Hyperspectral Image Classification via Kernel Sparse Representation

Yi Chen[1], Nasser M. Nasrabadi[2], *Fellow, IEEE,* and Trac D. Tran[1], *Senior Member, IEEE*

[1]Department of Electrical and Computer Engineering, The Johns Hopkins University
3400 N. Charles Street, Baltimore, MD 21218

[2]US Army Research Laboratory
2800 Powder Mill Road, Adelphi, MD 20783

Abstract

In this paper, a novel nonlinear technique for hyperspectral image classification is proposed. Our approach relies on sparsely representing a test sample in terms of all training samples in a feature space induced by a kernel function. For each test pixel in the feature space, a sparse representation vector is obtained by decomposing the test pixel over a training dictionary, also in the same feature space, by using a kernel-based greedy pursuit algorithm. The recovered sparse representation vector is then used directly to determine the class label of the test pixel. Projecting the samples into a high-dimensional feature space and kernelizing the sparse representation improves the data separability between different classes, providing a higher classification accuracy compared to the more conventional linear sparsity-based classification algorithms. Moreover, the spatial coherency across neighboring pixels is also incorporated through a kernelized joint sparsity model, where all of the pixels within a small neighborhood are jointly represented in the feature space by selecting a few common training samples. Several kernel greedy optimization algorithms are suggested in this paper to solve the kernel versions of the single-pixel and multi-pixel joint sparsity-based recovery problems. Experimental results on several hyperspectral images show that the proposed technique outperforms the linear sparsity-based classification technique, as well as the classical Support Vector Machines and sparse kernel logistic regression classifiers.

This paper is partially supported by ARO Grant 58110-MA-II and NSF Grant CCF-0728893.

I. Introduction

Hyperspectral imaging sensors capture digital images in hundreds of continuous narrow spectral bands, spanning the visible to infrared spectrum. Each pixel in a hyperspectral image (HSI) is represented by a vector whose entries correspond to various spectral-band responses. Different materials usually reflect electromagnetic energy differently at specific wavelengths. This enables discrimination of materials based on their spectral characteristics. One of the most important applications of HSI is image classification, where pixels are labeled to one of the classes based on their spectral characteristics, given a small set of training data for each class. Various techniques have been developed for HSI classification. Among the previous approaches, the support vector machine (SVM) [1], [2] has proven to be a powerful tool to solve supervised classification problems and has shown good performances in hyperspectral classification, as well [3], [4]. Variations of SVM-based algorithms have also been proposed to improve the classification accuracy. These variations include semi-supervised learning, which exploits both labeled and unlabeled samples [5], post-processing of the individually-labeled samples based on certain decision rules [6], [7], and incorporating spatial information directly in the SVM kernels [8], [9]. Multinomial logistic regression [10] is another widely used classifier, which fits the posterior probability to the logistic function. A fast algorithm for sparse multinomial logistic regression has been developed [11] and successfully adopted in semi-supervised HSI segmentation [12]. More recent HSI classification techniques can be found in [13]–[19].

Recently, sparse representation has also been proposed to solve many computer vision tasks [20]–[25], where the usage of sparsity as a prior often leads to state-of-the-art performance. Sparse representation has also been applied to HSI target detection and classification [26], relying on the assumption that hyperspectral pixels belonging to the same class lie in the same low-dimensional subspace. Thus, an unknown pixel can be sparsely represented by a few training samples from a given dictionary and the corresponding sparse representation vector encodes the class information.

It is well known that for the classical HSI image classification and target detection algorithms, the kernel methods can significantly improve their performance [27], [28], since the kernel-based algorithms implicitly exploit the higher-order structure of the given data which may not be captured by the linear models. Therefore, if the dataset is not linearly separable, kernel methods [29]–[32] can be applied to project the data into a nonlinear feature space in which the data becomes more separable. In practical implementation, the kernel trick [1] is often used in order to avoid explicitly evaluating the data in the feature space.

In this paper, we propose a new HSI classification algorithm based on kernel sparse representation by assuming that a test pixel can be linearly represented by a few training samples in the feature space. The proposed approach is different from the previous kernel regression approaches in [29], [30], where a function is learned as a sparse linear combination of basis which are in the form of kernel functions. Therefore, the target vector for fitting consists of the observations of the function value at the training points and the dictionary is the kernel matrix. In our proposed approach, a kernel sparse representation vector is obtained by decomposing the test pixel represented in a high dimensional feature space over a structured dictionary consisting of training samples from all of the classes in the same feature space. The contextual correlation between pixels within a small spatial neighborhood is also incorporated into the kernel sparse representation through a joint sparsity model [33], where all neighboring pixels are simultaneously represented by a linear combination of a few common training samples in the feature space. Each pixel, although sharing the same common support, might have weighting coefficients taking on different values. In this way, the smoothness across neighboring spectral pixels is enforced directly in the classification stage, and no post-processing steps are performed. Efficient kernel-based optimization algorithms are also discussed in this paper for the recovery of the kernel sparse representations for both single-pixel and multi-pixel joint sparsity models.

Notation-wise, vectors and matrices are denoted by lower- and upper-case bold letters, respectively. For a vector $\boldsymbol{\alpha} \in \mathbb{R}^N$ and an index set $\Lambda \subseteq \{1,\ldots,N\}$ with $|\Lambda| = t$, $\boldsymbol{\alpha}_\Lambda \in \mathbb{R}^t$ is the portion of $\boldsymbol{\alpha}$ indexed on Λ. For a matrix $\boldsymbol{S} \in \mathbb{R}^{N_1 \times N_2}$, index sets $\Lambda_1 \subseteq \{1,\ldots,M\}$ with $|\Lambda_1| = t_1$, and $\Lambda_2 \subseteq \{1,\ldots,N_2\}$ with $|\Lambda_2| = t_2$, $\boldsymbol{S}_{\Lambda_1,:} \in \mathbb{R}^{t_1 \times N_2}$ is a submatrix of $\boldsymbol{S}$ consisting of the t_1 rows in $\boldsymbol{S}$ indexed on Λ_1, $\boldsymbol{S}_{:,\Lambda_2} \in \mathbb{R}^{N_1 \times t_2}$ consists of the t_2 columns in $\boldsymbol{S}$ indexed on Λ_2, and $\boldsymbol{S}_{\Lambda_1,\Lambda_2} \in \mathbb{R}^{t_1 \times t_2}$ is formed by the rows and columns of $\boldsymbol{S}$ indexed on Λ_1 and Λ_2, respectively.

The remainder of this paper is structured as follows. Section II briefly introduces the sparsity-based HSI classification technique. Section III then defines the sparsity models in the feature space and describes how to solve the kernel sparse recovery problems. Experimental results are discussed in Section IV, and conclusions are drawn in Section V.

II. Sparsity-based HSI Classification

This section briefly introduces the sparsity-based algorithm for HSI classification, and more details can be found in [26]. It is assumed that the spectral signatures of pixels belonging to the same class approximately lie in the same low-dimensional subspace. Thus, an unknown test sample $\boldsymbol{x} \in \mathbb{R}^B$, where B is the number of spectral bands, can be written as a sparse linear combination of all of the training

pixels as

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{\alpha}, \tag{1}$$

where $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \end{bmatrix} \in \mathbb{R}^{B\times N}$ is a structured dictionary whose columns $\{\boldsymbol{a}_i\}_{i=1,2,\ldots,N}$ are N training samples (referred to as atoms) from all classes, and $\boldsymbol{\alpha} \in \mathbb{R}^N$ is an unknown sparse vector. The index set on which $\boldsymbol{\alpha}$ have nonzero entries is the support of $\boldsymbol{\alpha}$. The number of nonzero entries in $\boldsymbol{\alpha}$ is called the sparsity level K of $\boldsymbol{\alpha}$ and denoted by $K = \|\boldsymbol{\alpha}\|_0$. Given the dictionary $\boldsymbol{A}$, the sparse coefficient vector $\boldsymbol{\alpha}$ is obtained by solving

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\alpha}\|_2 \quad \text{subject to} \quad \|\boldsymbol{\alpha}\|_0 \leq K_0, \tag{2}$$

where K_0 is a preset upper bound on the sparsity level. The problem in (2) is NP-hard, which can be approximately solved by greedy algorithms, such as Orthogonal Matching Pursuit (OMP) [34] or Subspace Pursuit (SP) [35]. Both OMP and SP algorithms are used to locate the support of the sparse vector $\hat{\boldsymbol{\alpha}}$, but the difference between these two algorithms is in the way the atoms are selected from the dictionary. The OMP algorithm augments the support set by one index at each iteration until K_0 atoms are selected or the approximation error is within a preset threshold. The SP algorithm maintains a set of K_0 indices. At each iteration, the index set is refined by adding K_0 new candidates to the current list and then discarding K_0 insignificant ones from the list of $2K_0$ candidates. With the backtracking mechanism, SP is able to find the K_0 most significant atoms. The class label of $\boldsymbol{x}$ is determined by the minimal residual between $\boldsymbol{x}$ and its approximation from each class sub-dictionary:

$$\text{Class}(\boldsymbol{x}) = \arg \min_{m=1,\ldots,M} \|\boldsymbol{x} - \boldsymbol{A}_{:,\Omega_m}\hat{\boldsymbol{\alpha}}_{\Omega_m}\|_2, \tag{3}$$

where $\Omega_m \subset \{1,2,\ldots,N\}$ is the index set associated with the training samples belonging to the mth class.

In HSI, pixels within a small neighborhood usually consist of similar materials and, thus, their spectral characteristics are highly correlated. The spatial correlation between neighboring pixels can be incorporated through a joint sparsity model [26], [33] by assuming the underlying sparse vectors associated with these pixels share a common sparsity pattern as follows. Let $\{\boldsymbol{x}_t\}_{t=1,\ldots,T}$ be T pixels in a spatial neighborhood centered at $\boldsymbol{x}_1$. These pixels can be compactly represented as

$$\begin{aligned} \boldsymbol{X} &= \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}\boldsymbol{\alpha}_1 & \boldsymbol{A}\boldsymbol{\alpha}_2 & \cdots & \boldsymbol{A}\boldsymbol{\alpha}_T \end{bmatrix} \\ &= \boldsymbol{A}\underbrace{\begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \cdots & \boldsymbol{\alpha}_T \end{bmatrix}}_{\boldsymbol{S}} = \boldsymbol{A}\boldsymbol{S}. \end{aligned} \tag{4}$$

In the joint sparsity model, the sparse vectors $\{\boldsymbol{\alpha}_t\}_{t=1,\ldots,T}$ share the same support Λ and, thus, $\boldsymbol{S}$ is a sparse matrix with only $|\Lambda|$ nonzero rows. The row-sparse matrix $\boldsymbol{S}$ can be recovered by solving the

following optimization problem

$$\hat{\boldsymbol{S}} = \arg\min \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|_F \quad \text{subject to} \quad \|\boldsymbol{S}\|_{\text{row},0} \leq K_0, \tag{5}$$

where $\|\boldsymbol{S}\|_{\text{row},0}$ denotes the number of non-zero rows of $\boldsymbol{S}$ and $\|\cdot\|_F$ denotes the Frobenius norm. The problem in (5) can be approximately solved by the simultaneous versions of OMP (SOMP) [33] or SP (SSP). The label of the center pixel $\boldsymbol{x}_1$ is then determined by the minimal total residual:

$$\text{Class}(\boldsymbol{x}_1) = \arg\min_{m=1,\ldots,M} \left\| \boldsymbol{X} - \boldsymbol{A}_{:,\Omega_m} \hat{\boldsymbol{S}}_{\Omega_m,:} \right\|_F, \tag{6}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

III. Kernel Sparse Representation

If the classes in the dataset are not linearly separable, then the kernel methods can be used to project the data into a feature space, in which the classes become linearly separable [1]. The kernel function $\kappa : \mathbb{R}^B \times \mathbb{R}^B \mapsto \mathbb{R}$ is defined as the inner product

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle . \tag{7}$$

Commonly used kernels include the radial Basis Function (RBF) kernel $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)$ with $\gamma > 0$ controlling the width of the RBF, and order$-d$ homogeneous and inhomogeneous polynomial kernels $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j)^d$ and $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^d$, respectively. In this section, we describe how the sparsity models in Section II can be extended to a feature space induced by a kernel function.

A. Pixel-wise Sparsity in Feature Space

Let $\boldsymbol{x} \in \mathbb{R}^B$ be the data point of interest and $\phi(\boldsymbol{x})$ be its representation in the feature space. The kernel sparse representation of a sample $\boldsymbol{x}$ in terms of training atoms $\boldsymbol{a}_i$'s can be formulated as

$$\phi(\boldsymbol{x}) = \underbrace{\begin{bmatrix} \phi(\boldsymbol{a}_1) & \cdots & \phi(\boldsymbol{a}_N) \end{bmatrix}}_{\boldsymbol{A}_\phi} \underbrace{\begin{bmatrix} \alpha'_1 & \cdots & \alpha'_N \end{bmatrix}^T}_{\boldsymbol{\alpha}'} = \boldsymbol{A}_\phi \boldsymbol{\alpha}', \tag{8}$$

where the columns of $\boldsymbol{A}_\phi$ are the representations of training samples in the feature space and $\boldsymbol{\alpha}'$ is assumed to be a sparse vector.

Similar to the linear sparse recovery problem in (2), $\boldsymbol{\alpha}'$ can be recovered by solving

$$\hat{\boldsymbol{\alpha}}' = \arg\min \left\| \phi(\boldsymbol{x}) - \boldsymbol{A}_\phi \boldsymbol{\alpha}' \right\|_2 \quad \text{subject to} \quad \|\boldsymbol{\alpha}'\|_0 \leq K_0. \tag{9}$$

The problem in (9) can be approximately solved by kernelizing the OMP and SP algorithms (denoted by KOMP and KSP, respectively).

In KOMP and KSP, essentially each dot product operation in OMP/SP is replaced by the kernel trick in (7). Let $\boldsymbol{K}_{\boldsymbol{A}} \in \mathbb{R}^{N \times N}$ be the kernel matrix whose (i,j)th entry is $\kappa(\boldsymbol{a}_i, \boldsymbol{a}_j)$, and $\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}} \in \mathbb{R}^N$ be the vector

whose ith entry is $\kappa(\boldsymbol{a}_i,\boldsymbol{x})$. Using the feature representations, the correlation (dot product) between a pixel $\phi(\boldsymbol{x})$ and a dictionary atom $\phi(\boldsymbol{a}_i)$ is then computed by

$$c_i = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{a}_i) \rangle = \kappa(\boldsymbol{x},\boldsymbol{a}_i) = (\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}})_i, \tag{10}$$

the orthogonal projection coefficient of $\phi(\boldsymbol{x})$ onto a set of selected dictionary atoms $\{\phi(\boldsymbol{a}_n)\}_{n\in\Lambda}$ is given as

$$\boldsymbol{p}_\Lambda = \left((\boldsymbol{K}_{\boldsymbol{A}})_{\Lambda,\Lambda}\right)^{-1} (\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}})_\Lambda, \tag{11}$$

and the residual vector between $\phi(\boldsymbol{x})$ and its approximation from $\{\phi(\boldsymbol{a}_n)\}_{n\in\Lambda} = (\boldsymbol{A}_\phi)_{:,\Lambda}$ is then expressed as

$$\phi(\boldsymbol{r}) = \phi(\boldsymbol{x}) - (\boldsymbol{A}_\phi)_{:,\Lambda} \left((\boldsymbol{K}_{\boldsymbol{A}})_{\Lambda,\Lambda}\right)^{-1} (\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}})_\Lambda. \tag{12}$$

Note that the feature representation of the residual vector $\phi(\boldsymbol{r})$ in (12) cannot be evaluated explicitly. However, the correlation between $\phi(\boldsymbol{r})$ and an atom $\phi(\boldsymbol{a}_i)$ can be computed by

$$c_i = \langle \phi(\boldsymbol{r}), \phi(\boldsymbol{a}_i) \rangle = (\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}})_i - (\boldsymbol{K}_{\boldsymbol{A}})_{i,\Lambda} \left((\boldsymbol{K}_{\boldsymbol{A}})_{\Lambda,\Lambda}\right)^{-1} (\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}})_\Lambda. \tag{13}$$

The details of the KOMP and KSP algorithms are summarized in Algorithms 1 and 2, respectively. The step for computing the residual vector (12) is incorporated into the computation of the correlation vector in Step (1) of both KOMP and KSP.

Once the sparse vector $\hat{\boldsymbol{\alpha}}'$ is recovered, the residual between the test sample and the mth-class reconstruction in the high-dimensional feature space is then computed by

$$\begin{aligned} r_m(\boldsymbol{x}) &= \left\| \phi(\boldsymbol{x}) - (\boldsymbol{A}_\phi)_{:,\Omega_m} \hat{\boldsymbol{\alpha}}'_{\Omega_m} \right\| \\ &= \left\langle \phi(\boldsymbol{x}) - (\boldsymbol{A}_\phi)_{:,\Omega_m} \hat{\boldsymbol{\alpha}}'_{\Omega_m}, \phi(\boldsymbol{x}) - (\boldsymbol{A}_\phi)_{:,\Omega_m} \hat{\boldsymbol{\alpha}}'_{\Omega_m} \right\rangle^{1/2} \\ &= \left(\kappa(\boldsymbol{x},\boldsymbol{x}) - 2\hat{\boldsymbol{\alpha}}'^T_{\Omega_m} (\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}})_{\Omega_m} + \hat{\boldsymbol{\alpha}}'^T_{\Omega_m} (\boldsymbol{K}_{\boldsymbol{A}})_{\Omega_m,\Omega_m} \hat{\boldsymbol{\alpha}}'_{\Omega_m} \right)^{1/2}, \end{aligned} \tag{14}$$

where $\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}}$ and $\boldsymbol{K}_{\boldsymbol{A}}$ are as defined in the initialization step in Algorithms 1 and 2, and Ω_m is the index set associated with the mth class. The class label of $\boldsymbol{x}$ is determined as

$$\text{Class}(\boldsymbol{x}) = \arg \min_{m=1,\ldots,M} r_m(\boldsymbol{x}). \tag{15}$$

B. Joint Sparsity in Feature Space

The joint sparsity model in (4) can also be extended to the feature space as follows:

$$\begin{aligned} \boldsymbol{X}_\phi &= \begin{bmatrix} \phi(\boldsymbol{x}_1) & \cdots & \phi(\boldsymbol{x}_T) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_\phi \boldsymbol{\alpha}'_1 & \cdots & \boldsymbol{A}_\phi \boldsymbol{\alpha}'_T \end{bmatrix} \\ &= \boldsymbol{A}_\phi \underbrace{\begin{bmatrix} \boldsymbol{\alpha}'_1 & \cdots & \boldsymbol{\alpha}'_T \end{bmatrix}}_{\boldsymbol{S}'} = \boldsymbol{A}_\phi \boldsymbol{S}', \end{aligned} \tag{16}$$

Input: $B \times N$ dictionary $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \end{bmatrix}$, test sample $\boldsymbol{x}$, kernel function κ, and a stopping criterion

---

Initialization: compute kernel matrix $\boldsymbol{K_A} \in \mathbb{R}^{N \times N}$ whose (i,j)th entry is $\kappa(\boldsymbol{a}_i, \boldsymbol{a}_j)$, and vector $\boldsymbol{k_{A,x}} \in \mathbb{R}^N$ whose ith entry is $\kappa(\boldsymbol{a}_i, \boldsymbol{x})$. Set index set Λ_0 to be index corresponding to the largest entry in $\boldsymbol{k_{A,x}}$ and iteration counter $t = 1$.

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

(1) Compute the correlation vector $\boldsymbol{c} = \begin{bmatrix} c_1 & \cdots & c_N \end{bmatrix}^T$ by

$$\boldsymbol{c} = \boldsymbol{k_{A,x}} - (\boldsymbol{K_A})_{:,\Lambda_{t-1}} \left((\boldsymbol{K_A})_{\Lambda_{t-1},\Lambda_{t-1}} \right)^{-1} (\boldsymbol{k_{A,x}})_{\Lambda_{t-1}}$$

(2) Select the new index as $\lambda_t = \arg \max_{i=1,\ldots,N} |c_i|$

(3) Update the index set $\Lambda_t = \Lambda_{t-1} \bigcup \{\lambda_t\}$

(4) $t \leftarrow t + 1$

end while

Output: Index set $\Lambda = \Lambda_{t-1}$, the sparse representation $\hat{\boldsymbol{\alpha}}'$ whose nonzero entries indexed by Λ are $\hat{\boldsymbol{\alpha}}'_\Lambda = (\boldsymbol{K}_{\Lambda,\Lambda})^{-1} (\boldsymbol{k_{A,x}})_\Lambda$

Algorithm 1: Kernelized Orthogonal Matching Pursuit (KOMP)

where the vectors $\{\boldsymbol{\alpha}'_t\}_{t=1,\ldots,T}$ share the same support. The row-sparse matrix $\boldsymbol{S}'$ is recovered by solving

$$\hat{\boldsymbol{S}}' = \arg\min \left\| \boldsymbol{X}_\phi - \boldsymbol{A}_\phi \boldsymbol{S}' \right\|_F \quad \text{subject to} \quad \|\boldsymbol{S}'\|_{\text{row},0} \leq K_0. \tag{17}$$

In this paper, we propose kernelized SOMP (KSOMP) and kernelized SSP (KSSP) to approximately solve the above joint sparse recovery problem in (17).

In KSOMP, at every iteration, the atom that simultaneously yields the best approximation to all the T pixels (or residuals after initialization) is selected. Specifically, let $\boldsymbol{C} \in \mathbb{R}^{N \times T}$ be the correlation matrix whose (i,j)th entry is the correlation between $\phi(\boldsymbol{a}_i)$ and $\phi(\boldsymbol{r}_j)$, where $\phi(\boldsymbol{r}_j)$ is the residual vector of $\phi(\boldsymbol{x}_j)$. The new atom is then selected as the one associated with the row of $\boldsymbol{C}$, which has the maximal ℓ_p-norm for some $p \geq 1$. The KSOMP algorithm is summarized in Algorithm 3.

Similarly, KSSP is a simultaneous version of KSP where the K_0 atoms that best simultaneously approximate all of the T residuals in terms of the ℓ_p-norm are chosen. The KSSP algorithm is summarized in Algorithm 4.

Once the matrix $\hat{\boldsymbol{S}}'$ is recovered, the total residual between the T neighboring pixels and their approx-

Input: $B \times N$ dictionary $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_N \end{bmatrix}$, test sample $\boldsymbol{x}$, kernel function κ, sparsity K_0, and a stopping criterion

---

Initialization: compute kernel matrix $\boldsymbol{K}_{\boldsymbol{A}} \in \mathbb{R}^{N \times N}$ whose (i,j)th entry is $\kappa(\boldsymbol{a}_i, \boldsymbol{a}_j)$, and vector $\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}} \in \mathbb{R}^N$ whose ith entry is $\kappa(\boldsymbol{a}_i, \boldsymbol{x})$. Set index set $\Lambda_0 = \left\{ K_0 \text{ indices corresponding to the } K_0 \text{ largest entries in } \boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}} \right\}$ and iteration counter $t = 1$.

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

(1) Compute the correlation vector $\boldsymbol{c} = \boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}} - (\boldsymbol{K}_{\boldsymbol{A}})_{:,\Lambda_{t-1}} \left((\boldsymbol{K}_{\boldsymbol{A}})_{\Lambda_{t-1},\Lambda_{t-1}} \right)^{-1} (\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}})_{\Lambda_{t-1}} \in \mathbb{R}^N$

(2) Find the index set $I = \left\{ K_0 \text{ indices corresponding to the } K_0 \text{ largest entries in } |\boldsymbol{c}| \right\}$

(3) Update the candidate index set $\tilde{\Lambda}_t = \Lambda_{t-1} \bigcup I$

(4) Compute $\boldsymbol{p} = \left((\boldsymbol{K}_{\boldsymbol{A}})_{\tilde{\Lambda}_t,\tilde{\Lambda}_t} \right)^{-1} (\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}})_{\tilde{\Lambda}_t} \in \mathbb{R}^{2K_0}$

(5) Update the index set $\Lambda_t = \left\{ K_0 \text{ indices in } \tilde{\Lambda}_t \text{ corresponding to the } K_0 \text{ largest entries in } |\boldsymbol{p}| \right\}$

(6) $t \leftarrow t+1$

end while

Output: Index set $\Lambda = \Lambda_{t-1}$, the sparse representation $\hat{\boldsymbol{\alpha}}'$ whose nonzero entries indexed by Λ are $\hat{\boldsymbol{\alpha}}'_{\Lambda} = (\boldsymbol{K}_{\Lambda,\Lambda})^{-1} (\boldsymbol{k}_{\boldsymbol{A},\boldsymbol{x}})_{\Lambda}$

Algorithm 2: Kernelized Subspace Pursuit (KSP)

imations from the mth-class training samples is computed by

$$r_m(\boldsymbol{x}_1) = \left(\sum_{i=1}^{T} \left(\kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2\hat{\boldsymbol{S}}'^{T}_{\Omega_m,i} (\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}})_{\Omega_m,i} + \hat{\boldsymbol{S}}'^{T}_{\Omega_m,i} (\boldsymbol{K}_{\boldsymbol{A}})_{\Omega_m,\Omega_m} \hat{\boldsymbol{S}}'_{\Omega_m,i} \right) \right)^{1/2}, \tag{18}$$

where $\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}}$ and $\boldsymbol{K}_{\boldsymbol{A}}$ are as defined in Algorithms 3 and 4, and $\Omega_m \in \{1, 2, \ldots, N\}$ is the index set associated with the mth class. The label for the center pixel $\boldsymbol{x}_1$ is then determined by the total residual

$$\text{Class}(\boldsymbol{x}_1) = \arg \min_{m=1,\ldots,M} r_m(\boldsymbol{x}_1). \tag{19}$$

IV. Experimental Results

In this section, we demonstrate the effectiveness of the proposed algorithms on classification of several hyperspectral datasets. For each image, we solve the sparse recovery problems in (2), (5), (9), and (17) for each test sample, and then determine the class by the minimal residual (the results are denoted by OMP/SP, KOMP/KSP, SOMP/SSP, and KSOMP/KSSP, respectively). The classification results are then compared visually and quantitatively to those obtained by the classical SVM classifier and sparse

Input: $B \times N$ dictionary $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_N \end{bmatrix}$, $B \times T$ data matrix $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_T \end{bmatrix}$, kernel function κ, and a stopping criterion

---

Initialization: compute the kernel matrices $\boldsymbol{K}_{\boldsymbol{A}}$ in Algorithm 1 (Initialization) and $\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}} \in \mathbb{R}^{N \times T}$ whose (i,j)th entry is $\kappa(\boldsymbol{a}_i, \boldsymbol{x}_j)$. Set index set $\Lambda_0 = \arg\max_{i=1,\ldots,N} \left\| (\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}})_{i,:} \right\|_p$ with some $p \geq 1$ and iteration counter $t = 1$.

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

(1) Compute the correlation matrix

$$\boldsymbol{C} = \boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}} - (\boldsymbol{K}_{\boldsymbol{A}})_{:,\Lambda_{t-1}} \left((\boldsymbol{K}_{\boldsymbol{A}})_{\Lambda_{t-1},\Lambda_{t-1}} \right)^{-1} (\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}})_{\Lambda_{t-1},:} \in \mathbb{R}^{N \times T}$$

(2) Select the new index as $\lambda_t = \arg\max_{i=1,\ldots,N} \|\boldsymbol{C}_{i,:}\|_p$, $p \geq 1$

(3) Update the index set $\Lambda_t = \Lambda_{t-1} \bigcup \{\lambda_t\}$

(4) $t \leftarrow t + 1$

end while

Output: Index set $\Lambda = \Lambda_{t-1}$, the sparse representation $\hat{\boldsymbol{S}}'$ whose nonzero rows indexed by Λ are $\hat{\boldsymbol{S}}'_{\Lambda,:} = (\boldsymbol{K}_{\Lambda,\Lambda})^{-1} (\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}})_{\Lambda,:}$

Algorithm 3: Kernelized Simultaneous Orthogonal Matching Pursuit (KSOMP)

multinomial kernel logistic regression (KLR). For SVM and KLR classifiers, we use a spectral-only kernel (denoted by SVM/KLR), as well as a composite kernel [8] (denoted by SVMCK/KLRCK). The composite kernel takes into account the spatial correlation between neighboring pixels by combining the spectral and spatial information via a weighted kernel summation, and SVMCK has been shown to outperform the spectral-only SVM in HSI classification [9]. The parameters for KLR, KLRCK, SVM, and SVMCK are obtained by cross-validation.

The first hyperspectral image in our experiments is the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) image Indian Pines [36]. The AVIRIS sensor generates 220 bands across the spectral range from 0.2 to 2.4 μm. In the experiments, the number of bands is reduced to 200 by removing 20 water absorption bands. This image has spatial resolution of 20 m per pixel and spatial dimension 145×145. It contains 16 ground-truth classes. For each class, we randomly choose around 10% of the labeled samples for training and use the remaining 90% for testing, as seen in Table I and Fig. 1. Radial Basis Function (RBF) kernels are used in all kernel-based classifiers (i.e., SVM, SVMCK, KLR,

Input: $B \times N$ dictionary $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_N \end{bmatrix}$, $B \times T$ data matrix $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_T \end{bmatrix}$, kernel function κ, and a stopping criterion

---

Initialization: compute the kernel matrices $\boldsymbol{K}_{\boldsymbol{A}}$ in Algorithm 1 (Initialization) and $\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}} \in \mathbb{R}^{N \times T}$ whose (i,j)th entry is $\kappa(\boldsymbol{a}_i, \boldsymbol{x}_j)$. Set index set $\Lambda_0 = \Big\{ K_0$ indices corresponding to the K_0 largest numbers in $\left\| (\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}})_{i,:} \right\|_p$, $p \geq 1$, $i = 1, \ldots, N \Big\}$, and set iteration counter $t = 1$.

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

(1) Compute the correlation matrix

$$\boldsymbol{C} = \boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}} - (\boldsymbol{K}_{\boldsymbol{A}})_{:,\Lambda_{t-1}} \left((\boldsymbol{K}_{\boldsymbol{A}})_{\Lambda_{t-1},\Lambda_{t-1}} \right)^{-1} (\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}})_{\Lambda_{t-1},:} \in \mathbb{R}^{N \times T}$$

(2) Find the index set $I = \Big\{ K_0$ indices corresponding to the K_0 largest numbers in $\|\boldsymbol{C}_{i,:}\|_p$, $p \geq 1$, $i = 1, \ldots, N \Big\}$

(3) Update the candidate index set $\tilde{\Lambda}_t = \Lambda_{t-1} \bigcup I$

(4) Compute the projection coefficients $\boldsymbol{P} = \left((\boldsymbol{K}_{\boldsymbol{A}})_{\tilde{\Lambda}_t,\tilde{\Lambda}_t} \right)^{-1} (\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}})_{\tilde{\Lambda}_t,:} \in \mathbb{R}^{2K_0 \times T}$

(5) Update the index set $\Lambda_t = \Big\{ K_0$ indices in $\tilde{\Lambda}_t$ corresponding to the K_0 largest numbers in $\|\boldsymbol{P}_{i,:}\|_p$, $p \geq 1$, $i = 1, \ldots, N \Big\}$

(6) $t \leftarrow t + 1$

end while

Output: Index set $\Lambda = \Lambda_{t-1}$, the sparse representation $\hat{\boldsymbol{S}}'$ whose nonzero rows indexed by Λ are $\hat{\boldsymbol{S}}'_{\Lambda,:} = (\boldsymbol{K}_{\Lambda,\Lambda})^{-1} (\boldsymbol{K}_{\boldsymbol{A},\boldsymbol{X}})_{\Lambda,:}$

Algorithm 4: Kernelized Simultaneous Subspace Pursuit (KSSP)

KLRCK, KOMP, KSP, KSOMP, and KSSP). Since this image consists of large homogenous regions, a large spatial window of size 9×9 ($T = 81$) is used in classifiers with a composite kernel and the joint sparsity models (4) and (16).

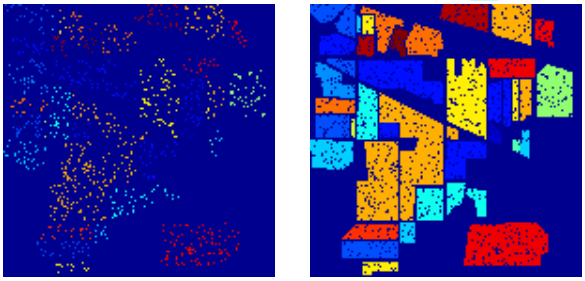
The classification accuracy for each class, overall accuracy (OA), average accuracy (AA), and the κ coefficient measure [37] on the test set are shown in Table II. The OA is computed by the ratio between correctly classified test samples and the total number of test samples, and the AA is the mean of the 16 class accuracies. The classification maps on labeled pixels are presented in Fig. 2. One can clearly see from Table II that the KSOMP and KSSP algorithms associated with the kernelized joint sparsity model yield the best classification performance for most of the classes. Incorporating the contextual correlation

and operating in the feature space both have significantly improved the classification accuracy.

TABLE I

THE 16 GROUND-TRUTH CLASSES IN AVIRIS INDIAN PINES.

Class		Samples	
No	Name	Train	Test
1	Alfalfa	6	48
2	Corn-notill	144	1290
3	Corn-min	84	750
4	Corn	24	210
5	Grass/Pasture	50	447
6	Grass/Trees	75	672
7	Grass/Pasture-mowed	3	23
8	Hay-windrowed	49	440
9	Oats	2	18
10	Soybeans-notill	97	871
11	Soybeans-min	247	2221
12	Soybean-clean	62	552
13	Wheat	22	190
14	Woods	130	1164
15	Building-Grass-Trees-Drives	38	342
16	Stone-steel Towers	10	85
Total		1043	9323



(a) (b)

Fig. 1. (a) Training and (b) test sets for Indian Pines.

Now we examine the effect of the sparsity level K_0 and the RBF kernel parameter γ in the proposed algorithms on the classification performance on Indian Pines. We use the same training and test set in Table I, and a 9×9 window for KSOMP and KSSP, and then vary K_0 from 5 to 80 and γ from 2^{-3} to

TABLE II

CLASSIFICATION ACCURACY (%) FOR INDIAN PINES.

Class	SVM	SVMCK	KLR	KLRCK	OMP	KOMP	SOMP	KSOMP	SP	KSP	SSP	KSSP
1	81.25	95.83	64.58	75.00	68.75	72.92	85.42	**97.92**	68.75	72.92	81.25	91.67
2	86.28	96.67	89.46	96.43	65.97	86.36	94.88	97.21	74.65	87.91	95.74	**97.98**
3	72.80	90.93	70.67	95.47	60.67	77.47	94.93	96.67	63.20	78.53	92.80	**97.73**
4	58.10	85.71	67.14	86.19	38.57	62.86	91.43	93.33	40.00	62.86	82.38	**96.67**
5	92.39	93.74	90.60	**96.42**	89.49	90.38	89.49	95.75	89.04	90.60	93.29	94.85
6	96.88	97.32	98.07	98.66	95.24	97.17	98.51	**99.55**	95.98	96.88	98.81	98.96
7	43.48	69.57	17.39	82.61	21.74	21.74	**91.30**	60.87	21.74	21.74	82.61	17.39
8	98.86	98.41	98.86	97.95	97.05	98.18	99.55	**100**	99.09	98.64	99.77	**100**
9	50.00	55.56	16.67	50.00	33.33	55.56	0	0	**61.11**	55.56	0	0
10	71.53	93.80	74.97	93.80	68.20	77.61	89.44	**94.60**	70.72	79.33	91.27	94.37
11	84.38	94.37	84.87	95.54	75.96	85.68	97.34	**99.28**	77.94	86.90	97.43	98.33
12	85.51	93.66	81.16	91.85	54.53	77.90	88.22	95.65	61.23	78.44	89.13	**97.46**
13	**100**	99.47	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.47	**100**
14	93.30	99.14	95.02	96.56	92.87	95.70	99.14	99.83	95.62	95.96	99.05	**99.91**
15	64.91	87.43	61.70	88.01	41.23	55.85	**99.12**	91.81	48.25	55.56	97.95	97.08
16	88.24	**100**	57.65	88.24	94.12	92.94	96.47	91.76	92.94	94.12	92.94	94.12
OA	84.52	94.86	84.78	95.10	74.78	85.26	95.28	97.33	78.10	86.09	95.34	**97.46**
AA	79.24	**90.73**	73.05	89.55	68.61	78.02	88.45	88.39	72.52	78.50	87.12	86.03
κ	0.823	0.941	0.826	0.944	0.712	0.832	0.946	0.970	0.749	0.841	0.947	**0.971**

2^{12} in KOMP, KSP, KSOMP, and KSSP. The OA on the test set are shown in Fig. 3. One can observe from Figs. 3(a) and (b) that for the pixel-wise kernel sparsity model, $\gamma = 512$ leads to the highest OA at all sparsity levels. For a fixed γ, the performance of KOMP and KSP generally improves as K_0 increases, and tends to saturate as K_0 reaches 30-50. For KSOMP and KSSP, as shown in Figs. 3(c) and (d), the same tendency cannot be observed. However, the kernel joint sparsity model is more stable than the pixel-wise model, as for a large range of sparsity level K_0 and sufficiently large γ, the overall accuracy is always around 96% with a small variance.

The next two hyperspectral images used in our experiments, University of Pavia and Center of Pavia, are urban images acquired by the Reflective Optics System Imaging Spectrometer (ROSIS). The ROSIS sensor generates 115 spectral bands ranging from 0.43 to 0.86 μm and has a spatial resolution of 1.3-meters per pixel [9]. The University of Pavia image consists of 610×340 pixels, each having 103 bands, with the 12 most noisy bands removed. There are nine ground-truth classes of interests, as shown in

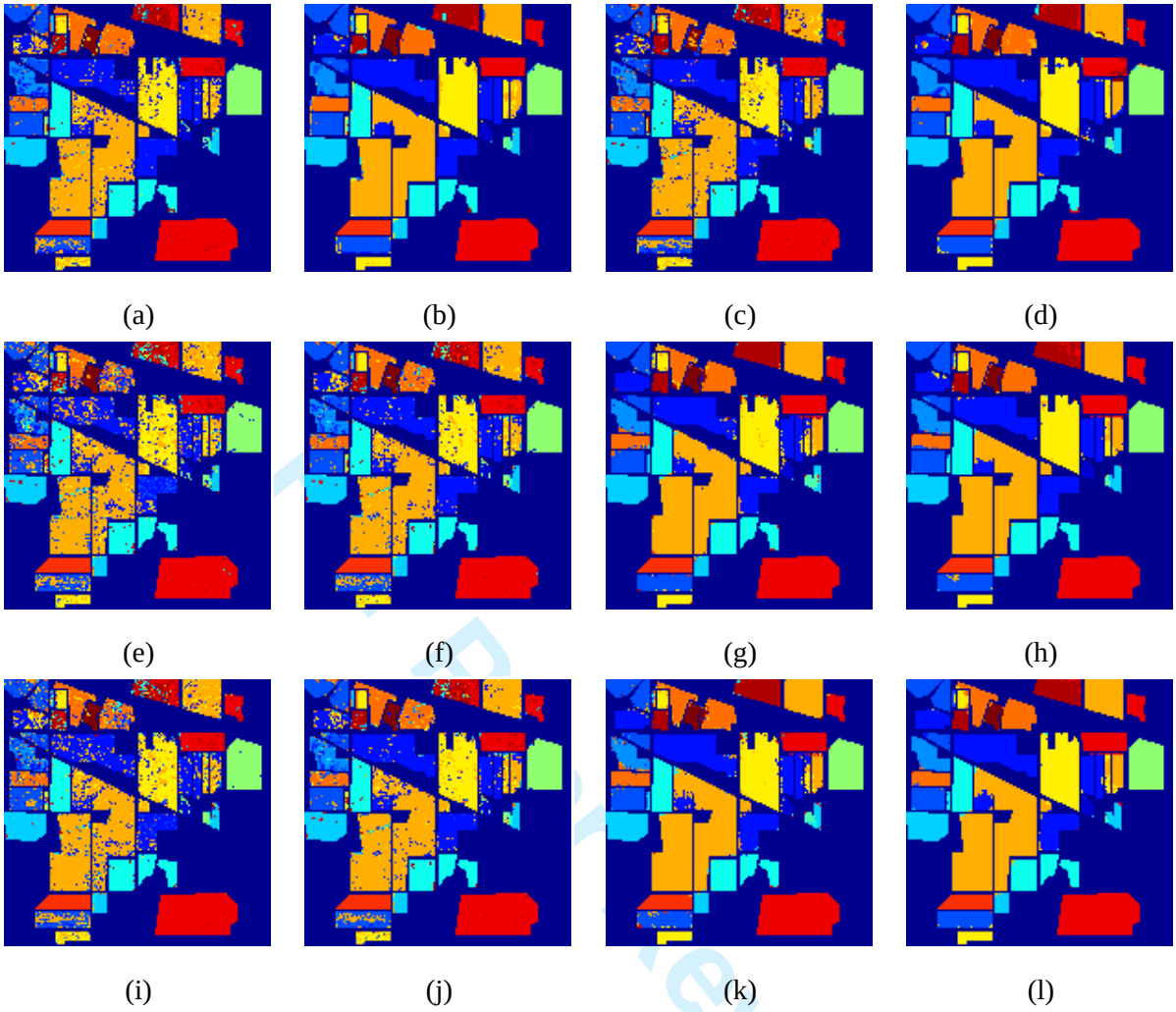


Fig. 2. Classification maps for Indian Pines using (a) SVM, (b) SVMCK, (c) KLR, (d) KLRCK, (e) OMP, (f) KOMP, (g) SOMP, (h) KSOMP, (i) SP, (j) KSP, (k) SSP, and (l) KSSP.

Table III. For this image, we follow the same experiment settings for the training and test sets as used in [7], in which about 9% of labeled data are used as training and the rest are used for testing, as shown in Table III and Fig. 4.

The classification accuracies and the κ coefficients on the test set using various techniques are shown in Table IV, and the classification maps for all labeled pixels are presented in Fig. 5. Again, the RBF kernel is used for all kernel-based algorithms. This urban image lacks the large spatial homogeneity and, therefore, a smaller neighborhood of size 5×5 is optimal for algorithms using a composite kernel, and the linear and kernel joint sparsity models. Similar to the Indian Pines image, the proposed KSOMP/KSSP algorithms achieve better or comparable performance when compared with the SVMCK classifier for

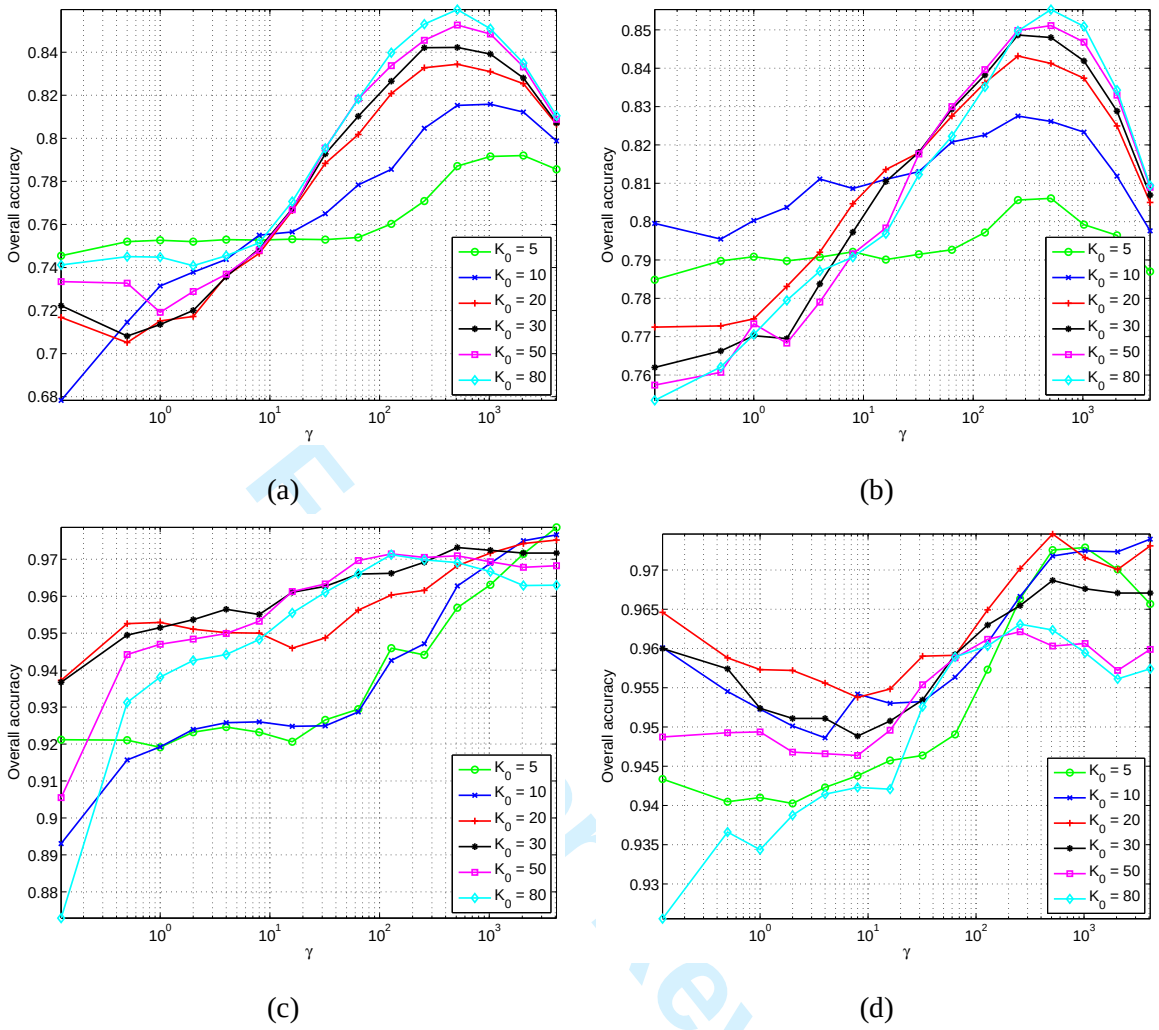


Fig. 3. Effect of sparsity level K_0 and RBF kernel parameter γ on Indian Pines using (a) KOMP, (b) KSP, (c) KSOMP, and (d) KSSP.

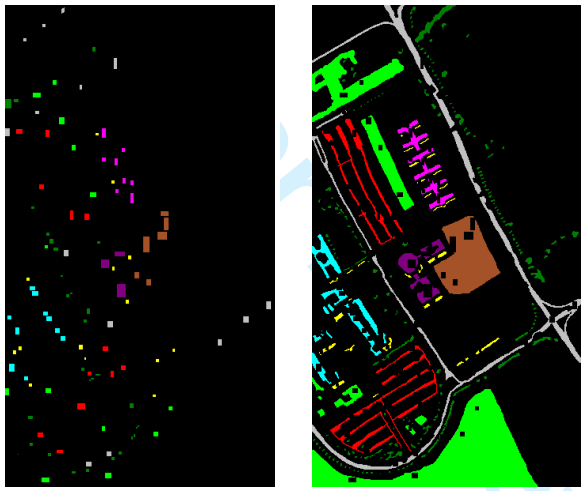
most of the classes. KSOMP and KSSP yield the best accuracy in five and three classes out of the total nine classes, respectively, and KSSP has the highest OA, AA, and κ coefficient. The accuracy for the second class representing Meadows, which contains more than 45% of the samples in the entire test set, for KSOMP and KSSP is 5%-9% lower than that for SVMCK and KLRCK.

In what follows, we examine how the number of training samples affects the classification performance for various algorithms on the Indian Pines and University of Pavia images. The algorithm parameters are fixed to be the same as those used to generate the results in Tables II and IV. For Indian Pines, in each test, we randomly choose 1% to 30% of the labeled data in each class as the training samples and the remaining samples as the test ones. The classification accuracy plots under various conditions are

TABLE III

THE 9 GROUND-TRUTH CLASSES IN UNIVERSITY OF PAVIA.

Class		Samples	
No	Name	Train	Test
1	Asphalt	548	6304
2	Meadows	540	18146
3	Gravel	392	1815
4	Trees	524	2912
5	Metal sheets	265	1113
6	Bare soil	532	4572
7	Bitumen	375	981
8	Bricks	514	3364
9	Shadows	231	795
Total		3921	40002



(a) (b)

Fig. 4. (a) Training and (b) test sets for University of Pavia.

shown in Fig. 6(a) for Indian Pines, where the x-axis denotes the percentage of training samples from the total available labeled samples, and the y-axis is the OA on the test set. The accuracies are averaged over five runs at each percentage level to avoid any bias induced by random sampling. For the University of Pavia image, we create a balanced dictionary by randomly choosing $L = 10, 20, 30, 50, 100$, and 200 training samples per class from the entire training set shown in Fig. 4(a). The classification accuracy plots using the sparsity-based algorithms are shown in Fig. 6(b), where the x-axis denotes the number of training samples per class, and the y-axis is the overall classification accuracy on the test set. Again,

TABLE IV

CLASSIFICATION ACCURACY (%) FOR UNIVERSITY OF PAVIA.

Class	SVM	SVMCK	KLR	KLRCK	OMP	KOMP	SOMP	KSOMP	SP	KSP	SSP	KSSP
1	84.30	79.85	82.96	74.40	68.23	76.09	59.33	**94.23**	69.78	76.67	69.59	89.56
2	67.01	84.86	83.34	**85.91**	67.04	69.61	78.15	76.74	67.90	70.92	72.31	79.98
3	68.43	81.87	64.13	61.71	65.45	72.12	83.53	79.23	69.20	73.39	74.10	**85.45**
4	97.80	96.36	96.33	96.22	97.29	98.11	96.91	95.12	96.77	98.15	95.33	**98.66**
5	99.37	99.37	99.19	99.10	99.73	99.73	99.46	**100**	99.64	99.82	99.73	99.91
6	92.45	93.55	80.05	84.45	73.27	87.66	77.41	**99.50**	78.96	89.70	86.72	95.76
7	89.91	90.21	84.51	85.32	87.26	88.07	98.57	**99.80**	88.18	88.28	90.32	97.96
8	92.42	92.81	83.17	93.37	81.87	89.51	89.09	**98.78**	83.68	87.54	90.46	96.43
9	97.23	95.35	89.81	96.48	95.97	93.96	91.95	29.06	94.59	95.22	90.94	**98.49**
OA	79.15	87.18	83.56	84.77	73.30	78.33	79.00	85.67	74.86	79.18	78.39	**87.65**
AA	87.66	90.47	84.83	86.33	81.79	86.10	86.04	85.83	83.19	86.63	85.50	**93.58**
κ	0.737	0.833	0.784	0.799	0.661	0.725	0.728	0.815	0.681	0.735	0.724	**0.840**

the accuracies are averaged over five runs at each L. It is obvious that in most cases the OA increases monotonically as the number of training samples increases. For University of Pavia, the performance at $L = 50$ is almost the same as that at $L = 100$ for all classifiers. The SVMCK classifier consistently outperforms all of the other classifiers when the number of training samples is small. It should also be pointed out that during the training stage of algorithms using a composite kernel (i.e., SVMCK and KLRCK), in order to extract the spatial features for each training sample, one requires knowledge of the neighboring pixels or the location of the training sample, which may not be available in the training set. Moreover, the proposed sparsity-based algorithms rely on the approximation accuracy from each class sub-dictionary. Therefore, if the size of the sub-dictionary is too small, the training samples may not be sufficient to faithfully represent the subspace associated with each class, leading to a lower classification accuracy than the discriminative classifier SVM.

The third image in our experiments, Center of Pavia, is the other urban image collected by the ROSIS sensor over the center of the Pavia city. This image consists of 1096×492 pixels, each having 102 spectral bands after 13 noisy bands are removed. The nine ground-truth classes and the number of training and test samples for each class are shown in Table V and illustrated in Fig. 7. For this image, about 5% of the labeled data are used as training samples. The classification results are summarized in Table VI, and the classification maps are shown in Fig. 8. KLRCK achieves a 100% accuracy on the first class of water, which occupies 66% of the test set, and thus yields the best OA. The KSOMP and KSSP work

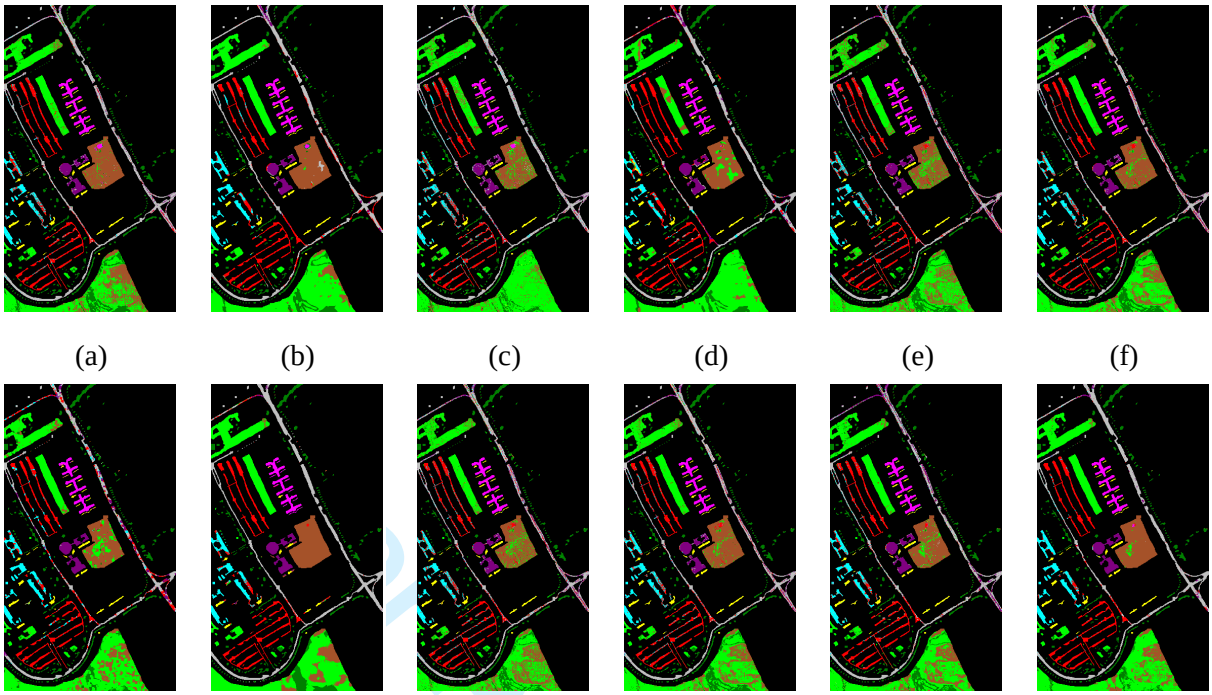


Fig. 5. Classification maps for University of Pavia using (a) SVM, (b) SVMCK, (c) KLR, (d) KLRCK, (e) OMP, (f) KOMP, (g) SOMP, (h) KSOMP, (i) SP, (j) KSP, (k) SSP, and (l) KSSP.

very well on the other classes, except that KSSP fails at the ninth class (Shadow).

In general, one can observe from the experimental results on these three images that the incorporation of contextual information improves the classification performance (e.g., SP vs. SSP, KSP vs. KSSP, SVM vs. SVMCK, etc). Moreover, operating in the feature space also significantly improve the accuracy (e.g., SP vs. KSP, SSP vs. KSSP, etc).

V. Conclusions

In this paper, we propose a new HSI classification technique based on sparse representations in a nonlinear feature space induced by a kernel function. The spatial correlation between neighboring pixels is incorporated through a joint sparsity model. Experimental results on AVIRIS and ROSIS hyperspectral images show that in most cases the proposed algorithm outperforms the conventional classifiers and the linear sparsity-based classification algorithms.

TABLE V

THE 9 GROUND-TRUTH CLASSES IN CENTER OF PAVIA AND THE TRAINING AND TEST SETS.

Class		Samples	
No	Name	Train	Test
1	Water	745	64533
2	Trees	785	5722
3	Meadow	797	2094
4	Brick	485	1667
5	Soil	820	5729
6	Asphalt	678	6847
7	Bitumen	808	6479
8	Tile	223	2899
9	Shadow	195	1970
Total		5536	97940

TABLE VI

CLASSIFICATION ACCURACY (%) FOR CENTER OF PAVIA.

Class	SVM	SVMCK	KLR	KLRCK	OMP	KOMP	SOMP	KSOMP	SP	KSP	SSP	KSSP
1	99.19	97.46	99.63	**100**	98.91	98.13	99.32	99.07	98.20	98.09	97.79	99.26
2	77.74	93.08	93.18	**95.39**	86.75	92.76	92.38	95.30	86.98	91.17	92.82	91.23
3	86.74	97.09	96.18	95.89	96.04	97.04	95.46	97.09	96.61	97.28	**97.80**	97.71
4	40.38	77.02	81.76	89.80	81.22	88.84	85.66	89.68	84.16	86.86	78.52	**95.26**
5	97.52	98.39	96.25	**98.59**	94.40	94.89	96.37	97.56	94.01	95.76	95.81	97.45
6	94.77	94.32	93.91	96.67	91.94	96.13	92.83	**98.31**	92.92	95.82	96.52	97.41
7	74.37	97.50	95.22	97.31	93.18	95.40	94.68	**98.80**	93.80	95.57	95.96	97.82
8	98.94	99.83	99.52	98.41	98.62	99.34	99.69	**99.93**	98.79	99.24	99.79	99.90
9	**100**	99.95	99.90	99.49	98.07	99.39	98.68	**100**	99.34	99.39	98.83	71.42
OA	94.63	96.81	97.99	**98.92**	96.68	97.19	97.66	98.53	96.40	97.08	96.93	97.82
AA	85.52	94.96	95.06	96.84	93.24	95.77	95.01	**97.30**	93.87	95.47	94.87	94.16
κ	0.899	0.943	0.963	**0.980**	0.940	0.949	0.958	0.973	0.935	0.947	0.945	0.960

VI. ACKNOWLEDGEMENT

The authors would like to thank the University of Pavia and the HySenS project for kindly providing the ROSIS images of University of Pavia and Center of Pavia, and Prof. J. M. Bioucas-Dias for providing the sparse multinomial logistic regression code.

REFERENCES

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. Fifth Annual Workshop on Computational Learning Theory*, 1992, pp. 144–152.

[2] V. N. Vapnik, *The nature of statistical learning theory*, Springer, 1995.

[3] J. A. Gualtieri and R. F. Cromp, "Support vector machines for hyperspectral remote sensing classification," in *Proc. SPIE*, Jan. 1998, vol. 3584, pp. 221–232.

[4] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[5] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for the semisupervised classification of remote sensing images," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3363–3373, Nov. 2006.

[6] F. Bovolo, L. Bruzzone, and M. Marconcini, "A novel context-sensitive SVM for classification of remote sensing images," in *Proc. of IEEE International Geoscience and Remote Sensing Symposium*, July 2006, pp. 2498–2501.

[7] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, "Spectra-spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2973–2987, Aug. 2009.

[8] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 93–97, Jan. 2006.

[9] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sensing of Environment*, vol. 113, Supplement 1, pp. S110–S122, Sept. 2009.

[10] D. Böhning, "Multinomial logistic regression algorithm," *Ann. Inst. Stat. Math.*, vol. 44, no. 1, pp. 197–200, Mar. 1992.

[11] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 957–968, June 2005.

[12] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4085–4098, Nov. 2010.

[13] H.-Y. Huang and B.-C. Kuo, "Double nearest proportion feature extraction for hyperspectral-image classification," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4034–4046, Nov. 2010.

[14] H. R. Kalluri, S. Prasad, and L. M. Bruce, "Decision-level fusion of spectral reflectance and derivative information for robust hyperspectral land cover classification," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4047–4058, Nov. 2010.

[15] B. D. Bue, E. Merényi, and B. Csathó, "Automated labeling of materials in hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4059–4070, Nov. 2010.

[16] B. Demir and S. Ertürk, "Empirical mode decomposition of hyperspectral images for support vector machine classification," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4071–4084, Nov. 2010.

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

[18] W. Kim and M. M. Crawford, "Adaptive classification for hyperspectral image data using manifold regularization kernel machines," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4110–4121, Nov. 2010.

[19] Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Multiple spectralcspatial classification approach for hyperspectral data," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4122–4132, Nov. 2010.

[20] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[21] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 2790–2797.

[22] J. Yang, J. Wright, T. Huang, and Y. Ma;, "Image super-resolution as sparse representation of raw image patches," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[23] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. on Image Processing*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[24] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[25] J. Wright, A. Y. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[26] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Simultaneous joint sparsity model for target detection in hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, July 2011.

[27] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, June 2005.

[28] H. Kwon and N. M. Nasrabadi, "A comparative analysis of kernel subspace target detectors for hyperspectral imagery," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 193–193, Jan. 2007.

[29] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Machine Learning*, vol. 48, pp. 165–187, 2002.

[30] V. Guigue, A. Rakotomamonjy, and S. Canu, "Kernel basis pursuit," *New Methods in Machine Learning*, vol. 20, no. 6, pp. 757–774, 2006.

[31] S. Gao, I. W. Tsang, and L.-T. Chia, "Kernel sparse representation for image classification and face recognition," in *Proc. of the 11th European Conference on Computer Vision*, 2010, vol. 6314.

[32] X.-T. Yuan and S. Yan, "Visual classification with multi-task joint sparse representation," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June 2010.

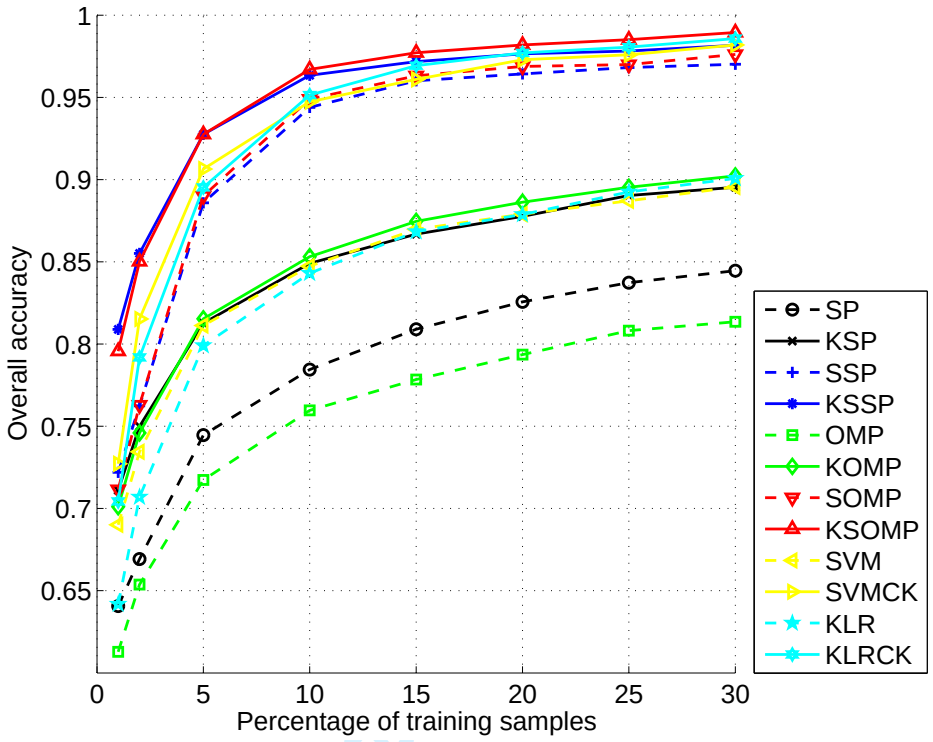
[33] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Processing, special issue on Sparse approximations in signal and image processing*, vol. 86, pp. 572–588, Mar. 2006.

[34] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

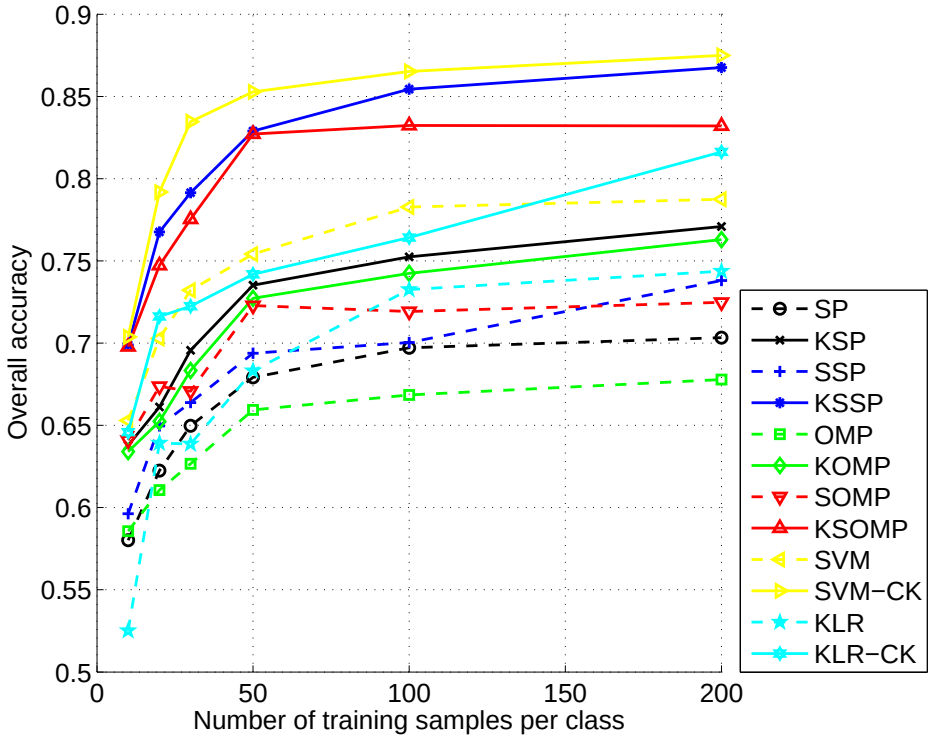
[35] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. on Information Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[36] "AVIRIS NW Indiana's Indian Pines 1992 data set," http://cobweb.ecn.purdue.edu/~biehl/MultiSpec/documentation.html.

[37] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*, Springer, fourth edition, 2006.

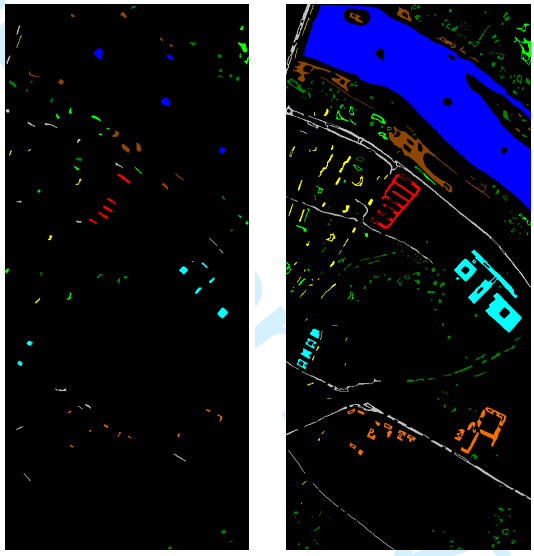


(a)



(b)

Fig. 6. Effect of dictionary size for (a) Indian Pines and (b)University of Pavia.



(a) (b)

Fig. 7. (a) Training and (b) test sets for Center of Pavia.

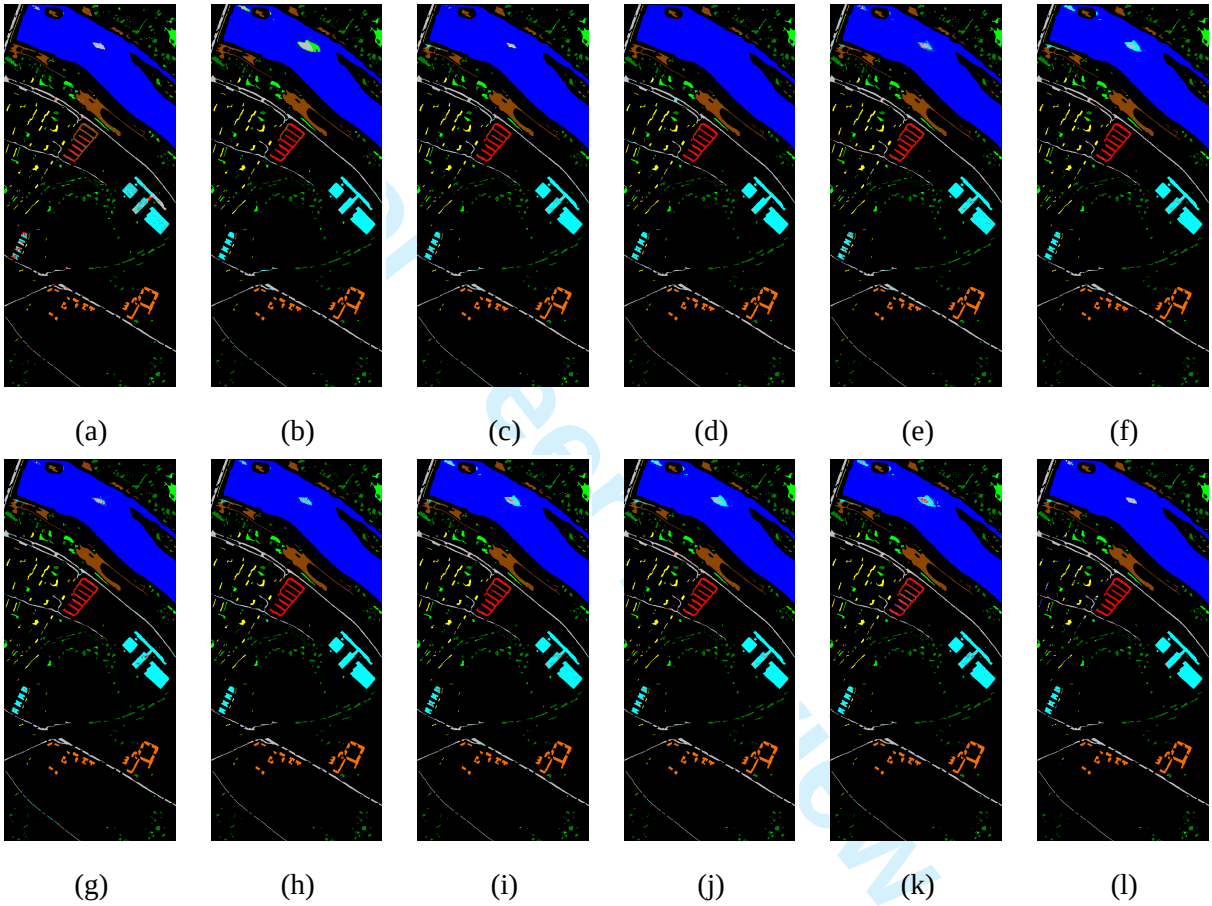


Fig. 8. Classification maps for Center of Pavia using (a) SVM, (b) SVMCK, (c) KLR, (d) KLRCK, (e) OMP, (f) KOMP, (g) SOMP, (h) KSOMP, (i) SP, (j) KSP, (k) SSP, and (l) KSSP.