Neurocomputing 178 (2016) 71–86

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Weighted multifeature hyperspectral image classification via kernel joint sparse representation

Erlei Zhang [a], Xiangrong Zhang [a,b,*], Licheng Jiao [a], Hongying Liu [a], Shuang Wang [a], Biao Hou [a]

[a] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, PR China
[b] Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 31 January 2015
Received in revised form
19 May 2015
Accepted 5 July 2015
Communicated by: Chennai Guest Editor
Available online 6 November 2015

Keywords:
Hyperspectral imagery classification
kernel trick
joint sparse representation
feature extraction

ABSTRACT

The advantage of using multifeature information for classification has been widely recognized. Representation-based methods with multifeature combination learning have only recently attracted increasing attention for hyperspectral classification. However, nonlinearity in data and the computational load of processing multifeature information and contextual information have been two thorny issues. In this paper, we present a fast joint sparse representation model with multifeature combination learning and its kernel extensions for hyperspectral imagery classification. For several complementary features (spectral, shape, and texture), the proposed model simultaneously acquires a representation vector for each type of feature and encourages the representation vectors to share a common sparsity pattern by imposing the joint sparsity $\ell_{row,0}$-norm regularization. Thus, the cross-feature information can be taken into account. For different features, different weights are assigned since they may not contribute equally to the final decision. Furthermore, kernel joint sparse representation model is presented to handle nonlinearity in the data. Kernel model projects the data into a high-dimensional space to improve the separability, achieving a better performance than the linear version. At the same time, we incorporate contextual neighborhood knowledge into the learned models. Experiments on several real hyperspectral images indicate that the proposed algorithms with much less memory requirements perform significantly faster than state-of-the-art algorithms, while exhibit highly competitive classification accuracy.

1. Introduction

Hyperspectral imagery (HSI), like other spectral image, is obtained by imaging spectrometer. The goal of HSI is to collect the spectrum for each pixel in the image of a scene. In HSI, every pixel is represented by hundreds of values, which correspond to different narrow wavelengths spanning the visible spectrum to the infrared one [1]. Since it is of high spectral resolution, these values offer fine spectral differences between different materials of interest. Therefore, HSI has paved the way for remote sensing applications in various fields, such as military affairs, precision agriculture [2], environmental protection [3,4], and so on. Classification of HSI is an extremely important task for image understanding.

The objective of HSI classification is to convert spectral data into meaningful information where pixels are classified into certain classes. Many spectral feature-based classifiers for HSI have been developed and applied into practice successfully [5–16]. Notably, in the supervised classification case, the class label of each pixel can be determined by a given training set from each class. For instance, some methods are based on the support vector machine (SVM) [9–11], and some on multinomial logistic regression [12,13]. There are also a few methods based on nature-inspired computation [14–16]. Although these classifiers can make full advantages of the spectral information of HSI, obtained classification maps often appear noisy. Therefore, the spatial context information or the advanced feature should be taken into consideration. Recently, numerous works have been made to combine the spatial information of HSI with spectral information [17–21]. All these methods are based on the assumption that pixels within a local region with similar spectral characteristics usually represent the same material. [29] ingeniously utilize the non-local similarity over space and the global correlation across

* Corresponding author. Tel.: +86 029 88202279.
E-mail addresses: zhangerlei0123@163.com (E. Zhang), xrzhang@mail.xidian.edu.cn (X. Zhang), lchjiao@mail.xidian.edu.cn (L. Jiao), hyliu@xidian.edu.cn (H. Liu), shwang@mail.xidian.edu.cn (S. Wang), avcodec@163.com (B. Hou).

http://dx.doi.org/10.1016/j.neucom.2015.07.114
0925-2312/

spectrum under tensor dictionary learning framework. Furthermore, HSI classification works [22–28] also focus on feature extraction or feature reduction approaches for getting effective feature representation (e.g., differential morphological profiles (DMP) [25], gray-level co-occurrence matrix (GLCM) [26], and three dimensional wavelet transform (3D WT) [27]). An overview of both conventional and advanced feature reduction or feature extraction methods can be found in [28].

Recently, sparse representation techniques have been widely used in pattern recognition and computer vision areas [29–34]. Although pixels in HSI are high dimensionality signals, those in the common class usually lie in a low-dimensional subspace, which is spanned by the same class atoms among one dictionary (training samples). Based on this observation, sparse representation-based approaches have been used to HSI classification [27,35–37]. In classification, a test pixel can be approximately represented by a few atoms from the training dictionary. The recovered sparse coefficients determine the class label of the tested pixels by the minimal reconstruction error. In addition, collaborative representation-based classifier (CRC) also can be used for HSI classification [38,39].

Although the above representation-based methods have good performance, they are based on one single type of feature. It is obvious that one type of feature can only depict the HSI from one perspective and different feature descriptors have different discriminative power [40,41,45]. Therefore, some representation-based methods have been extended to multitask learning (MTL) [46–51]. The MTL framework has been proved to be beneficial by a good deal of works in theory [42–44]. The basic idea of representation-based multitask learning is to find out several training samples that are most correlated to the testing sample from different representation tasks. Since different tasks (e.g., each feature type is one task) may support different sparse representation coefficients, the constraint of joint sparsity across different tasks provides additional useful information to solve the classification problem. Thus, the joint sparsity constraint may enforce the coefficient estimation more robust. As is shown in [47,48], the authors built multitask joint sparse representation models by extending the sparsity-inducing ℓ_1-norm from single task learning to MTL via $\ell_{1,2}$ regularization term. However, time-consumption of joint sparsity constraint with $\ell_{1,2}$-norm is still a problem. In [50], the authors proposed a joint CR model with multitask learning for HSI classification. However, multitask learning and the extended contextual information highly increases the computational load. When dealing with a large-scale training sample set as the dictionary, CR-based MTL (CRC-MTL and JCRC-MTL), utilizing the matrix inverse operation, require more computational time and memory cost. Thus, even though these methods achieve state-of-the-art performance, it is necessary to develop a fast method. In addition, most MTL methods assign average weights to each task, which may restrain the overall performance because they have unequal contributions to the final decision.

Furthermore, if the classes are linearly inseparable in the feature space, or the features are encoded as similarity or kernel matrices, the kernel techniques [7,19,20] are good solutions for HSI classification. Kernel methods map the data from the original space to another higher dimensional space in which the high-order structure of the given data can be captured. Thus, they often show a significant improvement.

Motivated by the multitask joint representation models [47,48,50] and the kernel trick [37], in this paper, we propose a weighted multifeature joint sparse representation classifier (WMF-JSRC) for HSI classification. The proposed method not only can utilize the information across multifeature representation tasks by joint sparsity penalty, but also can obtain significant gains in speed with much less memory requirements. Once getting several complementary features (spectral, shape and texture), the proposed method simultaneously acquires a representation vector for each type of feature and imposes the joint sparsity $\ell_{row,0}$-norm regularization on the representation coefficients. The regularization encourages the coefficients to share a common sparsity pattern, which can preserve the cross-feature information. In this process, appropriate weights are assigned to different feature representation tasks. Furthermore, kernel extension (WKMF-JSRC) is proposed to solve the nonlinearity in hyperspectral image. The proposed method uses the kernel trick to map the data into a nonlinear feature space in which the data become more separable. At the same time, contextual neighborhood knowledge is integrated in the multifeature joint sparse representation framework by utilizing neighborhood pixels around the unlabeled pixel. The formulated objective consists of a squared reconstruction error term and a sparse $\ell_{row,0}$-norm regularization term, which is a NP-hard problem. In this paper, we adopt an algorithm based on simultaneous orthogonal matching pursuit (SOMP) with strong convergence guarantee to solve above problems. Another advantage of SOMP is that it can avoid the large matrix inverse operation because of parts of dictionaries selected for computation. That is why our methods have less computational complexity and memory requirements. The validity of the proposed algorithms is confirmed by experiments on several hyperspectral images.

Compared with the preliminary work [56], the new weighted multifeature sparse representation methods, presented in this paper, take the contributions of different feature representation tasks to the final decision into consideration, and kernel extension of the algorithm is employed to solve nonlinear cases in hyperspectral image. Furthermore, extensive theoretical analyses and experimental evaluations are presented here.

The paper is organized as follows. The related works are introduced in Section 2. The details of the weighted multifeature joint sparsity representation model are described in Section 3. And, kernel-view extensions are described in Section 4. Then, the efficiency and effectiveness of the proposed methods are demonstrated in Section 5 by experimental results on several real hyperspectral images. Finally, Section 6 makes a summarization and some closing remarks.

2. Related work

We first introduce some necessary notions and preliminaries as follows. Suppose a classification problem includes C distinct classes, and the c-th class has N_c training samples which form the subdictionary $\boldsymbol{D}_c \in \Re^{b\times N_c}$. Let $\boldsymbol{D}$ be the dictionary whose columns (atoms) are extracted from the training samples $\boldsymbol{X}_{train}$, i.e., $\boldsymbol{D}=[\boldsymbol{D}_1,\cdots,\boldsymbol{D}_c,\cdots,\boldsymbol{D}_C]\in\Re^{b\times N}(N=\sum_{c=1}^{C}N_c)$. Let vector $\boldsymbol{x}\in\Re^b$ denote a b dimensional pixel. $\boldsymbol{\alpha}\in\Re^N$ is sparse representation coefficient vector for a testing sample $\boldsymbol{x}$ in reconstruction. In the multifeature case, we suppose that each pixel has S different kinds of features. $\boldsymbol{x}^s\in\Re^{b^s}(s=1,\cdots,S)$ represents the s-th type of feature vector of the unlabeled pixel $\boldsymbol{x}$. $\boldsymbol{D}^s=[\boldsymbol{D}_1^s,\cdots,\boldsymbol{D}_c^s,\cdots,\boldsymbol{D}_C^s]$ is the dictionary of the s-th type of feature ($\boldsymbol{D}_c^s\in\Re^{b^s\times N_c}$ is the c-th subdictionary; b^s is the dimension of the s-th feature vector). $\boldsymbol{\alpha}^s\in\Re^N$ is the coefficient vector of $\boldsymbol{x}^s$ over $\boldsymbol{D}^s$.

2.1. Pixelwise sparse representation model

Sparse representation classifier (SRC) [34] was proposed for face recognition. It uses a given dictionary (all the training data) to approximately represent the test sample through sparse coding framework. In other words, a query pixel in HSI can be

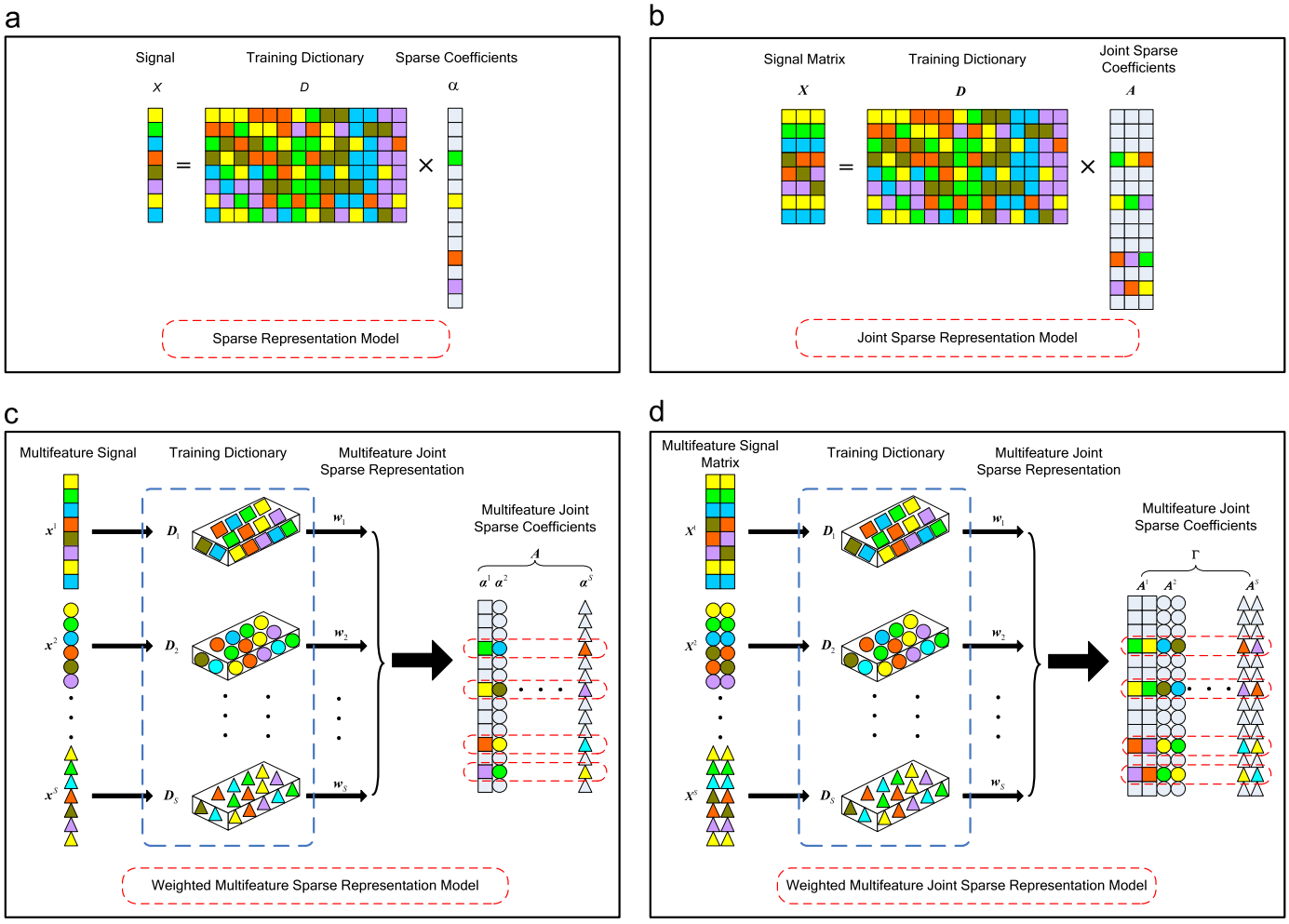


Fig. 1. Illustration of different sparse models. Each algorithm represents the test data by a sparse linear combination of training data. Each patch (square, circle and triangular) in the column is a coefficient value, and each column represents a sparse coefficient vector corresponding to one pixel, in which white blocks denote zero values, whereas color blocks stand for nonzero values. (a) Pixelwise SRC [see formula (2)]. (b) JSRC [see formula (5)]. (c) WMF-SRC [see formula (7)]. (d) WMF-JSRC [see formula (10)].

approximated by a combination of a few atoms from the given dictionary. The overview of pixelwise sparse representation model is shown in Fig. 1(a). For dictionary $\boldsymbol{D}$, the sparse representation coefficient $\boldsymbol{\alpha}$ satisfying $D\boldsymbol{\alpha}=\boldsymbol{x}$ is obtained by solving the following optimization problem:

$$\hat{\boldsymbol{\alpha}}=\arg\min \|\boldsymbol{\alpha}\|_0 \quad s.t. \quad \boldsymbol{D\alpha}=\boldsymbol{x} \tag{1}$$

In order to meet the real data requirements of measurement errors or noise in the sensing process, the above problem can be transformed into minimizing the approximation error within a certain sparsity level:

$$\hat{\boldsymbol{\alpha}}=\arg\min \|\boldsymbol{D\alpha}-\boldsymbol{x}\|_2 \quad s.t. \quad \|\boldsymbol{\alpha}\|_0\le K \tag{2}$$

where K is a predefined upper bound on the sparsity level, representing the maximum number of the nonzero coefficients in $\hat{\boldsymbol{\alpha}}$. The class label of pixel $\boldsymbol{x}$ is decided by the minimal representation error (between $\boldsymbol{x}$ and its approximation from the sub-dictionary D_i of each class) when $\hat{\boldsymbol{\alpha}}$ is obtained, i.e.,

$$\begin{aligned} label(x)&=\arg\min_{i=1,\cdots,C} r_i(x)\\ &=\arg\min_{i=1,\cdots,C}\|x-D_i\hat{\boldsymbol{\alpha}}_i\|_2 \end{aligned} \tag{3}$$

where $\hat{\boldsymbol{\alpha}}_i$ contains the coefficients in $\hat{\boldsymbol{\alpha}}$ belonging to i-th class.

2.2. Joint sparse representation model

In HSI, neighboring pixels usually are strongly correlated with each other. It means they probably belong to the same material. In [35], the joint sparse representation classifier (JSRC) captures such spatial correlations by assuming that neighboring pixels within a region of fixed size can be jointly represented by a few common atoms from a structural dictionary. Fig. 1(b) shows the overview of joint sparse representation model. Specifically, the size of a region centered at test pixel $\boldsymbol{x}_t$ is denoted by $l\times l$, and pixels within such a region are denoted by $\{\boldsymbol{x}_i\}, i=1,\cdots,l\times l$. These pixels can also be stacked into a matrix $X=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_t,\cdots,\boldsymbol{x}_{l\times l}]\in\Re^{b\times l^2}$. The matrix can be compactly represented as

$$\begin{aligned} \boldsymbol{X}&=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_t,\cdots,\boldsymbol{x}_{l\times l}]=[\boldsymbol{D\alpha}_1,\cdots,\boldsymbol{D\alpha}_t,\cdots,\boldsymbol{D\alpha}_{l\times l}]\\ &=\boldsymbol{D}[\boldsymbol{\alpha}_1,\cdots,\boldsymbol{\alpha}_t,\cdots,\boldsymbol{\alpha}_{l\times l}]=\boldsymbol{DA} \end{aligned} \tag{4}$$

where $\boldsymbol{A}=[\boldsymbol{\alpha}_1,\cdots,\boldsymbol{\alpha}_t,\cdots,\boldsymbol{\alpha}_{l\times l}]\in\Re^{N\times l^2}$ is the sparse coefficients matrix corresponding to $\boldsymbol{X}$. Due to the indexes of the selected atoms in $\boldsymbol{D}$ are determined by the positions of nonzero coefficients in $[\boldsymbol{\alpha}_1,\cdots,\boldsymbol{\alpha}_t,\cdots,\boldsymbol{\alpha}_{l\times l}]$, by enforcing a few nonzero rows on the sparse coefficients matrix $\boldsymbol{A}$, neighboring pixels $\boldsymbol{X}$ can be represented by a small set of common atoms. Then, matrix $\boldsymbol{A}$ can be obtained by solving the following optimization problem:

$$\hat{\boldsymbol{A}}=\arg\min_{\boldsymbol{A}}\|\boldsymbol{X}-\boldsymbol{DA}\|_F \quad s.t. \quad \|\boldsymbol{A}\|_{\text{row},0}\le K \tag{5}$$

Table 1
Parameters for features.

Datasets	Features	Parameters	Num. of Dimensions
Indian Pines/Pavia University/Pavia Center images	GLCM	Base image: PC1, PC2, PC3, PC4; Measure: contrast, entropy, variance, correlation, angular second moment; Window: 3,7,11; Direction: averaging the extracted features over four directions.	60
	DMP	Base image: PC1, PC2; Size of structuring elements: 3,5,7,9; Morphological operators: opening and closing.	16
	3D WT	Base image:PC1, PC2, PC3, PC4; Level: 2.	60

where $||\boldsymbol{A}||_{\text{row},0}$ denotes the joint sparse norm, which is used to select a number of the most representative nonzero rows in $\boldsymbol{A}$, and $\|\bullet\|_F$ is the Frobenius norm. After $\hat{\boldsymbol{A}}$ is recovered, the label of the test pixel $\boldsymbol{x}_t$ can be decided by the minimal total error, i.e.,

$$\begin{aligned} label(\boldsymbol{x}_t) &= \arg\min_{i=1,\cdots,C} r_i(\boldsymbol{X}) \\ &= \arg\min_{i=1,\cdots,C} \|\boldsymbol{X} - \boldsymbol{D}_i\hat{\boldsymbol{A}}_i\|_2 \end{aligned} \tag{6}$$

where $\hat{\boldsymbol{A}}_i$ denotes the rows in $\hat{\boldsymbol{A}}$ associated with the i-th class.

2.3. Multiple feature extraction

Since hyperspectral images usually have complex contents, the combination of multiple kinds of features would be helpful for the classification task. An overview of feature extraction methods can be found in [28]. Four types of features, namely, original spectral value feature, GLCM [26], DMP [25], and 3D WT [27], are used in our experiments. Each type of feature of a pixel is denoted as a vector. The spectral feature of a pixel is obtained by arranging the value of all bands. The parameters for different kinds of features are set to be the same as the corresponding references and listed in Table 1.

3. Weighted multifeature joint sparse representation model

Compared with SRC model, the JSRC incorporating contextual knowledge of local regions can deliver much better classification results in terms of accuracy. However, single spectral information (or the single type of feature) greatly affects the classification performance. In addition, simple spatial structure information makes a limited contribution to classification in the JSRC. Therefore, we propose a weighted multifeature joint sparse representation classifier (WMF-JSRC) for HSI classification as follows.

3.1. Weighted multifeature pixelwise sparse representation model (WMF-SRC)

The overview of WMF-SRC model is shown in Fig. 1(c). For one unlabeled hyperspectral pixel $\boldsymbol{x}$, multiple different types of features $\{\boldsymbol{x}^s\}_{s=1,\cdots,S}$ are extracted from the different perspectives. Since each kind of feature of the same training samples constructs corresponding dictionary $\{\boldsymbol{D}^s\}_{s=1,\cdots,S}$ respectively, it is reasonable that the representation coefficients $\{\boldsymbol{\alpha}^s\}_{s=1,\cdots,S}$ of these features over their associated dictionaries should be similar. Here, we assume that the positions of nonzero coefficients tend to be the same, while the representation coefficients need not to be identical because of differences among various kinds of features. Thus, they can preserve additional useful information for classification. Under this assumption, WMF-SRC can be formulated as

$$\hat{\boldsymbol{A}} = \arg\min \textstyle\sum_{s=1}^{S} w^s \|\boldsymbol{x}^s - \boldsymbol{D}^s\boldsymbol{\alpha}^s\|_2 \quad s.t. \quad \|\boldsymbol{A}\|_{\text{row},0} \le K \tag{7}$$

where $\boldsymbol{A} = [\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \cdots, \boldsymbol{\alpha}^S]$, and w^s is the weight assigned to the s-th feature. In order to balance the similarity and the diversity of the different features in the linear representation, weights are used to constrain the representation error. Once $\hat{\boldsymbol{A}}$ is obtained, the label of $\boldsymbol{x}$ can be determined as follows

$$\begin{aligned} label(\boldsymbol{x}) &= \arg\min_{i=1,\cdots,C} r_i(\boldsymbol{x}) \\ &= \arg\min_{i=1,\cdots,C} \sum_{s=1}^{S} w^s \|\boldsymbol{x}^s - \boldsymbol{D}_i^s\hat{\boldsymbol{\alpha}}_i^s\|_2 \end{aligned} \tag{8}$$

where $\hat{\boldsymbol{\alpha}}_i^s$ is the subset of the coefficient vector $\hat{\boldsymbol{\alpha}}^s$ associated with class i.

3.2. Weighted multifeature joint sparse representation model (WMF-JSRC)

The WMF-SRC model (proposed in Section 3.1) is a pixelwise classification method. Since HSI pixels in a small spatial neighborhood tend to be highly correlated and share many similarities, neighboring pixels are always assumed consisting of similar materials. Based on the framework of JSRC (described in Section 2.2), these hyperspectral pixels in a neighborhood can be simultaneously approximated by a linear combination of common training pixels. Similarly, in multifeature cases, these neighboring pixels can also be used for every kind of feature (e.g., spectral value, GLCM, DMP, and 3D WT for HSI). That is, adjacent pixels $\{\boldsymbol{x}_t^s\}_{t=1,\cdots,L}$ (where L is the neighborhood size) of every kind of feature are approximated by a linear combination of common atoms from the given dictionary $\boldsymbol{D}^s$. A visual illustration of the classification scheme for HSI with the proposed WMF-JSRC is shown in Fig. 1(d). The model represents the test data by a sparse linear combination of training data. Sparse coefficients of pixels in a small region from different types of features share the same sparsity pattern.

Suppose we get a matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_L]$ of size L via simultaneously stacking all the pixels in the neighborhood pixels centered at the hyperspectral pixel $\boldsymbol{x}_t$. For S different types of features, in the same way, we can construct the joint signal matrix set $\{\boldsymbol{X}^s\}_{s=1,\cdots,S} = \{[\boldsymbol{x}_1^s, \cdots, \boldsymbol{x}_t^s, \cdots, \boldsymbol{x}_L^s]\}_{s=1,\cdots,S}$ which contains S matrices sized $b^s \times L$ for each neighborhood patch. Using the joint sparse representation model, $\{\boldsymbol{X}^s\}_{s=1,\cdots,S}$ can be represented by

$$\boldsymbol{X}^s = [\boldsymbol{x}_1^s, \boldsymbol{x}_2^s, \cdots, \boldsymbol{x}_L^s] = \boldsymbol{D}^s\boldsymbol{A}^s \tag{9}$$

where $\{\boldsymbol{A}^s\}_{s=1,\cdots,S} \in \Re^{N\times L}$ is a set of the coding coefficient matrices associated with the corresponding feature dictionary $\{\boldsymbol{D}^s\}_{s=1,\cdots,S}$. Let $\boldsymbol{\Gamma} = \left[\boldsymbol{A}^1, \boldsymbol{A}^2, \cdots, \boldsymbol{A}^S\right]$ be the matrix formed by concatenation of the coefficient matrices. Then, following joint model with

multifeature learning, we can determine the row-sparse matrix $\mathbf{\Gamma}$:

$$\hat{\mathbf{\Gamma}} = \arg\min_{\Gamma} \sum_{s=1}^{S} w^s \|\boldsymbol{X}^s - \boldsymbol{D}^s \boldsymbol{A}^s\|_F \quad s.t. \quad \|\mathbf{\Gamma}\|_{\text{row},0} \le K \tag{10}$$

In formula (10), the objective function aims at minimizing the total representation error for all features; the constraint condition directs the minimization task towards the sparsest possible representation. After obtaining $\hat{\mathbf{\Gamma}}$, we can determine the label of the center pixel $\boldsymbol{x}_t$ by computing the total residual errors between $\boldsymbol{X}^s$ and the approximations obtained over their corresponding subdictionaries $\{\boldsymbol{D}_i^s\}_{\substack{s=1,\cdots,S \\ i=1,\cdots,C}}$ as follows:

$$\begin{aligned} label(\boldsymbol{x}_t) &= \arg\min_{i=1,\cdots,C} r_i(\boldsymbol{X}^s) \\ &= \arg\min_{i=1,\cdots,C} \sum_{s=1}^{S} w^s \|\boldsymbol{X}^s - \boldsymbol{D}_i^s \boldsymbol{A}_i^s\|_F \end{aligned} \tag{11}$$

where $\boldsymbol{A}_i^s$ is the subset of the coefficient matrix $\boldsymbol{A}^s$ associated with class i.

3.3. Weights and optimization algorithm

In WMF-JSRC algorithm, the influence of a specific feature to a certain task can be obtained from prior knowledge. Thus, the weights $\boldsymbol{w} = [w^1, w^2, \cdots, w^S]$ can be pre-learned by using a training dataset [51]. In this paper, the weights in the testing phase could be simply determined by Fisher discrimination criterion [57].

$$\begin{aligned} \mathbf{P}^s &= \mathbf{P}_{BC}^s(\boldsymbol{X}_{train}^s) / \mathbf{P}_{WC}^s(\boldsymbol{X}_{train}^s) \\ w^s &= \mathbf{P}^s / \sum_{s=1}^{S} \mathbf{P}^s \end{aligned} \tag{12}$$

where $\mathbf{P}_{BC}^s(\boldsymbol{X}_{train}^s)$ is the between-class scatter of the training data $\boldsymbol{X}_{train}^s$, and $\mathbf{P}_{WC}^s$ is the within-class scatter. Detailed research and computational process of Fisher criterion can be found in [57].

The aforementioned problems (2) and (5) are NP-hard, but they can be approximately solved by greedy pursuit algorithms [52,53], such OMP and SOMP [54]. While problems (7) and (10) are more complex which request a simultaneous sparse approximation of several input signals over multiple dictionaries (features), we cannot use OMP or SOMP algorithm to solve them directly. In this paper, we make some changes based on SOMP to solve the problems. In greedy pursuit process, the support of the solution is sequentially updated as well as SOMP. The major difference is the selected atom that simultaneously yields the best approximation to all of the residuals of S feature dictionaries. The implementation details for WMF-JSRC are summarized in Algorithm 1. Besides optimization with strong convergence guarantee, it can avoid the large matrix inverse operation because it augments the support set by choosing one atom at each time until the approximation error decreases to the preset threshold or K atoms are selected. That means only parts of dictionaries are selected for computation.

Algorithm 1. : Weighted multifeature joint sparse representation model

Input: multifeature test matrix $\left\{\boldsymbol{X}^s \in \Re^{b^s \times L}\right\}_{s=1,\cdots,S}$, structural dictionary $\left\{\boldsymbol{D}^s \in \Re^{b^s \times N}\right\}_{s=1,\cdots,S}$, number of classes C, sparsity level K, pre-learned weight vector for every feature $\boldsymbol{w}$.

Output: joint sparse representation coefficients matrix $\hat{\mathbf{\Gamma}}$.

Initialization: set iteration counter $iter = 1$; index set $\Lambda_0 = \varnothing$; residual matrix $\{\boldsymbol{R}_0^s\}_{s=1,\cdots,S} = \{\boldsymbol{X}^s\}_{s=1,\cdots,S}$.

While stopping criterion has not been met **do**

Step 1: Compute residual correlation matrix for each type of feature: $Z(s,i) = \|(R_{iter-1}^s)^T \boldsymbol{d}_i^s\|_p$, $s = 1,\cdots,S$, $i = 1,\cdots,N$, $p \ge 1$, $\boldsymbol{d}_i^s$ is i-th atom of $\boldsymbol{D}^s$.

Step 2: Find index of the atom that best approximates all residual: $\lambda_{iter} = \arg\max_{i=1,2,\cdots,N} \|\boldsymbol{WZ}(:,i)\|_q$, $q \ge 1$.

where $\boldsymbol{W}$ is a diagonal matrix obtained from the weight vector $\boldsymbol{w}$.

Step 3: Update the index set: $\Lambda_{iter} = \Lambda_{iter-1} \cup \lambda_{iter}$.

Step 4: Estimate the sparse representation coefficients:

$$\boldsymbol{A}_{iter}^s = ((\boldsymbol{D}_{\Lambda_{iter}}^s)^T \boldsymbol{D}_{\Lambda_{iter}}^s)^{-1} (\boldsymbol{D}_{\Lambda_{iter}}^s)^T \boldsymbol{X}^s, \quad s = 1,\cdots,S.$$

Step 5: Update the residual matrix:

$$\boldsymbol{R}_{iter}^s = \boldsymbol{X}^s - \boldsymbol{D}_{\Lambda_{iter}}^s \boldsymbol{A}_{iter}^s, \quad s = 1,\cdots,S.$$

Step 6: Update joint sparse representation coefficients:

$$\hat{\mathbf{\Gamma}} = \left[\boldsymbol{A}_{iter}^1, \boldsymbol{A}_{iter}^2, \cdots, \boldsymbol{A}_{iter}^S\right], \quad s = 1,\cdots,S.$$

Step 7: $iter \leftarrow iter + 1$.

End while

4. Kernel-view extension of WMF-JSRC

If the classes may not be linearly separable in the feature space, or the features are encodes as similarity or kernel matrices, then the kernel extension of WMF-JSRC makes sense for combining multiple feature kernels. The aim of the extension is to encode the features in a Reproducing Kernel Hilbert Space (RKHS) by the kernel trick. The basic idea is to use a non-linear function ϕ to map the data from the original space to another higher dimensional RKHS in which the classes become linearly separable [55]. We define $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$ for some given kernel function κ.

4.1. Kernel-view extension of WMF-JSRC (WKMF-JSRC)

Considering the general case of S different types of features with $\{\boldsymbol{X}^s\}_{s=1,\cdots,S}$, the kernel space representation can be written as $\Phi(\boldsymbol{X}^s) = [\phi(\boldsymbol{x}_1^s), \phi(\boldsymbol{x}_2^s), \cdots, \phi(\boldsymbol{x}_L^s)]$. Similarly, the dictionary of training samples for s-th type of feature can be represented in kernel space as $\Phi(\boldsymbol{D}^s)$. In this new space, we can write the problem (10) as:

$$\hat{\mathbf{\Gamma}} = \arg\min_{\Gamma} \sum_{s=1}^{S} w^s \|\Phi(\boldsymbol{X}^s) - \Phi(\boldsymbol{D}^s)\boldsymbol{A}^s\|_F \quad s.t. \quad \|\mathbf{\Gamma}\|_{\text{row},0} \le K \tag{13}$$

where $\mathbf{\Gamma} = \left[\boldsymbol{A}^1, \boldsymbol{A}^2, \cdots, \boldsymbol{A}^S\right]$. It is clear that the information from all kinds of features is integrated via the shared sparsity pattern of the matrix $\mathbf{\Gamma}$. This can be reformulated in terms of kernel matrices as

$$\begin{aligned} \hat{\mathbf{\Gamma}} = \arg\min_{\Gamma} \sum_{s=1}^{S} w^s \left(\text{trace}(\boldsymbol{A}^{s^T} \mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s} \boldsymbol{A}^s) - 2\text{trace}(\boldsymbol{A}^{s^T} \mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s})\right) \\ s.t. \quad \|\mathbf{\Gamma}\|_{\text{row},0} \le K \end{aligned} \tag{14}$$

where $\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s}$ is the kernel matrix whose (i,j)-th entry is $\kappa(\boldsymbol{d}_i^s, \boldsymbol{d}_j^s)$, and $\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s}$ is the matrix whose (i,j)-th entry is $\kappa(\boldsymbol{d}_i^s, \boldsymbol{x}_j^s)$. Once $\mathbf{\Gamma}$ is obtained, classification can be done by assigning the class label as

$$\begin{aligned} label(\boldsymbol{x}) &= \arg\min_{i=1,\cdots,C} r_i(\boldsymbol{X}^s) \\ &= \arg\min_{i=1,\cdots,C} \sum_{s=1}^{S} w^s \|\Phi(\boldsymbol{X}^s) - \Phi(\boldsymbol{D}_i^s)\boldsymbol{A}_i^s\|_F \end{aligned} \tag{15}$$

or in terms of kernel matrices as

$$\begin{aligned} label(\boldsymbol{x}) = \arg\min_{i=1,\cdots,C} \sum_{s=1}^{S} w^s (\text{trace}(\mathbf{K}_{\boldsymbol{X}^s,\boldsymbol{X}^s}) - 2\text{trace}(\boldsymbol{A}_i^{s^T} \mathbf{K}_{\boldsymbol{D}_i^s,\boldsymbol{X}^s}) \\ + \text{trace}(\boldsymbol{A}_i^{s^T} \mathbf{K}_{\boldsymbol{D}_i^s,\boldsymbol{D}_i^s} \boldsymbol{A}_i^s)) \end{aligned} \tag{16}$$

Here, $\boldsymbol{D}_i^s$ is the subdictionary associated with the i-th class, and $\boldsymbol{A}_i^s$ is the subset of the coefficient matrix $\boldsymbol{A}^s$ associated with class i.

4.2. Optimization algorithm

Similarly to the linear fusion method, we apply the kernelized SOMP method (denoted as KSOMP) to efficiently solve the problem for kernel extension. The implementation details of KSOMP are summarized in Algorithm 2. In KSOMP, firstly the correlation (dot product) between a pixel $\phi(\boldsymbol{x}^s)$ and a dictionary atom $\phi(\boldsymbol{d}_i^s)$ is computed by $\kappa(\boldsymbol{d}_i^s, \boldsymbol{x}^s) = \langle \phi(\boldsymbol{d}_i^s), \phi(\boldsymbol{x}^s) \rangle$. Let the kernel matrix $\mathbf{K}_{\boldsymbol{A},\boldsymbol{B}}$ denote $\mathbf{K}_{\boldsymbol{A},\boldsymbol{B}}(i,j) = \langle \phi(\boldsymbol{a}_i), \phi(\boldsymbol{b}_i) \rangle$, where $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are i-th and j-th columns of matrix $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. Thus, the correlation between $\boldsymbol{X}^s$ and $\boldsymbol{D}^s$ is computed by $\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s}$. Secondly, the orthogonal projection coefficients of $\Phi(\boldsymbol{X}^s)$ onto a set of selected dictionary atoms $\{\phi(\boldsymbol{d}_i^s)\}_{i \in \boldsymbol{\Lambda}}$ is given as

$$\boldsymbol{A}_{\boldsymbol{\Lambda}}^s = \left((\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s})_{\boldsymbol{\Lambda},\boldsymbol{\Lambda}} \right)^{-1} (\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s})_{\boldsymbol{\Lambda},:} \tag{17}$$

where $\boldsymbol{\Lambda}$ is the index set of selected dictionary atoms. When computing the projection $\boldsymbol{A}_{\boldsymbol{\Lambda}}^s$, a regularization term $\mu \boldsymbol{I}$ is added in order to have a stable inversion.

$$\boldsymbol{A}_{\boldsymbol{\Lambda}}^s = \left((\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s})_{\boldsymbol{\Lambda},\boldsymbol{\Lambda}} + \mu \boldsymbol{I} \right)^{-1} (\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s})_{\boldsymbol{\Lambda},:} \tag{18}$$

where $\boldsymbol{I}$ is an identity matrix whose dimensionality should be clear from the context, μ is a small scalar and is chosen as $\mu = 10^{-5}$ in our implementation. This parameter does not seriously affect the classification performance because the matrix is usually invertible and regularization is not really needed.

Thirdly, the residual matrix between $\Phi(\boldsymbol{X}^s)$ and its approximation using the selected atoms $\{\phi(\boldsymbol{d}_i^s)\}_{i \in \boldsymbol{\Lambda}} = \Phi(\boldsymbol{D}^s)_{:,\boldsymbol{\Lambda}}$ is then expressed as

$$\begin{aligned} \Phi(\boldsymbol{R}^s) &= \Phi(\boldsymbol{X}^s) - \Phi(\boldsymbol{D}^s)_{:,\boldsymbol{\Lambda}} \left((\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s})_{\boldsymbol{\Lambda},\boldsymbol{\Lambda}} + \mu \boldsymbol{I} \right)^{-1} (\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s})_{\boldsymbol{\Lambda},:} \\ &= \Phi(\boldsymbol{X}^s) - \Phi(\boldsymbol{D}^s)_{:,\boldsymbol{\Lambda}} \boldsymbol{A}_{\boldsymbol{\Lambda}}^s \end{aligned} \tag{19}$$

Note that the residual matrix $\Phi(\boldsymbol{R}^s)$ in (19) cannot be calculated directly. However, the correlation between $\Phi(\boldsymbol{R}^s)$ and dictionary $\Phi(\boldsymbol{D}^s)$ can be computed by

$$\begin{aligned} \boldsymbol{U}^s &= \langle \Phi(\boldsymbol{D}^s), \Phi(\boldsymbol{R}^s) \rangle \\ &= \mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s} - (\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s})_{:,\boldsymbol{\Lambda}} \left((\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s})_{\boldsymbol{\Lambda},\boldsymbol{\Lambda}} + \mu \boldsymbol{I} \right)^{-1} (\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s})_{\boldsymbol{\Lambda},:} \\ &= \mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s} - (\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s})_{:,\boldsymbol{\Lambda}} \boldsymbol{A}_{\boldsymbol{\Lambda}}^s \end{aligned} \tag{20}$$

4.3. Computational complexity analysis

Suppose that multifeature test matrix is $\left\{ \boldsymbol{X}^s \in \Re^{b^s \times L} \right\}_{s=1,\cdots,S}$, structural dictionary is $\left\{ \boldsymbol{D}^s \in \Re^{b^s \times N} \right\}_{s=1,\cdots,S}$, and sparsity level is K. Multifeature learning and the extension of contextual information inevitably require much more computational load, especially for large-scale dictionary cases. Most of the computational costs in sparse recovery are the inversion of a matrix. In Algorithm 1, the support set is sequentially augmented by one index at a time, thus the inversion of a matrix of at most size $K \times K$ is the most intensive part. Therefore, the complexity is $\boldsymbol{O}\left(\sum_{s=1}^{S} KNLb^s \right)$. The kernelized joint sparsity model has to compute the kernel matrix $\{\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s}\}_{s=1,\cdots,S}$ and $\{\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s}\}_{s=1,\cdots,S}$. The computational complexity will therefore increase. Note that the matrix $\{\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s}\}_{s=1,\cdots,S}$ can be pre-computed, and $\{\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s}\}_{s=1,\cdots,S}$costs little because the number of neighboring pixel L is usually a small number. Therefore, the kernel trick does not significantly increase the complexity while providing significant performance improvements, as will be seen in Section 5.

Algorithm 2. : Weighted kernel multifeature joint sparse representation model

Input: multifeature test matrix $\{\boldsymbol{X}^s\}_{s=1,\cdots,S}$, structural dictionary $\{\boldsymbol{D}^s\}_{s=1,\cdots,S}$, number of classes C, sparsity level K, pre-learned weight vector for every feature $\boldsymbol{w}$.

Output: joint sparse representation coefficients matrix $\hat{\boldsymbol{\Gamma}}$.

Initialization: compute the kernel matrices $\{\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s}\}_{s=1,\cdots,S}$ and $\{\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s}\}_{s=1,\cdots,S}$, set iteration counter $iter = 1$; index set $\boldsymbol{\Lambda}_0 = \varnothing$; residual matrix $\{\boldsymbol{U}_0^s\}_{s=1,\cdots,S} = \{\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s}\}_{s=1,\cdots,S}$.

While stopping criterion has not been met **do**

Step 1: Compute residual correlation matrix for each type of feature: $\boldsymbol{Z}(i,s) = \|\boldsymbol{U}_{iter-1}^s(i,:)\|_p$, $i = 1,\cdots,N$, $s = 1,\cdots,S$, $p \geq 1$.

Step 2: Find index of the atom that best approximates all residual: $\lambda_{iter} = \arg \max_{i=1,2,\cdots,N} \|\boldsymbol{Z}(i,:)\boldsymbol{W}\|_q$, $q \geq 1$, $i = 1,\cdots,N$. $\boldsymbol{W}$ is a diagonal matrix obtained from the weight vector $\boldsymbol{w}$.

Step 3: Update the index set: $\boldsymbol{\Lambda}_{iter} = \boldsymbol{\Lambda}_{iter-1} \cup \lambda_{iter}$.

Step 4: Estimate the sparse representation coefficients:

$\boldsymbol{A}_{iter}^s = \left((\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s})_{\boldsymbol{\Lambda}_{iter},\boldsymbol{\Lambda}_{iter}} + \mu \boldsymbol{I} \right)^{-1} (\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s})_{\boldsymbol{\Lambda}_{iter}},:$

Step 5: Update the residual matrix:

$\boldsymbol{U}_{iter}^s = \mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{X}^s} - (\mathbf{K}_{\boldsymbol{D}^s,\boldsymbol{D}^s}) : ,_{\boldsymbol{\Lambda}_{iter}} \boldsymbol{A}_{iter}^s, s = 1,\cdots,S$

Step 6: Update joint sparse representation coefficients:

$\hat{\boldsymbol{\Gamma}} = [\boldsymbol{A}_{iter}^1, \boldsymbol{A}_{iter}^2, \cdots, \boldsymbol{A}_{iter}^S], s = 1,\cdots,S.$

Step 7: $iter \leftarrow iter + 1$.

End while

5. Experiments and performance comparisons

In this section, we demonstrate the effectiveness and efficiency of the proposed WMF-JSRC and WKMF-JSRC on three real hyperspectral images. The classification results are then compared visually and quantitatively to those obtained by existing methods (including CR-based and SR-based methods), which have shown good performance in HSI classification.

5.1. Classifiers for comparison and quantitative metrics

Here, the proposed WMF-JSRC and WKMF-JSRC are compared with the widely used classification methods, including classical SVM-based methods [9], CR-based methods (CRC [38], JCRC [39], CRC-MTL and JCRC-MTL [50]) and SR-based methods (SRC-Pixel-wise [34], JSRC [35], SRC-MTL [47], MF-SRC, and MF-JSRC [56]). The SVM classifier is implemented with Radial Basis Function (RBF) kernel using support vector machines library. The regularization parameters for CRC, JCRC, CRC-MTL, JCRC-MTL and SRC-MTL algorithms range from $1e^{-6}$ to $1e^{-1}$. RBF kernel is used in WKMF-JSRC, and parameter γ ranges from $1e^{-3}$ to $1e^{3}$.

In these experiments, the classification accuracy for each class, the overall accuracy (OA), average accuracy (AA), and the Kappa coefficient measure (Kappa) are adopted by different classifiers on the test set to evaluate the performance of classification. The accuracy for each class measures the percentage computed by the rate between correctly classified testing samples and the total number of testing samples of each class. The OA represents the ratio between the number of correctly classified testing samples and the number of all testing samples, and the AA is the mean of the percentage of correctly classified pixels for each class. The Kappa coefficient is a robust measure of the degree of agreement that computed by weighting the measured accuracies.

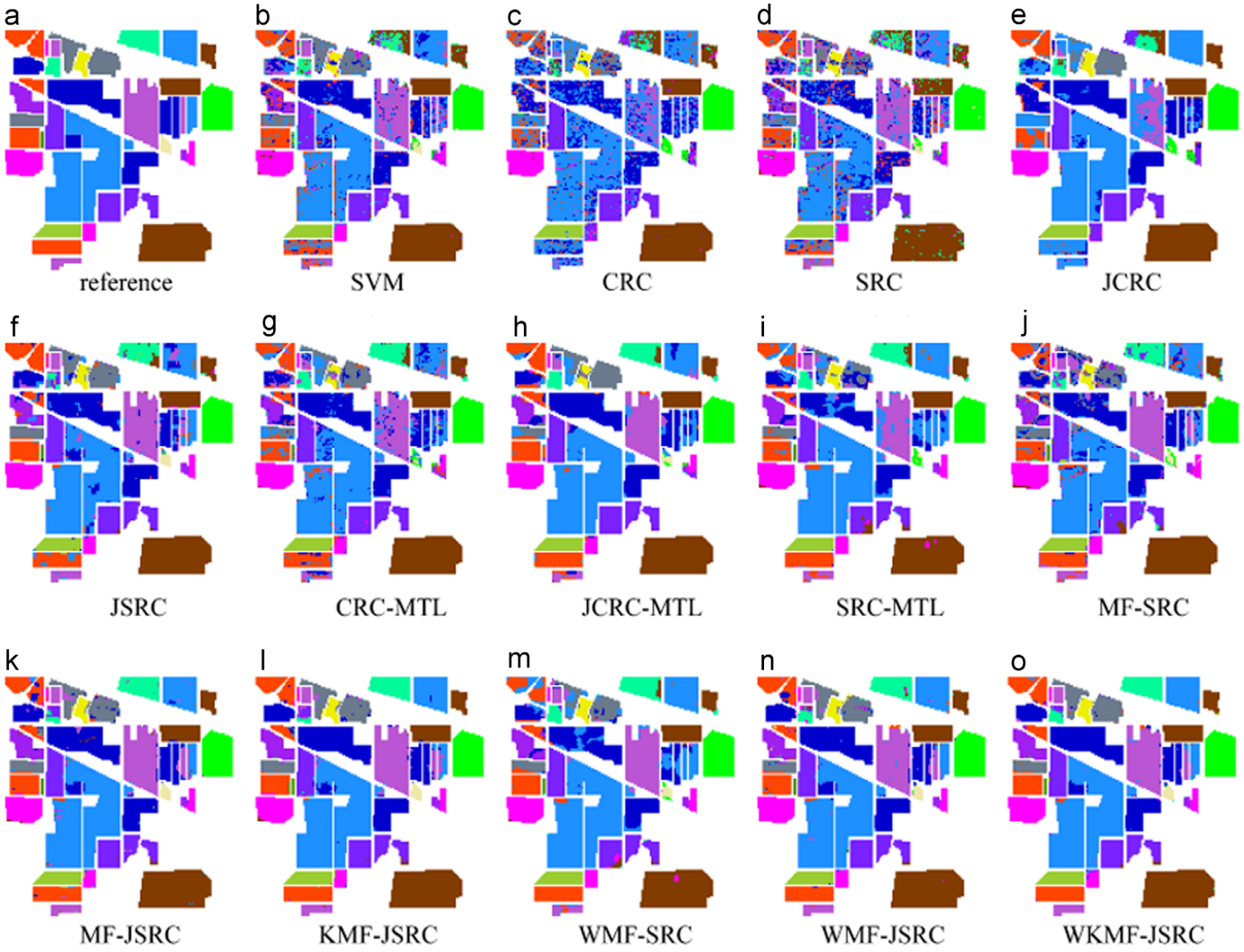


Fig. 2. Indian Pines image: (a) reference image. The classification maps obtained by the (b) SVM, (c) CRC, (d) SRC, (e) JCRC, (f) JSRC, (g) CRC-MTL, (h) JCRC-MTL, (i) SRC-MTL, (j) MF-SRC, (k) MF-JSRC, (l) KMF-JSRC, (m) WMF-SRC, (n) WMF-JSRC, and (o) WKMF-JSRC.

5.2. *Experiment results and analysis*

5.2.1. *AVIRIS data set: Indian Pines image*

The first image in our experiments was captured by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over the agricultural Indian Pines in northwestern Indiana. In the experiments, the number of available bands is 200 by removing 20 water absorption bands. The spatial resolution of Indiana Pines image is 20m per pixel, and the size is 145×145. This image contains 16 ground-truth classes (Fig. 2(a)), most of which are different types of crops (e.g., corns, soybeans, and wheat).

First, we investigate the complementary properties of the multiple features and the necessity of combining multiple features with weights. Around 5% of the labeled samples (523 samples) are chosen randomly as training set and the remaining samples for testing. The classification results of SVM, SRC and JSRC with different types of features are shown in Table 2. In Table 2, the best results for each quality index are labeled in bold.

Comparing these classification results in Table 2, it can be seen that almost all of the features can achieve the best performance on certain classes. For instance, with the JSRC method, the spectral value feature achieves the best classification results for Classes 5, 6, 7, 8, 10, and 12; the GLCM texture feature obtains the best results for Classes 7 and 9; the DMP shape feature performs the best on Classes 2 and 11; and 3D WT feature obtains the best results for Classes 1 and 9. Similar observations can be made for the other classification methods. In other words, different features are complementary to each other, and they reflect different aspects of the discriminative information of the HSI. Thus, it is reasonable to combine the multiple features to improve the performance of classification. Whereas the original spectral value feature and the DMP-feature-based classifiers lead to better overall accuracies than the others. It means that different features may not contribute equally to the final decision. It is reasonable to assign weights to each feature.

Next, we compare the classification results of the proposed WMF-JSRC and WKMF-JSRC with those obtained by other multifeature learning methods. The single-feature-based algorithms for comparison include CR-based methods (CRC and JCRC) and SR-based methods (SRC and JSRC). Each result of single-feature-based algorithm is obtained with the original spectral value feature. The multifeature learning methods include CR-based algorithms (CRC-MTL and JCRC-MTL), SRC-MTL and the proposed methods (non-weighted and weighted). The neighborhood size L for JCRC, JSRC, JCRC-MTL, MF-JSRC, KMF-JSRC, WMF-JSRC, and WKMF-JSRC is set 5×5. The classification results shown in Table 3 are averaged over 10 runs for each classifier, and the classification maps are shown in Fig. 2. Firstly, it shows that all multifeature learning methods dramatically improve the classification accuracy comparing with

Table 2
Classification accuracy (%) using single feature for the Indian Pines image on the test set.

Methods	SVM				SRC				JSRC			
Features	Spectral	GLCM	DMP	3D WT	Spectral	GLCM	DMP	3D WT	Spectral	GLCM	DMP	3D-WT
1	58.82	56.86	84.31	13.73	50.98	68.63	82.35	39.22	82.35	86.27	82.35	**92.16**
2	78.49	79.96	87.15	67.77	59.40	72.83	90.09	69.75	89.57	89.65	**94.27**	86.71
3	72.60	70.33	86.87	38.89	61.62	59.22	**89.65**	49.49	88.64	79.55	85.10	76.01
4	52.70	64.86	**94.14**	13.06	40.99	59.91	89.19	62.61	77.48	74.32	89.19	77.93
5	90.68	90.47	92.37	58.47	86.86	80.72	92.58	92.16	**96.19**	93.43	91.10	93.86
6	92.38	90.55	93.23	63.05	89.99	91.40	94.22	71.23	**96.90**	96.47	92.95	88.72
7	45.83	54.17	87.50	25.00	54.17	75.00	91.67	91.67	**100**	**100**	95.83	75.00
8	98.71	96.34	98.28	88.36	98.49	93.97	97.20	98.28	**100**	99.35	96.55	98.28
9	31.58	42.11	63.16	21.05	31.58	42.11	63.16	63.16	68.42	**94.74**	84.21	**94.74**
10	74.10	80.41	73.45	77.58	74.32	76.82	84.33	82.92	**93.47**	91.19	88.36	89.77
11	83.24	83.07	88.10	92.79	71.30	79.23	89.72	79.19	89.42	95.99	**97.23**	93.60
12	75.13	53.69	70.15	29.67	44.43	53.34	79.93	45.11	**83.53**	75.64	81.65	75.30
13	**99.50**	98.51	99.00	73.63	**99.50**	95.52	99.00	93.53	97.01	90.05	98.01	95.52
14	98.37	91.38	**98.78**	80.80	93.41	95.52	98.62	84.70	98.29	91.21	98.21	91.21
15	48.75	65.65	93.35	14.68	39.34	57.89	**94.18**	60.94	70.08	79.50	87.53	82.55
16	74.44	44.44	85.56	15.56	74.44	67.78	**88.89**	83.33	86.67	77.78	73.33	86.67
OA	82.01	80.81	88.29	67.87	72.22	77.53	90.86	74.54	90.98	90.19	**92.74**	88.54
AA	73.46	72.67	87.21	48.38	66.93	73.12	89.05	72.96	88.63	88.45	**89.74**	87.38
Kappa	79.44	78.07	86.64	61.59	68.31	74.31	89.60	70.86	89.70	88.78	**91.70**	86.89

Table 3
Classification accuracy (%) for the Indian Pines image on the test set.

Methods	Single feature methods				Multifeature methods								
								Non-Weighted			Weighted		
	CRC	JCRC	SRC	JSRC	CRC-MTL	JCRC -MTL	SRC-MTL	MF- SRC	MF- JSRC	KMF-JSRC	WMF-SRC	WMF-JSRC	WKMF-JSRC
1	0	0	50.98	82.35	13.73	56.86	21.57	54.90	**98.04**	94.12	72.55	86.27	88.24
2	65.49	81.42	59.40	89.57	83.04	84.88	64.83	82.45	95.23	96.70	72.39	**97.36**	**97.36**
3	31.44	47.73	61.62	88.64	77.78	92.68	79.92	74.75	86.74	92.80	85.10	92.93	**96.97**
4	1.35	0	40.99	77.48	62.61	74.32	86.04	72.52	93.24	93.24	90.09	91.89	**97.75**
5	71.61	67.58	86.86	96.19	91.53	94.07	98.31	94.49	94.28	94.28	**99.36**	99.15	96.61
6	93.23	**99.72**	89.99	96.90	98.31	99.44	89.84	90.83	97.32	99.01	92.52	98.31	97.18
7	0	0	54.17	**100**	41.67	37.50	**100**	95.83	91.67	**100**	**100**	**100**	91.67
8	**100**	**100**	98.49	**100**	**100**	**100**	**100**	99.14	99.78	**100**	**100**	99.78	**100**
9	0	0	31.58	68.42	57.89	0	94.74	**100**	**100**	**100**	94.74	94.74	**100**
10	31.88	39.83	74.32	93.47	68.55	80.63	75.08	88.36	**96.19**	95.43	87.16	94.45	95.32
11	79.23	93.09	71.30	89.42	85.80	93.73	95.44	86.31	95.69	95.91	97.40	97.48	**98.17**
12	34.65	54.03	44.43	83.53	61.23	72.56	49.40	54.89	83.53	87.65	67.58	85.25	**93.65**
13	98.01	99.50	99.50	97.01	99.00	99.50	99.50	99.00	99.50	**100**	**100**	99.00	99.50
14	97.64	**100**	93.41	98.29	98.78	98.70	98.29	97.64	99.27	99.76	98.37	99.67	99.59
15	26.59	45.98	39.34	70.08	83.93	90.30	84.49	80.33	83.66	86.43	91.14	88.64	**91.97**
16	82.22	91.11	74.44	86.67	56.67	61.11	91.11	82.22	90.00	88.89	94.44	93.33	**97.78**
OA	66.31	76.38	72.22	90.98	84.05	90.03	84.69	85.49	94.46	95.67	89.72	96.10	**97.27**
AA	50.83	57.50	66.93	88.63	73.78	77.27	83.03	84.60	94.01	95.26	90.18	94.89	**96.36**
Kappa	60.61	72.37	68.31	89.70	81.69	88.58	82.33	83.45	93.69	95.07	88.18	95.55	**96.89**

single-feature-based algorithms. Thus, it is reasonable and necessary to combine the multiple features.

Secondly, as shown in Table 3, the classification accuracy (JCRC, JSRC, JCRC-MTL, MF-JSRC, and WMF-JSRC) can be further improved than those methods without considering neighboring pixels (CRC, SRC, CRC-MTL, SRC-MTL, MF-SRC, and WMF-SRC). It demonstrates that contextual neighborhood knowledge is helpful to HSI classification. Finally, the proposed WMF-JSRC outperforms the other compared linear approaches in terms of OA, AA, and Kappa coefficients. Compared with JCRC-MTL and SRC-MTL, the gains (in OA, AA, and Kappa coefficients) of the WMF-JSRC are more than 6% and 11% respectively, which show the proposed WMF-JSRC successfully integrates multifeature information and contextual neighborhood information in joint sparse representation framework. Compared with MF-SRC, MF-JSRC and KMF-JSRC (without weights), WMF-SRC, WMF-JSRC and WKMF-JSRC lead to better results respectively, which validates that appropriate weights are beneficial to the final decision. In addition, it is also shown that the

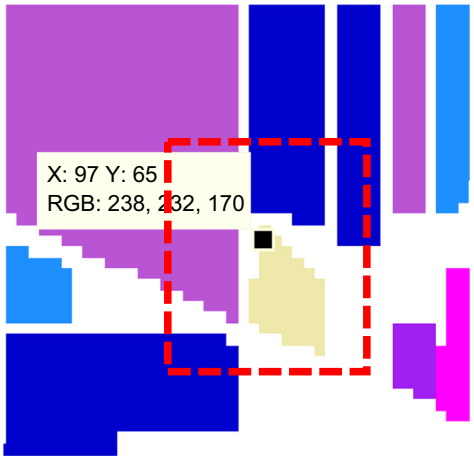


Fig. 3. The pixel is located at (97, 65) in the Indian Pines image.

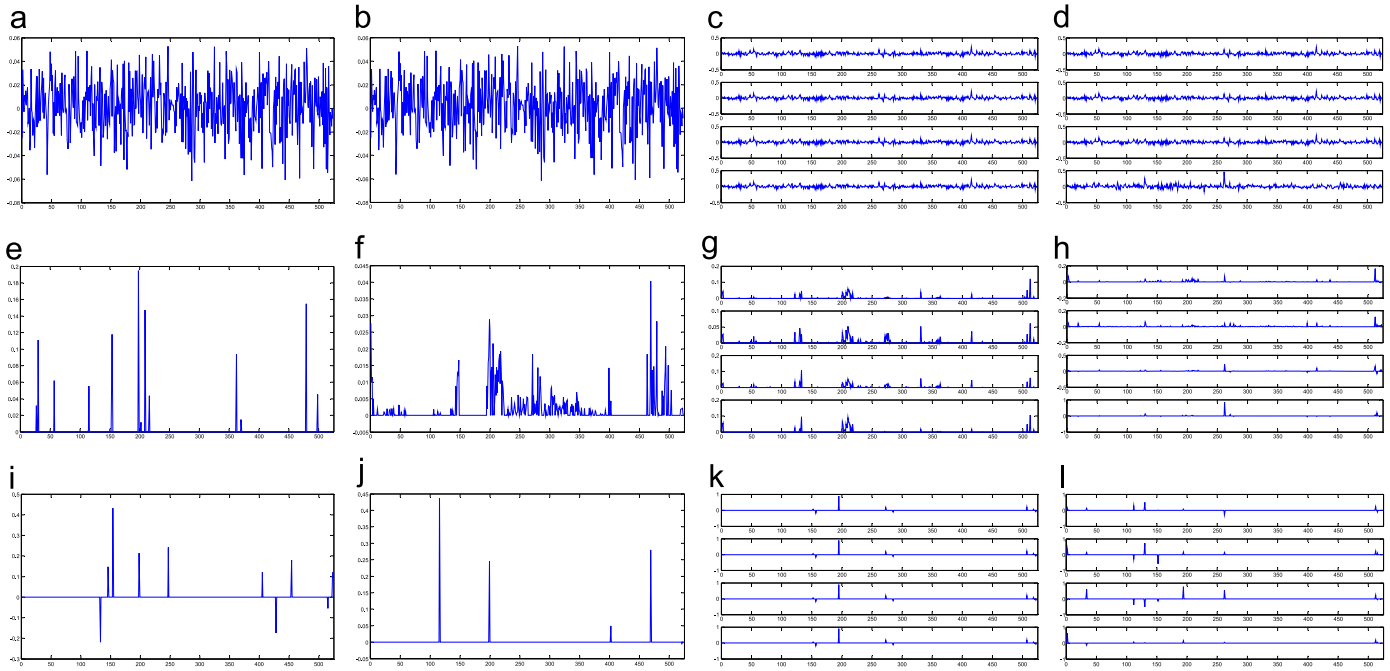


Fig. 4. Estimate reconstruction coefficients for the pixel located at (97, 65) in the Indian Pines image: (a) CRC algorithm with the ℓ_2 − normconstraint, (b) JCRC with ℓ_F − normconstraint, (c) Multitask CRC with the ℓ_2 − norm regularization, (d) Multitask joint CRC with the ℓ_F − norm regularization, (e) SRC with the ℓ_1 constraint, (f) JSRC algorithm with the $\ell_{1,2}$ − norm constraint, (g) multitask SRC with the $\ell_{1,2}$ − norm constraint, (h) multitask spatial joint sparse representation with the $\ell_{1,2}$ − norm, (i) SRC with the ℓ_0 constraint, (j) spatial JSRC algorithm with the $\ell_{row,0}$ − norm constraint, (k) multifeature SRC with the $\ell_{row,0}$ − normconstraint, and (l) multifeature joint SRC with the $\ell_{row,0}$ − norm regularization.

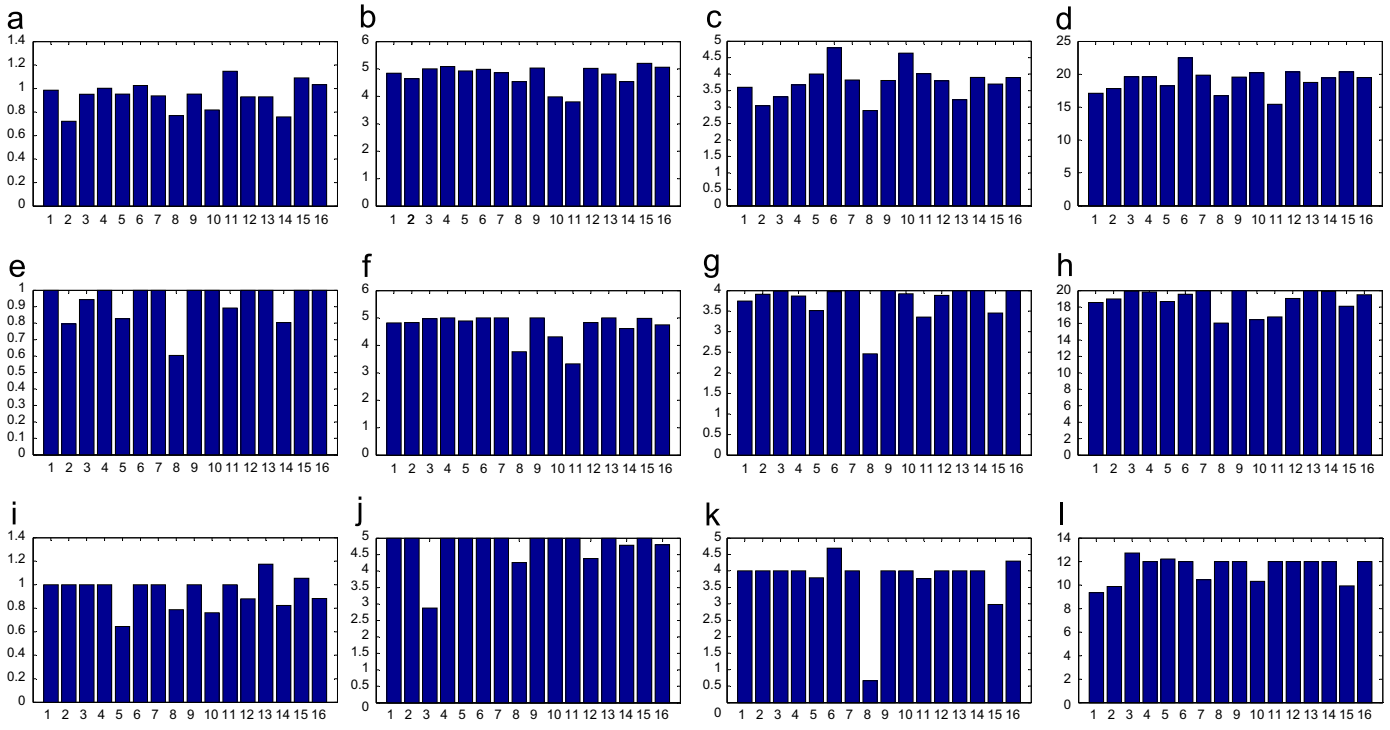


Fig. 5. The residuals for each class for the pixel located at (97, 65) in the Indian Pines image: (a) CRC algorithm with the ℓ_2 − normconstraint, (b) JCRC with ℓ_F − normconstraint, (c) Multitask CRC with the ℓ_2 − norm regularization, (d) Multitask joint CRC with the ℓ_F − norm regularization, (e) SRC with the ℓ_1 constraint, (f) JSRC algorithm with the $\ell_{1,2}$ − norm constraint, (g) multitask SRC with the $\ell_{1,2}$ − norm constraint, (h) multitask spatial joint sparse representation with the $\ell_{1,2}$ − norm, (i) SRC with the ℓ_0 constraint, (j) spatial JSRC algorithm with the $\ell_{row,0}$ − norm constraint, (k) multifeature SRC with the $\ell_{row,0}$ − normconstraint, and (l) multifeature joint SRC with the $\ell_{row,0}$ − norm regularization.

KMF-JSRC and WKMF-JSRC obtain best results for most classes in terms of OA, AA and Kappa. Thus, the kernel extensions are helpful in advancing classification. Overall, the proposed algorithms achieve a satisfying performance.

We further investigate the relationship of constraints considered by CR-based and SR-based methods. We randomly select a test pixel which belongs to Class 1 and is located at (97, 65) in the Indian Pines image. As is shown in Fig. 3, the black patch denotes

Table 4
Running time (seconds) for the classification of the Indian Pines image by the representation-based methods at different sample size.

Methods		5%	10%	15%	20%	25%	30%	35%	40%
Single feature methods	CRC	0.12	0.28	0.38	0.55	1.02	1.39	–	–
	JCRC	0.45	0.82	1.22	1.61	2.37	2.58	–	–
	SRC	0.15	0.30	0.44	0.60	1.07	1.53	1.81	2.02
	JSRC	0.49	0.81	1.22	1.76	2.55	2.93	3.24	3.85
Multiple feature methods	CRC-MTL	23.21	163.97	530.83	–	–	–	–	–
	JCRC-MTL	24.00	169.59	542.23	–	–	–	–	–
	SRC-MTL	28.24	76.47	143.09	230.61	336.10	460.41	–	–
	MF-SRC	0.37	0.64	1.34	2.06	2.50	3.08	3.50	3.88
	WMF-SRC								
	MF-JSRC	1.02	1.83	2.84	4.13	5.32	5.99	7.83	7.88
	WMF-JSRC								
	KMF-JSRC	2.43	4.69	7.26	10.25	13.39	19.06	26.47	36.83
	WKMF-JSRC								

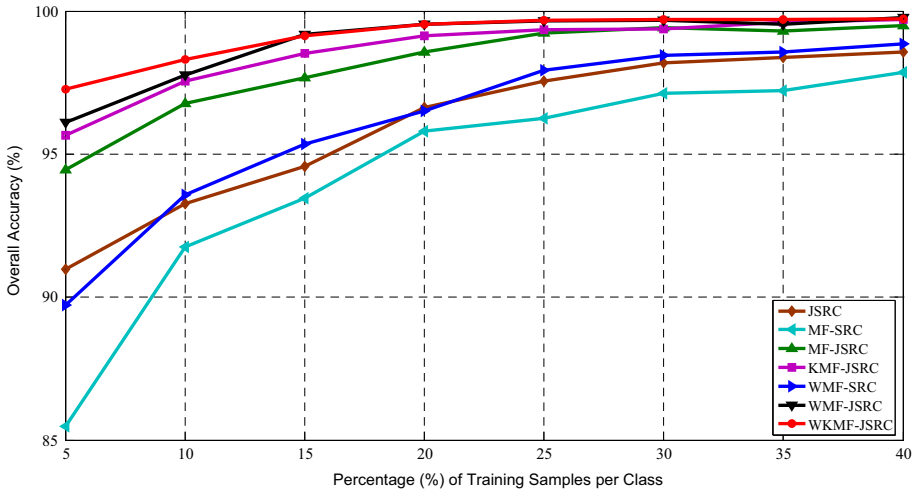


Fig. 6. Effects of the number of training samples on JSRC, MF-SRC, MF-JSRC, KMF-JSRC, WMF-SRC, WMF-JSRC and WKMF-JSRC for the Indian Pine image.

the pixel and there are three types of crops around the pixel. Thus, it is difficult to make a right decision. We calculate the reconstruction coefficients and residuals of the pixel through various CR-based and SR-based methods as shown in Figs. 4 and 5 where the reconstruction coefficients of the current test pixel under various norms regularization are shown in Fig. 4, and their corresponding residuals are shown in Fig. 5. In Fig. 4, the *X*-axis is the index of training samples, and the *Y*-axis is reconstruction coefficients. In Fig. 5, the *X*-axis is the index of classes (from 1 to 16), and the *Y*-axis is residuals.

It can be observed in Fig. 4 that the coefficient vectors obtained by each algorithm are largely different. For CR-based methods, the representation coefficients obtained with ℓ_2-norm orℓ_F-norm constraint are not sparse as shown in Fig. 4(a)-(d); there is little difference between the residuals of all the classes, which probably leads poor classification results of CR-based methods. For the ℓ_1-norm constraint and its matrix form (the $\ell_{1,2}$-norm), the sparse coefficient vectors are illustrated in Fig. 4(e)-(h); we can also see that the sparse coefficient vectors of multifeature cases are more concentrated and more discriminative in terms of residuals; however, those methods do not give a correct label for this pixel. Compared with coefficient vectors obtained with ℓ_2-norm and ℓ_1-norm regularizations, the methods with ℓ_0-norm and the $\ell_{row,0}$-norm have least nonzero coefficients. Although it may not be the best reconstruction performance, the difference between the residuals of all the classes is large, which means it has a better discrimination performance. In all the methods, only MF-JSRC can determine a correct class label of the pixel as shown in Fig. 4(l) and Fig. 5(l). As is shown in Fig. 4(l), the classification result based on spectral value feature is incorrectly, but the combination makes the right decision. Thus, it is proved that the combination of multifeature information can fuse their benefits and achieve an improved classification performance.

Next, we compare the running time of the various classification algorithms. Table 4 shows the running time of four single-feature algorithms and nine multifeature algorithms. All the programs are executed using MATLAB in the environment of an Intel Core i3-550 CPU 3.20 GHz and 4 GB of RAM. For the Indian Pines image, different percentages (from 5% to 40% per class) of samples are randomly chosen to construct a training sample set, and 100 samples are randomly chosen as the test samples. Table 4 shows the average running time for each case of each classifier in detail. The "–" symbol means the lack of results because of insufficient memory during calculation.

As shown in Tables 3 and 4, CRC is the fastest but serves the worst classification performance; JCRC and JSRC have better classification performance than CRC and SRC because of incorporating spatial neighborhood information, while they require more computational time; JSRC costs comparable time and achieves better classification results than JCRC. For multifeature learning methods, CRC-MTL and JCRC-MTL are slower than SRC-MTL. The reason is that CRC-MTL and JCRC-MTL include the matrix inverse operation, while SRC-MTL utilizes the accelerated optimize method [48]. It should also be noted that CR-based methods and SRC-MTL algorithm lack results when dealing with a large-scale training sample set as the dictionary. There are some reasons for explaining this

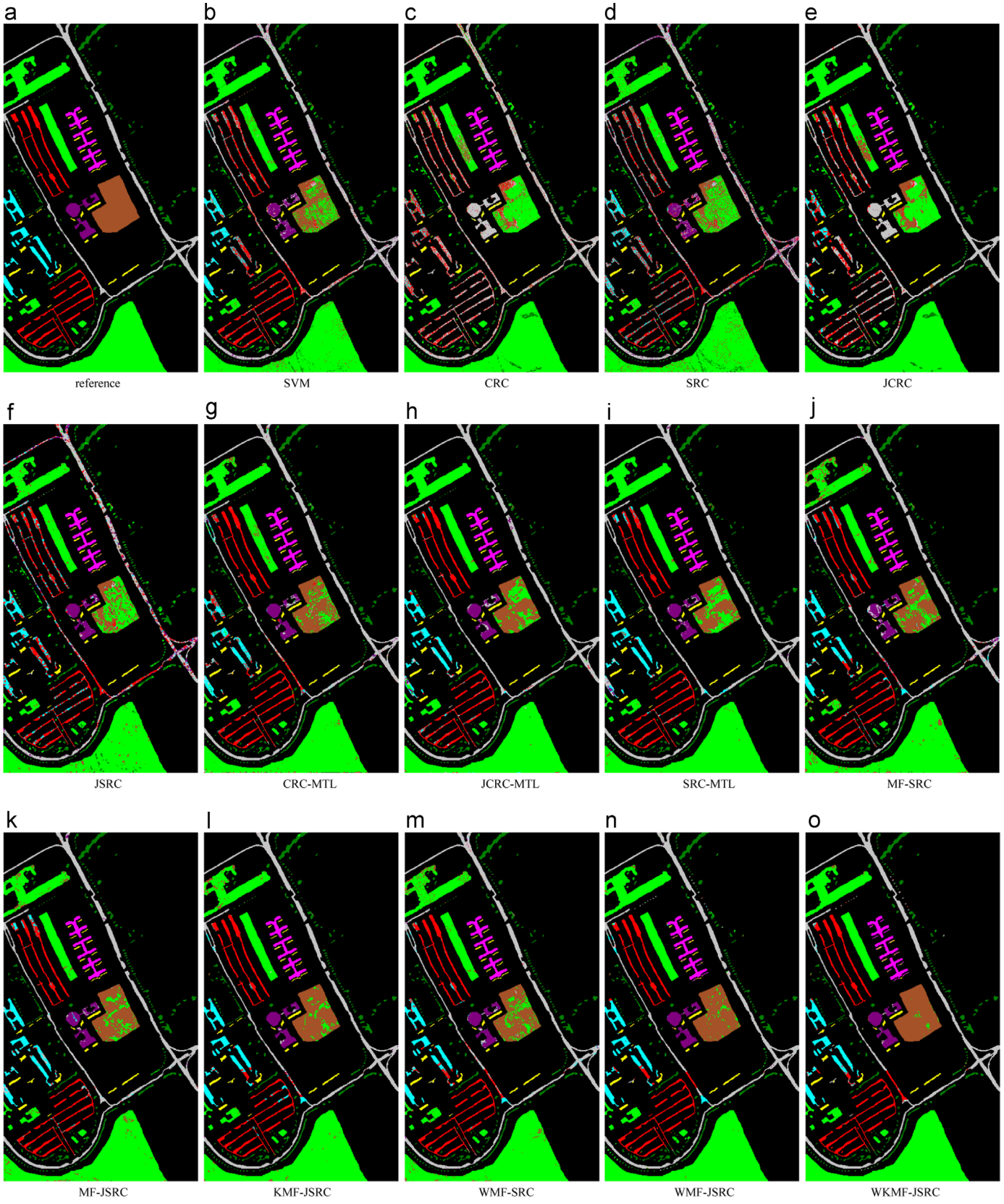


Fig. 7. The University of Pavia image: (a) reference image. The classification maps obtained by the (b) SVM, (c) CRC, (d) SRC, (e) JCRC, (f) JSRC, (g) CRC-MTL, (h) JCRC-MTL, (i) SRC-MTL, (j) MF-SRC, (k) MF-JSRC, and (l) KMF-JSRC, (m) WMF-SRC, (n) WMF-JSRC, and (o) WKMF-JSRC.

phenomenon. Multitask learning and the extended contextual information increases the computational load [50]. Especially, dealing with a large-scale training sample set as the dictionary, CRC-MTL and JCRC-MTL would require more computing time and memory due to including the matrix inverse operation. For SRC-MTL method, satisfying recognition accuracy can be obtained

Table 5
Classification accuracy (%) using single feature for the University of Pavia image on the test set.

Methods	SVM				SRC				JSRC			
Features	Spectral	GLCM	DMP	3D WT	Spectral	GLCM	DMP	3D WT	Spectral	GLCM	DMP	3D-WT
1	86.25	**94.78**	89.29	87.33	69.46	81.14	81.55	86.29	60.38	85.41	87.59	89.81
2	96.07	96.61	**98.20**	88.81	91.65	90.77	96.14	83.47	96.45	92.41	97.98	88.75
3	**67.15**	22.80	66.38	57.67	55.46	35.35	58.06	54.98	61.52	46.22	64.07	66.52
4	88.89	91.99	**93.14**	83.29	79.33	88.26	85.40	78.41	77.88	83.62	91.96	86.49
5	99.92	**100**	91.14	98.35	99.77	**100**	79.05	99.92	**100**	99.77	88.74	99.62
6	61.50	55.55	73.59	68.33	40.61	70.03	59.37	69.87	43.62	**77.26**	69.95	74.51
7	80.56	38.27	88.15	82.69	74.41	37.20	76.84	82.08	88.76	44.42	86.64	**92.71**
8	80.52	85.35	71.36	76.24	65.02	83.95	57.06	79.75	76.79	88.15	66.94	**88.23**
9	93.81	99.47	91.66	95.41	89.65	98.39	80.47	98.93	91.25	98.93	89.01	**100**
OA	86.80	84.93	**89.01**	83.42	76.93	82.15	82.07	81.04	79.67	85.16	87.44	86.65
AA	83.85	76.09	84.77	82.01	73.93	76.12	74.88	81.52	77.41	79.58	82.50	**87.40**
Kappa	82.25	79.56	**85.32**	77.81	68.94	76.27	75.95	75.05	72.47	80.36	83.14	82.35

Table 6
Classification accuracy (%) for the University of Pavia image on the test set.

Methods	Single feature methods				Multifeature methods								
								Non-Weighted			Weighted		
	CRC	JCRC	SRC	JSRC	CRC-MTL	JCRC -MTL	SRC-MTL	MF- SRC	MF- JSRC	KMF-JSRC	WMF-SRC	WMF-JSRC	WKMF-JSRC
1	86.44	95.34	69.46	60.38	90.66	97.73	98.05	92.03	**98.57**	95.95	94.01	95.10	96.39
2	95.52	95.93	91.65	96.45	96.88	99.50	99.49	93.42	96.46	97.75	96.21	99.62	**99.68**
3	17.17	31.36	55.46	61.52	66.43	78.69	82.20	83.16	86.72	**90.72**	83.98	89.37	89.51
4	80.59	81.51	79.33	77.88	96.57	98.25	**98.52**	97.56	98.48	97.36	**98.52**	92.25	94.86
5	99.85	**100**	99.77	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.92	99.02	**100**
6	18.46	27.31	40.61	43.62	78.31	70.52	62.62	75.60	81.28	88.61	79.33	93.93	**98.19**
7	0	0	74.41	88.76	86.33	84.13	90.05	78.13	90.66	94.15	89.37	**98.79**	95.29
8	42.96	47.22	65.02	76.79	96.10	93.36	92.29	92.26	91.41	95.36	96.10	**97.94**	97.86
9	71.40	90.93	89.65	91.25	83.88	94.02	99.79	99.36	**100**	97.87	97.01	84.10	75.88
OA	72.25	76.41	76.93	79.67	91.63	93.59	93.12	90.67	94.24	95.78	93.36	96.69	**97.34**
AA	56.94	63.29	73.93	77.41	88.35	90.69	91.45	90.17	93.73	**95.31**	92.72	94.46	94.19
Kappa	61.32	67.23	68.92	72.47	88.83	91.37	90.71	87.63	92.34	94.40	91.18	95.59	**96.47**

within a few hundred times of iteration [48]. Thus, SRC-MTL requires much computing time. Since the weights of the features can be pre-learned, WMF-SRC, WMF-JSRC, and WKMF-JSRC cost as much time as the one without weight, respectively. As analysis of computational complexity described in Section 4.3, the optimization problems in the proposed WMF-SRC, WMF-JSRC, and WKMF-JSRC methods are too complex, but they can be approximately solved by greedy algorithms [52,53]. In this paper, they are solved by a modified SOMP which has strong convergence guarantee and can avoid large matrix inverse operation. Thus, the proposed methods can achieve comparable classification accuracy and perform tens to hundreds of times faster than other multifeature methods.

Finally, we show the effects of different numbers of training samples on the classification accuracy of the JSRC, MF-SRC, MF-JSRC and the proposed methods (KMF-JSRC, WMF-JSRC, WKMF-JSRC), as shown in Fig. 6. In the Indian Pines image, we randomly choose different percentages (from 5% to 40% per class) of samples as the training sample set, and the remainder as the test samples. The results of OA averaged over 10 runs are shown in Fig. 6. In Fig. 6, with the training samples increase, the performances of all the classifiers generally improve. It is also shown in Fig. 6 that the proposed WMF-JSRC and WKMF-JSRC can give the state-of-the-art results even when the number of training samples is limited, which further verifies the superiority of the combination of multifeature information. Overall, the proposed WMF-JSRC and WKMF-JSRC can consistently outperform the other approaches on all cases with different training samples.

5.2.2. ROSIS urban data over Pavia, Italy

The second image is University of Pavia which is an urban scene. This scene was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS). This image has a spatial resolution of 1.3 m per pixel and consists of 610×340 pixels. Each pixel has 103 available bands with the 12 noisiest bands removed. There are nine ground-truth classes of interests shown in Fig. 7(a). For this image, around 1% samples selected randomly from each class are used in training and the rest 42,351 samples are used for testing. With each single type of feature, the classification results of the SVM, SRC, and JSRC are shown in Table 5.

In Table 5, the best results for each quality index are marked in bold. Similar conclusions as Indian Pines image's can be obtained. It can be seen that each kind of features can achieve the best performance on certain classes. That is to say, all kinds of features can reflect various aspects of the discriminative information of the HSI and are complementary to each other. The classification results using single-feature-based classification algorithm (CRC, JCRC, SRC and JSRC with spectral feature) and multifeature-based methods are summarized in Table 6. The classification maps are presented in Fig.7.

From Table 6, similar conclusions as Indian Pines image's can be drawn. Multifeature methods outperform single-feature methods in terms of accuracy. Compared with CRC-MTL and SRC-MTL, MF-JSRC leads to much better results. Different from the results on Indian Pines image, MF-JSRC only leads to a little improvement over JCRC-MTL, while WMF-JSRC leads to much improvement. It validates that appropriate weights are helpful to classification.

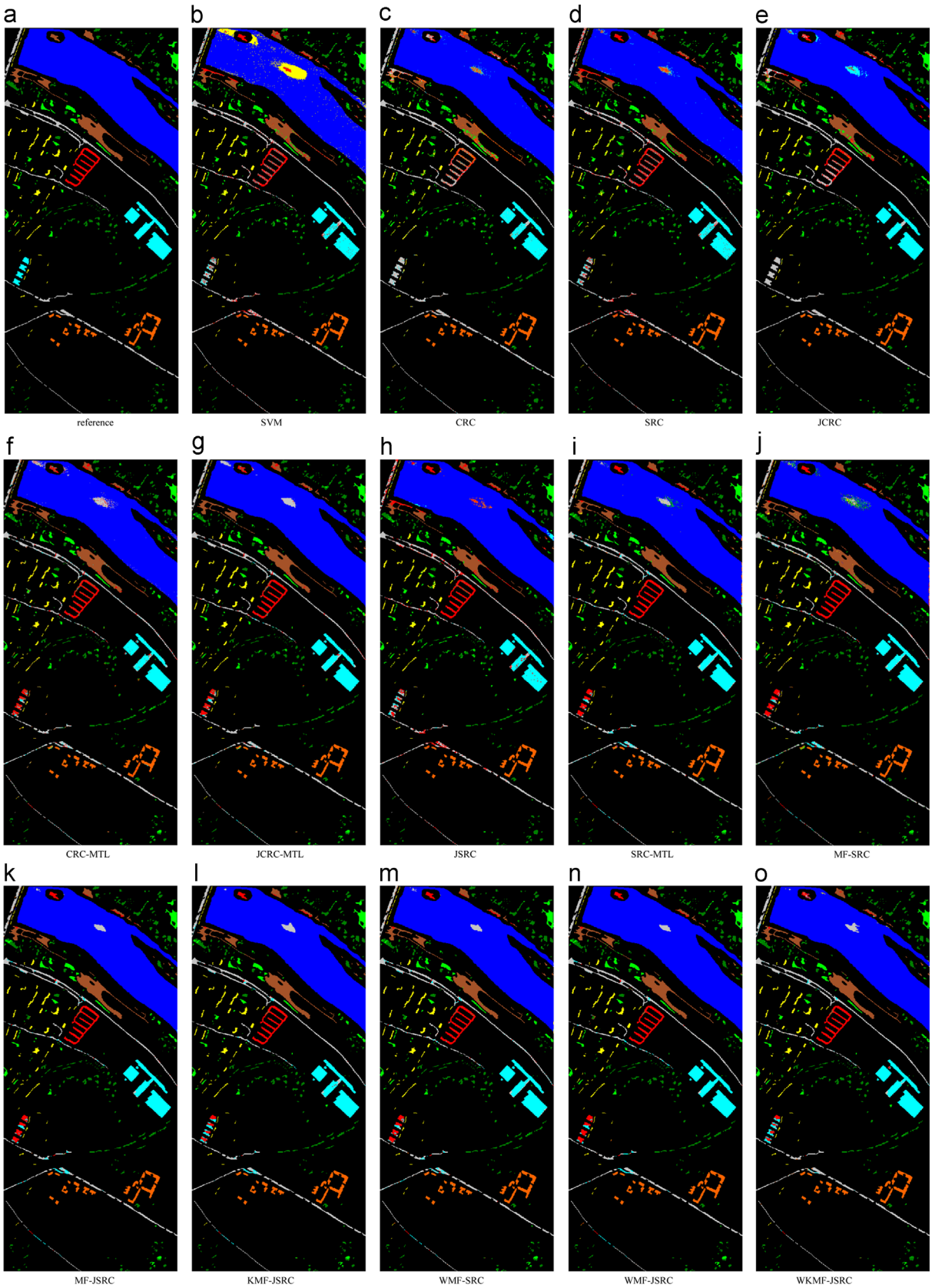


Fig. 8. The Center of Pavia image: (a) reference image. The classification maps obtained by the (b) SVM, (c) CRC, (d) SRC, (e) JCRC, (f) JSRC, (g) CRC-MTL, (h) JCRC-MTL, (i) SRC-MTL, (j) MF-SRC, (k) MF-JSRC, and (l) KMF-JSRC, (m) WMF-SRC, (n) WMF-JSRC, and (o) WKMF-JSRC.

Moreover, KMF-JSRC and WKMF-JSRC obtain better performance than the other multifeature learning methods in terms of accuracy. It demonstrates that the kernel extensions are necessary for HSI classification. It further demonstrates that the proposed WMF-JSRC, KMF-JSRC and WKMF-JSRC can successfully integrate multi-feature information and contextual neighborhood knowledge in joint sparse representation framework.

The third hyperspectral image is Center of Pavia collected by the ROSIS sensor over the center of Pavia City. A patch sized 1096×492 pixels which have 102 available spectral bands is used for experiments. This image contains nine ground-truth classes, as shown in Fig. 8(a). For this image, about 15 samples of each class of the labeled data are used as training samples and the rest samples are used for testing.

The classification results are summarized in Tables 7 and 8, and a part of classification maps are shown in Fig. 8. The classification results of the SVM, SRC, and JSRC with each single feature are shown in Table 7. We further analyze the classification results of the proposed WMF-JSRC, KMF-JSRC and WKMF-JSRC with other multifeature methods as shown in Table 8. In Tables 7 and 8, the best results for each quality index are labeled in bold. In Table 7, it can be seen that the spectral value feature and 3D WT feature-based classifiers lead to better accuracies than the others. It also can be concluded that different features can reflect different aspects of the discriminative information of the HSI and have unequal contributions to the final decision. Thus, it is reasonable to combine multiple features with appropriate weights to improve the performance of classification. It can come to similar conclusions with Indian Pines and Pavia University images. In experiments, it indicates that the performances of WMF-JSRC and WKMF-JSRC are better than the other multifeature learning methods. The proposed KMF-JSRC and WKMF-JSRC have the best accuracy for Class 3 ('Meadow'), Class 7 ('Bitumen'), Class 8 ('Tile') and Class 9 ('Shadow'). Especially, Meadow pixels (Class 3) cover a very narrow region and are dispersed on the scene. It is a challenge for other methods while KMF-JSRC gives a satisfying classification results. Moreover, WKMF-JSRC yields the best overall performance and a smoother visual effect. It is proved that the kernel extension is a good solution for solving nonlinear cases.

6. Conclusion

This paper has presented a new method for hyperspectral image classification using joint sparse representation with multi-feature learning and fusing non-linear kernel extensions. In the proposed approach, multiple helpful prior knowledge and latent structure information underneath data are simultaneously integrated in the joint sparse representation framework for classification. Moreover, non-linear kernel extensions have been employed to solve the nonlinear cases in the data. In this paper, an efficient algorithm based on SOMP is used to solve the optimization problems. On real hyperspectral images, the extensive experimental results confirm the efficiency and effectiveness of the proposed methods.

Table 7
Classification accuracy (%) using single feature for the Center of Pavia image on the test set.

Methods	SVM				SRC				JSRC			
Features	Spectral	GLCM	DMP	3D WT	Spectral	GLCM	DMP	3D WT	Spectral	GLCM	DMP	3D-WT
1	93.07	95.90	96.67	96.85	98.78	94.36	96.26	**98.95**	98.78	98.01	98.63	98.42
2	85.11	82.44	**93.21**	88.06	84.49	82.20	89.71	88.77	85.95	86.39	82.92	92.88
3	**90.55**	69.76	81.97	83.77	88.69	54.98	74.57	82.87	90.10	56.12	77.58	84.88
4	78.31	84.38	76.85	**97.55**	74.26	86.02	87.62	94.49	88.28	91.25	86.45	93.79
5	81.94	**91.35**	40.97	87.71	75.91	81.71	55.69	80.15	82.34	82.02	62.11	90.31
6	85.01	76.59	59.97	**85.55**	81.53	75.26	61.52	74.25	79.76	77.97	58.61	77.15
7	**86.29**	77.35	57.40	85.31	85.45	75.61	71.27	83.80	85.71	81.77	72.18	83.69
8	96.62	97.52	83.55	98.58	96.78	95.17	83.78	95.78	**98.81**	97.36	86.10	98.13
9	96.88	98.33	**99.81**	92.88	66.60	95.12	99.35	93.44	82.14	96.93	99.67	96.56
OA	90.61	91.18	86.34	93.70	92.95	88.85	87.89	93.50	93.82	92.31	89.35	**94.45**
AA	88.20	85.96	76.71	**90.70**	84.01	82.27	79.98	88.06	87.99	85.31	80.47	90.65
Kappa	84.58	85.23	77.04	89.38	87.97	81.51	79.69	88.90	89.47	86.94	81.87	**90.53**

Table 8
Classification accuracy (%) for the Center of Pavia image on the test set.

Methods	Single feature methods				Multifeature methods								
								Non-Weighted			Weighted		
	CRC	JCRC	SRC	JSRC	CRC-MTL	JCRC -MTL	SRC-MTL	MF- SRC	MF- JSRC	KMF-JSRC	WMF-SRC	WMF-JSRC	WKMF-JSRC
1	99.01	98.40	98.78	98.78	97.67	98.52	98.47	97.68	99.26	99.03	99.22	**99.41**	99.06
2	81.63	83.63	84.49	85.95	87.85	90.99	93.36	93.29	93.19	93.02	93.62	93.65	**96.01**
3	90.80	92.42	88.69	90.10	91.21	92.91	89.00	88.37	91.97	**96.92**	90.28	92.25	92.91
4	30.68	54.73	74.26	88.28	80.75	88.38	96.66	96.61	97.46	97.36	96.66	**97.55**	97.13
5	58.52	71.72	75.91	82.34	90.77	**93.50**	91.66	90.65	93.37	93.37	92.24	93.46	93.25
6	97.19	**99.29**	81.53	79.76	85.81	89.79	85.06	80.74	88.27	92.73	86.68	90.85	95.75
7	83.39	84.41	85.45	85.71	84.02	85.59	85.64	85.13	87.89	90.55	86.06	88.70	**91.87**
8	99.68	**100**	96.78	98.81	96.43	98.58	99.07	97.68	99.52	**100**	99.71	99.52	99.10
9	66.33	79.35	66.60	82.14	79.44	96.98	96.74	88.51	98.88	99.72	98.19	97.44	**99.81**
OA	91.86	93.45	92.95	93.82	93.84	95.78	95.52	94.36	96.66	97.19	96.28	97.02	**97.56**
AA	78.61	84.88	84.01	87.99	88.22	92.80	92.85	90.96	94.42	95.86	93.63	94.76	**96.10**
Kappa	86.07	88.84	87.97	89.47	89.56	92.81	92.37	90.44	94.28	95.19	93.63	94.89	**95.82**

It should be pointed out that the proposed methods could still be further improved in certain aspects. For instance, bridging different features based on semantic information, learning a more compact and more discriminative dictionary, realizing parallel computing, etc.

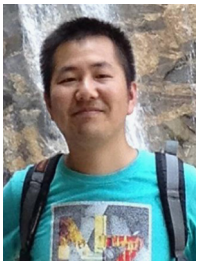
Acknowledgment

This work was supported by the National Basic Research Program of China (973 Program, no. 2013CB329402), the Nation Natural Science Foundation of China (Nos. 61272282, 61203303, 61272279, 61377011, 61373111 and 61502369), the Program for New Century Excellent Talents in University (NCET-13-0948), and the Program for New Scientific and Technological Star of Shaanxi Province (No. 2014KJXX-45).

References

[1] E. Christophe, D. Leger, C. Mailhes, Quality criteria benchmark for hyperspectral imagery, IEEE Trans. Geosci. Remote Sens 43 (2005) 2103–2114.
[2] P.K. Goel, S.O. Prasher, R.M. Patel, J.A. Landry, R.B. Bonnell, A.A. Viau, Classification of hyperspectral data by decision trees and artificial neural networks to identify weed stress and nitrogen status of corn, Comput. Electron. Agric. 39 (2003) 67–93.
[3] J.A. Benediktsson, J.A. Palmason, J.R. Sveinsson, Classification of hyperspectral data from urban areas based on extended morphological profiles, IEEE Trans. Geosci. Remote Sens. 43 (2005) 480–491.
[4] J.A. Palmason, J.A. Benediktsson, J.R. Sveinsson, J. Chanussot, Classification of hyperspectral data from urban areas using morphological preprocessing and independent component analysis, in: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, 2005, pp. 25–29.
[5] M. Volpi, G. Matasci, M. Kanevski, D. Tuia, Semi-supervised multiview embdding for hyperspectral data classification, Neurocomputing 145 (2014) 427–437.
[6] J.C. Harsanyi, I.C. Chein, Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach, IEEE Trans. Geosci. Remote Sens. 32 (1994) 779–785.
[7] G. Camps-Valls, L. Bruzzone, Kernel-based methods for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 43 (2005) 1351–1362.
[8] C.-I. Chang, Hyperspectral Data Exploitation: Theory and Applications, Wiley, Hoboken, NJ, USA, 2007.
[9] F. Melgani, L. Bruzzone, Classification of hyperspectral remote sensing images with support vector machines, IEEE Trans. Geosci. Remote Sens. 42 (2004) 1778–1790.
[10] L. Bruzzone, C. Mingmin, M. Marconcini, A novel transductive SVM for semisupervised classification of remote-sensing images, IEEE Trans. Geosci. Remote Sens vol. 44 (2006) 3363–3373.
[11] C. Mingmin, L. Bruzzone, Semisupervised classification of hyperspectral images by SVMs optimized in the primal, IEEE Trans. Geosci. Remote Sens vol. 45 (2007) 1870–1880.
[12] J. Li, J.M. Bioucas-Dias, A. Plaza, Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields, IEEE Trans. Geosci. Remote Sens vol. 50 (2012) 809–823.
[13] J. Li, J.M. Bioucas-Dias, A. Plaza, Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression, IEEE Geosci. Remote Sens. Lett. 10 (2013) 318–322.
[14] F. Ratle, G. Camps-Valls, J. Weston, Semisupervised neural networks for efficient hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 48 (2010) 2271–2282.
[15] Y. Zhong, L. Zhang, An adaptive artificial immune network for supervised classification of multi-/hyperspectral remote sensing imagery, IEEE Trans. Geosci. Remote Sens. 50 (2012) 894–909.
[16] H. Jiao, Y. Zhong, L. Zhang, Artificial DNA computing-based spectral encoding and matching algorithm for hyperspectral remote sensing data, IEEE Trans. Geosci. Remote Sens. 50 (2012) 4085–4104.
[17] M. Fauvel, Y. Tarabalka, J.A. Benediktsson, J. Chanussot, J.C. Tilton, Advances in spectral-spatial classification of hyperspectral images, Proc. IEEE 101 (2013) 652–675.
[18] Y. Tarabalka, J.A. Benediktsson, J. Chanussot, Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques, IEEE Trans. Geosci. Remote Sens 47 (2009) 2973–2987.
[19] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, J. Calpe-Maravilla, Composite kernels for hyperspectral image classification, IEEE Geosci. Remote Sens. Lett. vol. 3 (2006) 93–97.
[20] J. Li, P. Reddy Marpu, A. Plaza, J.M. Bioucas-Dias, J. Atli Benediktsson, Generalized composite kernel framework for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. vol. 51 (2013) 4816–4829.
[21] J. Li, J.M. Bioucas-Dias, A. Plaza, Hyperspectral image segmentation using a new bayesian approach with active learning, IEEE Trans. Geosci. Remote Sens. 49 (2011) 3947–3960.
[22] S. Prasad, L.M. Bruce, Limitations of principal components analysis for hyperspectral target recognition, IEEE Geosci. Remote Sens. Lett. 5 (2008) 625–629.
[23] H.-Y. Huang, B.-C. Kuo, Double nearest proportion feature extraction for hyperspectral-image classification, IEEE Trans. Geosci. Remote Sens. 48 (2010) 4034–4046.
[24] L. Zhang, Y. Zhong, B. Huang, J. Gong, P. Li, Dimensionality reduction based on clonal selection for hyperspectral imagery, IEEE Trans. Geosci. Remote Sens. 45 (2007) 4172–4186.
[25] D. Tuia, F. Pacifici, M. Kanevski, W.J. Emery, Classification of very high spatial resolution imagery using mathematical morphology and support vector machines, IEEE Trans. Geosci. Remote Sens. 47 (2009) 3866–3879.
[26] F. Pacifici, M. Chini, W.J. Emery, A neural network approach using multi-scale textural metrics from very high-resolution panchromatic imagery for urban land-use classification, Remote Sens. Environ. vol. 113 (2009) 1276–1292.
[27] Y. Qian, M. Ye, J. Zhou, Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features, IEEE Trans. Geosci. Remote Sens. 51 (2013) 2276–2291.
[28] X. Jia, B.-C. Kuo, M.M. Crawford, Feature mining for hyperspectral image classification, Proc. IEEE vol. 101 (2013) 676–697.
[29] Y. Peng, D.Y. Meng, Z.B. Xu, C.Q. Gao, Y. Yang, B. Zhang, Decomposable Nonlocal Tensor Dictionary Learning for Multispectral Image Denoising, in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2949-2956.
[30] E.J. Candes, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inf. Theory 52 (2006) 489–509.
[31] J. Wright, M. Yi, J. Mairal, G. Sapiro, T.S. Huang, Y. Shuicheng, Sparse representation for computer vision and pattern recognition, Proc. IEEE 98 (2010) 1031–1044.
[32] M. Elad, Sparse and redundant representations: From Theory To Applicationsin Signal and Image Processing, Springer-Verlag, New York, USA, 2010.
[33] A.M. Bruckstein, D.L. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images, SIAM Rev. 51 (2009) 34–81.
[34] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, M. Yi, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell 31 (2009) 210–227.
[35] Y. Chen, N.M. Nasrabadi, T.D. Tran, Hyperspectral image classification using dictionary-based sparse representation, IEEE Trans. Geosci. Remote Sens. 49 (2011) 3973–3985.
[36] U. Srinivas, Y. Chen, V. Monga, N.M. Nasrabadi, T.D. Tran, Exploiting sparsity in hyperspectral image classification via graphical models, IEEE Geosci. Remote Sens. Lett. 10 (2013) 505–509.
[37] Y. Chen, N.M. Nasrabadi, T.D. Tran, Hyperspectral image classification via kernel sparse representation, IEEE Trans. Geosci. Remote Sens 51 (2013) 217–231.
[38] L. Zhang, M. Yang, X. Feng, Y. Ma, D. Zhang, *Collaborative Representation based Classification for Face Recognition*, 2012, arXiv preprint arXiv: 1204.2358.
[39] J. Li, H. Zhang, Y. Huang, L. Zhang, Hyperspectral image classification by nonlocal joint collaborative representation with a locally adaptive dictionary, IEEE Trans. Geosci. Remote Sens vol. 52 (2014) 3707–3719.
[40] X. Huang, L. Zhang, An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery, IEEE Trans. Geosci. Remote Sens vol. 46 (2008) 4173–4185.
[41] X. Huang, L. Zhang, An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery, IEEE Trans. Geosci. Remote Sens vol. 51 (2013) 257–272.
[42] R. Caruana, Multi-task learning, Mach. Learn. vol. 28 (1997) 430–445.
[43] T. Kato, H. Kashima, M. Sugiyama, K. Asai, Multi-task learning via conic programming, in: J.C. Platt, D. Koller, Y. Singer, S.T. Roweis (Eds.), Advances in Neural Information Processing Systems, 2007.
[44] S. Ozawa, A. Roy, D. Roussinov, A multitask learning model for online pattern recognition, IEEE Trans. Neural Netw. 20 (2009) 430–445.
[45] L. Zhang, L. Zhang, D. Tao, X. Huang, On combining multiple features for hyperspectral remote sensing image classification, IEEE Trans. Geosci. Remote Sens. 50 (2012) 879–893.
[46] S. Shekhar, V.M. Patel, N.M. Nasrabadi, R. Chellappa, Joint sparse representation for robust multimodal biometrics recognition, IEEE Trans. Pattern Anal. Mach. Intell 36 (2014) 113–126.
[47] X. Zheng, X. Sun, K. Fu, H. Wang, Automatic annotation of satellite images via multifeature joint sparse coding with spatial relation constraint, IEEE Geosci. Remote Sens. Lett. 10 (2013) 652–656.
[48] X.-T. Yuan, X. Liu, S. Yan, Visual classification with multitask joint sparse representation, IEEE Trans. Image Process 21 (2012) 4349–4360.
[49] L. Fang, S. Li, X. Kang, J.A. Benediktsson, Spectral-spatial hyperspectral image classification via multiscale adaptive sparse representation, IEEE Trans. Geosci. Remote Sens. 52 (2014) 7738–7749.
[50] J. Li, H. Zhang, L. Zhang, X. Huang, L. Zhang, Joint collaborative representation with multitask learning for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens 52 (2014) 5923–5936.
[51] Y. Meng, L. Zhang, D. Zhang, and S. Wang, ".R.elaxed collaborative representation for pattern classification, in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 2224-2231.

[52] J.A. Tropp, A.C. Gilbert, M.J. Strauss, Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit, Signal Process.-Special Issue on Sparse Approximations in Signal and Image Processing vol. 86 (2006) 572–588.
[53] S.F. Cotter, B.D. Rao, K. Engan, K. Kreutz-Delgado, Sparse solutions to linear inverse problems with multiple measurement vectors, IEEE Trans. Signal Process. vol. 53 (2005) 2477–2488.
[54] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inf. Theory vol. 53 (2007) 4655–4666.
[55] B.E. Boser, I.M. Guyon, and V.N. Vapnik, A training algorithm for optimal margin classifiers," in: Proceedings of 5th Annual Workshop on Computer Learning Theory, ACM.Pittsburgh, PA, USA, 1992, pp. 144–152.
[56] E.L. Zhang, X.R. Zhang, H.Y. Liu, L.C. Jiao, Fast multifeature joint sparse representation for hyperspectral image classification, IEEE Geosci. Remote Sens. Lett. 12 (2015) 1397–1401.
[57] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification (2nd ed.), Wiley-Interscience.

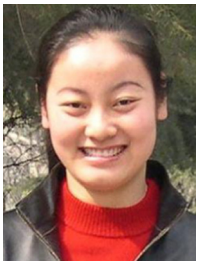


Erlei Zhang received the B.S. degree from the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiao Zuo, China, in 2010. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems from the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, China. His current research interests include remote sensing image analysis, pattern recognition, and machine learning.

Xiangrong Zhang received the B.S. and M.S. degrees from the School of Computer Science, Xidian University, Xi'an, China, in 1999 and 2003, respectively, and the Ph.D. degree from the School of Electronic Engineering, Xidian University, in 2006. Currently, she is a professor in the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University, China. She is a visiting scientist in Computer Science and Artificial Intelligence Laboratory, MIT since Feb. 2015. Her research interests include pattern recognition, machine learning, and image analysis and understanding.

Licheng Jiao received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982 and the M. S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively. He is the author or coauthor of more than 150 scientific papers. His current research interests include signal and image processing, nonlinear circuit and systems theory, learning theory and algorithms, optimization problems, wavelet theory, and data mining.

Hongying Liu received her B.E. and M.S. degrees in Computer Science and Technology from Xi'an University of Technology, China, in 2002 and 2006, respectively, and Ph.D. in Engineering from Waseda University, Japan in 2012. Currently, she is with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, China. Her major research interests include intelligent signal processing, machine learning, compressive sampling, etc.



Shuang Wang received the B.S. degree and M.S. degrees from Xidian University, Xi'an, China, in 2000 and 2003, respectively, and received the Ph.D. in circuits and systems from Xidian University, Xi'an, China, in 2007. Now she is a Professor in the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University. She is a member of IEEE. Her main research interests are machine learning, image processing and SAR/POLSAR image processing, etc.

Biao Hou received the B.S. and M.S. degrees in mathematics from Northwest University, Xi'an, China, in 1996 and 1999, respectively, and the Ph.D. degree in circuits and systems from Xidian University, Xi'an, in 2003. Since 2003, he has been with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, where he is currently a Professor. His research interests include compressive sensing and Synthetic Aperture Radar image interpretation, etc.