{z}} = \frac{1}{m} \sum_{i=1}^m \mathbf{z}_i$.

2.3.2. Local Correntropy Matrix Representation

Given two random variables u and v , the correntropy between them is defined as [31]:

$$\mathbf{V}_{\sigma}(u, v) = \mathbf{E}[\kappa_{\sigma}(u, v)], \quad (5)$$

where \mathbf{E} is the expectation operator, and $\kappa_{\sigma}(\cdot)$ is the kernel function and usually uses the Gaussian kernel $\kappa_{\sigma}(u, v) = \exp(-(u-v)^2/\sigma^2)$.

In the case of Gaussian kernel function, by the Taylor series expansion of Equation (5), we can obtain

$$\mathbf{V}_{\sigma}(u, v) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n! \sigma^{2n}} \mathbf{E}[(u-v)^{2n}]. \quad (6)$$

The correntropy with the Gaussian kernel contains all the even order moment information of $u-v$ and hence can reflect the high-level similarities between variables.

In practice, the joint probability density function of u and v is often unknown, so the expectation operation in Equation (5) cannot be computed. Given limited empirical data $\{(u_i, v_i)\}_{i=1}^m$, the correntropy can be estimated by the following empirical correntropy:

$$\hat{\mathbf{V}}_{m,\sigma}(u, v) = \frac{1}{m} \sum_{i=1}^m \kappa_{\sigma}(u_i, v_i) \tag{7}$$

Correntropy can be used to measure nonlinear similarity between variables. In the local region \mathcal{R} or matrix $\mathbf{Z}_{\mathcal{R}}$, we denote $z_{i,p}$ as the p -th component of pixel \mathbf{z}_i and $\mathbf{b}_p = [z_{1,p}, \dots, z_{m,p}]^T \in \mathbb{R}^{m \times 1}$ as a spectral variable or vector. If we regard spectral vectors \mathbf{b}_p and \mathbf{b}_q as two variables, then the correntropy between them can be defined as [30]:

$$\mathbf{V}(\mathbf{b}_p, \mathbf{b}_q) = \kappa_{\sigma}(\mathbf{b}_p - \mathbf{b}_q) / m = \frac{1}{m} \exp\left(-\frac{\|\mathbf{b}_p - \mathbf{b}_q\|_2^2}{\sigma^2}\right) \tag{8}$$

The local correntropy matrix representation of features in the local region \mathcal{R} is:

$$\mathbf{C}_{\mathbf{Z}_{\mathcal{R}}} = \{\mathbf{V}(\mathbf{b}_p, \mathbf{b}_q)\}_{p,q=1}^d \tag{9}$$

2.3.3. Local Matrix Feature

To make the covariance matrix $\Sigma_{\mathbf{Z}_{\mathcal{R}}}$ or correntropy matrix $\mathbf{C}_{\mathbf{Z}_{\mathcal{R}}}$ strictly positive definite, regularization is applied to the original matrix as [7,27]: $\mathbf{C} = \mathbf{C} + 10^{-3} * \text{trace}(\mathbf{C}) * \mathbf{I}$, where \mathbf{I} is the identity matrix. To measure the distance between symmetric positive definite (SPD) matrices, the logarithmic operation is performed on the matrices and the Log-Euclidean distance between two SPD matrices \mathbf{C}_1 and \mathbf{C}_2 is [37,38]:

$$d_{\log}(\mathbf{C}_1, \mathbf{C}_2) = \|\log(\mathbf{C}_1) - \log(\mathbf{C}_2)\|_F. \tag{10}$$

If we consider $\log(\mathbf{C}_1)$ as the feature corresponding to matrix \mathbf{C}_1 , then Equation (10) measures the Euclidean distance between logarithmic transformed features in a Riemannian manifold [38].

Although both the covariance and correntropy matrices can reflect the relations of features in a local region, they are different features. The covariance matrix $\Sigma_{\mathbf{Z}_{\mathcal{R}}}$ mainly measures the correlation between neighboring pixels, while the correntropy matrix $\mathbf{C}_{\mathbf{Z}_{\mathcal{R}}}$ measures the nonlinear similarity (low-level and high-level similarities) between spectral variables. To make full use of local features from the pixel and variable aspects, we combine the local covariance and correntropy matrix features and design a new local matrix feature for a region \mathcal{R} as

$$\mathbf{L}_{\mathcal{R}} = \mu \log(\Sigma_{\mathbf{Z}_{\mathcal{R}}}) + (1 - \mu) \log(\mathbf{C}_{\mathbf{Z}_{\mathcal{R}}}), \tag{11}$$

where μ is a weighting parameter.

2.4. Local Matrix Feature Based Kernel Joint Sparse Representation

2.4.1. Joint Sparse Representation (JSR)

For a testing pixel \mathbf{h} , all pixels in a spatial $w_2 \times w_2$ neighborhood centered at \mathbf{h} form a matrix $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_T]$ ($T = w_2^2$). In the joint sparse representation (JSR) model [19], all neighboring pixels are assumed to be similar and can be simultaneously represented by a common dictionary as:

$$\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_T] = [\mathbf{X}\beta_1, \dots, \mathbf{X}\beta_T] = \mathbf{X}\mathbf{B}, \tag{12}$$

where $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_M]$ is a dictionary matrix consisting of all training pixels, and $\mathbf{B} = [\beta_1, \dots, \beta_T]$ is a coefficient matrix. Assume the coefficient matrix \mathbf{B} is row-sparse;

the simultaneous orthogonal matching pursuit (SOMP) algorithm can be used to solve it as [19]:

$$\hat{\mathbf{B}} = \arg \min_{\mathbf{B}} \|\mathbf{H} - \mathbf{X}\mathbf{B}\|_F^2, \quad \text{s.t. } \|\mathbf{B}\|_{row,0} \leq K, \tag{13}$$

where $\|\mathbf{B}\|_{row,0}$ refers to the number of non-zero rows of \mathbf{B} , and K is a parameter to reflect the sparsity level.

2.4.2. Kernel Joint Sparse Representation (KJSR)

It is clear that Equation (12) only performs linear representations for neighboring pixels. To measure nonlinear relations between neighboring pixels and the training dictionary, the kernel method is used [33]. A feature map ϕ is used to project all pixels onto feature space, and the kernel-based JSR (KJSR) model is:

$$\mathbf{H}_\phi = [\phi(\mathbf{h}_1), \dots, \phi(\mathbf{h}_T)] = [\mathbf{X}_\phi \boldsymbol{\beta}_1, \dots, \mathbf{X}_\phi \boldsymbol{\beta}_T] = \mathbf{X}_\phi \mathbf{B}, \tag{14}$$

where $\mathbf{X}_\phi = [\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_M)]$. Similar to the JSR, the matrix \mathbf{B} can be solved by the following optimization problem [33]:

$$\hat{\mathbf{B}} = \arg \min_{\mathbf{B}} \|\mathbf{H}_\phi - \mathbf{X}_\phi \mathbf{B}\|_F^2, \quad \text{s.t. } \|\mathbf{B}\|_{row,0} \leq K. \tag{15}$$

2.4.3. Local Matrix Feature Based Kernel Joint Sparse Representation (LMFKJSR)

In the JSR-based model, the local neighborhood should be first constructed for the joint representation. Here, we use the same strategy as shown in Section 2.2 to construct a spatial local neighborhood [27]. That is, in a $w_2 \times w_2$ window centered at a testing pixel \mathbf{h} , m_2 most similar pixels (i.e., $\mathbf{h}_1, \dots, \mathbf{h}_{m_2}$) are picked to form local neighborhood pixel set. Then, we can generate the local matrix feature for these neighboring pixels as $\mathbf{L}_{\mathbf{h}_k}$ ($k = 1, \dots, m_2$). For training pixels $\mathbf{x}_1, \dots, \mathbf{x}_M$, the corresponding local matrix features are $\mathbf{L}_{\mathbf{x}_i}$ ($i = 1, \dots, M$).

By performing the KJSR on the local matrix features, one can generate:

$$\mathbf{H}_\Phi = [\Phi(\mathbf{L}_{\mathbf{h}_1}), \dots, \Phi(\mathbf{L}_{\mathbf{h}_{m_2}})] = [\mathbf{X}_\Phi \boldsymbol{\beta}_1, \dots, \mathbf{X}_\Phi \boldsymbol{\beta}_{m_2}] = \mathbf{X}_\Phi \mathbf{B}, \tag{16}$$

where $\Phi(\mathbf{L}_{\mathbf{h}_k}) = \mathbf{X}_\Phi \boldsymbol{\beta}_k$ ($k = 1, \dots, m_2$), $\mathbf{X}_\Phi = [\Phi(\mathbf{L}_{\mathbf{x}_1}), \dots, \Phi(\mathbf{L}_{\mathbf{x}_M})]$ is the feature representation of training set, and $\mathbf{B} = [\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_{m_2}]$ is the sparse representation coefficient.

The row-sparse matrix \mathbf{B} can be obtained by solving the following problem:

$$\hat{\mathbf{B}} = \arg \min_{\mathbf{B}} \|\mathbf{H}_\Phi - \mathbf{X}_\Phi \mathbf{B}\|_F^2, \quad \text{s.t. } \|\mathbf{B}\|_{row,0} \leq K. \tag{17}$$

For solving the problem Equation (17), a key step is the computation the correlation between $\Phi(\mathbf{L}_{\mathbf{x}_i})$ and $\Phi(\mathbf{L}_{\mathbf{h}_j})$,

$$\langle \Phi(\mathbf{L}_{\mathbf{x}_i}), \Phi(\mathbf{L}_{\mathbf{h}_j}) \rangle = \kappa(\mathbf{L}_{\mathbf{x}_i}, \mathbf{L}_{\mathbf{h}_j}) = \text{tr}(\mathbf{L}_{\mathbf{x}_i} \cdot \mathbf{L}_{\mathbf{h}_j}), \tag{18}$$

where tr is the matrix trace operator, and the linear kernel is used.

Denote $\mathbf{K}_{\mathbf{X},\mathbf{H}} \in \mathbb{R}^{M \times m_2}$ as the kernel matrix between the training samples and the neighboring pixels whose (i, j) -th entry is $\kappa(\mathbf{L}_{\mathbf{x}_i}, \mathbf{L}_{\mathbf{h}_j})$, and $\mathbf{K}_{\mathbf{X},\mathbf{X}} \in \mathbb{R}^{M \times M}$ as the kernel matrix for training samples with (i, j) -th entry $\kappa(\mathbf{L}_{\mathbf{x}_i}, \mathbf{L}_{\mathbf{x}_j})$. The sparse coefficient matrix $\hat{\mathbf{B}}$

can be solved by the kernel-based SOMP algorithm [33]. Then, the reconstruction residual of the c -th class can be computed:

$$\begin{aligned} r^c(\mathbf{h}) &= \|\mathbf{H}_\Phi - (\mathbf{X}_\Phi)_{:\Omega_c} \widehat{\mathbf{B}}_{\Omega_c} \|_F = \sqrt{\sum_{k=1}^{m_2} \|\Phi(\mathbf{L}_{\mathbf{h}_k}) - (\mathbf{X}_\Phi)_{:\Omega_c} \widehat{\mathbf{B}}_{\Omega_c,k} \|_F^2} \\ &= \sqrt{\sum_{k=1}^{m_2} \left(\kappa(\mathbf{L}_{\mathbf{h}_k}, \mathbf{L}_{\mathbf{h}_k}) - 2\widehat{\mathbf{B}}_{\Omega_c,k}^T (\mathbf{K}_{\mathbf{X},\mathbf{Z}})_{\Omega_c,k} + \widehat{\mathbf{B}}_{\Omega_c,k}^T (\mathbf{K}_{\mathbf{X},\mathbf{X}})_{\Omega_c,\Omega_c} \widehat{\mathbf{B}}_{\Omega_c,k} \right)}, \quad (19) \end{aligned}$$

where Ω_c is the index set of selected atoms.

Based on the reconstruction residuals, the testing pixel \mathbf{h} can be classified into the class with the minimal residual as:

$$\text{Class}(\mathbf{h}) = \arg \min_{c=1,\dots,C} r^c(\mathbf{h}). \quad (20)$$

The pseudo-code of LMFJKSR is shown in Algorithm 1.

Algorithm 1 LMFJKSR.

Input: Dictionary $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]$, parameter K

Output: The label of all testing pixels.

1. Reduce the dimensionality by the MNF
2. Compute the matrix features
3. Compute the training kernel $\mathbf{K}_{\mathbf{X},\mathbf{X}}$
4. Perform KJSR for each testing pixel \mathbf{h} :
 - 4.1. Construct local neighborhood $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_{m_2}]$
 - 4.2. Construct local matrix features $\mathbf{L}_{\mathbf{h}_k} (k = 1, \dots, m_2)$
 - 4.3. Compute kernels $\mathbf{K}_{\mathbf{H},\mathbf{H}}$ and $\mathbf{K}_{\mathbf{X},\mathbf{H}}$
 - 4.4. Solve the coefficient matrix:

$$\widehat{\mathbf{B}} = \arg \min_{\mathbf{B}} \|\mathbf{H}_\Phi - \mathbf{X}_\Phi \mathbf{B}\|_F^2, \quad \text{s.t. } \|\mathbf{B}\|_{\text{row},0} \leq K$$

- 4.5. Compute the class reconstruction residual:

$$r^c(\mathbf{h}) = \|\mathbf{H}_\Phi - (\mathbf{X}_\Phi)_{:\Omega_c} \widehat{\mathbf{B}}_{\Omega_c} \|_F$$

- 4.6. Classify the testing pixel \mathbf{h} :

$$\text{Class}(\mathbf{h}) = \arg \min_{c=1,\dots,C} r^c(\mathbf{h})$$

3. Experiments

3.1. Data Sets

- (1) Indian Pines (IP) (<ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/92AV3C.tif.zip>, accessed on 8 October 2015): These data have a size of 145×145 pixels and 220 spectral bands. After 20 bad bands are removed, the remaining 200 bands are used. The IP data contain 16 land cover classes. The false color composite image and ground-truth map are shown in Figure 2.
- (2) University of Pavia (UP) (https://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes, accessed on 8 June 2013): The UP data has the size of 610×340 pixels and 115 spectral bands. After 12 bad bands are discarded, 103 bands are retained. The data contain nine ground-truth classes. The false color composition image and the ground-truth map are shown in Figure 3.
- (3) Salinas (SA) (https://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes, accessed on 8 June 2013): The SA data have a size of 512×217 pixels

and 204 spectral bands. The data contain 16 ground-truth classes. The false color composite image and the ground-truth map are shown in Figure 4.

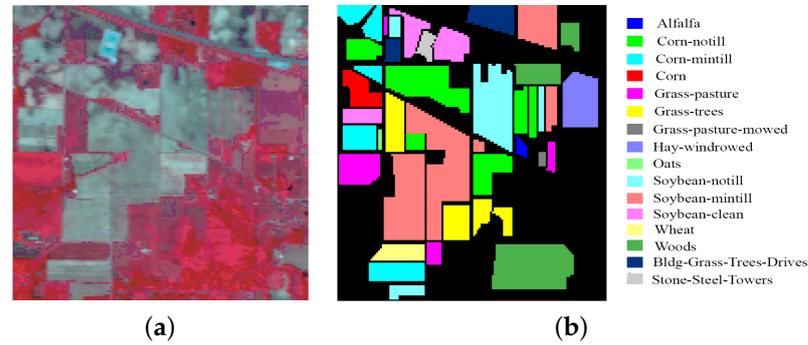


Figure 2. IP data set: (a) RGB composite image; (b) Ground-truth map.

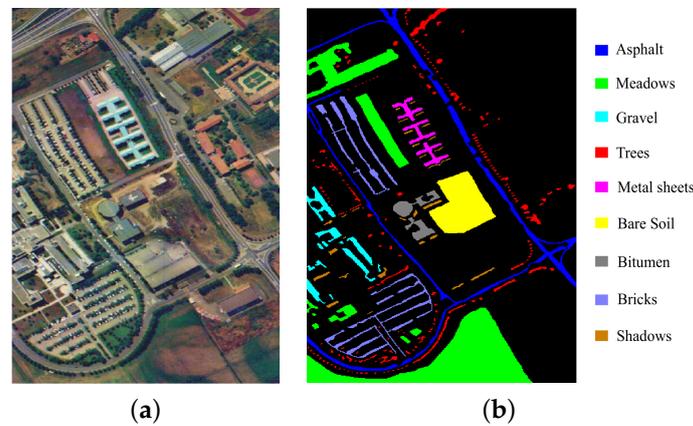


Figure 3. UP data set: (a) RGB composite image; (b) Ground-truth map.

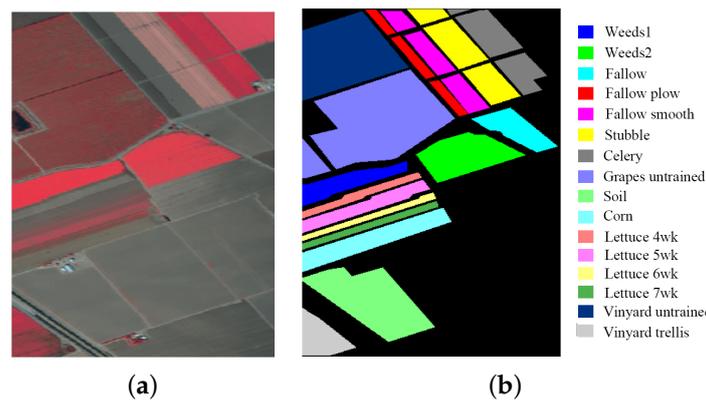


Figure 4. SA data set: (a) RGB composite image; (b) Ground-truth map.

3.2. Experimental Setting

The proposed LMFKJSR is compared with the following classification methods: SVM with composite kernel (SVM-CK) [15], local covariance matrix representation (LCMR) [7], local correntropy matrix representation (LCEM) [30], attention-based adaptive spectral-spatial kernel improved residual network (A2S2K-ResNet, A2S2K for short) [24], NoiseCNN [26], JSR [19], kernel-based JSR (KJSR) [33], self-paced KJSR (SPKJSR) [39], covariance feature-based KJSR (LogEKJSR, here called as CovKJSR) [27], correntropy feature-based KJSR (CEKJSR). It should be noted that LCMR and LCEM are SVM classifiers performed on the local covariance matrix features and local correntropy matrix features, respectively.

CovKJSR and CEKJSR are KJSR classifiers performed on local covariance and correntropy matrix features, respectively. CovKJSR and CEKJSR are special cases of the proposed LMFKJSR (CEKJSR is also our proposed method). A2S2K and NoiseCNN are recently proposed deep learning HSI classification methods. The class-specific accuracy (CA), average accuracy (AA), overall accuracy (OA), and κ coefficient on the testing set were used for comparison.

Next, in [27], in the local neighborhood construction for local feature representation and joint sparse representation, the window sizes w_1 and w_2 are set to 9, and the number of similar pixels are set as $m_1 = 70$ and $m_2 = 30$, respectively. The sparsity level in KJSR is set as $K = 40$.

3.3. Classification Results

3.3.1. Results from IP

For IP data, 1% labeled samples per class were randomly selected for training (in total, 115 training samples) and the other samples were used for testing. All methods were randomly run ten times, and the averaged classification are reported in Table 1.

Table 1. Classification results on the IP data set with 1% labeled training samples (115 training samples). The bold values indicate the highest accuracy among all methods.

Class	Train	Test	SVMCK	LCMR	LCEM	A2S2K	NoiseCNN	JSR	KJSR	SPKJSR	CovKJSR	CEKJSR	LMFKJSR
1	3	51	64.12	99.22	93.94	86.67	46.15	77.84	88.04	96.67	97.06	92.16	95.69
2	14	1420	75.45	75.68	89.66	68.12	48.86	58.40	70.57	78.11	83.68	90.80	94.05
3	8	826	61.11	57.03	71.44	88.32	37.07	40.11	58.68	66.23	69.62	76.27	83.00
4	3	231	56.02	73.85	78.92	89.26	72.09	36.36	58.70	77.19	82.90	80.65	84.11
5	5	492	70.87	71.69	83.46	86.96	58.33	68.72	75.22	77.76	80.77	83.44	84.59
6	7	740	85.54	78.12	95.28	92.08	57.58	96.95	95.18	87.58	85.20	95.42	97.43
7	3	23	93.04	100.0	100.0	49.02	8.54	58.70	85.65	100.0	100.0	100.0	100.0
8	5	484	88.59	94.36	97.12	97.42	82.99	89.67	97.91	98.31	98.51	98.12	99.79
9	3	17	89.41	100.0	100.0	34.69	22.22	31.18	51.18	90.00	100.0	100.0	100.0
10	10	958	62.46	57.39	76.81	80.52	31.46	40.45	76.82	80.04	71.36	86.10	88.80
11	25	2443	79.78	83.98	88.69	71.05	49.26	78.70	86.39	88.26	89.82	89.83	91.47
12	6	608	45.76	51.89	71.93	90.83	26.69	42.48	56.81	66.60	70.54	83.11	82.68
13	3	209	83.21	84.88	98.42	87.07	42.36	98.95	99.95	84.40	84.55	99.38	99.86
14	13	1281	92.45	94.04	94.36	86.89	79.27	98.51	99.02	98.09	98.75	95.15	96.66
15	4	376	54.39	70.45	79.34	70.70	46.51	46.52	37.26	54.76	76.99	83.22	83.80
16	3	92	96.52	93.80	95.76	75.86	47.62	94.13	87.07	84.46	94.57	99.35	99.24
Overall accuracy			74.75	76.53	86.43	78.90	51.58	69.18	79.34	82.66	84.82	89.16	91.36
Average accuracy			74.92	80.40	88.44	78.47	47.33	66.10	76.53	83.03	87.15	90.81	92.57
Coefficient κ			0.712	0.730	0.845	0.757	0.439	0.644	0.762	0.801	0.826	0.876	0.902

From the results, we can see that:

- (1) Among the three SVM-based classifiers, the matrix-feature-based classifiers (LCMR and LCEM) show much better results than the vector-feature-based SVM-CK. This demonstrates that the local covariance or correntropy matrix feature representation is more effective than the vector feature representation. In addition, due to the strong nonlinearity similarity representation ability, LCEM shows much better results than LCMR.
- (2) The recently proposed deep HSI classification methods (A2S2K and NoiseCNN) show poor results due to the limited number of training samples. In particular, for Classes 7 and 9 with only three training samples, the OA of each method is lower than 50%. To achieve satisfactory results, deep learning methods usually need a large number of training samples.
- (3) By mining nonlinear relations between pixels, KJSR improves JSR. By further selecting similar pixels in the spatial neighborhood based on self-paced learning, SPKJSR improves KJSR. By exploiting matrix representations, CovKJSR and CEKJSR improve the traditional vector-feature-based KJSRs. CEKJSR shows better results than CovKJSR.

- (4) Comparing CovKJSR with LCMR (or CEKJSR with LCEM), it can be seen that the KJSR methods show better results than SVM-based methods on these data. For IP data, there are many large homogeneous regions and region-based characteristics that can be used to improve the classification performance. Different from LCMR and LCEM, which only use the region-based matrix features, CovKJSR and CEKJSR use region-based characteristics in both the feature and classification parts. Therefore, KJSR methods show relatively better results.
- (5) By combining the local covariance and correntropy matrix features, the proposed LMFKJSR improves both methods and provides the best results. It demonstrates that the local covariance and correntropy features are complementary.
- (6) On the subclasses of “Corn” (Classes 2, 3, 4) and “Soybean” (Classes 10, 11, 12), the proposed LMFKJSR provides overall better results than the other methods (i.e., the best results on Classes 2, 10, and 11, the second best results on Classes 3 and 4). The results demonstrate that the local matrix feature representations exploiting both spectral correlation with covariance features and band similarity with correntropy features are more effective in distinguishing the subtle differences between similar materials.

The classification maps of different methods are shown in Figure 5. By seeing the highlighted elliptic and rectangle regions in the CovKJSR, CEKJSR, and LMFKJSR maps, we can find that LMFKJSR takes advantages of CovKJSR and CEKJSR. CovKJSR shows better results on the elliptic region, while CEKJSR is much better in the rectangular region. LMFKJSR shows consistently better results in both regions. In general, the classification map of LMFKJSR is more consistent with the groundtruth map.

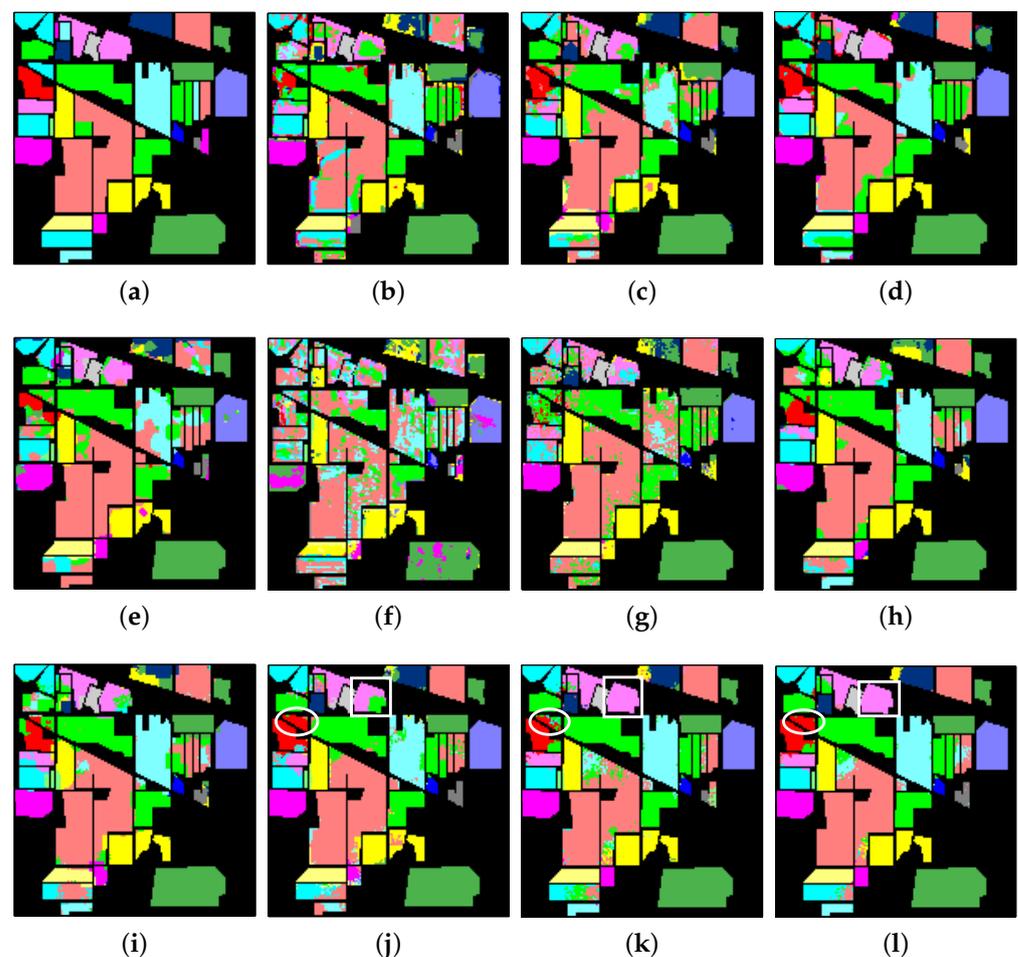


Figure 5. Classification maps on IP: (a) Groundtruth, (b) SVMCK, (c) LCMR, (d) LCEM, (e) A2S2K, (f) NoiseCNN, (g) JSR, (h) KJSR, (i) SPKJSR, (j) CovKJSR, (k) CEKJSR, (l) LMFKJSR.

3.3.2. Results from UP

Because the UP data have a large number of samples, only 0.1% labeled samples per class were randomly selected for training (in total, 50 training samples), and the other samples were used for testing. The averaged classification results over ten runs are recorded in Table 2. When there were only 50 training samples, the traditional JSR-based methods showed worse results than SVM-based methods because the dictionary representation ability is insufficient in the case of limited training samples (i.e., the number of dictionary atoms is limited). The deep learning methods produce very poor results because of the lack of training samples. By combining the local covariance and correntropy matrix features, the proposed LMFKJSR method provides the best results. Compared with the KJSR-based methods, LMFKJSR improves the OA by 10% and the κ coefficient by about 13%. Although LMFKJSR provides the best results on only two classes, it has the highest AA. This shows that LMFKJSR can generate more consistent and stable results on different classes.

Figure 6 shows the classification maps of different methods, where the proposed LMFKJSR produces a relatively better map than other methods with little “salt and pepper” noise.

Table 2. Classification results on the UP data set with 0.1% labeled training samples (50 training samples). The bold values indicate the highest accuracy among all methods.

Class	Train	Test	SVMCK	LCMR	LCEM	A2S2K	NoiseCNN	JSR	KJSR	SPKJSR	CovKJSR	CEKJSR	LMFKJSR
1	7	6624	74.54	83.47	84.67	76.78	64.24	31.01	68.13	73.91	80.91	77.72	83.38
2	19	18630	88.81	99.14	89.71	73.86	83.82	90.03	90.48	89.89	99.64	84.26	97.92
3	3	2096	63.19	66.82	68.76	56.74	29.10	67.56	65.84	67.29	51.57	73.78	79.90
4	3	3061	74.93	86.23	63.76	97.61	82.98	75.26	74.01	64.40	85.47	61.00	80.00
5	3	1342	99.61	89.52	97.76	98.82	97.27	99.69	99.84	99.33	99.30	99.52	99.57
6	5	5024	50.63	69.77	83.35	74.85	61.61	31.69	44.96	50.59	55.55	77.59	82.95
7	3	1327	79.16	77.12	96.67	73.24	13.20	80.78	66.44	74.15	74.62	100.0	99.01
8	4	3678	62.78	62.49	64.85	57.24	48.68	64.83	47.61	49.93	46.81	66.53	73.13
9	3	944	86.29	98.55	93.81	55.55	31.19	78.88	77.78	12.03	69.41	71.04	90.91
Overall accuracy			77.60	86.59	83.71	74.05	65.81	69.46	74.85	75.29	82.17	79.43	89.53
Average accuracy			75.55	81.46	82.59	73.85	56.89	68.86	73.71	64.61	73.70	79.05	87.42
Coefficient κ			0.702	0.819	0.786	0.633	0.541	0.590	0.661	0.667	0.755	0.733	0.860

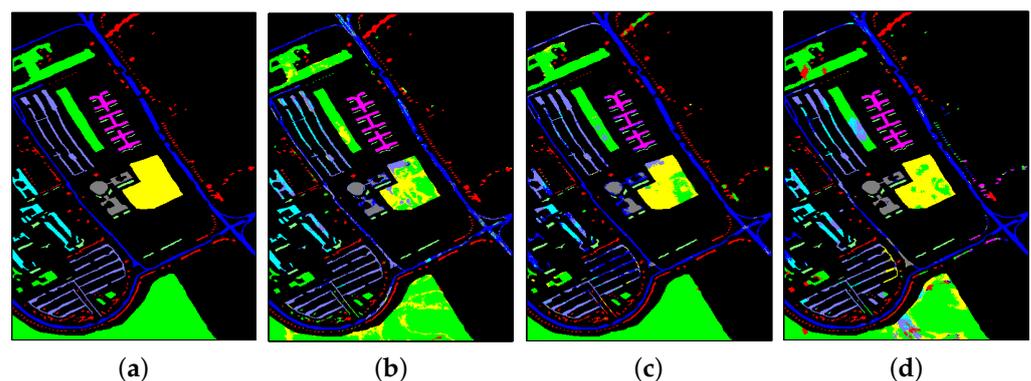


Figure 6. Cont.

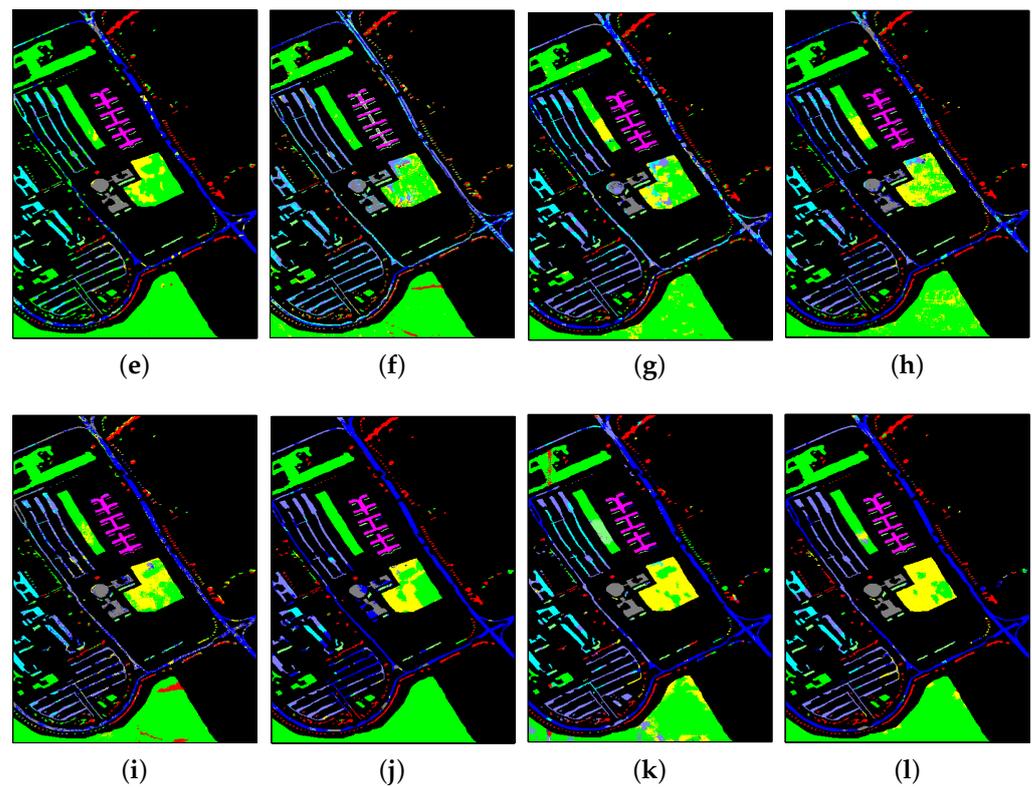


Figure 6. Classification maps on IP: Classification maps on IP: (a) Groundtruth, (b) SVMCK, (c) LCMR, (d) LCEM, (e) A2S2K, (f) NoiseCNN, (g) JSR, (h) KJSR, (i) SPKJSR, (j) CovKJSR, (k) CEKJSR, (l) LMFKJSR.

3.3.3. Results from SA

For SA data, only 0.1% labeled samples per class were randomly selected for training (totally, 66 training samples), and the other samples were used for testing. The averaged classification results over ten runs are recorded in Table 3. Except for the NoiseCNN, all methods provide OAs higher than 80%. The LMFKJSR and A2S2K provide the best and second best results. The classification of Classes 8 and 15 (i.e., “Grapes untrained” and “Vinyard untrained”) is relatively difficult for the SA data. The traditional JSR methods show poor results on Class 15, while the matrix-feature-based KJSR methods improve the traditional JSR methods by almost 20% in OA. From the classification maps in Figure 7, it can be seen that Classes 8 and 15 are located on the upper left of the image and are spatially adjacent; our proposed LMFKJSR provides better results in these two classes.

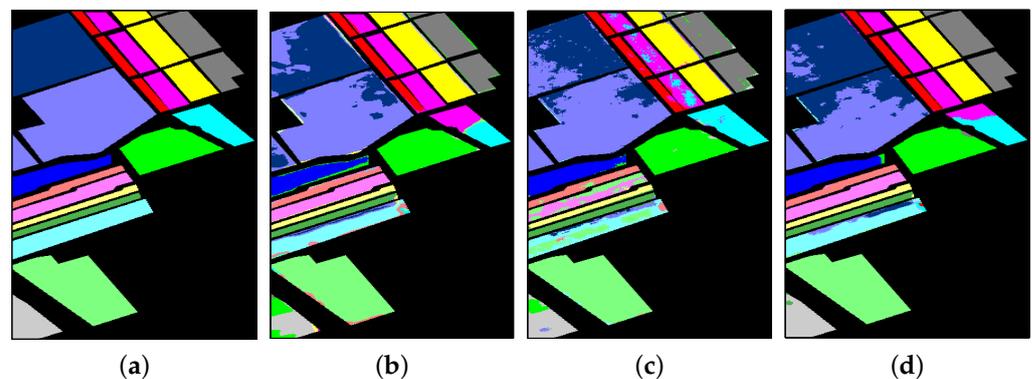


Figure 7. Cont.

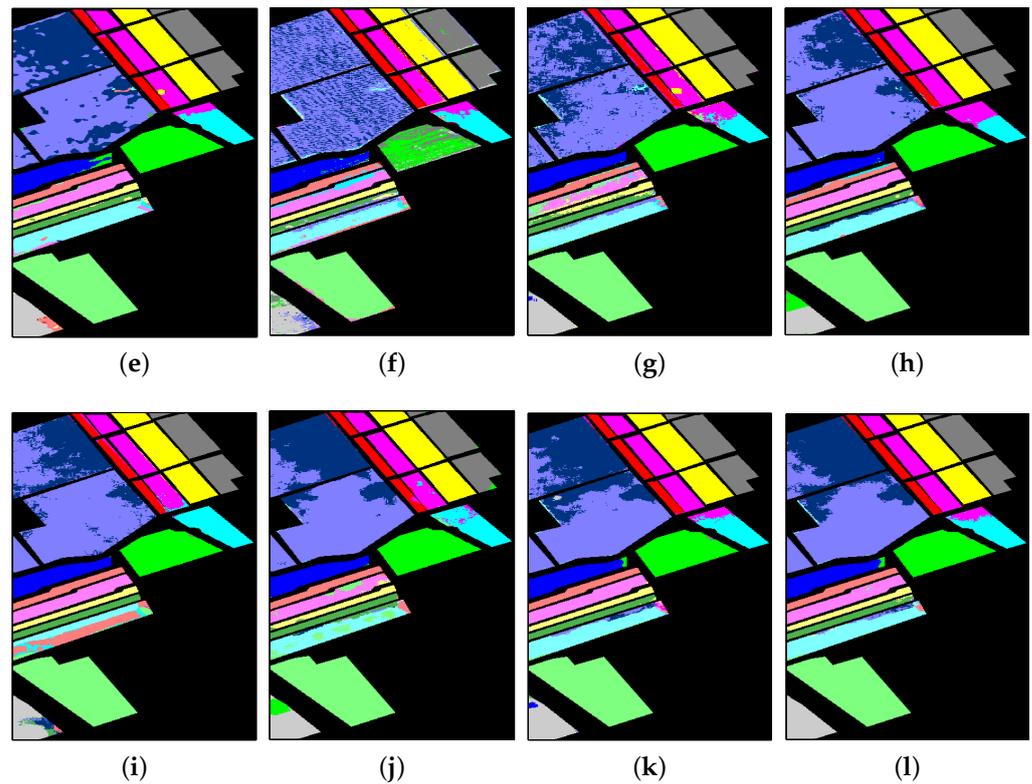


Figure 7. Classification maps on SA. Classification maps on IP: (a) Groundtruth, (b) SVMCK, (c) LCMR, (d) LCEM, (e) A2S2K, (f) NoiseCNN, (g) JSR, (h) KJSR, (i) SPKJSR, (j) CovKJSR, (k) CEKJSR, (l) LMFKJSR.

Table 3. Classification results for the Salinas data set with 0.1% labeled training samples (66 training samples). The bold values indicate the highest accuracy among all methods.

Class	Train	Test	SVMCK	LCMR	LCEM	A2S2K	NoiseCNN	JSR	KJSR	SPKJSR	CovKJSR	CEKJSR	LCKJSR
1	3	2006	89.82	98.84	99.40	99.97	88.63	99.79	99.86	99.91	99.91	99.27	99.21
2	4	3722	94.34	90.74	98.76	98.32	92.16	99.01	99.81	99.27	93.59	99.24	99.86
3	3	1973	74.35	96.15	93.58	97.17	74.27	78.86	76.90	83.02	95.39	95.97	95.48
4	3	1391	92.58	93.90	97.28	91.40	83.05	98.07	94.28	86.24	97.04	98.27	98.68
5	3	2675	79.97	64.85	99.19	86.71	84.22	85.76	94.05	95.90	73.79	99.27	98.37
6	4	3955	96.93	95.63	99.90	99.80	99.18	99.87	100.0	99.86	99.81	99.90	99.89
7	4	3575	94.95	92.58	100.0	99.50	73.89	99.93	99.68	100.0	98.88	99.99	99.94
8	11	11260	81.95	87.48	77.89	81.61	54.79	90.76	87.07	89.96	88.09	82.24	89.09
9	6	6197	94.07	98.48	99.91	98.92	99.11	99.95	99.78	99.97	99.85	99.97	100.0
10	3	3275	52.67	53.95	72.23	97.70	85.30	92.48	59.95	63.76	64.65	70.56	70.36
11	3	1065	90.01	80.45	96.85	90.78	65.57	94.42	87.19	99.59	98.57	99.25	99.66
12	3	1924	85.47	71.83	91.49	95.89	81.43	79.07	97.42	94.45	83.85	94.10	93.79
13	3	913	97.02	94.66	99.12	93.40	88.25	79.89	99.67	96.90	95.49	97.83	99.19
14	3	1067	93.87	85.81	97.23	92.33	98.02	95.49	98.86	99.76	87.67	97.96	95.91
15	7	7261	78.02	65.75	78.89	78.57	37.78	54.31	53.66	52.04	73.56	74.73	73.13
16	3	1804	66.46	79.00	92.21	97.96	69.14	99.05	81.01	85.10	88.87	87.88	88.65
Overall accuracy			84.29	83.84	89.71	90.95	73.27	83.64	86.30	87.27	88.44	90.10	91.29
Average accuracy			85.16	84.38	93.37	93.75	79.67	84.74	89.32	90.36	89.94	93.53	93.83
Coefficient κ			0.825	0.819	0.885	0.899	0.701	0.817	0.847	0.858	0.871	0.890	0.903

4. Discussions

4.1. The Effect of the Number of Training Samples

Here, the effect of the number of training samples on different methods is analyzed. For IP, the ratios of labeled samples per class are set as 1%, 2%, 3%, 4%, and 5%. As UP and SA have more labeled samples than IP, and the ratios of labeled samples per class are relatively smaller, i.e., 0.1%, 0.2%, 0.3%, 0.4%, and 0.5%. The OAs of different methods such as the changes in the ratio of training samples per class are shown in Figure 8. It can be seen that the proposed LMFKJSR shows consistently better results than other methods in different numbers of training samples.

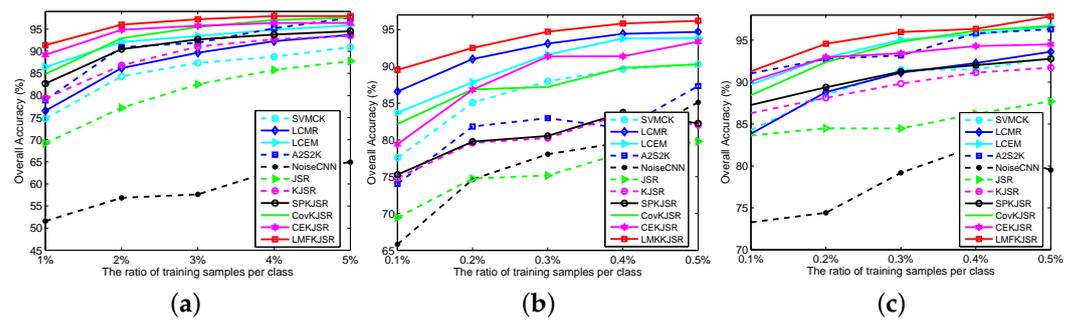


Figure 8. The OA versus the ratio of training samples per class: (a) IP; (b) UP; (c) SA.

4.2. The Effect of the MNF Dimension

Figure 9 shows the OA of LMFKJSR versus the MNF dimension d . It can be clearly seen that the OA dramatically increases as the of the number of dimension d increases and becomes stable when the dimension d is larger than 25. In the experiments, the dimension d is set as 25.

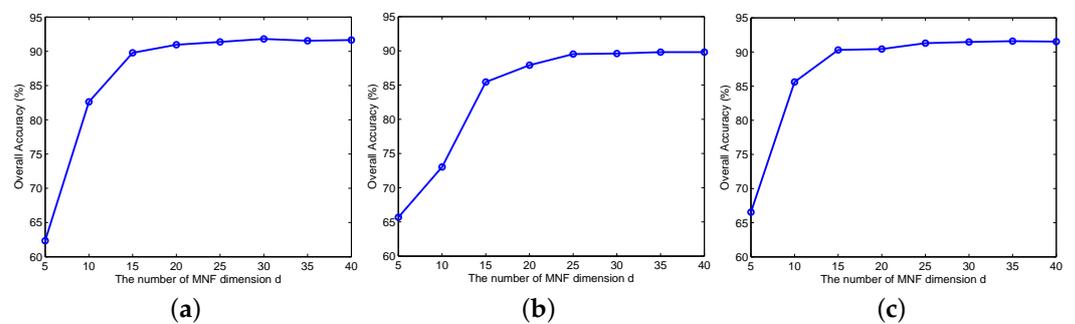


Figure 9. The OA versus the MNF dimension d : (a) IP; (b) UP; (c) SA.

4.3. The Effect of the Weighting Coefficient

The proposed LMFKJSR exploits the local matrix feature, which is a combination of local covariance matrix feature and local correntropy matrix feature. The combination coefficient μ in Equation (11) measures the importance of local covariance and correntropy features. Figure 10 shows the effect of parameter μ on the LMFKJSR, where μ changes from 0 to 1 with an increment 0.1. The best μ values for three data sets are 0.9, 0.7, and 0.7, respectively. It should be noted that LMFKJSR is reduced to CEKJSR and CovKJSR in the case of $\mu = 0$ and $\mu = 1$, respectively. The OA of LMFKJSR at the optimal parameter is obviously better than either CEKJSR or CovKJSR.

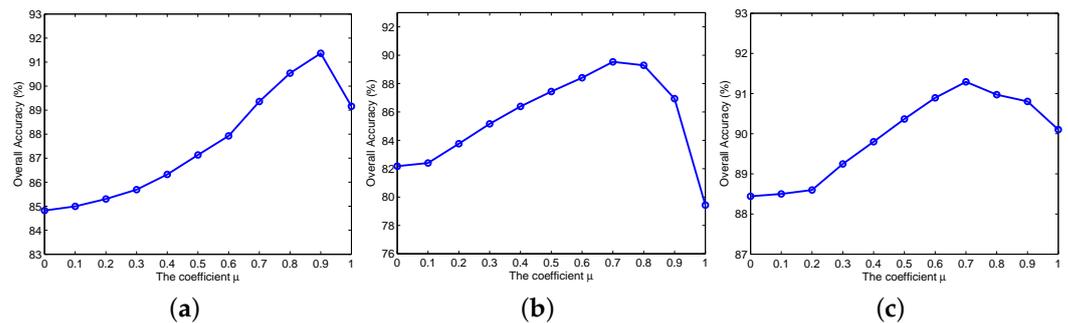


Figure 10. The OA versus the weighting coefficient μ : (a) IP; (b) UP; (c) SA.

5. Conclusions

In this paper, a local matrix-feature-based kernel joint sparse representation (LMFKJSR) model has been proposed for hyperspectral image classification. In the proposed LMFKJSR, a novel local matrix feature (LMF) is designed to reflect both the correlation between spectral pixels and the spectral bands. In detail, the local matrix feature is a linear combination of the local covariance matrix feature and the local correntropy matrix feature, where the former can describe the correlation between spectral pixels and the latter measures the similarity between spectral bands in a local spatial neighborhood. Based on the constructed LMFs, a simple linear kernel is introduced to measure the similarity between them, and a KJSR model is performed for classification. Compared with existing vector-feature-based and matrix-feature-based SVM and JSR methods, the proposed LMFKJSR shows better results on three well-known HSI data sets.

Author Contributions: Conceptualization, X.C., N.C., J.P. and W.S.; Methodology, X.C., N.C. and J.P.; Software, X.C. and J.P.; Validation, X.C. and J.P.; Formal analysis, X.C., N.C., J.P. and W.S.; Investigation, X.C. and J.P.; Resources, J.P. and W.S.; Writing—original draft preparation, X.C., N.C. and J.P.; Writing—review and editing, X.C., N.C., J.P. and W.S.; Supervision, J.P. and W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant Nos. 42171351, 42122009, 41971296, and by the Natural Science Foundation of Hubei Province under Grant 2021CFA087.

Data Availability Statement: Indian Pines data set is available at <ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/92AV3C.tif.zip>, accessed on 5 October 2015. University of Pavia and Salinas data sets are available at http://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes, accessed on 8 June 2013.

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

References

1. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P.; Nasrabadi, N.; Chanussot, J. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36. [CrossRef]
2. He, L.; Li, J.; Liu, C.; Li, S. Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1579–1597.
3. Peng, J.; Sun, W.; Li, H.; Li, W.; Meng, X.; Ge, C.; Du, Q. Low-rank and sparse representation for hyperspectral image processing: A review. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 10–43.
4. Benediktsson, J.A.; Pesaresi, M.; Arnason, K. Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1940–1949. [CrossRef]
5. Huang, X.; Zhang, L. An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 257–272. [CrossRef]
6. Zhou, Y.; Peng, J.; Chen, C.L.P. Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 1082–1095.
7. Fang, L.; He, N.; Li, S.; Plaza, A.; Plaza, J. A new spatial-spectral feature extraction method for hyperspectral images using local covariance matrix representation. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 3534–3546. [CrossRef]

8. Rasti, B.; Hong, D.; Hang, R.; Ghamisi, P.; Kang, X.; Chanussot, J.; Benediktsson, J.A. Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 60–88. [[CrossRef](#)]
9. Jia, S.; Zhan, Z.; Zhang, M.; Xu, M.; Huang, Q.; Zhou, J.; Jia, X. Multiple feature-based superpixel-level decision fusion for hyperspectral and LiDAR data classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 1437–1452.
10. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [[CrossRef](#)]
11. Camps-Valls, G.; Bruzzone, L. Kernel-based methods for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 1351–1362. [[CrossRef](#)]
12. Shen, L.; Jia, S. Three-dimensional Gabor wavelets for pixel-based hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 5039–5046. [[CrossRef](#)]
13. Li, W.; Chen, C.; Su, H.; Du, Q. Local binary patterns and extreme learning machine for hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3681–3693. [[CrossRef](#)]
14. Borzov, S.; Potaturkin, O. Increasing the classification efficiency of hyperspectral images due to multi-scale spatial processing. *Comput. Opt.* **2020**, *44*, 937–943.
15. Camps-Valls, G.; Gomez-Chova, L.; Munoz-Mare, J.; Vila-Frances, J.; Calpe-Maravilla, J. Composite kernels for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2006**, *3*, 93–97. [[CrossRef](#)]
16. Peng, J.; Zhou, Y.; Chen, C.L.P. Region-kernel-based support vector machines for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4810–4824.
17. Zhou, Y.; Peng, J.; Chen, C.L.P. Extreme learning machine with composite kernels for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2351–2360.
18. Gu, Y.; Chanussot, J.; Jia, X.; Benediktsson, J.A. Multiple kernel learning for hyperspectral image classification: A review. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 6547–6565. [[CrossRef](#)]
19. Chen, Y.; Nasrabadi, N.; Tran, T. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3973–3985.
20. Peng, J.; Du, Q. Robust joint sparse representation based on maximum correntropy criterion for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 7152–7164.
21. Peng, J.; Sun, W.; Du, Q. Self-paced joint sparse representation for the classification of hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1183–1194. [[CrossRef](#)]
22. Li, W.; Du, Q. Joint within-class collaborative representation for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2200–2208. [[CrossRef](#)]
23. Zhang, L.; Zhang, L.; Du, B. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40. [[CrossRef](#)]
24. Roy, S.; Manna, S.; Song, T.; Bruzzone, L. Attention-based adaptive spectral–spatial kernel ResNet for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 7831–7843. [[CrossRef](#)]
25. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [[CrossRef](#)]
26. Gong, Z.; Zhong, P.; Qi, J.; Hu, P. A CNN with noise inclined module and denoise framework for hyperspectral image classification. *arXiv* **2022**, arXiv:2205.12459.
27. Yang, W.; Peng, J.; Sun, W.; Du, Q. Log-Euclidean kernel-based joint sparse representation for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 5023–5034. [[CrossRef](#)]
28. Tabia, H.; Laga, H. Covariance-based descriptors for efficient 3D shape matching, retrieval, and classification. *IEEE Trans. Multimed.* **2015**, *17*, 1591–1603. [[CrossRef](#)]
29. Tuzel, O.; Porikli, F.; Meer, P. Region covariance: A fast descriptor for detection and classification. In Proceedings of the European Conference on Computer Vision (ECCV), Graz, Austria, 7–13 May 2006; pp. 589–600.
30. Zhang, X.; Wei, Y.; Cao, W.; Yao, H.; Peng, J.; Zhou, Y. Local correntropy matrix representation for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5525813. [[CrossRef](#)]
31. Liu, W.; Pokharel, P.P.; Principe, J.C. Correntropy: Properties and applications in non-Gaussian signal processing. *IEEE Trans. Signal Process.* **2007**, *55*, 5286–5298. [[CrossRef](#)]
32. Peng, J.; Li, L.; Tang, Y. Maximum likelihood estimation based joint sparse representation for the classification of hyperspectral remote sensing images. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 1790–1802. [[CrossRef](#)] [[PubMed](#)]
33. Chen, Y.; Nasrabadi, N.; Tran, T. Hyperspectral image classification via kernel sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 217–231. [[CrossRef](#)]
34. Green, A.; Berman, M.; Switzer, P.; Craig, M.D. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. *IEEE Trans. Geosci. Remote Sens.* **1998**, *26*, 65–74. [[CrossRef](#)]
35. Zhang, H.; Li, J.; Huang, Y.; Zhang, L. A nonlocal weighted joint sparse representation classification method for hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2057–2066. [[CrossRef](#)]
36. Fang, L.; Li, S.; Duan, W.; Ren, J.; Benediktsson, J.A. Classification of hyperspectral images by exploiting spectral–spatial information of superpixel via multiple kernels. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 6663–6674. [[CrossRef](#)]

37. Li, P.; Wang, Q.; Zeng, H.; Zhang, L. Local Log-Euclidean multivariate Gaussian descriptor and its application to image classification. *IEEE Trans. Pattern Anal. Machine Intell.* **2017**, *39*, 803–817. [[CrossRef](#)] [[PubMed](#)]
38. Li, P.; Wang, Q.; Zuo, W.; Zhang, L. Log-Euclidean kernels for sparse representation and dictionary learning. In Proceedings of the International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013; pp. 1601–1608.
39. Hu, S.; Peng, J.; Fu, Y.; Li, L. Kernel joint sparse representation based on self-paced learning for hyperspectral image classification. *Remote Sens.* **2019**, *11*, 1114. [[CrossRef](#)]