International Journal of
Geo-Information

MDPI

Article

Spatial-Spectral Graph Regularized Kernel Sparse Representation for Hyperspectral Image Classification

Jianjun Liu [1,*], Zhiyong Xiao [1], Yufeng Chen [2] and Jinlong Yang [1]

[1] School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China; zhiyong.xiao@jiangnan.edu.cn (Z.X.); yjlgedeng@163.com (J.Y.)

[2] School of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, China; chenyf_cn@189.cn

* Correspondence: liuofficial@163.com

Academic Editor: Wolfgang Kainz
Received: 12 June 2017; Accepted: 18 August 2017; Published: 22 August 2017

Abstract: This paper presents a spatial-spectral method for hyperspectral image classification in the regularization framework of kernel sparse representation. First, two spatial-spectral constraint terms are appended to the sparse recovery model of kernel sparse representation. The first one is a graph-based spatially-smooth constraint which is utilized to describe the contextual information of hyperspectral images. The second one is a spatial location constraint, which is exploited to incorporate the prior knowledge of the location information of training pixels. Then, an efficient alternating direction method of multipliers is developed to solve the corresponding minimization problem. At last, the recovered sparse coefficient vectors are used to determine the labels of test pixels. Experimental results carried out on three real hyperspectral images point out the effectiveness of the proposed method.

Keywords: classification; sparse representation; hyperspectral image; kernel; regularization

1. Introduction

Hyperspectral imaging sensors capture digital images in hundreds of narrow and contiguous spectral bands spanning the visible to infrared spectrum. Each pixel of a hyperspectral image can be represented by a vector whose entries correspond to various spectral-band responses. Different materials usually reflect electromagnetic energy differently at specific wavelengths. The wealth of spectral information of hyperspectral images promotes the development of many application domains, such as military [1], agriculture [2], and mineralogy [3]. Among the information processing procedures of these applications, classification is an important one, where pixels are assigned to one of the available classes according to a set of given training pixels. Various classifiers have been developed for hyperspectral image classification and have shown good performance. Some popular classifiers include support vector machine (SVM) [4], multinomial logistic regression [5], and sparse representation classification (SRC) [6].

Hyperspectral images usually have large homogeneous regions within which the neighboring pixels consist of the same type of materials and have similar spectral characteristics. Previous works have highlighted that hyperspectral image classification should not only focus on analyzing spectral features, but also incorporate the information of spatially-adjacent pixels. For instance, the authors in [7] incorporate the spectral and spatial information into SVM by taking advantage of the composite kernel technique. In [5], the spatial information is incorporated into multinomial logistic regression by some spatial feature extraction techniques. In [6], a joint sparsity model is introduced to SRC, where the neighborhood pixels are assumed to have the same labels. Very recently, some deep models (e.g., stacked autoencoder [8] and convolutional neural network [9–12]) have been proposed for the spatial-spectral classification of hyperspectral images and have shown good performance in terms of accuracy and flexibility.

Among the aforementioned classifiers, SRC is an excellent one and can directly assign a class label to a test pixel without a training process. Although SRC can achieve good performance in hyperspectral image classification [6], it is hard to classify the data that is not linearly separable. To overcome this drawback, kernel SRC (KSRC) is proposed in [13] to capture the nonlinear similarity of features. In view of this, this paper considers using KSRC for the classification of hyperspectral images. However, KSRC is also a pixel-wise classifier, which does not consider the spectral similarity of spatially-adjacent pixels. In the literature, some methods have been proposed to incorporate the spatial information into KSRC. In [13], a joint sparsity model is introduced to KSRC, which can be seen as a kernel extension of [6]. In [14], the spatial-spectral information is incorporated into KSRC by taking advantage of the neighboring filtering kernel technique. In [15], a centralized quadratic constraint that describes the spectral similarity of spatially-adjacent pixels is appended to the sparse recovery model of KSRC.

This paper considers incorporating the spatial-spectral information of hyperspectral images in the regularization framework of KSRC. Different from the relevant methods, a spatial-spectral graph is designed to describe the contextual information of hyperspectral images, where the spectral similarity of spatially-adjacent pixels (i.e., nodes) is measured by the weight of the edge. In addition, the acquisition of labeled pixels in most hyperspectral scenes is often an expensive procedure. Thus, the number of training pixels that can be used for hyperspectral image classification is often very limited. To deal with this problem, we assume that the training pixels are labeled by experts and then exploit the wealth of these training pixels. In detail, a spatial location constraint is introduced to capture the prior knowledge of the location information of training pixels and is thereby appended to the sparse recovery model of KSRC. As for the final classification procedure, the sparse coefficient vectors obtained by solving the corresponding regularization problem are used to determine the labels of test pixels.

The remainder of this paper is organized as follows. Section 2 briefly reviews the formulations of KSRC. In Section 3, we present the formulations of the proposed SSGL method and the corresponding optimization algorithm. The effectiveness of the proposed method is demonstrated in Section 4 by conducting experiments on three real hyperspectral images. Finally, Section 5 discusses and concludes this paper.

2. KSRC

This section briefly introduces the notations of kernel sparse representation for hyperspectral image classification. As with the assumption of SRC, KSRC also assumes that the features belonging to the same class approximately lie in the same low-dimensional subspace [16,17]. Suppose the given hyperspectral scene includes C classes, and that there exists a feature mapping function ϕ which maps an L-band test pixel $\mathbf{x} \in \mathbb{R}^L$ and a J-sized training dictionary $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_J] \in \mathbb{R}^{L \times J}$ to the high-dimensional feature space: $\mathbf{x} \rightarrow \phi(\mathbf{x})$, $\mathbf{A} \rightarrow \Phi(\mathbf{A}) = [\phi(\mathbf{a}_1), ..., \phi(\mathbf{a}_J)]$, where the columns of $\mathbf{A}$ are composed of J training pixels from all classes. Then, for an unknown test pixel $\mathbf{x} \in \mathbb{R}^L$, it can be represented as follows:

$$\phi(\mathbf{x}) = \Phi(\mathbf{A})\mathbf{s}, \tag{1}$$

where $\mathbf{s} \in \mathbb{R}^J$ denotes an unknown sparse coefficient vector.

Suppose the dictionary $\mathbf{A}$ is given; the unknown sparse coefficient vector $\mathbf{s}$ can be obtained by solving the following optimization problem:

$$\min_{\mathbf{s}} \frac{1}{2}||\phi(\mathbf{x}) - \Phi(\mathbf{A})\mathbf{s}||_2^2 + \lambda||\mathbf{s}||_1, \tag{2}$$

where $\lambda > 0$ is used to control the sparsity of $\mathbf{s}$. Then, the class label of $\mathbf{x}$ can be determined by the following classification rule:

$$\text{class}(\mathbf{x}) = \arg \min_{c=1,\ldots,C} ||\phi(\mathbf{x}) - \Phi(\mathbf{A})\delta_c(\mathbf{s})||_2, \tag{3}$$

where $\delta_c(\cdot)$ is the characteristic function that selects coefficients related with the c-th class and makes the rest zero.

Since all ϕ mappings used in KSRC occur in the form of inner products, the kernel function K can be defined for any two pixels $\mathbf{x}_i \in \mathbb{R}^L$ and $\mathbf{x}_j \in \mathbb{R}^L$ as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle . \tag{4}$$

Some commonly-used kernel functions are: linear kernel ($K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$), polynomial kernel ($K(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^d$, $d \in \mathbb{Z}^+$), and Gaussian radial basis function (RBF) kernel ($K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||_2^2)$, $\gamma \in \mathbb{R}^+$). In this paper, only the RBF kernel is considered. After introducing (4) into (2), the corresponding optimization problem can be rewritten as follows:

$$\min_{\mathbf{s}} \frac{1}{2}\mathbf{s}^T\mathbf{Q}\mathbf{s} - \mathbf{s}^T\mathbf{p} + \lambda||\mathbf{s}||_1 + \text{Const}, \tag{5}$$

where Const $= (1/2)K(\mathbf{x}, \mathbf{x})$ is a constant that can be dropped in the optimization problem (5), $\mathbf{Q} = \langle \Phi(\mathbf{A}), \Phi(\mathbf{A}) \rangle$ is a $J \times J$ positive semi-definite matrix with entry $\mathbf{Q}_{ij} = K(\mathbf{a}_i, \mathbf{a}_j)$, and $\mathbf{p} = [K(\mathbf{a}_1, \mathbf{x}), ..., K(\mathbf{a}_J, \mathbf{x})]^T \in \mathbb{R}^J$. Analogously, the classification rule (3) can be rewritten as follows:

$$\text{class}(\mathbf{x}) = \arg \min_{c=1,\ldots,C} \delta_c^T(\mathbf{s})\mathbf{Q}\delta_c(\mathbf{s}) - 2\delta_c^T(\mathbf{s})\mathbf{p}. \tag{6}$$

3. Proposed Approach

3.1. Proposed Model

Suppose the observed hyperspectral image $\mathbf{X}$ consists of I pixels $\{\mathbf{x}_i \in \mathbb{R}^L\}_{i=1}^I$; i.e., $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_I] \in \mathbb{R}^{L \times J}$. Then, the corresponding sparse recovery model of (2) for the observed hyperspectral image $\mathbf{X}$ can be written as follows:

$$\min_{\mathbf{S}} \frac{1}{2}||\Phi(\mathbf{X}) - \Phi(\mathbf{A})\mathbf{S}||_F^2 + \lambda||\mathbf{S}||_{1,1}, \tag{7}$$

where $||\cdot||_F$ is the Frobenius norm, $||\mathbf{S}||_{1,1} = \sum_{j,i}|\mathbf{S}_{ji}|$, $\Phi(\mathbf{X}) = [\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_I)]$ denotes the corresponding data of $\mathbf{X}$ under the mapping function ϕ and $\mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_I] \in \mathbb{R}^{J \times I}$ is the unknown sparse coefficient matrix, with $\mathbf{s}_i \in \mathbb{R}^J$ being the sparse coefficient vector of pixel $\mathbf{x}_i$. Similarly, the corresponding sparse recovery model of (5) can be written as follows:

$$\min_{\mathbf{S}} \frac{1}{2}\text{Tr}(\mathbf{S}^T\mathbf{Q}\mathbf{S}) - \text{Tr}(\mathbf{S}^T\mathbf{P}) + \lambda||\mathbf{S}||_{1,1}, \tag{8}$$

where the constant term is dropped and $\text{Tr}(\cdot)$ denotes the trace of a matrix, $\mathbf{P} = \langle \Phi(\mathbf{A}), \Phi(\mathbf{X}) \rangle = [\mathbf{p}_1, ..., \mathbf{p}_I] \in \mathbb{R}^{J \times I}$ with entry $\mathbf{P}_{ij} = K(\mathbf{a}_i, \mathbf{x}_j)$. The corresponding classification rule of (6) can be written as follows:

$$\text{class}(\mathbf{x}_i) = \arg \min_{c=1,\ldots,C} \delta_c^T(\mathbf{s}_i)\mathbf{Q}\delta_c(\mathbf{s}_i) - 2\delta_c^T(\mathbf{s}_i)\mathbf{p}_i, \forall i = 1, 2, \cdots, I. \tag{9}$$

The matrix $\mathbf{X}$ is not just a set of pixels; it is essentially a three-dimensional image; since for any i in the complete set $\mathcal{I} = \{1, 2, \cdots, I\}$, $\mathbf{s}_i$ is the sparse coefficient vector of pixel $\mathbf{x}_i$. Then, the matrix $\mathbf{S}$ can also be seen as a three-dimensional image. That is to say, the spatial relationship between every two pixels $\mathbf{x}_i$ and $\mathbf{x}_j$ is also suitable for that of the sparse coefficient vectors $\mathbf{s}_i$ and $\mathbf{s}_j$. It is natural to incorporate the spatial information of $\mathbf{X}$ by enforcing $\mathbf{S}$. However, directly enforcing $\mathbf{S}$ is too strict; it does not consider the variation of training pixels within each class. In view of this, the summation of sparse coefficients that belong to each class is considered. Specifically, the spatial information is incorporated by enforcing **TS**, where $\mathbf{T} \in \mathbb{R}^{C \times I}$ is defined as:

$$\mathbf{T}_{cj} = \begin{cases} 1 & \text{if class}(\mathbf{a}_j) = c \\ 0 & \text{else} \end{cases}, \forall c = 1, 2, \cdots, C; j = 1, 2, \cdots, J. \tag{10}$$

Let us define a spatial-spectral graph as $\mathcal{G} = \{\mathbf{V}, \mathbf{E}, \mathbf{W}\}$, where $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_I]$ and **E** are the sets of vertices and edges, respectively, and $\mathbf{W} \in \mathbb{R}^{I \times I}$ is a weight matrix on the graph. In this paper, every node $\mathbf{v}_i$ is connected to its eight spatially-adjacent nodes (see Figure 1), and the weight between $\mathbf{v}_i$ and $\mathbf{v}_i$ is defined as follows:

$$\mathbf{W}_{ij} = \begin{cases} \exp(-\beta||\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j||_2) + 10^{-6} & \text{if } \mathbf{v}_i \text{ and } \mathbf{v}_i \text{ are connected} \\ 0 & \text{else} \end{cases}, \forall i, j \tag{11}$$

where $\beta > 0$ and $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_j$ are the pixels of the first three principal components of the hyperspectral image **X**. It is apparent that for any two similar nodes, a large weight will be given. In this paper, the graph-based spatially-smooth constraint term is defined as follows:

$$\mathcal{S}(\mathbf{TS}) = \mathrm{Tr}\left((\mathbf{TS})\mathbf{L}(\mathbf{TS})^T\right), \tag{12}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian and **D** is a diagonal matrix whose entries are column sums of **W** (i.e., $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$). After appending (12) to the sparse recovery model (8), we can get the following optimization problem:

$$\min_{\mathbf{S}} \frac{1}{2}\mathrm{Tr}(\mathbf{S}^T\mathbf{QS}) - \mathrm{Tr}(\mathbf{S}^T\mathbf{P}) + \lambda||\mathbf{S}||_{1,1} + \frac{\alpha}{2}\mathrm{Tr}\left((\mathbf{TS})\mathbf{L}(\mathbf{TS})^T\right), \tag{13}$$

where $\alpha > 0$. In order to effectively work with limited training pixels, the prior knowledge of the location information of training pixels is included by a spatial location constraint term. The final sparse recovery model can be written as follows:

$$\min_{\mathbf{S}} \frac{1}{2}\mathrm{Tr}(\mathbf{S}^T\mathbf{QS}) - \mathrm{Tr}(\mathbf{S}^T\mathbf{P}) + \lambda||\mathbf{S}||_{1,1} + \frac{\alpha}{2}\mathrm{Tr}\left((\mathbf{TS})\mathbf{L}(\mathbf{TS})^T\right) + \iota_0(\mathbf{TS}_\Lambda - \mathbf{T}), \tag{14}$$

where $\Lambda \subset \mathcal{I}$ is a set with $\mathbf{S}_\Lambda$ being the corresponding coefficient matrix of the labeled pixels **A** and $\iota_0(\cdot)$ denotes the indicator function; i.e., $\iota_0(\cdot)$ is zero if the given matrix is equal to zero, and infinity otherwise. By using the location information of training pixels in the model (14), these training pixels can be treated as anchor samples whose coefficients are fixed throughout the classification process, and the graph-based spatially-smooth constraint can spread their label information to their neighbors until a global stable state is achieved on the whole data set.

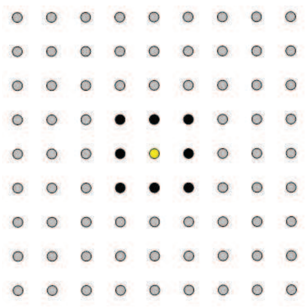


Figure 1. Illustration of the spatially-adjacent nodes. The yellow one denotes the given node and the black ones denote the eight spatially-adjacent nodes.

3.2. Optimization Algorithm

By introducing two variables $\mathbf{M} \in \mathbb{R}^{J \times I}$ and $\mathbf{N} \in \mathbb{R}^{C \times I}$, the proposed optimization model (14) can be rewritten as follows:

$$\begin{gathered}\min_{\mathbf{S},\mathbf{M},\mathbf{N}} f(\mathbf{S},\mathbf{M},\mathbf{N}) \\ \text{s.t.}\mathbf{M} = \mathbf{S}, \mathbf{N} = \mathbf{TS}.\end{gathered} \tag{15}$$

where
$$f(\mathbf{S},\mathbf{M},\mathbf{N}) = \frac{1}{2}\mathrm{Tr}(\mathbf{S}^T\mathbf{QS}) - \mathrm{Tr}(\mathbf{S}^T\mathbf{P}) + \lambda||\mathbf{M}||_{1,1} + \frac{\alpha}{2}\mathrm{Tr}(\mathbf{NLN}^T) + \iota_0(\mathbf{N}_\Lambda - \mathbf{T}). \tag{16}$$

The optimization problem (15) accords with the framework of the alternating direction method of multipliers (ADMM) [18,19]. The augmented Lagrangian function of (15) can be written as follows:

$$\mathcal{L}(\mathbf{S},\mathbf{M},\mathbf{N},\mathbf{H},\mathbf{B}) = f(\mathbf{S},\mathbf{M},\mathbf{N}) + \frac{\mu}{2}||\mathbf{S} - \mathbf{M} - \mathbf{H}||_F^2 + \frac{\mu}{2}||\mathbf{TS} - \mathbf{N} - \mathbf{B}||_F^2, \tag{17}$$

where $\mu > 0$ is the penalty parameter and $\mathbf{H} \in \mathbb{R}^{J \times I}$ and $\mathbf{B} \in \mathbb{R}^{C \times I}$ are two auxiliary variables. The corresponding iteration procedure can be written as follows:

$$\begin{cases} \mathbf{S}^t = \arg\min_{\mathbf{S}} \mathcal{L}(\mathbf{S},\mathbf{M}^{t-1},\mathbf{N}^{t-1},\mathbf{H}^{t-1},\mathbf{B}^{t-1}) \\ \mathbf{M}^t = \arg\min_{\mathbf{M}} \mathcal{L}(\mathbf{S}^t,\mathbf{M},\mathbf{N}^{t-1},\mathbf{H}^{t-1},\mathbf{B}^{t-1}) \\ \mathbf{N}^t = \arg\min_{\mathbf{N}} \mathcal{L}(\mathbf{S}^t,\mathbf{M}^t,\mathbf{N},\mathbf{H}^{t-1},\mathbf{B}^{t-1}) \\ \mathbf{H}^t = \mathbf{H}^{t-1} - (\mathbf{S}^t - \mathbf{M}^t) \\ \mathbf{B}^t = \mathbf{B}^{t-1} - (\mathbf{TS}^t - \mathbf{N}^t) \end{cases} \tag{18}$$

where $t > 0$ is the iteration number. The first step of (18) is to solve the **S**-subproblem, which can be derived as:

$$\mathbf{S}^t = (\mathbf{Q} + \mu\mathbf{I} + \mu\mathbf{T}^T\mathbf{T})^{-1}(\mathbf{P} + \mu(\mathbf{M}^{t-1} + \mathbf{H}^{t-1}) + \mu\mathbf{T}^T(\mathbf{N}^{t-1} + \mathbf{B}^{t-1})), \tag{19}$$

where **I** is the identity matrix. The second step of (18) is to solve the **M**-subproblem, which is the well-known soft threshold [20]:

$$\mathbf{M}^t = \mathrm{soft}(\mathbf{S}^t - \mathbf{H}^{t-1}, \lambda/\mu), \tag{20}$$

where $\mathrm{soft}(\cdot,\tau)$ denotes the component-wise application of the soft-threshold function $y = \mathrm{sign}(y)\max\{|y| - \tau, 0\}$. The third step of (18) is to solve a linear system, which can be written as:

$$\min_{\mathbf{N}} \frac{1}{2}\|\mathbf{N} - (\mathbf{TS}^t - \mathbf{B}^{t-1})\|_F^2 + \frac{\alpha}{2\mu}\mathrm{Tr}(\mathbf{NLN}^T) + \iota_0(\mathbf{N}_\Lambda - \mathbf{T}). \tag{21}$$

If removing the last term, the solution of this system can be written as:

$$\mathbf{N}^t = (\mathbf{TS}^t - \mathbf{B}^{t-1})\left(\frac{\alpha}{\mu}\mathbf{L} + \mathbf{I}\right)^{-1}. \tag{22}$$

Let $\bar{\Lambda}$ be the complementary set of Λ under the complete set $\mathcal{I}$. Then, the optimization problem (21) can be rewritten as:

$$\min_{\mathbf{N}_{\bar{\Lambda}}} \frac{1}{2}\|[\mathbf{N}_\Lambda\mathbf{N}_{\bar{\Lambda}}] - (\mathbf{T}[\mathbf{S}_\Lambda^t\mathbf{S}_{\bar{\Lambda}}^t] - [\mathbf{B}_\Lambda^{t-1}\mathbf{B}_{\bar{\Lambda}}^{t-1}])\|_F^2 + \frac{\alpha}{2\mu}\mathrm{Tr}\left([\mathbf{N}_\Lambda\mathbf{N}_{\bar{\Lambda}}]\begin{bmatrix}\mathbf{L}_{\Lambda\Lambda} & \mathbf{L}_{\Lambda\bar{\Lambda}} \\ \mathbf{L}_{\bar{\Lambda}\Lambda} & \mathbf{L}_{\bar{\Lambda}\bar{\Lambda}}\end{bmatrix}[\mathbf{N}_\Lambda\mathbf{N}_{\bar{\Lambda}}]^T\right), \tag{23}$$

where $\mathbf{L}_{\Lambda\bar{\Lambda}} = \mathbf{L}_{\bar{\Lambda}\Lambda}^T \in \mathbb{R}^{J \times (I-J)}$ and $\mathbf{N}_\Lambda = \mathbf{T}$. The solution of (23) can be derived as:

$$\mathbf{N}_{\bar{\Lambda}} = \left(\mathbf{TS}_{\bar{\Lambda}}^t - \mathbf{B}_{\bar{\Lambda}}^{t-1} - \frac{\alpha}{\mu}\mathbf{N}_\Lambda\mathbf{L}_{\Lambda\bar{\Lambda}}\right)\left(\frac{\alpha}{\mu}\mathbf{L}_{\bar{\Lambda}\bar{\Lambda}} + \mathbf{I}\right)^{-1}. \tag{24}$$

For both (22) and (24), one has to solve a large linear system. Fortunately, **L** is a sparse and diagonally dominant matrix, and one can take the Gauss–Seidel method to get approximate solutions to these two systems. The last two steps of (18) are designed to update the auxiliary variables. The procedure of the proposed algorithm is detailed as follows:

1. **Input**: A training dictionary $\mathbf{A} \in \mathbb{R}^{L\times J}$ and a hyperspectral data matrix $\mathbf{X} \in \mathbb{R}^{L\times I}$.
2. Choose β, and compute the weight matrix **W** according to (11).
3. Select the γ parameter for the RBF kernel and compute the matrices **Q** and **P**.
4. Set $t = 1$; choose μ, λ, α, $\mathbf{S}^1$, $\mathbf{M}^1$, $\mathbf{N}^1$, $\mathbf{H}^1$, $\mathbf{B}^1$.
5. **Repeat**
6. Compute $\mathbf{S}^t$, $\mathbf{M}^t$, $\mathbf{N}^t$, $\mathbf{H}^t$, $\mathbf{B}^t$ using (18).
7. $t = t + 1$.
8. **Until** some stopping criterion is satisfied.
9. **Output**: The estimated label of $\mathbf{x}_i$ using (6), $i = 1, ..., I$.

3.3. Analysis and Comparison

The proposed spatial-spectral graph regularization with location (SSGL) method is designed by appending two spatial-spectral constraint terms to the sparse recovery model of KSRC. One is a graph-based spatially-smooth constraint, and the other is a spatial location constraint. When using the two terms, the following two issues should be mentioned.

1. The graph-based spatially-smooth constraint is proposed by measuring the spatial relationship between every two spatially-adjacent pixels. If using this term, the test pixels should be arranged in the form of images.
2. The spatial location constraint is exploited by integrating the location information of anchor samples, which are assumed to be taken from the test area and labeled by experts.

As described in the introduction part, deep models attract a lot of attention and provide competitive classification results very recently. Although the proposed SSGL method and the deep learning methods belong to two distinct techniques, we show the difference between them considering a further improvement of this work. In Table 1, we compare the proposed SSGL method and the deep learning methods in the respects of parameter, training sample, computational cost, flexibility, and accuracy.

Table 1. Difference between the proposed spatial-spectral graph regularization with location information (SSGL) method and the deep learning methods.

	Deep Learning Methods	**SSGL**
No. of parameters	Several parameters needed to be trained.	Four model parameters: β, γ, λ and α. One algorithm parameter μ.
No. of training samples	Abundant training samples needed.	Moderate (or even limited) training samples needed.
Computational cost	Expensive	Moderate
Flexibility	Robust to the spectral distortion.	Restricted to the test area.
Accuracy	High	Moderate

4. Experimental Results

4.1. Datasets

To test the performance of the proposed method, three real hyperspectral images have been considered.

(1) Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Indian Pines: The first image in our experiments deals with the standard AVIRIS image taken over northwest Indiana's Indian Pine test site in June 1992. There are 220 bands in the image, covering the wavelength range of 0.4–2.5 µm. The spectral and spatial resolutions are 10 nm and 17 m, respectively. This image consists of 145 × 145 pixels and 16 ground truth classes ranging from 20–2468 pixels in size. The false color image and the ground truth map are shown in Figure 2. We removed 20 noisy bands covering the region of water absorption and worked with 200 spectral bands. In the experiments, about 5% of the labeled pixels were randomly chosen for training, and the rest were used for testing, as shown in Table 2.

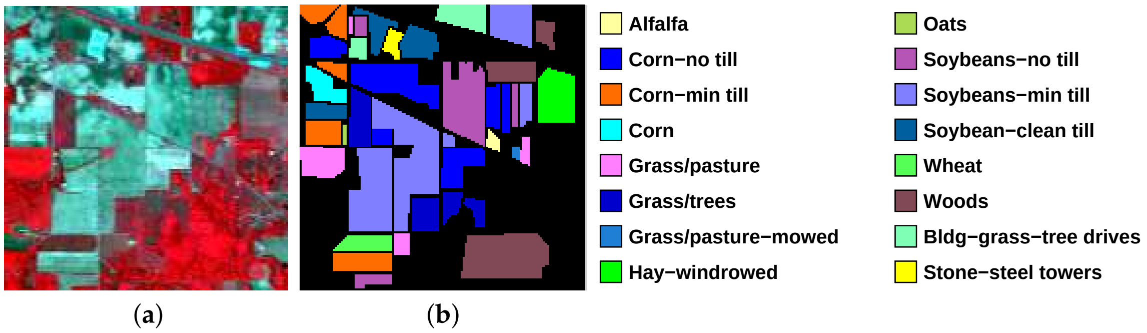


Figure 2. Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Indian Pines data set. (**a**) RGB composite image of three bands. (**b**) Ground-truth map.

Table 2. Sixteen ground-truth classes in the AVIRIS Indian Pines dataset and the number of training and test pixels used in the experiments.

Class		**Set**	
No.	**Name**	**Train**	**Test**
C01	Alfalfa	3	51
C02	Corn-no till	72	1362
C03	Corn-min till	42	792
C04	Corn	12	222
C05	Grass/pasture	25	472
C06	Grass/trees	38	709
C07	Grass/pasture-mowed	2	24
C08	Hay-windrowed	25	464
C09	Oats	2	18
C10	Soybeans-no till	49	919
C11	Soybeans-min till	124	2344
C12	Soybean-clean till	31	583
C13	Wheat	11	201
C14	Woods	65	1229
C15	Bldg-grass-tree drives	19	361
C16	Stone-steel towers	5	90
Total		525	9841

(2) Reflective Optics System Imaging Spectrometer (ROSIS) University of Pavia: This dataset is an urban image acquired by the ROSIS sensor, with spectral coverage ranging from 0.43–0.86 µm. The ROSIS sensor has a spatial resolution of 1.3 m per pixel, with 115 spectral bands. This image, with a size of 610 × 340 pixels, contains 103 spectral bands after the removal of noisy bands. There are nine ground truth classes of interest. The false color image and the ground truth map are shown in Figure 3. In the experiments, we randomly chose 40 samples per class for training and used the rest for testing, as shown in Table 3.

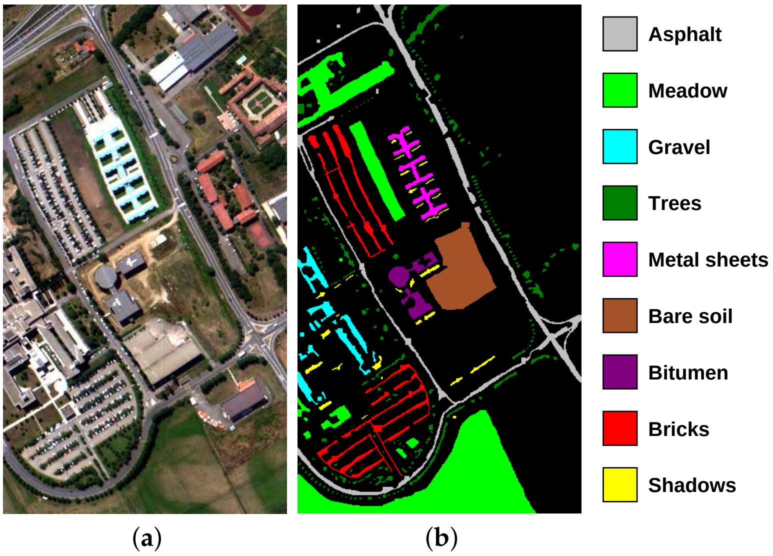


Figure 3. Reflective Optics System Imaging Spectrometer (ROSIS) University of Pavia dataset. (**a**) RGB composite image of three bands. (**b**) Ground-truth map.

Table 3. Nine ground-truth classes in the ROSIS University of Pavia dataset and the number of training and test pixels used in the experiments.

Class		Set	
No.	**Name**	**Train**	**Test**
C1	Asphalt	40	6812
C2	Meadow	40	18,646
C3	Gravel	40	2167
C4	Trees	40	3396
C5	Metal sheets	40	1338
C6	Bare soil	40	5064
C7	Bitumen	40	1316
C8	Bricks	40	3838
C9	Shadows	40	986
	Total	360	43,563

(3) AVIRIS Kennedy Space Center: This image was collected by the AVIRIS sensor over the Kennedy Space Center, Florida, in March 1996. There are 224 bands in the image, covering the wavelength range of 0.4–2.5 µm. The spectral and spatial resolutions are 10 nm and 18 m, respectively. This image acquired over an area of 512 × 614 pixels contains 176 spectral bands after removing water absorption and low signal-to-noise bands. Thirteen ground-truth classes ranging from 105–927 pixels in size that occur in this scene are defined for this image. The false color image and the ground truth

map are shown in Figure 4. In the experiments, 5% of the labeled pixels were randomly chosen for training, and the rest were used for testing, as shown in Table 4.

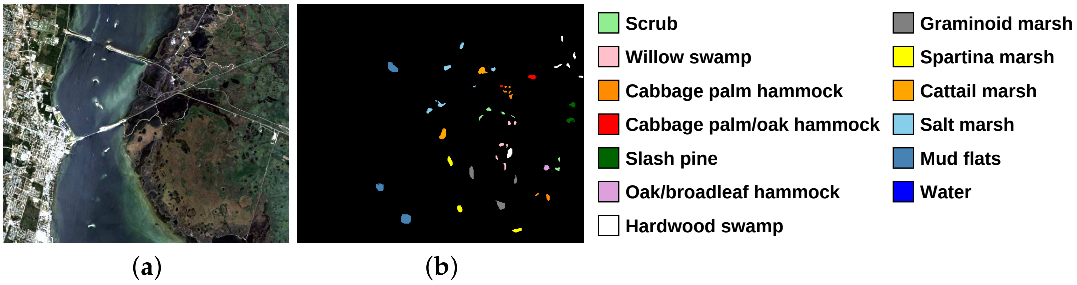


Figure 4. AVIRIS Kennedy Space Center dataset. (**a**) RGB composite image of three bands. (**b**) Ground-truth map.

Table 4. Thirteen ground-truth classes in the AVIRIS Kennedy Space Center dataset and the number of training and test pixels used in the experiments.

Class		**Set**	
No.	**Name**	**Train**	**Test**
C01	Scrub	39	722
C02	Willow swamp	13	230
C03	Cabbage palm hammock	13	243
C04	Cabbage palm/oak hammock	13	239
C05	Slash pine	9	152
C06	Oak/broadleaf hammock	12	217
C07	Hardwood swamp	6	99
C08	Graminoid marsh	22	409
C09	Spartina marsh	26	494
C10	Cattail marsh	21	383
C11	Salt marsh	21	398
C12	Mud flats	26	477
C13	Water	47	880
	Total	268	4943

4.2. Model Development and Experimental Setup

In the experiments, we compared the proposed SSGL method with four widely-used spatial-spectral classification methods based on the KSRC classifier. The first one is the majority voting (MV) method that performs a pixel-wise KSRC followed by majority voting within superpixel regions [21,22], where the superpixel regions are obtained by the entropy rate superpixel (ERS) algorithm [23]. The second one is the kernel joint SRC (KJSRC) method using kernel joint sparse representation classification [13,24], where the superpixel-based adaptive neighborhoods are used [24]. The third one is the condition random fields (CRF) method that combines pixel-wise KSRC and superpixel segmentation by using condition random fields [25]. The last one is the composite kernel with extended multi-attribute profile features (CKEMAP) method that integrates a composite kernel [7] into KSRC for the incorporation of spatial-spectral information, where the spatial features are extracted by EMAP [26,27]. In order to show the influence of the proposed two terms, we included a degeneration version (i.e., SSG) of SSGL for comparison, where the location information constraint term was dropped. Moreover, the pixel-wise KSRC [17] that only uses spectral features was also included as a baseline classifier. The corresponding optimization problems of the compared methods were all solved by ADMM. For the pixel-wise KSRC, the parameter γ for the RBF kernel was experimentally set to two for the AVIRIS Indian Pines dataset, 0.5 for the ROSIS University of Pavia dataset, and 0.125 for the AVIRIS Kennedy Space Center dataset, and the parameters λ and μ were empirically set to 10^{-4} and 10^{-3}, respectively. For further details, one can refer to the existing literature [14]. Considering that all compared methods are based on the KSRC classifier, their parameters brought by KSRC were experimentally set to be the same as those of the pixel-wise KSRC. For MV, the additional parameters brought by the ERS segmentation were carefully optimized by reference to [28]. For both KJSRC and CRF, the parameters brought by the ERS segmentation were fixed to be the same as those of MV, and the remaining parameters were obtained by cross-validation. For CKEMAP, the EMAP features were extracted from the first three principal components of the hyperspectral image, and the area and standard deviation attributes were considered to build EMAP, as reported in [29]. More specifically, for the area attribute the following values were selected for references: 49, 169, 361, 625, 961, 1369, 1849, and 2401. For the standard deviation, the EMAPs were built according to the following reference values with respect to the mean of the individual features: 2.5%, 5%, 7.5%, and 10%. As for the weight parameter and the kernel parameters in the composite kernel framework [7], they were obtained by cross-validation. For the proposed method, μ was set to 10^{-4}, α was set to 1, and β was experimentally set to 50 for the AVIRIS Indian Pines dataset and 100 for both the ROSIS University of Pavia dataset and the AVIRIS Kennedy Space Center dataset. As for SSG, the same parameters were used as SSGL.

Before the following experiments, the original data were scaled in the range [0, 1]. The classification accuracies are assessed with OA (overall accuracy), AA (average accuracy), and KA (kappa coefficient of agreement). The quantitative measures were obtained by averaging ten runs to avoid any bias induced by random sampling. All experiments were carried out in a 64-b quad-core CPU 2.40-GHz processor with 8 GB of memory.

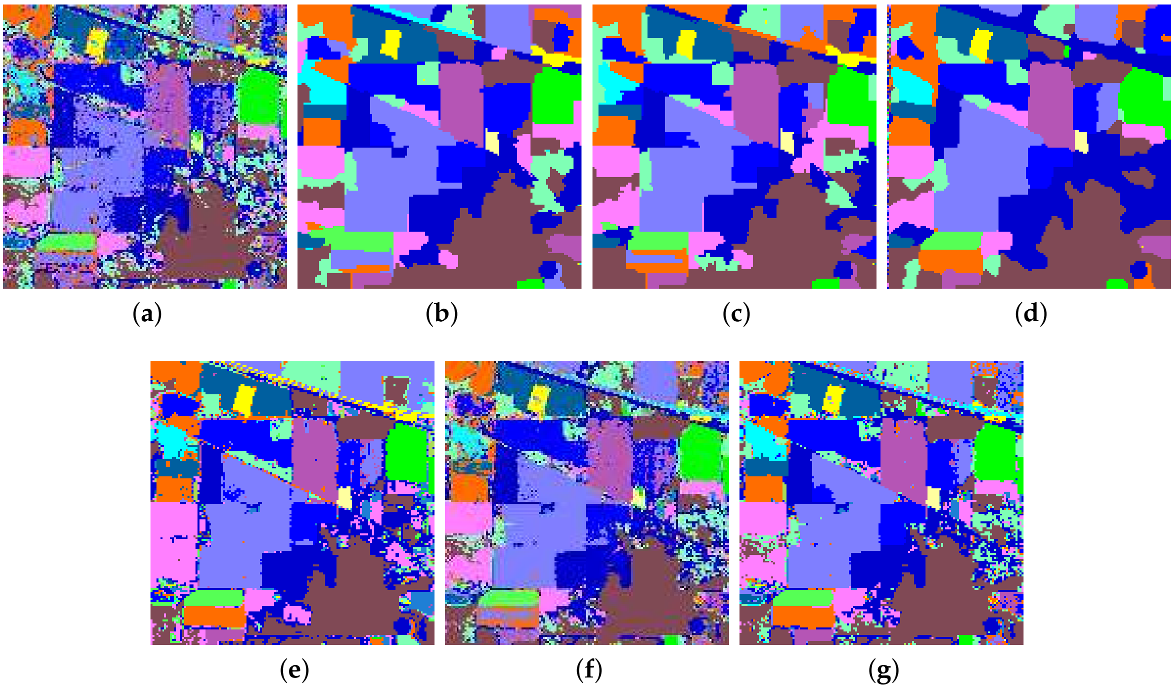
4.3. Numerical and Visual Comparisons

Table 5 summarizes the class-specific and global classification accuracies for the AVIRIS Indian Pines dataset. The processing times in seconds are also included for reference. Figure 5 shows the corresponding classification maps. From Table 5, it can be seen that all of the spatial-spectral methods yielded higher classification accuracies when compared to the pixel-wise KSRC. SSGL gave the highest global and most of the best class-specific accuracies, and CKEMAP performed the second highest. For the proposed methods SSG and SSGL, the improvement of SSGL over SSG was significant. From Figure 5, it can be seen that the maps of the spatial-spectral methods contain more homogeneous regions when compared with the map of the pixel-wise KSRC. The misclassified regions of SSGL and CKEMAP were smaller than those of the others.

Table 5. Classification accuracies for the AVIRIS Indian Pines dataset using different classification methods. The best results are highlighted in bold. AA: average accuracy; CKEMAP: composite kernel with extended multi-attribute profile features; CRF: condition random fields; KA: kappa coefficient of agreement; KSRC: kernel sparse representation classification; KJSRC: kernel joint SRC; MV: majority voting; OA: overall accuracy; SSG: spatial-spectral graph regularization with location information; SSGL: spatial-spectral graph regularization with location information.

Class Type	**KSRC**	**MV**	**KJSRC**	**CRF**	**CKEMAP**	**SSG**	**SSGL**
C01	56.67	72.35	81.57	80.78	89.61	68.43	**91.18**
C02	78.33	85.27	87.22	88.84	91.24	79.90	**95.01**
C03	64.31	82.15	92.29	89.31	**96.74**	76.98	96.48
C04	52.07	90.86	87.88	**91.98**	84.91	69.37	85.68
C05	89.03	90.97	90.97	88.88	92.65	90.15	**93.09**
C06	96.46	96.11	96.33	**99.31**	98.70	98.58	98.77
C07	74.58	19.17	19.17	56.25	96.67	88.33	**97.08**
C08	98.75	**99.78**	99.57	99.66	99.70	99.44	99.48
C09	57.78	30.00	40.00	20.00	**97.78**	51.67	93.33
C10	72.87	86.74	85.91	86.59	90.48	84.41	**92.76**
C11	82.43	**97.42**	93.95	97.11	96.92	93.78	97.42
C12	76.74	97.75	95.61	**97.99**	91.36	90.03	96.72
C13	98.76	**100**	99.00	99.60	99.50	99.45	99.55
C14	95.30	**99.78**	97.99	98.53	99.06	98.49	99.29
C15	53.80	69.56	86.26	90.61	**95.51**	73.16	90.42
C16	88.56	91.89	**96.78**	96.33	93.78	87.89	88.67
OA(%)	81.33	91.92	92.39	93.83	95.17	88.97	**96.16**
AA(%)	77.28	81.86	84.41	86.36	94.66	84.38	**94.68**
KA(%)	78.66	90.75	91.34	92.96	94.50	87.39	**95.62**
Time(s)	**17.03**	17.33	17.31	18.48	23.48	34.69	36.16

The class-specific and global classification accuracies for the ROSIS University of Pavia dataset are provided in Table 6, as well as the processing times. The corresponding classification maps are shown in Figure 6. From the table, it is clear that SSGL performed best followed by SSG, and the performances of the spatial-spectral methods were better than those of the pixel-wise method. The numerical comparisons are confirmed by inspecting the classification maps. Table 7 shows the class-specific and global classification accuracies and the processing times for the AVIRIS Kennedy Space Center dataset. It can be seen that Table 7 reveals almost the same results as Table 5. The classification maps shown in Figure 7 confirm the numerical comparisons further.



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**)

Figure 5. Classification maps and overall classification accuracies (in parentheses) obtained for the AVIRIS Indian Pines dataset using different classification methods. (**a**) KSRC (81.49), (**b**) MV (90.93), (**c**) KJSRC (92.79), (**d**) CRF (93.92), (**e**) CKEMAP (95.39), (**f**) SSG (88.30), (**g**) SSGL (96.44).

Table 6. Classification accuracies for the ROSIS University of Pavia dataset using different classification methods. The best results are highlighted in bold.

Class Type	KSRC	MV	KJSRC	CRF	CKEMAP	SSG	SSGL
C1	73.25	91.69	92.01	91.92	97.30	93.30	**97.49**
C2	83.24	93.23	95.25	94.77	94.94	97.01	**98.48**
C3	78.68	93.01	90.90	91.54	91.86	90.58	**97.70**
C4	91.65	86.39	89.32	91.58	**95.58**	94.44	94.90
C5	99.42	99.17	99.49	**99.89**	99.57	99.37	99.36
C6	84.77	98.20	99.26	**99.53**	96.85	98.70	98.90
C7	92.65	99.70	95.30	99.06	98.40	99.70	**99.86**
C8	77.99	98.00	70.73	94.72	95.52	96.11	**99.00**
C9	99.27	95.86	89.62	**99.73**	99.60	99.50	99.51
OA(%)	82.96	93.88	92.38	94.86	95.83	96.24	**98.19**
AA(%)	86.77	95.03	91.32	95.86	96.62	96.52	**98.36**
KA(%)	78.21	92.04	90.05	93.32	94.56	95.09	**97.63**
Time(s)	**98.80**	100.61	137.69	107.55	322.54	224.29	210.24

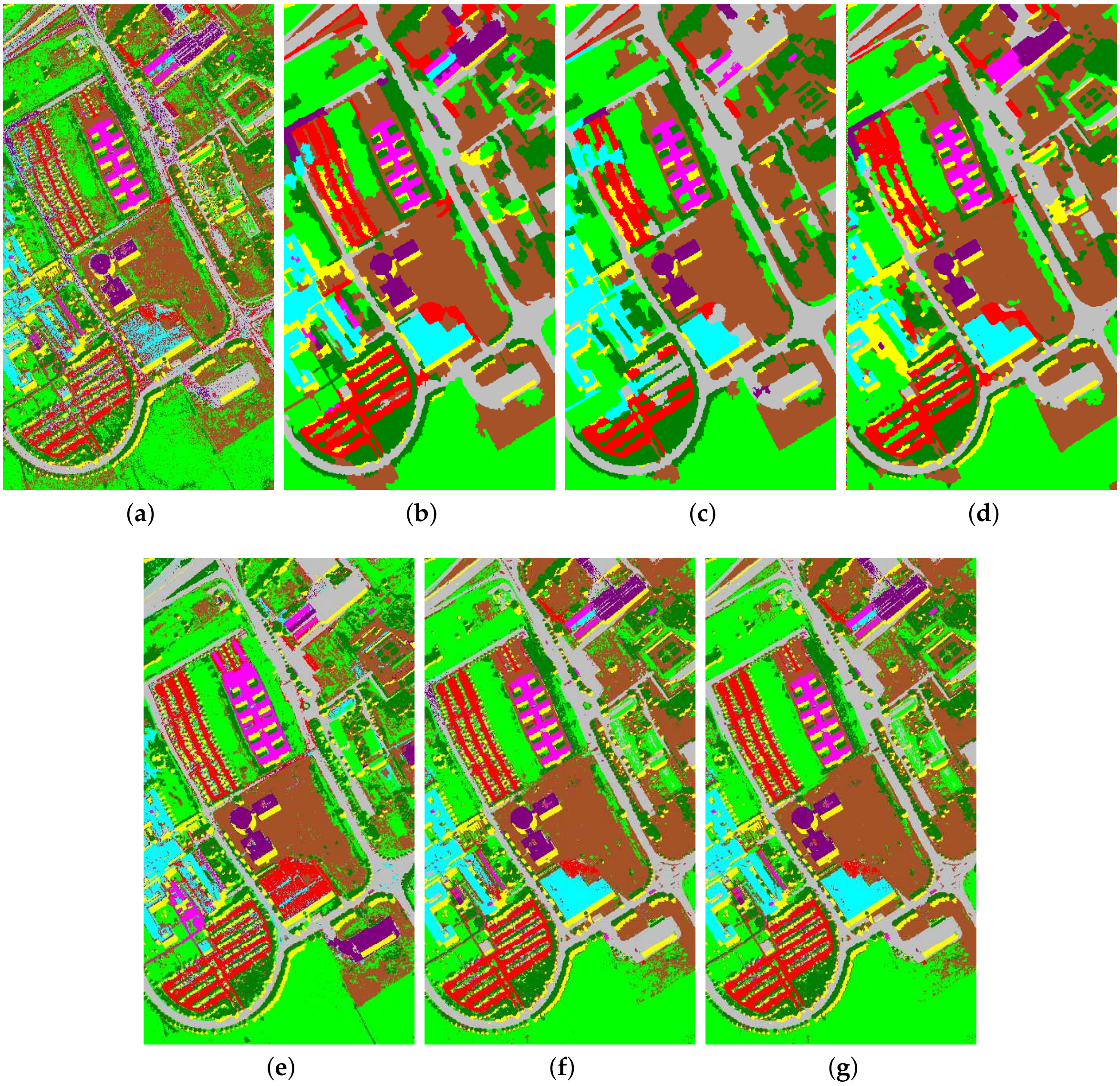
Table 7. Classification accuracies for the AVIRIS Kennedy Space Center dataset using different classification methods. The best results are highlighted in bold.

Class Type	KSRC	MV	KJSRC	CRF	CKEMAP	SSG	SSGL
C01	95.69	**100**	**100**	**100**	99.03	99.18	99.04
C02	85.09	79.43	80.61	88.30	88.39	90.22	**95.35**
C03	90.62	97.94	97.94	**98.31**	97.08	98.02	98.27
C04	46.78	63.05	50.92	56.90	86.86	45.48	**88.49**
C05	61.64	82.76	82.63	82.30	87.63	80.13	**92.57**
C06	45.58	73.04	69.45	60.05	89.08	35.02	**99.12**
C07	83.54	92.83	**100**	98.99	89.29	99.09	**100**
C08	89.29	99.12	**99.29**	98.78	98.07	98.26	99.05
C09	96.01	98.42	98.42	98.42	**98.54**	98.34	98.34
C10	95.30	87.08	87.57	**100**	99.19	99.63	99.71
C11	94.30	86.83	90.65	**98.94**	97.74	96.71	97.14
C12	88.05	**100**	78.07	97.17	97.04	95.30	97.63
C13	99.84	**100**	**100**	**100**	**100**	**100**	**100**
OA(%)	88.45	93.01	90.70	94.35	96.63	92.15	**98.01**
AA(%)	82.44	89.27	87.35	90.63	94.46	87.34	**97.29**
KA(%)	87.12	92.19	89.60	93.69	96.25	91.22	**97.78**
Time(s)	**111.68**	116.40	576.78	153.00	626.01	192.24	198.11

In this set of experiments, it can be seen that SSGL performed best in all scenes. This demonstrates the superiority of the spatial location constraint as well as the graph-based spatially-smooth constraint. However, for SSG, it is more suitable to classify the ROSIS scene. This is because the spatial resolution of the ROSIS scene is higher than those of the two AVIRIS scenes, and thus the spectral similarity of the spatially-adjacent pixels of which is more suitable for SSG to measure.

4.4. Analysis of Parameters

Among the parameters of the proposed method, β was set experimentally and differently for the three given datasets. In this set of experiments, we evaluated the influence of β by varying it from 10–200. Figure 8 shows the classification accuracies for both SSGL and SSG when applied to the three given datasets. It can be seen that when β is small, the classification performance of SSG increases monotonically as β increases; and the results typically saturate for large βs. When β is relatively large, the accuracies of SSGL decrease slightly. In all cases, we can choose β in a wide optimal region. Thus, the choice of β is robust.



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**)

Figure 6. Classification maps and overall classification accuracies (in parentheses) obtained for the ROSIS University of Pavia dataset using different classification methods. (**a**) KSRC (83.44), (**b**) MV (93.71), (**c**) KJSRC (94.55), (**d**) CRF (95.56), (**e**) CKEMAP (96.67), (**f**) SSG (97.57), (**g**) SSGL (98.28).

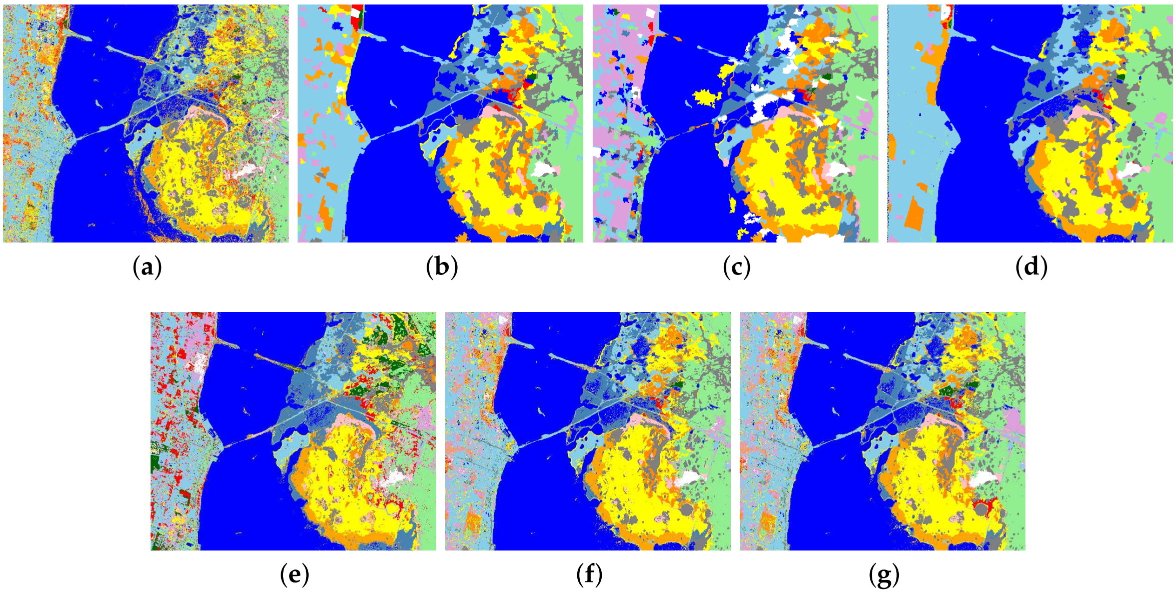


Figure 7. Classification maps and overall classification accuracies (in parentheses) obtained for the AVIRIS Kennedy Space Center dataset using different classification methods. (**a**) KSRC (89.86), (**b**) MV (93.30), (**c**) KJSRC (92.96), (**d**) CRF (93.49), (**e**) CKEMAP (97.43), (**f**) SSG (92.98), (**g**) SSGL (98.56).

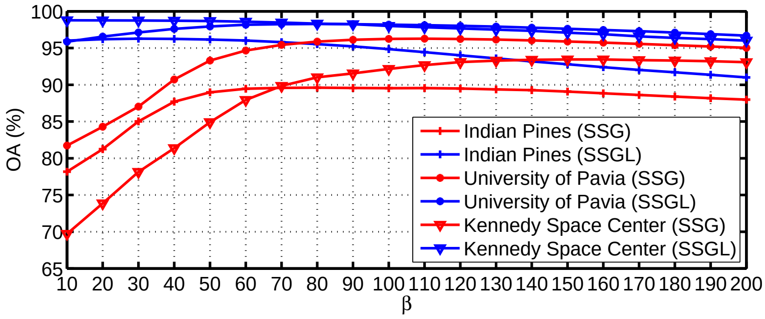


Figure 8. OA as a function of β.

4.5. Influence of Anchor Samples

In SSGL, the spatial location constraint is built on the prior knowledge of the location information of training samples. That is to say, all training samples are treated as anchor samples. In this set of experiments, we built the spatial location constraint term by using different percentages of training sample to show the influence of anchor samples. Figure 9 shows the classification accuracies for SSGL when applied to the three given datasets. From the figure, we can see that the accuracies increase monotonically as the percentage increases, but the accuracies reach the optimal values asymptotically when a moderate number of anchor samples are used. Thus, we can build the spatial location term by using a relatively small number of anchor samples.

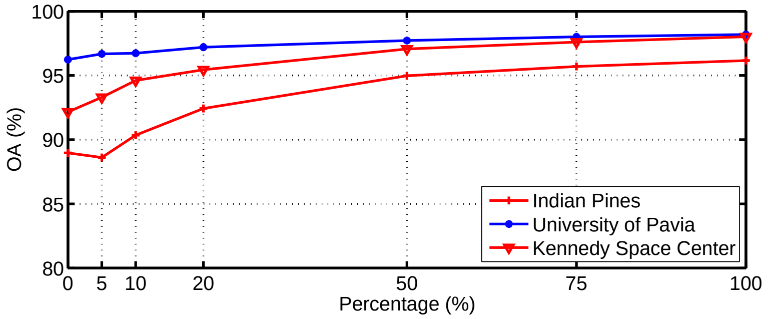
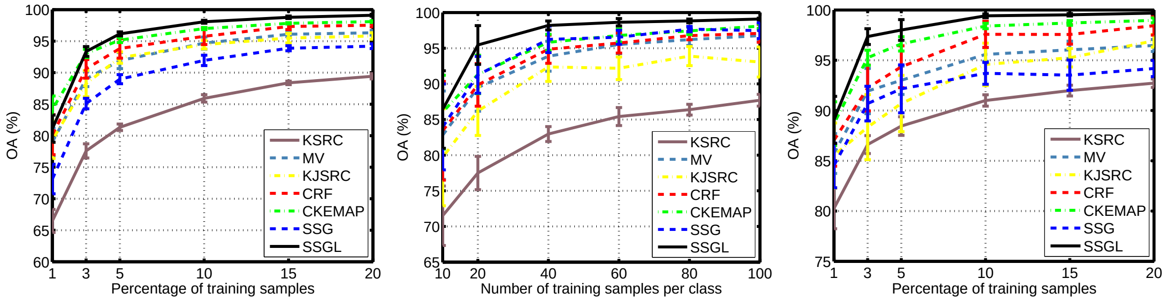


Figure 9. OA as a function of the number of anchor samples.

4.6. Different Numbers of Training Samples

In this set of experiments, we tested the performance of the compared classification methods in an ill-posed scenario, where different numbers of training pixels are used. Specifically, for both the AVIRIS Indian Pines dataset and the AVIRIS Kennedy Space Center dataset, we randomly chose 1–20% of the labeled pixels per class for training and the remaining pixels for testing. For the classes with limited labeled pixels, at least two pixels were chosen for training. For the ROSIS University of Pavia dataset, we built training sets by randomly choosing 10, 20, 40, 60, 80, and 100 training pixels per class and used the rest as test sets. Figure 10 shows the mean value and standard deviation of OA as a function of the number of training pixels for the three given datasets. From the figures, it can be seen that the proposed SSGL outperformed the compared methods in almost all cases. As expected, the performance of the spatial-spectral methods was better than that of the pixel-wise KSRC, and the classification accuracies decreased when the number of training pixels was reduced. When compared with the AVIRIS scene, SSG is more suitable for the ROSIS scene.



(**a**) (**b**) (**c**)

Figure 10. OA as a function of the number of training samples for different classification methods when applied to (**a**) the AVIRIS Indian Pines dataset, (**b**) the ROSIS University of Pavia data set, and (**c**) the AVIRIS Kennedy Space Center dataset.

5. Discussion and Conclusions

A regularized kernel sparse representation method for spatial-spectral classification of hyperspectral images has been presented in this paper, where the spatial-spectral information of hyperspectral images is incorporated by appending two spatial-spectral constraint terms to the sparse recovery model of KSRC. One is the graph-based spatially-smooth constraint, which is designed to measure the spectral similarity of spatially-adjacent pixels. The other is the spatial location constraint, which is used to exploit the wealth of training pixels and thereby to reduce the cost of acquiring a large number of labeled pixels. By introducing the two constraints, the spatial and label information of the hyperspectral pixels—especially that of the training pixels—can be spread to their neighbors, which is very useful for the discrimination of each class. Experiments conducted on three real hyperspectral images have demonstrated that the proposed method can yield accurate classification results. Although the results obtained by the proposed method are very encouraging in hyperspectral image classification, further enhancements such as more robust constraint terms and more reasonable regularization strategies should be pursued in future developments.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (Grant number 61601201) and the Natural Science Foundation of Jiangsu Province (Grant numbers BK20160188 and BK20150160). The authors would like to thank Prof. D. Landgrebe from Purdue University for providing the AVIRIS Indian Pines dataset and Prof. P. Gamba from the University of Pavia, Italy, for providing the ROSIS University of Pavia dataset.

Author Contributions: Jianjun Liu wrote the paper. Jianjun Liu and Jinlong Yang analyzed the data. Jianjun Liu and Zhiyong Xiao conceived of and designed the experiments. Yufeng Chen performed the experiments. Yunfeng Chen contributed analysis tools.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses or interpretation of the data; in the writing of the manuscript; nor in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

SVM	support vector machines
SRC	sparse representation classification
KSRC	kernel sparse representation classification
SSGL	spatial-spectral graph regularization with location information
RBF	Gaussian radial basis function
ADMM	alternating direction method of multipliers
AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
ROSIS	Reflective Optics System Imaging Spectrometer
MV	majority voting
KJSRC	kernel joint sparse representation classification
CRF	condition random fields
CKEMAP	composite kernel with extended multi-attribute profile features
EMAP	extended multi-attribute profile
ERS	entropy rate superpixel
SSG	spatial-spectral graph regularization
OA	overall accuracy
AA	average accuracy
KA	kappa coefficient of agreement

References

1. Manolakis, D.; Shaw, G. Detection algorithms for hyperspectral imaging applications. *IEEE Signal Process. Mag.* **2005**, *19*, 29–43.
2. Datt, B.; Mcvicar, T.R.; Van Niel, T.G.; Jupp, D.L.B. Preprocessing EO-1 Hyperion hyperspectral data to support the application of agricultural indexes. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1246–1259.
3. Horig, B.; Kuhn, F.; Oschutz, F.; Lehmann, F. HyMap hyperspectral remote sensing to detect hydrocarbons. *Int. J. Remote Sens.* **2001**, *22*, 1413–1422.
4. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790.
5. Li, J.; Huang, X.; Gamba, P.; Bioucas-Dias, J.M.; Zhang, L.; Benediktsson, J.A.; Plaza, A. Multiple feature learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 1592–1606.
6. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3973–3985.

7. Camps-Valls, G.; Gomez-Chova, L.; Munoz-Mari, J.; Vila-Frances, J.; Calpe-Maravilla, J. Composite kernels for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2006**, *3*, 93–97.
8. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep Learning-Based Classification of Hyperspectral Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107.
9. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Milan, Italy, 26–31 July 2015.
10. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251.
11. Wan, L.; Liu, N.; Huo, H.; Fang, T. Selective convolutional neural networks and cascade classifiers for remote sensing image classification. *Remote Sens. Lett.* **2017**, *8*, 917–926.
12. Wang, J.; Luo, C.; Huang, H.; Zhao, H.; Wang, S. Transferring Pre-Trained Deep CNNs for Remote Scene Classification with General Features Learned from Linear PCA Network. *Remote Sens.* **2017**, *9*, 225.
13. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral Image Classification via Kernel Sparse Representation. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 217–231.
14. Liu, J.; Wu, Z.; Wei, Z.; Xiao, L.; Sun, L. Spatial-spectral kernel sparse representation for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 2462–2471.
15. Yuan, H.; Tang, Y.Y.; Lu, Y.; Yang, L.; Luo, H. Hyperspectral Image Classification Based on Regularized Sparse Representation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2174–2182.
16. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Ma, Y. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 210–227.
17. Gao, S.; Tsang, I.W.H.; Chia, L.T. Kernel Sparse Representation for Image Classification and Face Recognition. In Proceedings of the 11th European Conference on Computer Vision: Part IV, Heraklion, Greece, 5–11 September 2010; pp. 1–14.
18. Goldstein, T.; Osher, S. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.* **2009**, *2*, 323–343.
19. Yang, J.; Zhang, Y. Alternating direction algorithms for l_1-problems in compressive sensing. *SIAM J. Sci. Comput.* **2011**, *33*, 250–278.
20. Combettes, P.L.; Wajs, V.R. Signal Recovery by Proximal Forward-Backward Splitting. *Multiscale Model. Simul.* **2005**, *4*, 1168–1200.
21. Tarabalka, Y.; Chanussot, J.; Benediktsson, J.A. Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recogition* **2010**, *43*, 2367–2379.
22. Liu, J.; Shi, X.; Wu, Z.; Xiao, L.; Xiao, Z.; Yuan, Y. Hyperspectral image classification via region-based composite kernels. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2015.
23. Liu, M.Y.; Tuzel, O.; Ramalingam, S.; Chellappa, R. Entropy rate superpixel segmentation. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011.
24. Feng, J.; Cao, Z.; Pi, Y. Polarimetric Contextual Classification of PolSAR Images Using Sparse Representation and Superpixels. *Remote Sens.* **2014**, *6*, 7158–7181.
25. Roscher, R.; Waske, B. Superpixel-based classification of hyperspectral data using sparse representation and conditional random fields. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada , 13–18 July 2014.
26. Mura, M.D.; Benediktsson, J.A.; Waske, B.; Bruzzone, L. Extended profiles with morphological attribute filters for the analysis of hyperspectral data. *Inte. J. Remote Sens.* **2010**, *31*, 5975–5991.
27. Mura, M.D.; Benediktsson, J.A.; Waske, B.; Bruzzone, L. Morphological Attribute Profiles for the Analysis of Very High Resolution Images. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 3747–3762.

28. Fang, L.; Li, S.; Kang, X.; Benediktsson, J.A. Spectral–Spatial Classification of Hyperspectral Images with a Superpixel-Based Discriminative Sparse Model. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4186–4201.
29. Li, J.; Marpu, P.R.; Plaza, A.; Bioucas-Dias, J.M.; Benediktsson, J.A. Generalized Composite Kernel Framework for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 4816–4829.