sensors

MDPI

Article

SLIC Superpixel-Based $l_{2,1}$-Norm Robust Principal Component Analysis for Hyperspectral Image Classification

Baokai Zu [1], Kewen Xia [1,*], Tiejun Li [2], Ziping He [1], Yafang Li [3], Jingzhong Hou [1] and Wei Du [4]

[1] School of Electronics and Information Engineering, Hebei University of Technology, Tianjin 300401, China; zubaokai@163.com (B.Z.); heziping1102@126.com (Z.H.); houjingzhong2002@163.com (J.H.)
[2] School of Mechanical Engineering, Hebei University of Technology, Tianjin 300401, China; 1993076@hebut.edu.cn
[3] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; yafangli@bjut.edu.cn
[4] College of Resources and Environment, Huazhong Agricultural University, Wuhan 430070, China; weidu@webmail.hzau.edu.cn
* Correspondence: kwxia@hebut.edu.cn

Received: 19 November 2018; Accepted: 17 January 2019; Published: 24 January 2019

Abstract: Hyperspectral Images (HSIs) contain enriched information due to the presence of various bands, which have gained attention for the past few decades. However, explosive growth in HSIs' scale and dimensions causes "Curse of dimensionality" and "Hughes phenomenon". Dimensionality reduction has become an important means to overcome the "Curse of dimensionality". In hyperspectral images, labeled samples are more difficult to collect because they require many labor and material resources. Semi-supervised dimensionality reduction is very important in mining high-dimensional data due to the lack of costly-labeled samples. The promotion of the supervised dimensionality reduction method to the semi-supervised method is mostly done by graph, which is a powerful tool for characterizing data relationships and manifold exploration. To take advantage of the spatial information of data, we put forward a novel graph construction method for semi-supervised learning, called SLIC Superpixel-based $l_{2,1}$-norm Robust Principal Component Analysis (SURPCA$_{2,1}$), which integrates superpixel segmentation method Simple Linear Iterative Clustering (SLIC) into Low-rank Decomposition. First, the SLIC algorithm is adopted to obtain the spatial homogeneous regions of HSI. Then, the $l_{2,1}$-norm RPCA is exploited in each superpixel area, which captures the global information of homogeneous regions and preserves spectral subspace segmentation of HSIs very well. Therefore, we have explored the spatial and spectral information of hyperspectral image simultaneously by combining superpixel segmentation with RPCA. Finally, a semi-supervised dimensionality reduction framework based on SURPCA$_{2,1}$ graph is used for feature extraction task. Extensive experiments on multiple HSIs showed that the proposed spectral-spatial SURPCA$_{2,1}$ is always comparable to other compared graphs with few labeled samples.

Keywords: Hyperspectral Image; Robust Principal Component Analysis (RPCA); Simple Linear Iterative Clustering (SLIC); superpixel segmentation

1. Introduction

Hyperspectral Images (HSIs) provide comprehensive spectral information of the materials' physical properties [1], which is applied in ecosystem monitoring [2], agricultural monitoring [3,4], environmental monitoring [5], forestry [6], urban growth analysis [7,8], and mineral identification [9,10]. The abundant hyperspectral image bands bring rich spectral information, but they result in the

"Hughes phenomenon" [11,12] that reduces the accuracy and efficiency of the classification task. By reducing the dimensionality of the data, more compact low-dimensional hyperspectral images can be obtained. Therefore, it is important and practically significant to perform appropriate dimensionality reduction on hyperspectral images.

Dimensionality reduction refers to finding the low-dimensional representation of the original data by eliminating the redundant elements and retaining their main features, which can overcome the "Curse of dimensionality" problem. Dimensionality reduction mainly includes feature selection and feature extraction [13]. Feature selection methods reduce the dimensions of the original data by selecting the most representative and distinguishing features [14]. Feature extraction methods combine multiple features linearly and non-linearly on the basis of maintaining the data structure information [15]. Recently, many dimensionality reduction algorithms have been put forward. The most classic ones are the early Principal Component Analysis (PCA) [16], Linear Discriminant Analysis (LDA) [17], and Independent Component Analysis (ICA) [18–20]. In recent years, feature extraction has quickly become one of the hotspots in machine learning and data mining. Compared with the traditional dimensionality reduction method, the main difference of manifold learning is its ability to maintain the invariance of the data structure. The most representative algorithms include Laplacian Eigenmap [21], Isomap [22], and Local Linear Embedding (LLE) [23]. A multimetric learning approach that combines feature extraction and active learning (AL) is proposed to deal with the high dimensionality of the input data and the limited number of the labeled samples simultaneously [24]. To simultaneously deal with the two issues mentioned above, a regularized multimetric active learning (AL) framework is proposed [25].

With the rapid development of data acquisition and storage technology, many unlabeled samples are available, but labeled samples are more difficult to collect. If the training samples are limited, the machine learning methods may be confronted with over-fitting problems when dealing with high-dimensional small sample size problems [26,27]. For the classification of hyperspectral images (HSIs), good classification results usually require many labeled samples. To resolve the over-fitting problem, semi-supervised learning was proposed to utilize both labeled samples and the unlabeled samples that conveyed the marginal distribution information to enhance the algorithmic performance [28–30]. Semi-supervised dimensionality reduction combines semi-supervised learning with dimensionality reduction, and has gradually become a new branch in the machine learning field. Semi-supervised dimensionality reduction methods not only use the supervised information of the data, but also maintain the structural information of the data. In the dimensionality reduction method, the promotion of the supervised dimensionality reduction method to a semi-supervised method is mostly done by graph-based methods. Therefore, graph-based semi-supervised learning methods have gained a great deal of attention, which construct adjacency graph to extract the local geometry of the data. Graph is a powerful data analysis tool, which is widely used in many research domains, such as dimension reduction [21–23], semi-supervised learning [31,32], manifold embedding, and machine learning [33,34]. In graph-based methods, each point is mapped to a low-dimensional feature vector, trying to maintain the connection between vertices [30]. The procedure of graph construction effectively determines the potential of the graph-based learning algorithms. Thus, for a specific task, a graph which models the data structure aptly will correspondingly achieve a good performance. Therefore, how to construct a good graph has been widely studied in recent years, and it is still an open problem [35].

Within the last several years, numerous algorithms for feature extraction have been put forward. PCA is a widely-used linear subspace algorithm that seeks a low-dimensional representation of high-dimensional data. PCA works against small Gaussian noise in data effectively, but it is highly sensitive to sparse errors of high magnitude. To solve this problem, Candès [36] and Wright et al. [37] proposed Robust Principal Component Analysis (RPCA), which intends to decompose the observed data into a low-rank matrix and a sparse noises matrix. In [38], a novel feature extraction method based on non-convex robust principal component analysis (NRPCA) was proposed for hyperspectral image classification. Nie et al. developed a novel model named graph-regularized tensor robust principal component analysis (GTRPCA) for denoising HSIs [39]. Chen et al. denoised hyperspectral images using principal component analysis and block-matching 4D filtering [40].

The above mentioned methods suggest that the pixels in a hyperspectral image lie in a low-rank manifold. However, the spatial correlation among pixels is not revealed. For HSIs, the adjacent pixels' correlation is typically extremely high [41], which has potential low-rank attributes. Pixels in homogenous areas usually consist of similar materials, whose spectral properties are highly similar and can be taken as being approximately in the same subspace. By extracting the low-rank matrix of the data, the low-dimensional structure of each pixel is revealed and we can classify each pixel more accurately. Xu et al. put forward a low-rank decomposition spectral-spatial algorithm, which incorporates global and local correlation [42]. Fan et al. integrated superpixel segmentation (SS) into Low-rank Representation (LRR) and proposed a novel denoising method called SS-LRR [43]. A multi-scale superpixel based sparse representation (MSSR) for HSIs' classification is proposed to overcome the disadvantages of utilizing structural information [44]. Sun et al. presented a novel noise reduction method based on superpixel-based low-rank representation for hyperspectral image [45]. In [46], Mei et al. proposed a new unmixing method with superpixel segmentation and LRR based on RGBM. Tong et al. proposed multiscale union regions adaptive sparse representation (MURASR) by multiscale patches and superpixels [47]. A novel method, robust regularization block low-rank discriminant analysis, is proposed for HSIs' feature extraction [48].

Spatial information can play a very important role in mapping data. However, many graph construction methods do not make full use of data's spatial information. For example, in hyperspectral images, adjacent pixels in a homogenous area usually belong to the same category. Therefore, how to consider the features of the actual data globally to further improve the semi-supervised dimensionality reduction performance is one of the main research components of this paper. Considering the spatial correlative and spectral low-rank characteristics of pixels in HSIs, we integrate the Simple Linear Iterative Clustering (SLIC) segmentation method into low-rank decomposition. Figure 1 shows the formulation of the proposed SLIC Superpixel-based $l_{2,1}$-norm Robust Principal Component Analysis (SURPCA$_{2,1}$) for hyperspectral image classification. As shown in Figure 1, we preprocess the hyperspectral image by the Image Fusion and Recursive Filtering feature (IFRF), which eradicates the noise and redundant information concurrently [12]. The proposed method divide the image into multiple homogeneous regions by the superpixel segmentation algorithm SLIC. The pixels in each homogeneous region may belong to the same ground object category. Therefore, we stack the pixels in the same homogeneous region into a matrix. Due to the low-rank property of the matrix, RPCA is used to recover the low-dimensional structure of all pixels in the homogeneous region. Then, we combine these low-rank matrices together to an integrated low-rank graph. In addition, we process the semi-supervised discriminant analysis for dimension reduction, which takes advantage of the labeled samples and the distribution of the whole samples. The k-Nearest Neighbor (kNN) algorithm is applied to handle the low-rank graph for the regularized graph of semi-supervised discriminant analysis. Finally, we implement the Nearest Neighbor classifier method. We summarize the main contributions of the paper in the following paragraphs.

- Inspired by robust principal component analysis and superpixel segmentation, we put forward a novel graph construction method, SLIC superpixel-based $l_{2,1}$-norm robust principal component analysis. The superpixel-based $l_{2,1}$-norm RPCA extracts the low-rank spectral structure of all pixels in each uniform region, respectively.
- The simple linear iterative clustering addresses the spatial characteristics of hyperspectral images. Consequently, the SURPCA$_{2,1}$ graph model can classify pixels more accurately.
- To investigate the performance of the SLIC superpixel-based $l_{2,1}$-norm robust principal component analysis graph model, we conducted extensive experiments on several real multi-class hyperspectral images.

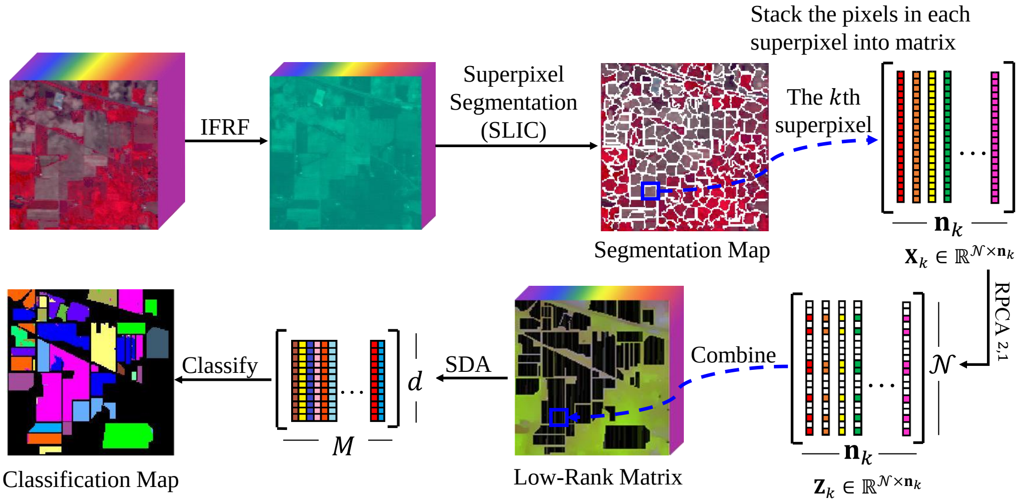

IFRF
Superpixel Segmentation (SLIC)
Segmentation Map
Stack the pixels in each superpixel into matrix
The kth superpixel
$\mathcal{N}$
$\mathbf{n}_k$
$\mathbf{X}_k \in \mathbb{R}^{\mathcal{N}\times\mathbf{n}_k}$
RPCA 2,1
$\mathbf{Z}_k \in \mathbb{R}^{\mathcal{N}\times\mathbf{n}_k}$
Combine
Low-Rank Matrix
SDA
d
M
Classify
Classification Map

Figure 1. Formulation of the proposed SLIC superpixel-based $l_{2,1}$-norm robust principal component analysis for HSIs classification.

We start with a scientific background in Section 2. Section 3 deciphers the HSIs classification with $l_{2,1}$-norm robust principal component analysis. Section 4 performs comparative experiments on real-world HSIs to examine the performance of the proposed graph. We provide the discussion in Section 5. Conclusions are presented in Section 6.

2. Scientific Background

2.1. Graph-Based Semi-Supervised Dimensionality Reduction Method

In the dimensionality reduction method, the promotion of supervised method to semi-supervised method is mostly done by graph-based methods. Maier et al. [49] showed that different graph structures obtain different results in the same clustering algorithm. A good graph construction method has great influence in graph-based semi-supervised learning. Therefore, how to construct a good graph sometimes seems to be more important than a good objective function [50].

Graph is composed by nodes and edges, which represents the structure of the entire dataset. The nodes represent the sample points, and the edges' weight represents the similarity between the nodes. Generally, the greater the weight is, the higher the similarity between the nodes will be. If the edge does not exist, the similarity between the two points will be zero. The similarity between nodes is usually measured by distance, such as Euclid, Mahalanobis, and Chebyshev distance [51]. Different graph construction methods have great impact on the performance of classification results. The graph construction process mainly includes the graph structure and the edges' weight function. Two commonly used graph structures are Fully Connected graphs and Nearest Neighbor graphs (k-nearest neighbor graph and ε-ball Graph).

- Fully Connected graphs [21,52]: In the fully connected graph, all nodes are connected by edges whose weights are not zero. The fully connected graphs are easy to construct and have good performance for semi-supervised learning methods. However, the disadvantage of the fully connected graphs is that it requires processing all nodes, which would lead to high computational complexity.
- k-nearest neighbor graph [21]: Each node in the k-nearest neighbor graph is only connected with k neighbor in a certain distance. The samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are considered as neighbor if $\mathbf{x}_i$ is among the k nearest neighbor of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the k-nearest neighbor of $\mathbf{x}_i$.
- ε-ball graph [21]: In the ε-ball graph, the connections between data points occur in the neighborhood of radius ε. That is to say, if the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ exists

$d(\mathbf{x}_i, \mathbf{x}_j) \leq \varepsilon, \varepsilon \in \mathbb{R}, \varepsilon > 0$, there will be a neighbor relationship between these two nodes. Therefore, the connectivity of the graph is largely influenced by the parameter ε.

It is also necessary to assign weight for the connected edges **W**. The followings are commonly used weight functions:

- Inverse of Euclidean distance [53],

$$\mathbf{W}_{ij} = \begin{cases} \left\|x_i - x_j\right\|^{-2}, & if \mathbf{x}_i \in N_k\left(\mathbf{x}_j\right) or \mathbf{x}_j \in N_k\left(\mathbf{x}_i\right) \\ 0, & otherwise. \end{cases} \tag{1}$$

- 0–1 weighting [21],

$$\mathbf{W}_{ij} = \begin{cases} 1, & if \mathbf{x}_i \in N_k\left(\mathbf{x}_j\right) or \mathbf{x}_j \in N_k\left(\mathbf{x}_i\right) \\ 0, & otherwise. \end{cases} \tag{2}$$

- Heat kernel weighting [21],

$$\mathbf{W}_{ij} = \begin{cases} \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right), & if \mathbf{x}_i \in N_k\left(\mathbf{x}_j\right) or \mathbf{x}_j \in N_k\left(\mathbf{x}_i\right) \\ 0, & otherwise. \end{cases} \tag{3}$$

kNN graph can make full use of the local information of adjacent nodes. Since kNN graph is sparse, it can solve the computational complexity and storage problem in the fully connected graph. The parameters k and ε depend on the data density, which is difficult to select. The choice of neighbor size parameter is the key to the effectiveness of the graph-based semi-supervised learning method. The Nearest Neighbor graph lacks global constraints of the data points, and its representation performance is greatly reduced when the data is seriously damaged [21].

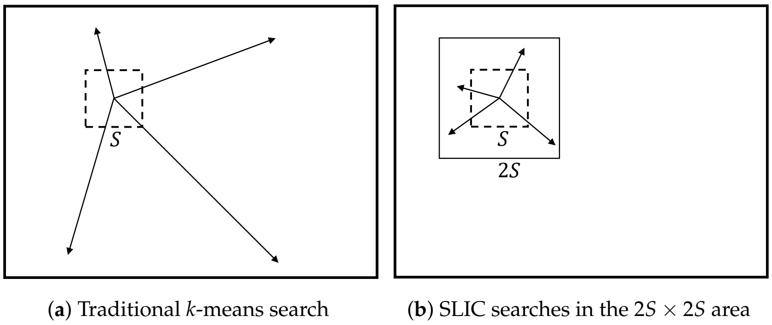
2.2. Simple Linear Iterative Clustering

The superpixel is an image segmentation technique proposed by Xiaofeng Ren in 2003 [54]. It refers to the image regions with certain visual meanings composed of adjacent pixels which have similar physical characteristics such as texture, color, brightness, etc. [55]. The superpixel segmentation methods segment pixels and replace a large number pixels with a small number of superpixels, which greatly reduces the complexity of image post-processing. It has been widely used in computer vision applications such as image segmentation, pose estimation, target tracking, and target recognition. The boundary information is relatively obvious between the superpixels. Achanta et al. introduced a simple linear iterative clustering algorithm to efficiently produce superpixels that are compact and nearly uniform [56].

The superpixel algorithms are broadly classified into graph-based and gradient-ascent-based algorithms. In graph-based algorithms, each pixel is treated as a node in a graph, and edge weights between two nodes are set proportional to the similarity between the pixels. Superpixel segmentations are extracted by effectively minimizing a cost function defined on the graph [57]. Simple linear iterative clustering was proposed in 2010, and is simple to use and understand. Simple linear iterative clustering is based on the k-means clustering algorithm [58], which is done in the five-dimensional *Labxy* space, where *Lab* is the pixel color vector in CIELAB color space, which is widely considered as perceptually uniform for small color distances, and x, y is the pixel position [57]. Starting from an initial rough clustering, the clusters from the previous iteration are refined to obtain better segmentation in each iteration by gradient ascent method until convergence. Then, the distance metrics of the five-dimensional feature vectors are constructed. Finally, the pixels are clustered in the image locally [57,59]. The method has a high comprehensive evaluation index in terms of computational speed and superpixel shape, which achieves good segmentation results. It has two significant advantages compared with other algorithms. One is that it restricts the search space to proportionate the size of the superpixels, which can significantly reduce the number of distance calculations during the optimization process. The other is that the weighted distance combines color and spatial metrics while control the number and compactness of superpixels.

The SLIC method performs a k-means-based local clustering algorithm in a five-dimensional space distance measurement, which achieves compactness and regularity in superpixel shapes. L represents luminosity, and ranges from 0 (black) to 100 (white). The color-associated elements a and b represent the range of colors from magenta to green and yellow to blue, respectively. The *Lab* color model not only contains the entire color gamut in RGB and $CMYK$, but also colors they cannot present. The *Lab* space can express all colors perceived by the human eyes, and is perceptually uniform for small color distance. Instead of directly using the Euclidean distance in this five-dimensional space, SLIC introduces a new distance measure that considers superpixels' size. For an image with N pixels, the SLIC algorithm takes a desired number of equally-sized superpixels N/K as input. There is a superpixel center at each grid interval $S = \sqrt{N/K}$. SLIC is easy in practice application because it only needs to set the unique superpixels' desired number K.

The clustering procedure begins with K initial cluster centers (seeds) $C_k = [l_k, a_k, b_k, x_k, y_k]^T$, which are sampled on regular grids with an S pixels interval. Each seed is moved to the lowest gradient position in its 3 × 3 neighborhood to avoid centering the superpixel at the edge position and reduce the chance of seeding with noisy pixels. We assume that the cluster center is located within a $2S \times 2S$ area since the spatial extent of a superpixel is about S^2. This strategy can accelerate the convergence process, as shown in Figure 2. The SLIC superpixel segmentation algorithm is a simple local k-means algorithm, whose search area is the $2S \times 2S$ area nearest to each cluster center. Since each pixel is searched by multiple seed pixels, the cluster center of the pixel is the seed with the minimum value.



(**a**) Traditional k-means search (**b**) SLIC searches in the $2S \times 2S$ area

Figure 2. Simple linear iterative clustering (SLIC) reducing the search regions.

Let $[l_i, a_i, b_i, x_i, y_i]^{\mathrm{T}}$ be the five-dimensional vector of a pixel. Calculate the distance between each pixel and the seeds. The distance between the pixel and seed C_k is shown as follows:

$$D_{lab}=\sqrt{(l_k-l_i)^2+(a_k-a_i)^2+(b_k-b_i)^2} \tag{4}$$

$$D_{xy}=\sqrt{(x_k-x_i)^2+(y_k-y_i)^2} \tag{5}$$

$$D_S=D_{lab}+\frac{l}{S}D_{xy} \tag{6}$$

The variable l controls the superpixels' compactness. The larger the value l is, the more compact the cluster will be.

The above steps are repeated iteratively until convergence, which means that the cluster center of each pixel is no longer changing. l ranges from 1 to 20. Here, we chose $l = 10$, which roughly matches the empirically perceptually meaningful *Lab* distance and provides good balance between spatial proximity and color similarity.

After the iterative optimization, superpixel multi-connected conditions may occur. Some superpixels' sizes may be too small, a single superpixel may be segmented into a plurality of discontinuous superpixels, etc., which can be solved by enhancing the connectivity of the superpixels. The solution is to reassign discrete and small-size superpixels to their adjacent superpixels until all the pixels have been traversed.

2.3. Robust Principal Component Analysis

Decomposing the matrix into low-rank and sparse parts can separate the main components from outliers or noise, which is suitable for mining low-dimensional manifold structures in high-dimensional data. We stack the pixels in the same homogeneous region into a matrix $\mathbf{X}_k \in \mathbb{R}^{\mathcal{N}\times\mathbf{n}_k}$, where $\mathbf{n}_k$ is the pixel number in the kth homogeneous region. Since the matrix has low-rank property, a low-rank matrix recovery algorithm is employed to recover the low-dimensional structure of all pixels in the homogeneous region. After recovering the low-rank matrix, the low-rank matrices are combined together into an integrated low-rank graph. The low-rank matrix restored by low-rank representation is a square matrix, while the dimensionality of the low-rank matrix restored by robust principal component analysis is the same as the original $\mathbf{X}_k \in \mathbb{R}^{\mathcal{N}\times\mathbf{n}_k}$. Since the number of pixels in each homogeneous region is different, robust principal component analysis is suitable for restoring the low-dimensional structures of all pixels in a homogeneous region here.

Principal Component Analysis is a fundamental operation in data analysis, which assumes that the data approximately lies in a low-dimensional linear subspace [16]. The success and popularity of PCA reveals the ubiquity of low-rank matrices. When the data are slightly damaged by small noise, it can be calculated stably and efficiently by singular value decomposition. However, noise and outliers limit PCA's performance and applicability in real scenarios.

Suppose we stack samples as column vectors of a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$. The problem of classical PCA is to seek the best estimate of $\mathbf{Z}$ that minimizes the difference between $\mathbf{Z}$ and $\mathbf{X}$:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} \ \|\mathbf{E}\|_F \\ &s.t. \ \ rank(\mathbf{Z}) \le r, \mathbf{X} = \mathbf{Z} + \mathbf{E} \end{aligned} \tag{7}$$

where $\|.\|_F$ denotes the Frobenius norm. r is the upper-bound rank of the low-rank matrix $\mathbf{Z}$. This problem can be efficiently solved via singular value decomposition (SVD) when the noise $\mathbf{E}$ is independent and identically distributed (i.i.d.) small Gaussian noise. A major disadvantage of classic PCA is its robustness to grossly corrupted or outlying observations [16]. In fact, even though only one element in the matrix changes, the obtained estimation matrix is far from the ground truth.

Recently, Wright et al. [37] proposed an idealized robust PCA model to extract low-rank structures from highly polluted data. In contrast to the traditional PCA, the proposed RPCA can exactly extract the low-rank matrix and the sparse error $\mathbf{E}$ (relative to $\mathbf{Z}$). The robust principal component analysis model is expressed as follows:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} \ rank(\mathbf{Z}) + \lambda\|\mathbf{E}\|_0 \\ &s.t. \ \mathbf{X} = \mathbf{Z} + \mathbf{E} \end{aligned} \tag{8}$$

where λ is to balance low-rank matrix $\mathbf{Z}$ and the error matrix $\mathbf{E}$. We recover the pair $(\mathbf{Z}_0, \mathbf{E}_0)$ that were generated from data $\mathbf{X}$. Unfortunately, Equation (8) is a very non-convex optimization problem with no valid solution. Replacing the rank function with the kernel norm, and replacing the l_0-norm with the l_1-norm, results in the following relaxing Equation (8) convex surrogate:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} \ \|\mathbf{Z}\|_* + \lambda\|\mathbf{E}\|_1 \\ &s.t. \ \mathbf{X} = \mathbf{Z} + \mathbf{E} \end{aligned} \tag{9}$$

The optimization in Equation (9) can be treated as a general convex optimization problem. Currently, the optimization algorithms of RPCA mainly contain Iterative Thresholding (IT), Accelerated Proximal Gradient (APG), and DUAL. However, the iterative thresholding algorithm proposed in [37] converges slowly. The APG algorithm is similar to the IT algorithm, but the number of iterations is significantly reduced. In addition, the DUAL algorithm does not require the complete singular value decomposition of the matrix, so it has better scalability than the APG algorithm. Recently, some researchers proposed Augmented Lagrangian Multiplier (ALM) algorithm, which has a faster speed than previous algorithms. The exact ALM (EALM) method turns out that it has a satisfactory Q-linear convergence speed, whereas the APG is theoretically only sub-linear. A slight improvement over the exact ALM leads to an inexact ALM (IALM) method, which converges as fast as the exact ALM. Therefore, the IALM method is applied to obtain the optimization here.

3. Methodology

In the above section, we discuss the simple linear iterative clustering and robust principle component analysis. Furthermore, labeled and unlabeled samples can be exploited simultaneously by depicting the underlying superpixels' low-rank subspace structure. Considering this, we have attempted to decipher our present work the superpixel segmentation and $l_{2,1}$-norm pobust principal component analysis feature extraction model.

3.1. *Superpixel Segmentation and $l_{2,1}$-Norm Robust Principal Component Analysis*

Hyperspectral images preserve low-rank properties, but the category composition corresponding to a hyperspectral image is very complicated in practical applications. The robust principal component analysis model supposes that the data lie in one low-rank subspace, which makes it difficult to characterize the structure of the data with multiple subspaces. Hence, we explore the robust principal component analysis via superpixel, which is usually defined as a homogeneous region. Here, we employ the SLIC superpixel segmentation method to segment the hyperspectral image into spatially homogeneous regions. Each superpixel is a unit with consistent visual perception, and the pixels in one superpixel are almost identical in features and belong to the same class. Initially, we extract the information of the HSIs by image fusion and recursive filtering [12].

Suppose $R = (r_1, r_2, \cdots, r_D) \in \mathbb{R}^{M\times D}$ represent the original hyperspectral image, which has M pixels and D-dimensional bands. We segment the whole hyperspectral bands into multiple subsets spectrally. Each subset is composed of L contiguous bands. The number of subsets $\mathcal{N} = \lfloor D/L \rfloor$, which represents the largest integer not greater than D/L. The ith ($i \in (1, 2, \cdots, \mathcal{N})$) subset is shown as follows:

$$\mathbf{P}^i = \begin{cases} (r_{L*(i-1)+1}, \cdots, r_{L*i}), \; if \; (L*i) \leq D \\ (r_{L*(i-1)+1}, \cdots, r_D), \; otherwise. \end{cases} \tag{10}$$

Afterwards, the adjacent bands of each subset are fused by an image fusion method (i.e., the averaging method). For example, the ith fusion band, that is, the image fusion feature F_i, is as follows:

$$Q^i = \frac{\sum_{n=1}^{N_i} \mathbf{P}_n^i}{N_i} \tag{11}$$

Here, $\mathbf{P}_n^i$ refers to the nth band in the ith subset of a hyperspectral image. N_i is the band number in the ith subset. After image fusion, we remove the noise pixels and redundant information for each subset. Then, we transform the domain recursive filtering algorithm on Q^i to obtain the ith feature:

$$O^i = \mathcal{RF}_{\delta_s, \delta_r}(Q^i) \tag{12}$$

Here, $\mathcal{RF}$ represents the domain recursive filtering algorithm. δ_s is the filter's spatial standard deviation, and δ_r is defined as the range standard deviation [60]. Then, we get the feature image $\mathbf{O} = [O^1, \cdots, O^{\mathcal{N}}] \in \mathbb{R}^{M \times \mathcal{N}}$.

Image fusion and recursive filtering (IFRF) is a simple yet powerful feature extraction method that aims at finding the best subset of hyperspectral bands that provide high classes' separability. In other words, the IFRF feature method is used to select better bands that remove the noise and redundant information simultaneously [12]. Hence, we preprocess the HSIs firstly by IFRF to eliminate redundant information.

Let $\mathbf{X} = \mathbf{O}^{\mathrm{T}} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M] \in \mathbb{R}^{\mathcal{N} \times M}$ be the preprocessed features vector, where $\mathbf{x}_i$ represents a pixel with $\mathcal{N}$ band numbers in the hyperspectral image, and M is the pixels number. We begin by sampling the K cluster centers and moving them to the lowest gradient position in the 3×3 neighborhood of the cluster centers. The squared Euclidean norm of the image is calculated by Equation (13):

$$\mathrm{G(x,y)} = \|\mathbf{I}(\mathrm{x}+1,\mathrm{y}) - \mathbf{I}(\mathrm{x}-1,\mathrm{y})\|^2 + \|\mathbf{I}(\mathrm{x},\mathrm{y}+1) - \mathbf{I}(\mathrm{x},\mathrm{y}-1)\|^2 \tag{13}$$

where $\mathbf{I}(\mathrm{x,y})$ represents the L, a, b vector corresponding to the pixel at position (x, y), and $\|.\|$ is the l_2-norm. This takes both color and intensity information into account.

Each pixel in the image is associated with the nearest cluster center whose search area overlaps the pixel. After each pixel is associated with its nearest cluster center, a new center is computed as the average *Labxy* vector of all the pixels in the cluster. Then, we iteratively associate the pixel with the nearest cluster center and recalculate the cluster center until convergence. We enforce the connection by relabeling the disjoint segmentations with their largest neighboring cluster's label. The simple linear iterative clustering algorithm is summarized in Algorithm 1.

Algorithm 1 SLIC superpixel segmentation.

```
Input: Processed HSIs image $\mathbf{X} \in \mathbb{R}^{\mathcal{N} \times M}$, Desired number of approximately superpixels $K$.
Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^{\mathrm{T}}$ by sampling pixels at regular grid steps $S$.
Move cluster centers to the lowest gradient position in a $3 \times 3$ neighborhood.
Set label $l(i) = -1$ for each pixel $i$.
Set distance $d(i) = \infty$ for each pixel $i$.
Repeat
for each cluster center $C_k$ do
    for each pixel $i$ in a $2S \times 2S$ region around $C_k$ do
        Compute the distance $D$ between $C_k$ and $i$.
        if $D < d(i)$ then
            set $d(i) = D$
            set $l(i) = k$
        end if
    end for
end for
Compute new cluster centers.
Compute residual error $E$.
Until $E \leq$threshold.
Enforce connectivity.
Output: Superpixel segmentations.
```

Hyperspectral images are segmented into many irregularly homogeneous regions. We stack the pixels in one homogeneous region into a matrix. We record the segment image as $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_m\}$, where $k \in \{1, 2, \cdots, m\}$ is the index of the superpixels. m is the exact total number of superpixels in $\mathbf{X}$. $\mathbf{n}_k$ is the number of pixels in the superpixel $\mathbf{X}_k$, and each column represents the bands of one pixel. Due to the low-rank attribute of the data, robust principal component analysis is used to extract the low-dimensional structure of all pixels in the homogeneous region. Then, the observed data matrix

$\mathbf{X}_k \in \mathbb{R}^{\mathcal{N} \times \mathbf{n}_k}$ can be decomposed into a low-rank matrix and a sparse matrix. That is, $\mathbf{X}_k = \mathbf{Z}_k + \mathbf{E}_k$, where $\mathbf{Z}_k$ represents the low-rank matrix of $\mathbf{X}_k$, and $\mathbf{E}_k$ indicates the sparse noise matrix. Further, the RPCA optimization problem for each superpixel region is converted to the following form:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} \ \|\mathbf{Z}_k\|_* + \lambda \|\mathbf{E}_k\|_1 \\ &s.t.\ \mathbf{X}_k = \mathbf{Z}_k + \mathbf{E}_k \end{aligned} \tag{14}$$

In the RPCA model, noise is required to be sparse, and the structural information of the noise is not considered. However, in machine learning or image processing fields, each column (or row) of the matrix has some meaning (e.g., a picture, and a signal data, etc.). Here, each column represents a pixel, which causes the noise to be sparse in column (or row). This structural information cannot be represented by the definition of an l_1-norm. To generate structural sparsity, the $l_{2,1}$-norm is proposed in the robust principal component analysis.

Unlike the l_1-norm, the $l_{2,1}$-norm produces sparsity based on columns (or rows). For the matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, its $l_{2,1}$- norm is [61,62]:

$$\|\mathbf{X}\|_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{m} \mathbf{x}_{i,j}^2} = \sum_{i=1}^{m} \|\mathbf{X}_{:,j}\|_2 \tag{15}$$

Thus, the $l_{2,1}$-norm is the sum of the l_2-norm of column vectors, which is also a measure of joint sparsity between vectors. For data that are interfered by structured noise, the following $l_{2,1}$-norm RPCA model in Equation (16) is very suitable:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} \ \|\mathbf{Z}_k\|_* + \lambda \|\mathbf{E}_k\|_{2,1} \\ &s.t.\ \mathbf{X}_k = \mathbf{Z}_k + \mathbf{E}_k \end{aligned} \tag{16}$$

In contrast to the original RPCA model, we call the error term with $l_{2,1}$-norm in Equation (16) as RPCA$_{2,1}$. Here, the inexact augmented Lagrangian multiplier (ALM) algorithm [63,64] is utilized for the optimal solution $\mathbf{Z}_k$.

The augmented Lagrangian multiplier method solves the optimization problem by minimizing the Lagrangian function as follows:

$$L(\mathbf{Z}, \mathbf{E}, \mathbf{Y}, \mu) = \|\mathbf{Z}\|_* + \lambda \|\mathbf{E}\|_{2,1} + \langle \mathbf{Y}, \mathbf{X} - \mathbf{Z} - \mathbf{E} \rangle + \tfrac{\mu}{2} \|\mathbf{X} - \mathbf{Z} - \mathbf{E}\|_F^2 \tag{17}$$

where $\mathbf{Y}$ is defined as Lagrangian multipliers and $\mu > 0$ is the penalty parameter. The sub-problem for all variables is convex, which can supply a relevant and unique solution. Alternate iterations $\mathbf{Z}_k$, $\mathbf{E}_k$, and $\mathbf{Y}$:

$$\begin{aligned} \mathbf{Z}_{i+1} &= \arg\min_{\mathbf{Z}} L(\mathbf{Z}, \mathbf{E}_i, \mathbf{Y}_i, \mu) \\ &= \arg\min_{\mathbf{Z}} \|\mathbf{Z}\|_* + \langle \mathbf{Y}_i, \mathbf{X} - \mathbf{Z} - \mathbf{E}_i \rangle + \tfrac{\mu}{2} \|\mathbf{X} - \mathbf{Z} - \mathbf{E}_i\|_F^2 \\ &= \arg\min_{\mathbf{Z}} \|\mathbf{Z}\|_* + \tfrac{\mu}{2} \left\|\mathbf{Z} - (\mathbf{X} - \mathbf{E}_i + \tfrac{\mathbf{Y}_i}{\mu})\right\|_F^2 \\ &= \arg\min_{\mathbf{Z}} \tfrac{1}{\mu} \|\mathbf{Z}\|_* + \tfrac{1}{2} \left\|\mathbf{Z} - (\mathbf{X} - \mathbf{E}_i + \tfrac{\mathbf{Y}_i}{\mu})\right\|_F^2 \end{aligned} \tag{18}$$

$$\begin{aligned} \mathbf{E}_{i+1} &= \arg\min_{\mathbf{E}} L(\mathbf{Z}_{i+1}, \mathbf{E}, \mathbf{Y}_i, \mu) \\ &= \arg\min_{\mathbf{E}} \lambda \|\mathbf{E}\|_{2,1} + \langle \mathbf{Y}_i, \mathbf{X} - \mathbf{Z} - \mathbf{E}_i \rangle + \tfrac{\mu}{2} \|\mathbf{X} - \mathbf{Z} - \mathbf{E}_i\|_F^2 \\ &= \arg\min_{\mathbf{E}} \lambda \|\mathbf{E}\|_{2,1} + \tfrac{\mu}{2} \left\|\mathbf{E} - (\mathbf{X} - \mathbf{Z} + \tfrac{1}{\mu}\mathbf{Y})\right\|_F^2 \\ &= \arg\min_{\mathbf{E}} \tfrac{\lambda}{\mu} \|\mathbf{E}\|_{2,1} + \tfrac{1}{2} \left\|\mathbf{E} - (\mathbf{X} - \mathbf{Z} + \tfrac{1}{\mu}\mathbf{Y})\right\|_F^2 \end{aligned} \tag{19}$$

Note that $\mathbf{Z}_{i+1}$ and $\mathbf{E}_{i+1}$ are convergent to ${\mathbf{Z}_{i+1}}^*$ and ${\mathbf{E}_{i+1}}^*$, respectively. Then, we update $\mathbf{Y}$ as follows:

$$\mathbf{Y}_{i+1} = \mathbf{Y}_i + \mu_i(\mathbf{X} - {\mathbf{Z}_{i+1}}^* - {\mathbf{E}_{i+1}}^*) \tag{20}$$

Finally, parameter μ is updated:

$$\mu_{i+1} = \begin{cases} \rho\mu_i, & if\ \mu_i \|{\mathbf{E}_{i+1}}^* - {\mathbf{E}_i}^*\|_F / \|\mathbf{X}\|_F < \varepsilon \\ \mu_i, & otherwise \end{cases} \tag{21}$$

where $\rho > 1$ is a constant and $\varepsilon > 0$ is a relatively small number.

The optimization process outline of the inexact ALM method [63] is given in Algorithm 2.

The initialization $\mathbf{Y}_0 = \mathbf{X}/J(\mathbf{X})$ is defined as

$$J(\mathbf{X}) = \max(\|\mathbf{X}\|_2, \lambda^{-1}\|\mathbf{X}\|_\infty) \tag{22}$$

where $\|\cdot\|_\infty$ is the maximum absolute value of the matrix entries.

Algorithm 2 Inexact ALM method for solving RPCA.

```
Input: Observation matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ and parameter $\lambda$ for local affinity.
 1: Initialize: $\mathbf{Y}_0 = \mathbf{X}/J(\mathbf{X}); \mathbf{E}_0 = \mathbf{0}; \mu_0 > 0; \rho > 1; k = 0$.
 2: while not converged do
 3:     //Lines 4-5 resolve $\mathbf{Z}_{i+1} = \arg\min_{\mathbf{Z}} L(\mathbf{Z}, \mathbf{E}_i, \mathbf{Y}_i, \mu_i)$.
 4:     $(\mathbf{U}, \mathbf{S}, \mathbf{V}) = svd(\mathbf{X} - \mathbf{E}_i + {\mu_i}^{-1}\mathbf{Y}_i)$;
 5:     $\mathbf{Z}_{i+1} = \mathbf{U}\mathbf{S}_{{\mu_i}^{-1}}[\mathbf{S}]\mathbf{V}^T$.
 6:     //Line 7 resolves $\mathbf{E}_{i+1} = \arg\min_{\mathbf{E}} L(\mathbf{Z}_{i+1}, \mathbf{E}, \mathbf{Y}_i, \mu_i)$.
 7:     $\mathbf{E}_{i+1} = \mathbf{S}_{{\mu_i}^{-1}}(\mathbf{X} - \mathbf{Z}_{i+1} + {\mu_i}^{-1}\mathbf{Y}_i)$.
 8:     $\mathbf{Y}_{i+1} = \mathbf{Y}_i + \mu_i(\mathbf{X} - {\mathbf{Z}_{i+1}}^* - {\mathbf{E}_{i+1}}^*)$ ; $\mu_{i+1} = \rho\mu_i$.
 9:     $i \leftarrow i + 1$.
10: end while
Output: The low-rank matrix $\mathbf{Z}$.
```

After RPCA$_{2,1}$, we get the low-rank matrix $\mathbf{Z}_k \in \mathbb{R}^{\mathcal{N} \times \mathbf{n}_k}$ of each superpixel. Then, we merge these low-rank matrices to a whole low-rank graph $\mathbf{Z} = \{\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_m\} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M] \in \mathbb{R}^{\mathcal{N} \times M}$, which is the low-rank representation of $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M]$.

3.2. HSIs' Classification Based on the SURPCA$_{2,1}$ Graph

To overcome the "Curse of dimensionality" caused by the high dimension, we apply semi-supervised discriminant analysis to reduce the dimension. SDA is derived from linear discriminant analysis (LDA). The rejection matrix $\mathbf{a}$ can represent a priori consistency assumptions according to the regularization term [65]:

$$\mathbf{a}_{opt} = \arg\max_{\mathbf{a}} \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_t \mathbf{a} + \alpha J(\mathbf{a})} \tag{23}$$

Here, S_t is the total class scatter matrix. Parameter α is used to balance the complexity and empirical loss of the model. Empirical loss $J(\mathbf{a})$ controls the learning complexity of the hypothesis family. The regularizer term $J(\mathbf{a})$ incorporates the prior knowledge into some particular applications. When a set of unlabeled examples available, we aim to construct a $J(\mathbf{a})$ incorporating the manifold structure.

Given a set of samples $\{\mathbf{z}_i\}_{i=1}^M$, we can construct the graph $\mathbf{G}$ to represent the relationship between nearby samples by kNN. The samples $\mathbf{z}_i$ and $\mathbf{z}_j$ are considered as k-nearest neighbor if $\mathbf{z}_i$ is among the

k-nearest neighbor of $\mathbf{z}_j$ or $\mathbf{z}_j$ is among the k-nearest neighbor of $\mathbf{z}_i$. Here, we employ the most simple 0–1 weighting [21] methods to assign weights for $\mathbf{S}$:

$$\mathbf{S}_{ij} = \begin{cases} 1, & if\ \mathbf{z}_i \in N_k(\mathbf{z}_j)\ or\ \mathbf{z}_j \in N_k(\mathbf{z}_i) \\ 0, & otherwise. \end{cases} \tag{24}$$

where $N_k(\mathbf{z}_i)$ denotes k-nearest neighbor of $\mathbf{z}_i$.

The SDA model incorporate the prior knowledge in the regularization term $J(\mathbf{a})$, as follows:

$$\begin{aligned} J(\mathbf{a}) &= \sum_{ij} (\mathbf{a}^T\mathbf{z}_i - \mathbf{a}^T\mathbf{z}_j)^2 \mathbf{S}_{ij} \\ &= 2\sum_i \mathbf{a}^T\mathbf{z}_i\mathbf{D}_{ii}\mathbf{z}_i{}^T\mathbf{a} - 2\sum_{ij}\mathbf{a}^T\mathbf{z}_i\mathbf{S}_{ij}\mathbf{z}_j^T\mathbf{a} \\ &= 2\mathbf{a}^T\mathbf{Z}(\mathbf{D}-\mathbf{S})\mathbf{Z}^T\mathbf{a} \\ &= 2\mathbf{a}^T\mathbf{Z}\mathbf{L}\mathbf{Z}^T\mathbf{a} \end{aligned} \tag{25}$$

The diagonal matrix $\mathbf{D}$, $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij}$ is column (or row, since $\mathbf{S}$ is symmetric) sum of $\mathbf{S}$. $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is the Laplacian matrix [66]. Then, SDA objective function is:

$$\mathbf{a}_{opt} = \max_{\mathbf{a}} \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T(S_t + \alpha\mathbf{Z}\mathbf{L}\mathbf{Z}^T)\mathbf{a}} \tag{26}$$

We obtain the projective vector $\mathbf{a}$ by maximizing the generalized eigenvalue problem where d is the weight matrix's rank for the labeled graph.

$$S_b\mathbf{a} = \lambda(S_t + \alpha\mathbf{Z}\mathbf{L}\mathbf{Z}^T)\mathbf{a} \tag{27}$$

The low-rank matrix extracted by RPCA captures the global correlation of the homogeneous region [37,39,67,68], whereas the k-nearest neighbor algorithm characterizes the local correlation of pixel points [69]. The probability that the neighboring pixels belong to the same category is large, which corresponds to the spatial distribution structure of the hyperspectral image. Therefore, the SURPCA$_{2,1}$ graph can achieve good feature representations for graph-based semi-supervised dimensionality reduction.

Given a set of samples $\{(\mathbf{z}_i, y_i)\}_{i=1}^{l}$ and unlabeled samples $\{\mathbf{z}_i\}_{i=l+1}^{m}$ with c classes, the l_k is the kth class's samples number. The algorithmic procedure of HSIs' classification by applying the SLIC superpixel-based $l_{2,1}$-norm robust principal component analysis method is stated in the following paragraphs:

Step 1 Construct the adjacency graph: Construct the block low-rank and kNN graph $\mathbf{S}$ in Equation (24) for the regularization term. Furthermore, calculate the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{S}$.

Step 2 Construct the labeled graph: For the labeled graph, construct the matrix as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{l\times l} & 0 \\ 0 & 0 \end{bmatrix} \tag{28}$$

Define identity matrix $\tilde{\mathbf{I}} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{l\times l}$.

Step 3 Eigen-problem: Calculate the eigenvectors for the generalized eigenvector problem:

$$\mathbf{Z}\mathbf{W}\mathbf{Z}^T\mathbf{a} = \lambda\mathbf{Z}(\tilde{\mathbf{I}} + \alpha\mathbf{L})\mathbf{Z}^T\mathbf{a} \tag{29}$$

d is the rank of $\mathbf{W}$, and $\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_d\}$ is the d eigenvectors.

Step 4 Regularize discriminant analysis embedding: Let the transformation matrix $A = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_d] \in \mathbb{R}^{\mathcal{N} \times d}$. Then, embedding the data into d-dimensional subspace,

$$\mathbf{z} \to \mathbf{z}' = A^{\mathrm{T}}\mathbf{z} \tag{30}$$

Step 5 Finally, apply the simple and ubiquitously-used classifiers nearest neighbor and Support Vector Machine (SVM) for classification.

Figure 3 shows a model diagram of the SURPCA$_{2,1}$ for hyperspectral image classification. The low-rank matrix measures the global correlation of the homogeneous regions, while the kNN algorithm preserves the local correlation of the pixels. The probability that adjacent pixels belong to the same category in the kNN is large, which corresponds to the spatial distribution structure of the hyperspectral image. Therefore, the SURPCA$_{2,1}$ graph can be used to achieve high-quality data representation.

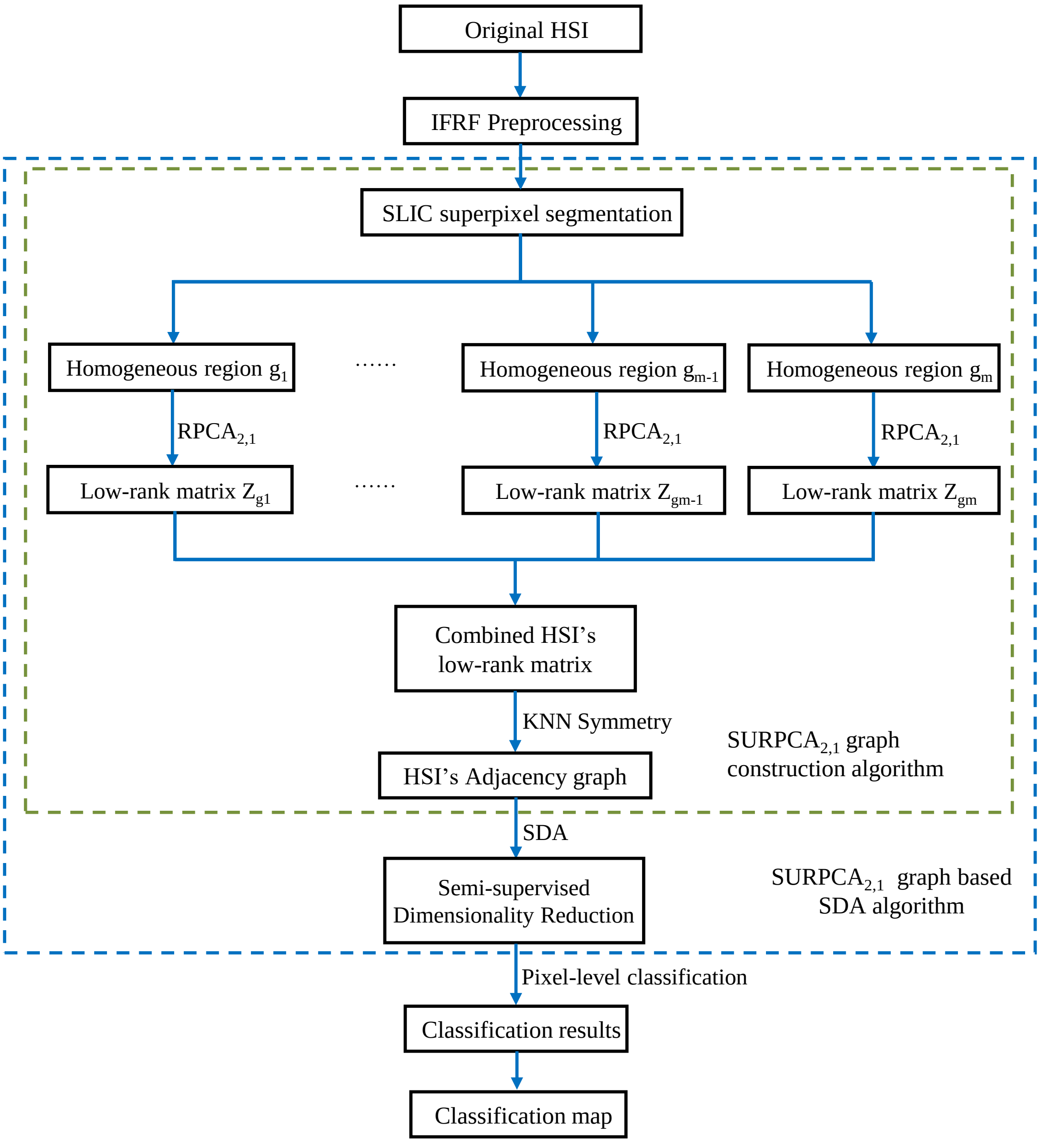

Original HSI
IFRF Preprocessing
SLIC superpixel segmentation
Homogeneous region g_1
……
Homogeneous region g_{m-1}
Homogeneous region g_m
RPCA$_{2,1}$
RPCA$_{2,1}$
RPCA$_{2,1}$
Low-rank matrix Z_{g1}
……
Low-rank matrix Z_{gm-1}
Low-rank matrix Z_{gm}
Combined HSI's low-rank matrix
KNN Symmetry
HSI's Adjacency graph
SURPCA$_{2,1}$ graph construction algorithm
SDA
Semi-supervised Dimensionality Reduction
SURPCA$_{2,1}$ graph based SDA algorithm
Pixel-level classification
Classification results
Classification map

Figure 3. Model diagram of the SURPCA$_{2,1}$ for hyperspectral image classification.

4. Experiments and Analysis

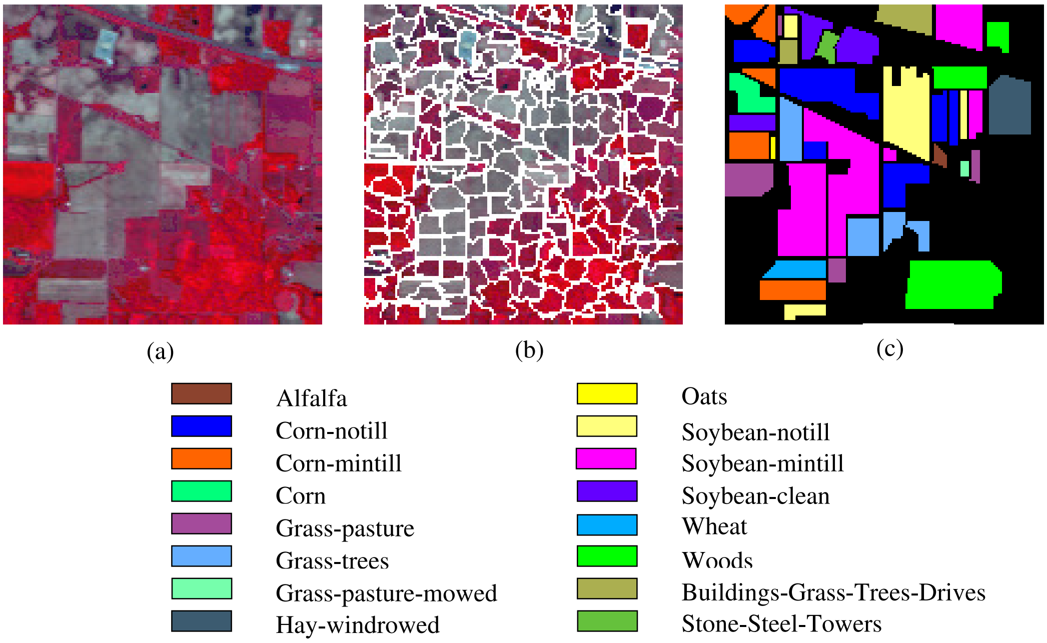
To investigate the performance of the SLIC superpixel-based $l_{2,1}$-norm robust principal component analysis graph model, we conducted extensive experiments on several real multi-class hyperspectral images. We performed experiments on a computer with CPU 2.60 GHz and 8 GB RAM.

4.1. Experimental Setup

4.1.1. Hyperspectral Images

The Indian Pines (https://purr.purdue.edu/publications/1947/usage?v=1), Pavia University scene, and Salinas (http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes) hyperspectral images were evaluated in the experiments.

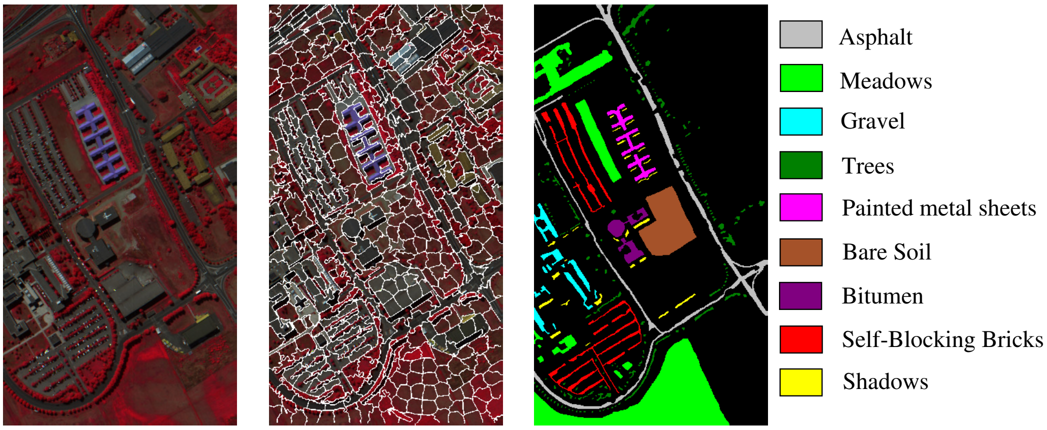
- The Indian Pines image is for the agricultural Indian Pine test site in Northwestern Indiana, which was a 220 Band AVIRIS Hyperspectral Image Data Set: 12 June 1992 Indian Pine Test Site 3. It was acquired over the Purdue University Agronomy farm northwest of West Lafayette and the surrounding area. The data were acquired to support soils research being conducted by Prof. Marion Baumgardner and his students [70]. The wavelength is from 400 to 2500 nm. The resolution is 145 × 145 pixels. Because some of the crops present (e.g., corn and soybean) are in the early stages of growth in June, the coverage is minuscule—approximately less than 5%. The ground truth is divided into sixteen classes, and are not all mutually exclusive. The Indian Pines false-color image and the ground truth image are presented in Figure 4.
- The Pavia University scene was obtained from an urban area surrounding the University of Pavia, Italy on 8 July 2002 with a spatial resolution of 1.3 m. The wavelength ranges from 0.43 to 0.86 µm. There are 115 bands with size 610 × 340 pixels in the image. After removing 12 water absorption bands, 103 channels were left for testing. The Pavia University scene's false-color image and the corresponding ground truth image are shown in Figure 5.



(a) (b) (c)

(d)

Figure 4. Indian Pines dataset: (**a**) false-color image; (**b**) segmentation map; and (**c**,**d**) ground truth image and reference data.



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

Figure 5. Pavia University scene; (**a**) false-color image; (**b**) segmentation map; and (**c**,**d**) ground truth image and reference data.

- The Salinas image is from the Salinas Valley, California, USA, which was obtained from an AVIRIS sensor with 3.7 m spatial resolution. The image includes a size of 512 × 217 with 224 bands. Twenty water absorption bands were discarded here. There are 16 classes containing vegetables, bare soil, vineyards, etc. Figure A1 displays the Salinas image's false-color and ground truth images.

4.1.2. Evaluation Criteria

We give some evaluation criteria to evaluate the proposed method for HSIs, as follows.

Classification accuracy (CA) is the classification accuracy of each category in the image. The confusion matrix [71] is often used in the remote sensing classification field, and its form is defined as: $M = [m_{ij}]_{n \times n}$, where m_{ij} indicates that the number of pixels labeled by the j class should belong to the i class. n is the class number. The reliability of classification depends on the diagonal values of the confusion matrix. The higher the values on the diagonal of the confusion matrix are, the better the classification results will be.

We also used the following three main indicators: overall accuracy (OA), average accuracy (AA), and the kappa coefficient [72]. OA refers to the percentage of the overall correct classification. AA estimates the average correct classification percentage for all categories. The kappa coefficient takes the chance agreement into account and fixes it, whereas OA and AA check how many pixels are classified correctly. Assuming that the reference classification (i.e., ground truth) is true, then how well they agree is measured. Here, we assumed that both classification and reference classification were independent class assignments of equal reliability. The advantage of the kappa coefficient over overall accuracy is that the kappa coefficient considers chance agreement and corrects it. The chance agreement is the probability that the classification and reference classification agree by mere chance. Assuming statistical independence, we obtained this probability estimation [73,74].

4.1.3. Comparative Algorithms

We give several comparative methods to illustrate the great improvement by the SLIC superpixel-based $l_{2,1}$-norm robust principal component analysis graph model in the HSIs' classification. For fairness, these comparative algorithms incorporate the simple linear iterative clustering (SLIC), which are shown below.

- RPCA (robust principal component analysis) method [63]: The original robust principal component analysis with l_1-norm.

- PCA (principal component analysis) method [63]: PCA [16] seeks the best low-rank representation of the given data matrix that minimizes the difference between **Z** and **X**:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} \ \|\mathbf{E}\|_F \\ &s.t. \ \ rank(\mathbf{Z}) \leq r, \mathbf{X} = \mathbf{Z} + \mathbf{E} \end{aligned} \tag{31}$$

- IFRF (image fusion and recursive filtering) algorithm.
- Origin: Original bands of the unprocessed image.

4.2. Classification of Hyperspectral Images

We performed experiments on the three hyperspectral images to examine the performance of the SURPCA$_{2,1}$ graph. Ten independent runs of each algorithm were evaluated by resampling the training samples in each run. We chose the mean values as the results. In practice, it is difficult to obtain labeled samples, while unlabeled samples are often available and in large numbers. Unlike most of the existing HSI classifications, we tested the performance of all comparative methods using only small rate of the labeled samples. Table 1 shows the training and testing sets for all datasets, where the training sets are chosen randomly. The training samples were approximately 4%, 3%, and 0.4% for the three images, which were minimal sets to the entire dataset. Considering the classes with a meager number of samples, we incorporated a minimum threshold of training samples. Here, we set the minimum threshold of training samples for each class as five, which can eradicate the difference between the classes with a low number of samples. The SLIC superpixels' number for the three HSIs is 200, 600, and 400, respectively. Figures 4b, 5b and A1b show the segmentation maps. The amount of reduced dimension is 30 for the three hyperspectral images. The spatial standard deviation and range standard deviation of the filter are δ_s and δ_r with 200 and 0.3, respectively. The parameter σ in k-nearest neighbor $\mathbf{S}_{ij} = \exp(-\frac{\|\mathbf{z_i}-\mathbf{z_j}\|^2}{2\sigma^2})$ is 0.1, which is provided randomly.

Table 1. Training and testing samples for the three hyperspectral images.

Class	Indian Pines Image			Pavia University Scene			Salinas Image		
	Train	Test	Sample No.	Train	Test	Sample No.	Train	Test	Sample No.
C1	7	39	46	205	6426	6631	14	1995	2009
C2	63	1365	1428	565	18,084	18,649	20	3706	3726
C3	39	791	830	69	2030	2099	13	1963	1976
C4	15	222	237	97	2967	3064	11	1383	1394
C5	25	458	483	46	1299	1345	16	2662	2678
C6	35	695	730	156	4873	5029	21	3938	3959
C7	7	21	28	45	1285	1330	20	3559	3579
C8	25	453	478	116	3566	3682	51	11,220	11,271
C9	6	14	20	34	913	947	30	6173	6203
C10	44	928	972	-	-	-	19	3259	3278
C11	104	2351	2455	-	-	-	10	1058	1068
C12	29	564	593	-	-	-	13	1914	1927
C13	14	191	205	-	-	-	9	907	916
C14	56	1209	1265	-	-	-	10	1060	1070
C15	21	365	386	-	-	-	35	7233	7268
C16	9	84	93	-	-	-	13	1794	1807
Total	702	9547	10,249	1762	41,014	42,776	305	53,824	54,129

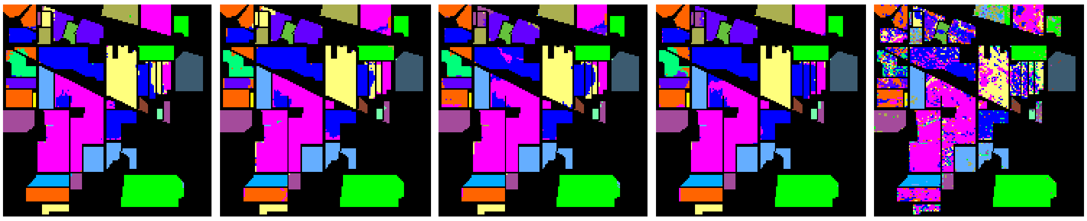
In the beginning, we utilized different manifold mapping graphs to obtain the manifold matrix of the hyperspectral images. Then, we applied the SDA algorithm for semi-supervised dimensionality reduction. Both the nearest neighbor and support vector machine algorithms were applied as the classifiers here to test and verify the proposed SURPCA$_{2,1}$ graph. Tables 2, 3 and A1 give detailed classification results for CA, OA, AA, and kappa coefficients received from different methods, where bold numbers represent the best ones for various graphs. Figures 6, 7 and A2 indicate classification maps of the hyperspectral images acquired by several methods (randomly selected from our experiments above) with the NN classifier. It was observed that the results in the figures were random for each method. We analyzed the running times of different models on the Indian Pines image, Pavia University scene, and Salinas scene image. Ten separate runs were calculated for the total time.

Table 2. Indian Pines image classification results.

Method	**RPCA$_{2,1}$**	**RPCA**	**PCA**	**IFRF**	**Origion**	**RPCA$_{2,1}$**	**RPCA**	**PCA**	**IFRF**	**Origin**
Classifier			**NN**					**SVM**		
C1	**1**	**1**	0.9870	0.9809	0.8051	**1**	**1**	**1**	**1**	0.9473
C2	0.9686	0.9442	0.9380	0.9487	0.4961	0.9676	**0.9701**	0.9295	0.9482	0.5169
C3	0.9667	0.9524	**0.9718**	0.9342	0.6572	0.9659	0.9679	0.9426	0.9574	0.3793
C4	0.9578	0.9698	0.9589	0.9122	0.3964	**0.9819**	0.9133	0.9383	0.9420	0.5671
C5	0.9868	0.9463	0.9384	0.9687	0.8225	**0.9933**	0.9856	0.9821	0.9868	0.8809
C6	0.9837	0.9833	0.9394	0.9483	0.7568	**1**	**1**	0.9986	0.9986	0.8476
C7	0.8686	0.7254	0.6164	0.8107	0.6786	**1**	0.9375	**1**	0.9886	0.9188
C8	**1**	**1**	**1**	**1**	0.9486	**1**	**1**	**1**	**1**	0.9367
C9	0.9800	0.7812	0.8496	0.7469	0.5042	**1**	**1**	**1**	**1**	0.8042
C10	0.9342	0.9139	0.9215	0.8742	0.6777	0.9647	0.9618	**0.9743**	0.9471	0.4916
C11	0.9721	0.9700	0.9748	**0.9749**	0.6933	0.9672	0.9604	0.9626	0.9677	0.5560
C12	0.9602	0.9024	0.9753	0.9344	0.6833	0.9854	**0.9908**	0.9658	0.9701	0.6454
C13	0.9705	**1**	0.9748	0.9935	0.9240	**1**	**1**	**1**	0.9987	0.9909
C14	0.9952	0.9895	0.9939	0.9915	0.9217	0.9954	**0.9996**	0.9975	0.9975	0.9190
C15	0.9624	0.9711	0.9768	0.9379	0.6666	0.9916	0.9937	**0.9958**	0.9897	0.7551
C16	**0.9905**	0.9881	0.9881	0.9881	0.9871	0.9866	0.9859	0.9712	0.9846	0.9825
OA	0.9706	0.9585	0.9594	0.9514	0.7077	**0.9786**	0.9759	0.9682	0.9713	0.6536
AA	0.9686	0.9399	0.9378	0.9341	0.7262	**0.9875**	0.9792	0.9787	0.9798	0.7587
Kappa	0.9664	0.9527	0.9536	0.9446	0.6633	**0.9755**	0.9724	0.9636	0.9672	0.6000

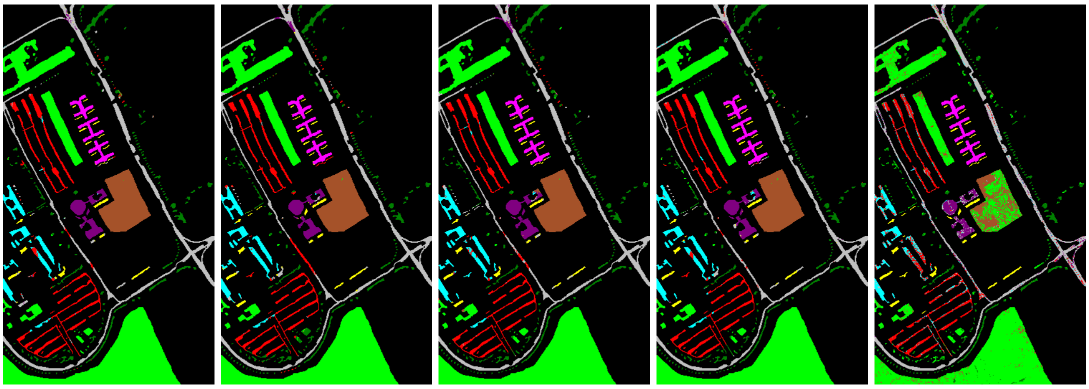
Table 3. Pavia University scene classification results.

Method	**RPCA$_{2,1}$**	**RPCA**	**PCA**	**IFRF**	**Origion**	**RPCA$_{2,1}$**	**RPCA**	**PCA**	**IFRF**	**Origin**
Classifier			**NN**					**SVM**		
C1	0.9581	0.9630	0.9632	0.9581	0.8969	**0.9771**	0.9745	0.9698	0.9657	0.8119
C2	0.9925	0.9942	0.9935	0.9944	0.8491	0.9957	**0.996**	0.9956	0.9951	0.8676
C3	0.9465	0.9420	0.9212	0.9543	0.6836	**0.9758**	0.9678	0.9715	0.9578	0.6425
C4	0.9923	0.9841	0.9865	0.9880	0.9360	0.9918	0.9889	**0.9938**	0.9895	0.8802
C5	**0.9993**	0.9989	0.9957	0.9985	0.9929	0.917	0.9377	0.9433	0.9355	0.9972
C6	0.9966	0.9953	0.9921	**0.9981**	0.7228	0.9965	0.9963	0.9953	0.9977	0.7286
C7	0.9177	0.9090	0.8921	0.8822	0.7385	**0.9751**	0.9689	0.9642	0.963	0.6193
C8	0.9169	0.9082	0.9182	0.9135	0.7990	**0.9656**	0.9589	0.9575	0.9574	0.7057
C9	0.9353	0.9302	0.9369	0.9509	**1**	0.964	0.9577	0.9546	0.9575	0.9996
OA	0.9751	0.9744	0.9734	0.9748	0.8425	**0.9849**	0.9839	0.9832	0.9815	0.8238
AA	0.9617	0.9583	0.9555	0.9598	0.8465	**0.9732**	0.9719	0.9717	0.9688	0.8058
Kappa	0.9670	0.9660	0.9647	0.9666	0.7869	**0.98**	0.9787	0.9777	0.9754	0.7628



(**a**) OA (0.9768) (**b**) OA (0.9644) (**c**) OA (0.9533) (**d**) OA (0.9601) (**e**) OA (0.7040)

Figure 6. Indian Pines image classification results: (**a**) robust principal component analysis (RPCA$_{2,1}$); (**b**) RPCA; (**c**) PCA; (**d**) image fusion and recursive filtering (IFRF); and (**e**) Origin. OA, overall accuracy.



(**a**) OA (0.9737) (**b**) OA (0.9619) (**c**) OA (0.9706) (**d**) OA (0.9703) (**e**) OA (0.8457)

Figure 7. Pavia University scene classification results: (**a**) RPCA$_{2,1}$; (**b**) RPCA; (**c**) PCA; (**d**) IFRF; and (**e**) Origin. OA, overall accuracy.

In Tables 2, 3, and A1, we can notice that the SURPCA$_{2,1}$ graph is superior to the other graph models. Therefore, it significantly improves the classification performance of hyperspectral images, which indicates that the SURPCA$_{2,1}$ graph is a superior HSIs feature extraction method with both NN and SVM classifiers. For example, in Table 2, although the overall accuracy and kappa coefficient of other graphs are high, the average accuracy is not good. The accuracy of Grass-pasture-mowed and Oats are more (i.e., 12–37%) improved with the NN classifier and with SVM classifier (i.e., 25–64%) than the compared graphs. In addition, the Corn, Soybean-notill, and Soybean-clean classes are significantly improved, especially the Grass-pasture-mowed, for the Indian Pines image. For the Salinas dataset, the classification of Fallow, Lettuce_romaine_6wk, and Vinyard_untrained classes are obviously improved. We can see that, although sometimes the overall accuracy is not very different, the average accuracy is far ahead of the compared graphs. This is because, when there are fewer samples in some classes, it is very easy to be confused. For the Indian Pines image, the accuracy of all categories is more than 89.81%, except for the 81.96% accuracy of the Grass-pasture-mowed with the NN classifier, and it is also more than 95.03% with the SVM classifier, which meets the general accuracy requirement of remote sensing. For the Pavia University scene and Salinas image, the accuracy of all categories is larger than 91.69% and 91.95%, and 91.7% and 97.45% with NN and SVM, respectively. We can also see that our graph performed well in all categories with both of the classifiers. However, for other methods, they might perform much worse in some confusing categories, especially with the NN classifier. This phenomenon shows that our proposed graph is robust to classifier methods.

We analyzed the running times of different models on the three hyperspectral images. We used ten separate runs to calculate the total time. We give the mean running time. As shown in Table 4, the execution time of the SURPCA$_{2,1}$ graph method is slightly longer than the other methods. Although our algorithm is slower than the traditional algorithms, the performance is much better than these baselines at an acceptable running time. Note that the RPCA$_{2,1}$ is significantly faster than the classic RPCA. This is owing to the superior ability of the $l_{2,1}$-norm to converge more quickly than the l_1-norm.

Although the classification results with NN are slightly worse with the SVM algorithm, the running time is much less than the SVM classifier. Moreover, in Tables 2, 3, and A1, we can see that the classification results with NN classifier are acceptable. Therefore, in the following experiments, the nearest neighbor is applied.

Table 4. Different graphs' running time on real-word hyperspectral images (HSIs) (unit:s). NN, nearest neighbor; SVM, support vector machine.

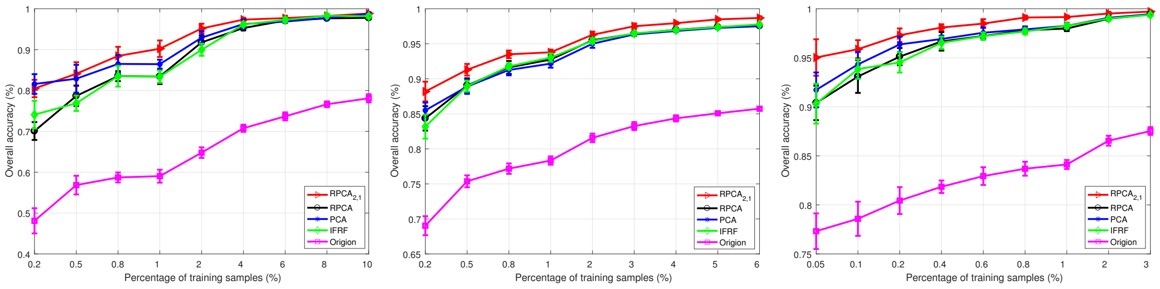
Method	**RPCA$_{2,1}$**	**RPCA**	**PCA**	**IFRF**	**Origion**	**RPCA$_{2,1}$**	**RPCA**	**PCA**	**IFRF**	**Origin**
Classifier	**NN**					**SVM**				
Indian Pines image	16.73	21.62	5.16	5.31	5.69	27.94	34.53	14.39	16.90	17.34
University of Pavia	83.89	100.31	54.28	56.92	47.11	115.84	135.86	88.52	90.26	116.61
Salinas image	15.55	18.14	4.24	4.27	3.22	28.56	32.61	14.39	14.32	15.29

4.3. Robustness of the SURPCA$_{2,1}$ Graph

We evaluated the performance of the proposed SURPCA$_{2,1}$ graph. To fully determine the superiority of the proposed graph model, we also analyzed the robustness of SURPCA$_{2,1}$. Considering the practical situation, we analyzed the robustness of SURPCA$_{2,1}$ on the size of the labeled samples and noise situation.

4.3.1. Labeled Size Robustness

We analyzed the impact of different sizes of training and testing sets. The experiments were carried out with ten independent runs for each algorithm. We calculated the mean values and standard deviations of the results. Figure 8 shows the classification results of different graph models. It compares the overall classification results with different training size samples in each class. The training samples percentages were 0.2–10%, 0.2–6%, and 0.05–3% for Indian Pines image, Pavia University scene, and Salinas image, respectively.



(**a**) Indian Pines image (**b**) Pavia University scene (**c**) Salinas image

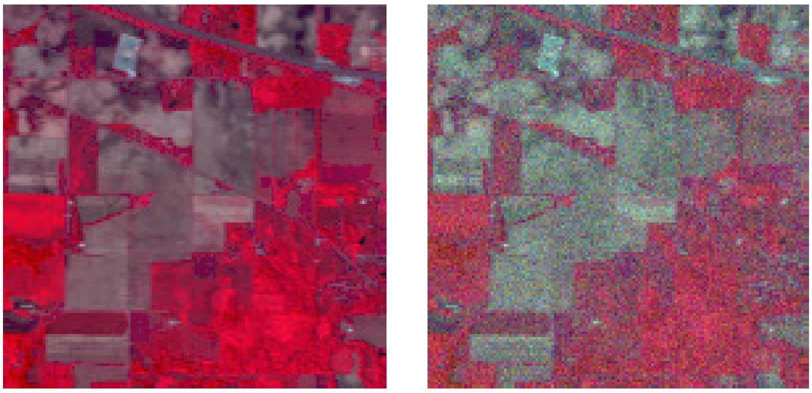
Figure 8. HSIs classification accuracy with varying labeled ratio.

In most cases, our graph model SURPCA$_{2,1}$ consistently achieves the best results, which are robust to the label percentage variations. With increasing size of the training sample, OA generally increases in all methods, showing a similar trend. When the labeled ratio is fixed, the SURPCA$_{2,1}$ method is usually superior to others. Similarly, the three classification criteria are increased with the increasing training samples simultaneously.

Note that, our proposed graph realize higher classification results, even with a low label ratio. The other comparison algorithms are especially not as robust as the SURPCA$_{2,1}$ graph with low label ratio. In particular, we can see that our graph model perform relatively better, and the difference between other comparison methods is larger when the labeling rate is smaller. Therefore, due to the high cost and difficulty of labeled samples, our proposed graph is much more robust and suitable for real-world hyperspectral images.

4.3.2. Noise Robustness

We evaluated the robustness to noise of the SURPCA$_{2,1}$ graph on the three hyperspectral images. We added zero-mean Gaussian noise with a signal-to-noise ratio (SNR) of 20 dB to each band. Figure 9 shows an example noise image of three randomly selected bands in the Indian Pines image, which is similar to the other two hyperspectral images.



(**a**) Clean image (**b**) Noisy image with SNR = 20 dB

Figure 9. False-color clean and noisy images of the Indian Pines image.

We evaluated different graphs' performance in the noisy situation. Each algorithm ran ten times independently, with re-sampling of training samples. We calculated the mean and standard deviation results. The labeled sampling ratios were 4%, 3%, and 0.4%, respectively, as described in Table 5. The bold numbers represent the best results for various graphs. Figures 10, 11, and A3 display classification maps of the three noise hyperspectral images correlated with OA values (randomly selected from our experiments above). It is observed that the results in the figures were random for each method.

From the results of these different graph models, we can see that our graph is robust to noise and the sample size of the labeled set. In a noisy environment, our method is relatively less degraded and more robust. With few labeled samples, the SURPCA$_{2,1}$ graph is more forceful than the other graphs due to the $l_{2,1}$-norm robust principal component analysis noise robustness. Moreover, the SURPCA$_{2,1}$ graph performs very well for all three experimental hyperspectral images.

Table 5. Classification results with Gaussian noise on three HSIs.

Images		**RPCA$_{2,1}$**	**RPCA**	**PCA**	**IFRF**	**Origin**
Indian Pines image	OA	0.9004 ± 0.0103	0.8615 ± 0.0137	0.8736 ± 0.0067	0.86261 ± 0.0937	0.5008 ± 0.006
	AA	0.9198 ± 0.0179	0.8481 ± 0.0261	0.8573 ± 0.024	0.8367 ± 0.0231	0.4530 ± 0.0256
	Kappa	0.8863 ± 0.011	0.8426 ± 0.0143	0.8561 ± 0.0257	0.8434 ± 0.00763	0.4230 ± 0.0069
University of Pavia image	OA	0.9261 ± 0.0041	0.9156 ± 0.004	0.9116 ± 0.0057	0.9182 ± 0.0035	0.6721 ± 0.0079
	AA	0.9212 ± 0.0054	0.9085 ± 0.0066	0.9055 ± 0.0107	0.9105 ± 0.0077	0.6162 ± 0.0088
	Kappa	0.9012 ± 0.0055	0.8871 ± 0.0055	0.8815 ± 0.0078	0.8906 ± 0.0048	0.5527 ± 0.0097
Salinas image	OA	0.9329 ± 0.0078	0.911 ± 0.0062	0.9125 ± 0.0072	0.9095 ± 0.0108	0.6658 ± 0.0101
	AA	0.9523 ± 0.0051	0.9268 ± 0.0054	0.9294 ± 0.0089	0.9247 ± 0.0102	0.7088 ± 0.0098
	Kappa	0.9253 ± 0.0086	0.901 ± 0.0069	0.9027 ± 0.008	0.8994 ± 0.012	0.625 ± 0.0106

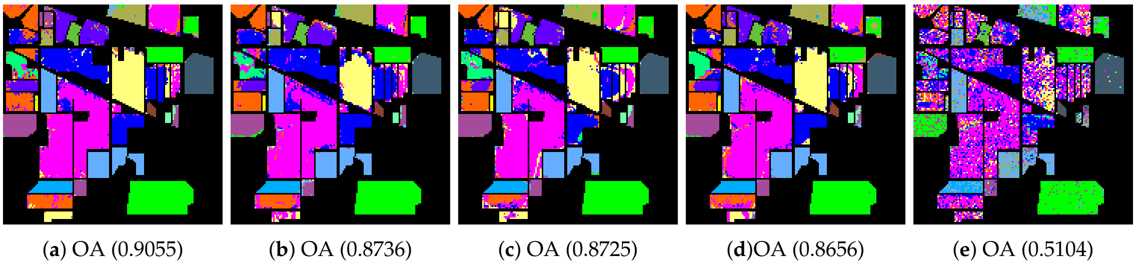


Figure 10. Noisy Indian Pines image classification results: (**a**) RPCA$_{2,1}$; (**b**) RPCA; (**c**) PCA; (**d**) IFRF; and (**e**) Origin. OA, overall accuracy.

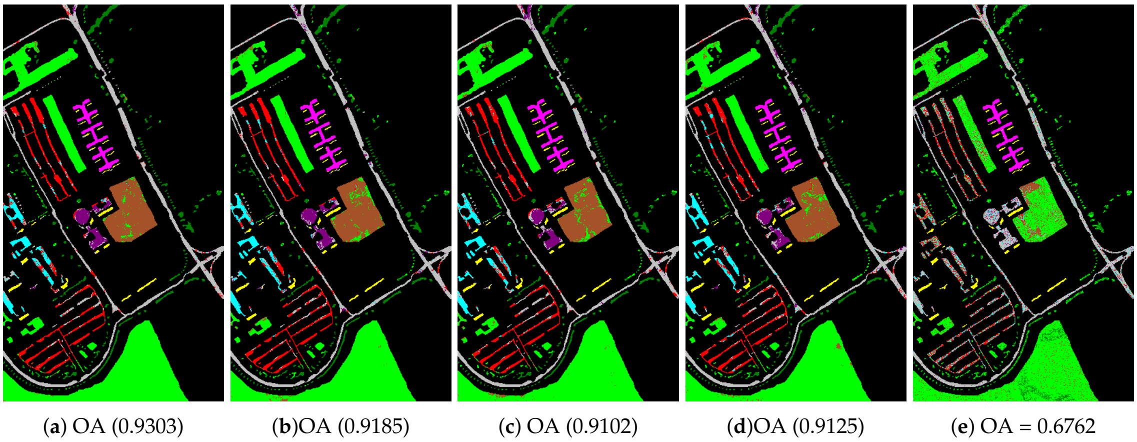


Figure 11. Noisy Pavia University scene classification results: (**a**) RPCA$_{2,1}$; (**b**) RPCA; (**c**) PCA; (**d**) IFRF; and (**e**) Origin. OA, overall accuracy.

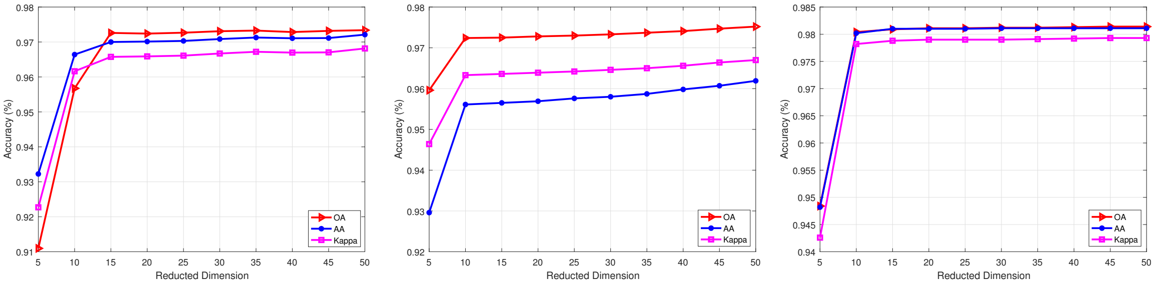
4.4. Parameters of SURPCA$_{2,1}$ Model

We evaluated the SURPCA$_{2,1}$ graph's parameters superpixel number m and the reduced dimensions of SDA. We conducted ten independent runs per algorithm and calculated the mean values and standard deviations of the results. Table 6 and Figure 12 show the performance of different superpixel number m and the reduced dimensions, respectively. For these three hyperspectral images, the labeled ratio was about 4%, 3%, and 0.4%, respectively. The superpixel segmentation numbers are shown in Table 6 for the three hyperspectral images.

Table 6. SURPCA$_{2,1}$ graph's classification accuracy with different segmentation numbers.

Images	No. m of Superpixels	OA	AA	Kappa	Time (s)
Indian Pines image	104	0.9633 ± 0.0072	0.9507 ± 0.027	0.9581 ± 0.0091	12.43
	226	0.9703 ± 0.006	0.9687 ± 0.032	0.9661 ± 0.0062	17.96
	329	0.9721 ± 0.0053	0.9657 ± 0.0254	0.9682 ± 0.0069	21.80
University of Pavia image	195	0.9697 ± 0.0035	0.9473 ± 0.0061	0.9598 ± 0.0047	75.15
	386	0.9711 ± 0.0043	0.9564 ± 0.0084	0.9616 ± 0.0058	78.33
	594	0.9753 ± 0.0026	0.9614 ± 0.0046	0.9672 ± 0.0035	81.20
	813	0.975 ± 0.0018	0.9596 ± 0.0053	0.9668 ± 0.0024	85.97
Salinas image	187	0.9666 ± 0.0059	0.9743 ± 0.0064	0.9629 ± 0.0065	87.71
	389	0.978 ± 0.0042	0.9813 ± 0.003	0.9755 ± 0.0047	92.96
	593	0.9823 ± 0.0037	0.9826 ± 0.003	0.9804 ± 0.0041	98.91

In general, the classification results are not greatly affected by the number of superpixel segmentations, indicating that our graph is robust to the number of superpixel segmentations. It is observed that the increase in superpixel numbers simultaneously accelerated the running time considerably. However, the impact is relatively small when the number of samples is large. Therefore, we could use relatively small superpixel numbers in actual situations for the purpose of efficiency with minimal overall loss of classification accuracy.



(**a**) Indian Pines image (**b**) Pavia University scene (**c**) Salinas image

Figure 12. Classification accuracy of HSIs with different reduced dimensions.

We can observe that the reduction dimension is robust in Figure 12. It will reach steady high accuracy at a relatively low dimension such as 10 or 15.

5. Discussion

The hyperspectral images' classification plays a pivotal role in HSIs' application. In the present work, we propose a novel graph construction model for HSIs' semi-supervised learning, SLIC superpixel-based $l_{2,1}$-norm robust principal component analysis. Our goal is to improve HSIs' classification results through useful graph model. The experimental results show that SURPCA$_{2,1}$ is a competitive graph model for hyperspectral images' classification.

Tables 2, 3, and A1 show that SURPCA$_{2,1}$ is an effective graph model that achieves the highest classification accuracy. Even with simple classifiers and few labeled samples, the performance is excellent. In most cases, the graph model SURPCA$_{2,1}$ obtains the highest classification results, which indicates that the SURPCA$_{2,1}$ graph is a good graph construction model and can notably enhance the hyperspectral images' classification performance. In some cases, other graph models (e.g., RPCA and PCA) may perform well in some categories, but they are not as robust as our graph in all categories. This is because the structural information cannot be well represented from the definition of l_1-norm, while the $l_{2,1}$-norm RPCA$_{2,1}$ produces sparsity based on columns (or rows). The SLIC superpixel segmentation method considers the spatial correlation of pixels in hyperspectral images. For an HSI, the correlation of the adjacent pixels is usually very high since their spectral features are approximately in the same subspace [41]. By extracting a low-rank matrix in each homogeneous region, we can reveal the low-dimensional structure of pixels and enhance the classification results. Consequently, the SURPCA$_{2,1}$ graph model is a superior hyperspectral image's feature graph which significantly improves the performance of HSIs' classification.

We analyzed the robustness to variations in the labeled samples ratio and noisy situations. Figure 8 shows that the SURPCA$_{2,1}$ graph model provides the best results, which are robust to the varying label percentage. Figure 8 also shows that the proposed graph achieves high classification accuracy even at meager label rates, while other graphs may not be as robust as the SURPCA$_{2,1}$ graph, particularly when the label rate is low. Therefore, since labeling data is a costly and difficult task, our SURPCA$_{2,1}$ model is more suitable and robust for real-world applications. From these results of different graph models, we can see that our graph is robust to the sample ratio of the labeled set as well as the noisy environments. Upon adding noise, the performance of the SURPCA$_{2,1}$ graph does not see a large decrease. However, for the comparative graphs, the overall accuracy drops much more than that of SURPCA$_{2,1}$. In addition to the India Pines image, we could see the same results from the other two hyperspectral images. This benefits from the robustness of the $l_{2,1}$-norm robust principal component analysis to sparse noise, whereas RPCA and PCA may be sensitive to sparse noise. Therefore, the

SURPCA$_{2,1}$ graph is robust to noisy environments and the labeled sample ratio, which indicates that it is highly competitive in practical situations.

We evaluated the performance of parameters with different superpixel segmentations number and reduction dimensions for the SDA method. In Table 6, we can see that the classification results are not greatly affected by the number of superpixel segmentations, indicating that our graph is robust to the number of superpixel segmentations. Therefore, we could use a relatively small number of superpixels in actual situations for the purpose of efficiency, with barely any classification accuracy loss. Moreover, in Figure 12, we can see that the classification results are robust after the dimension reached a certain numerical value. It reaches a high accuracy at a relatively low dimension (e.g., 10 or 15). Overall, our proposed SURPCA$_{2,1}$ graph is very much robust and shows excellent performance in the classification of HSIs.

6. Conclusions

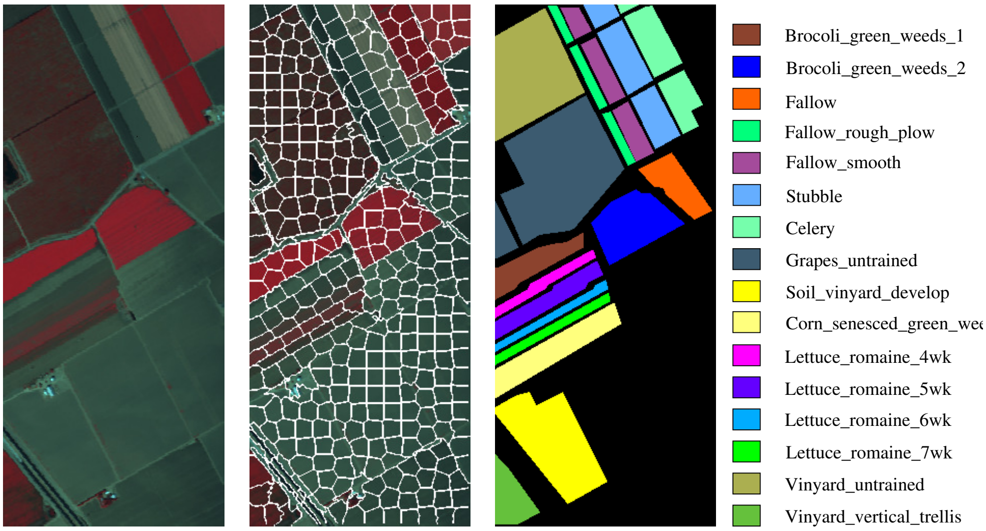
In this paper, we present a novel graph for HSI feature extraction, referred to as SLIC superpixel-based $l_{2,1}$-norm based robust principal component analysis. The SLIC superpixel segmentation method considers the spatial correlation of pixels in HSIs. The $l_{2,1}$-norm robust principal component analysis is used to extract the low-rank structure of all pixels in each homogeneous area, which captures the spectral correlation of the hyperspectral images. Therefore, the classification performance of the SURPCA$_{2,1}$ graph is much better than the compared methods RPCA, PCA, etc. Experiments on real-world HSIs indicated that the SURPCA$_{2,1}$ graph is a robust and efficient graph construction model.

Author Contributions: B.Z. conceived of and designed the experiments and wrote the paper. K.X. gave some comments and ideas for the work. T.L. helped revise the paper and improve the writing. Z.H. and Y.L. performed some experiments and analyzed the data. J.H. and W.D. contributed to the review and revision.

Funding: This work was supported by Key Supporting Project of the Joint Fund of the National Natural Science Foundation of China (No. U1813222), Tianjin Natural Science Foundation (No. 18JCYBJC16500), Hebei Province Natural Science Foundation (No. E2016202341), and Tianjin Education Commission Research Project (No. 2016CJ20).

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

Appendix A

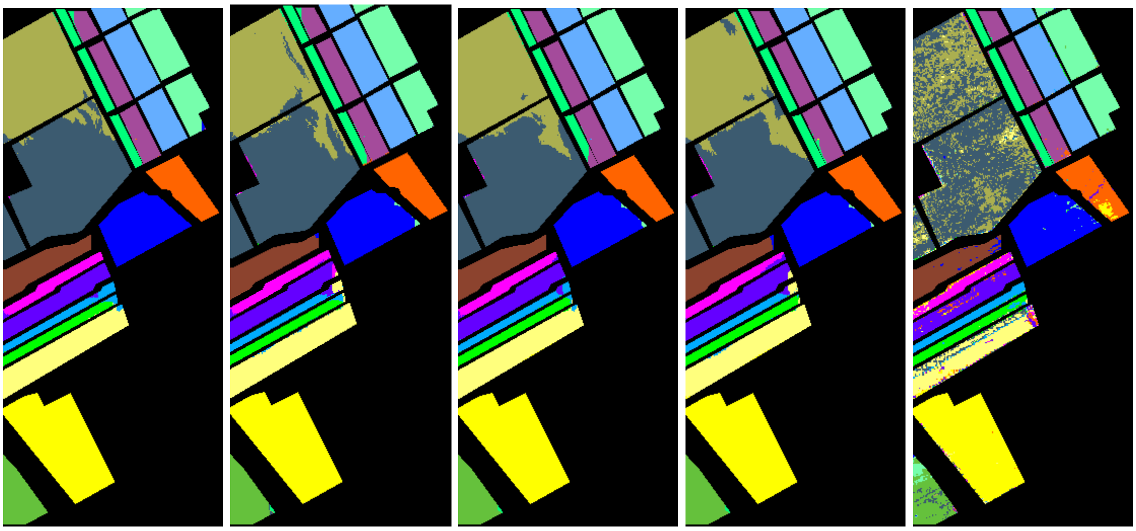


(a) (b) (c) (d)

Figure A1. Salinas dataset: (**a**) false-color image; (**b**) segmentation map; and (**c**,**d**) ground truth image and reference data.

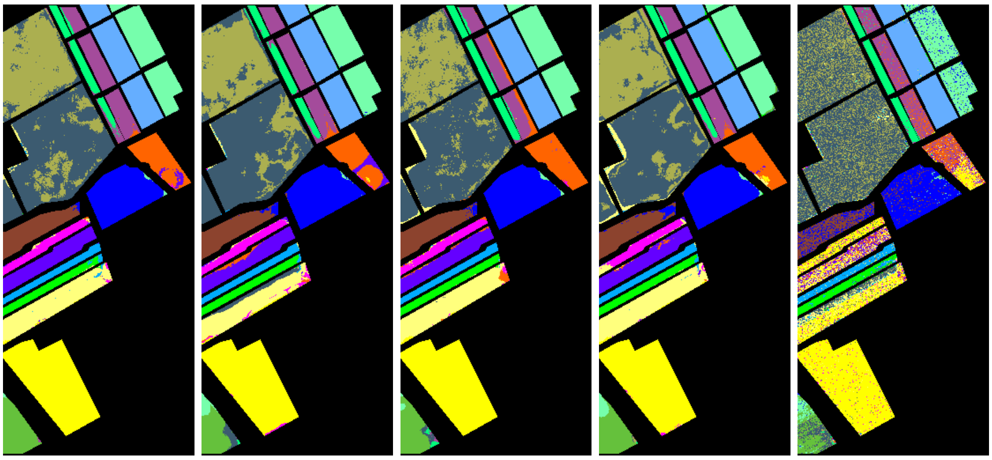
Table A1. Salinas image classification results.

Method	$RPCA_{2,1}$	RPCA	PCA	IFRF	Origion	$RPCA_{2,1}$	RPCA	PCA	IFRF	Origin
Classifier			**NN**					**SVM**		
C1	**1**	0.9999	**1**	0.932	0.9726	**1**	**1**	**1**	**1**	0.9982
C2	**0.9994**	0.9929	0.9952	0.9651	0.9796	0.9926	0.9959	0.9969	0.9789	0.9913
C3	0.9969	0.997	0.9949	0.9501	0.8262	**0.9990**	0.9984	0.9986	0.9982	0.9652
C4	0.9633	0.916	0.9081	0.8496	0.9703	**0.9745**	0.9718	0.9715	0.9542	0.9685
C5	0.9903	0.966	0.983	0.9592	0.9581	**0.9997**	0.9991	0.9986	0.9990	0.9001
C6	**1**	**1**	**1**	0.9952	0.9958	0.9997	0.9997	0.9998	0.9916	0.9997
C7	0.9971	0.9881	0.9901	0.9692	0.9202	**0.9991**	0.9975	0.9986	0.9900	0.9989
C8	0.9882	0.9666	0.9714	0.8995	0.6584	**0.9958**	0.9925	0.9927	0.9881	0.7304
C9	**0.9999**	0.9947	0.9976	0.9932	0.9408	0.9998	0.9995	0.9995	0.9979	0.9592
C10	**0.9969**	0.9905	0.9946	0.9681	0.8393	0.9957	0.9888	0.9931	0.9886	0.9283
C11	0.9961	0.982	0.9926	0.9548	0.851	**1**	0.9994	0.9989	0.9866	0.951
C12	0.9563	0.9481	0.9631	0.9259	0.8886	**0.9907**	0.9786	0.9819	0.9903	0.8949
C13	0.9394	0.8897	0.9354	0.7101	0.763	**0.9999**	0.9982	0.993	0.9406	0.8367
C14	**0.9865**	0.9845	0.9725	0.9791	0.9145	0.9759	0.9835	0.9688	0.9837	0.864
C15	0.9195	0.8994	0.8874	0.7298	0.5459	0.9980	0.9966	**0.9983**	0.9771	0.6202
C16	**1**	0.9998	0.9994	0.9944	0.9398	0.9996	0.9993	0.9999	0.9911	0.9955
OA	0.9809	0.9668	0.9692	0.9653	0.8185	**0.9964**	0.9947	0.9951	0.9867	0.866
AA	0.9831	0.9697	0.9741	0.9705	0.8727	**0.995**	0.9937	0.9931	0.9847	0.9126
Kappa	0.9787	0.963	0.9657	0.9614	0.7976	**0.996**	0.9941	0.9945	0.9852	0.8506



(**a**) OA (0.9833) (**b**) OA (0.9626) (**c**) OA (0.9683) (**d**)OA (0.9611) (**e**) OA (0.8288)

Figure A2. Salinas image Classification results: (**a**) $RPCA_{2,1}$; (**b**) RPCA; (**c**) PCA; (**d**) IFRF; and (**e**) Origin. OA, overall accuracy.



(**a**) OA (0.9303) (**b**) OA (0.8961) (**c**) OA (0.9129) (**d**)OA (0.9108) (**e**) OA (0.6768)

Figure A3. Noisy Salinas image classification results: (**a**) RPCA$_{2,1}$; (**b**) RPCA; (**c**) PCA; (**d**) IFRF; and (**e**) Origin. OA, overall accuracy.

References

1. Xu, L.; Zhang, H.; Zhao, M.; Chu, D.; Li, Y. Integrating spectral and spatial features for hyperspectral image classification using low-rank representation. In Proceedings of the 2017 IEEE International Conference on Industrial Technology (ICIT), Toronto, ON, Canada, 22–25 March 2017; pp. 1024–1029.
2. Ustin, S.L.; Roberts, D.A.; Gamon, J.A.; Asner, G.P.; Green, R.O. Using imaging spectroscopy to study ecosystem processes and properties. *AIBS Bull.* **2004**, *54*, 523–534. [CrossRef]
3. Haboudane, D.; Miller, J.R.; Pattey, E.; Zarco-Tejada, P.J.; Strachan, I.B. Hyperspectral vegetation indices and novel algorithms for predicting green LAI of crop canopies: Modeling and validation in the context of precision agriculture. *Remote Sens. Environ.* **2004**, *90*, 337–352. [CrossRef]
4. Dorigo, W.A.; Zurita-Milla, R.; de Wit, A.J.; Brazile, J.; Singh, R.; Schaepman, M.E. A review on reflective remote sensing and data assimilation techniques for enhanced agroecosystem modeling. *Int. J. Appl. Earth Obs. Geoinf.* **2007**, *9*, 165–193. [CrossRef]
5. Cocks, T.; Jenssen, R.; Stewart, A.; Wilson, I.; Shields, T. The HyMapTM airborne hyperspectral sensor: The system, calibration and performance. In Proceedings of the 1st EARSeL workshop on Imaging Spectroscopy (EARSeL), Paris, France, 6–8 October 1998; pp. 37–42.
6. Dalponte, M.; Bruzzone, L.; Vescovo, L.; Gianelle, D. The role of spectral resolution and classifier complexity in the analysis of hyperspectral images of forest areas. *Remote Sens. Environ.* **2009**, *113*, 2345–2355. [CrossRef]
7. De Morsier, F.; Borgeaud, M.; Gass, V.; Thiran, J.P.; Tuia, D. Kernel low-rank and sparse graph for unsupervised and semi-supervised classification of hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3410–3420. [CrossRef]
8. Hughes, G. On the mean accuracy of statistical pattern recognizers. *IEEE Trans. Inf. Theory* **1968**, *14*, 55–63. [CrossRef]
9. Heinz, D.C. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 529–545. [CrossRef]
10. Kruse, F.A.; Boardman, J.W.; Huntington, J.F. Comparison of airborne hyperspectral data and EO-1 Hyperion for mineral mapping. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1388–1400. [CrossRef]
11. Gu, Y.; Wang, Q.; Wang, H.; You, D.; Zhang, Y. Multiple kernel learning via low-rank nonnegative matrix factorization for classification of hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2739–2751. [CrossRef]
12. Kang, X.; Li, S.; Benediktsson, J.A. Feature extraction of hyperspectral images with image fusion and recursive filtering. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 3742–3752. [CrossRef]
13. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*, 2nd ed.; Wiley: New York, NY, USA, 2001; Volume 55.
14. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.

15. Sacha, D.; Zhang, L.; Sedlmair, M.; Lee, J.A.; Peltonen, J.; Weiskopf, D.; North, S.C.; Keim, D.A. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 241–250. [CrossRef] [PubMed]
16. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
17. Shuiwang, J.; Jieping, Y. Generalized linear discriminant analysis: a unified framework and efficient model selection. *IEEE Trans. Neural Netw.* **2008**, *19*, 1768. [CrossRef] [PubMed]
18. Stone, J.V. Independent component analysis: An introduction. *Trends Cogn. Sci.* **2002**, *6*, 59–64. [CrossRef]
19. Hyvärinen, A.; Hoyer, P.O.; Inki, M. Topographic independent component analysis. *Neural Comput.* **2001**, *13*, 1527–1558. [CrossRef]
20. Draper, B.A.; Baek, K.; Bartlett, M.S.; Beveridge, J.R. Recognizing faces with PCA and ICA. *Comput. Vis. Image Underst.* **2003**, *91*, 115–137. [CrossRef]
21. Belkin, M.; Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **2003**, *15*, 1373–1396. [CrossRef]
22. Tenenbaum, J.B.; De Silva, V.; Langford, J.C. A global geometric framework for nonlinear dimensionality reduction. *Science* **2000**, *290*, 2319–2323. [CrossRef]
23. Roweis, S.T.; Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **2000**, *290*, 2323–2326. [CrossRef]
24. Zhang, Z.; Pasolli, E.; Yang, L.; Crawford, M.M. Multi-metric Active Learning for Classification of Remote Sensing Data. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1007–1011. [CrossRef]
25. Zhang, Z.; Crawford, M.M. A Batch-Mode Regularized Multimetric Active Learning Framework for Classification of Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 1–16. [CrossRef]
26. Ul Haq, Q.S.; Tao, L.; Sun, F.; Yang, S. A fast and robust sparse approach for hyperspectral data classification using a few labeled samples. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 2287–2302. [CrossRef]
27. Kuo, B.C.; Chang, K.Y. Feature extractions for small sample size classification problem. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 756–764. [CrossRef]
28. Zhou, D.; Bousquet, O.; Lal, T.N.; Weston, J.; Schölkopf, B. Learning with local and global consistency. In Proceedings of the Advances in Neural Information Processing Systems, Whistler, BC, Canada, 9–11 December 2003; pp. 321–328.
29. Guo, Z.; Zhang, Z.; Xing, E.P.; Faloutsos, C. Semi-supervised learning based on semiparametric regularization. In Proceedings of the 2008 SIAM International Conference on Data Mining (SIAM), Atlanta, GA, USA, 24–26 April 2008; pp. 132–142.
30. Subramanya, A.; Talukdar, P.P. Graph-based semi-supervised learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2014**, *8*, 1–125. [CrossRef]
31. Zheng, M.; Bu, J.; Chen, C.; Wang, C.; Zhang, L.; Qiu, G.; Cai, D. Graph regularized sparse coding for image representation. *IEEE Trans. Image Process.* **2011**, *20*, 1327–1336. [CrossRef] [PubMed]
32. Tian, X.; Gasso, G.; Canu, S. A multiple kernel framework for inductive semi-supervised SVM learning. *Neurocomputing* **2012**, *90*, 46–58. [CrossRef]
33. Cai, D.; He, X.; Han, J.; Huang, T.S. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1548–1560. [PubMed]
34. Gao, S.; Tsang, I.W.H.; Chia, L.T.; Zhao, P. Local features are not lonely—Laplacian sparse coding for image classification. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
35. Liu, G.; Lin, Z.; Yu, Y. Robust subspace segmentation by low-rank representation. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 663–670.
36. Candès, E.J.; Li, X.; Ma, Y.; Wright, J. Robust principal component analysis? *J. ACM* **2011**, *58*, 11. [CrossRef]
37. Wright, J.; Ganesh, A.; Rao, S.; Peng, Y.; Ma, Y. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 7–10 December 2009; pp. 2080–2088.
38. Binge, C.; Zongqi, F.; Xiaoyun, X.; Yong, Z.; Liwei, Z. A Novel Feature Extraction Method for Hyperspectral Image Classification. In Proceedings of the 2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Changsha, China, 17–18 December 2016; pp. 51–54.

39. Nie, Y.; Chen, L.; Zhu, H.; Du, S.; Yue, T.; Cao, X. Graph-regularized tensor robust principal component analysis for hyperspectral image denoising. *Appl. Opt.* **2017**, *56*, 6094–6102. [CrossRef]
40. Chen, G.; Bui, T.D.; Quach, K.G.; Qian, S.E. Denoising hyperspectral imagery using principal component analysis and block-matching 4D filtering. *Can. J. Remote Sens.* **2014**, *40*, 60–66. [CrossRef]
41. Zhang, H.; He, W.; Zhang, L.; Shen, H.; Yuan, Q. Hyperspectral image restoration using low-rank matrix recovery. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 4729–4743. [CrossRef]
42. Xu, Y.; Wu, Z.; Wei, Z. Spectral–spatial classification of hyperspectral image based on low-rank decomposition. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2370–2380. [CrossRef]
43. Fan, F.; Ma, Y.; Li, C.; Mei, X.; Huang, J.; Ma, J. Hyperspectral image denoising with superpixel segmentation and low-rank representation. *Inf. Sci.* **2017**, *397*, 48–68. [CrossRef]
44. Zhang, S.; Li, S.; Fu, W.; Fang, L. Multiscale superpixel-based sparse representation for hyperspectral image classification. *Remote Sens.* **2017**, *9*, 139. [CrossRef]
45. Sun, L.; Jeon, B.; Soomro, B.N.; Zheng, Y.; Wu, Z.; Xiao, L. Fast Superpixel Based Subspace Low Rank Learning Method for Hyperspectral Denoising. *IEEE Access* **2018**, *6*, 12031–12043. [CrossRef]
46. Mei, X.; Ma, Y.; Li, C.; Fan, F.; Huang, J.; Ma, J. Robust GBM hyperspectral image unmixing with superpixel segmentation based low rank and sparse representation. *Neurocomputing* **2018**, *275*, 2783–2797. [CrossRef]
47. Tong, F.; Tong, H.; Jiang, J.; Zhang, Y. Multiscale union regions adaptive sparse representation for hyperspectral image classification. *Remote Sens.* **2017**, *9*, 872. [CrossRef]
48. Zu, B.; Xia, K.; Du, W.; Li, Y.; Ali, A.; Chakraborty, S. Classification of Hyperspectral Images with Robust Regularized Block Low-Rank Discriminant Analysis. *Remote Sens.* **2018**, *10*, 817. [CrossRef]
49. Maier, M.; Luxburg, U.V.; Hein, M. Influence of graph construction on graph-based clustering measures. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 7–10 December 2009; pp. 1025–1032.
50. Zhu, X.; Goldberg, A.B.; Brachman, R.; Dietterich, T. Introduction to Semi-Supervised Learning. *Semi-Superv. Learn.* **2009**, *3*, 130. [CrossRef]
51. Zhu, X. *Semi-Supervised Learning Literature Survey*; Department of Computer Science, University of Wisconsin-Madison: Madison, WI, USA, 2006; Volume 2, p. 4.
52. Stickel, M.E. *A Nonclausal Connection-Graph Resolution Theorem-Proving Program*; Technical Report; Sri International Menlo Park Ca Artificial Intelligence Center: Menlo Park, CA, USA, 1982.
53. Cortes, C.; Mohri, M. On transductive regression. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; Volume 19, p. 305.
54. Ren, X.; Malik, J. Learning a Classification Model for Segmentation. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; pp.10–17.
55. Moore, A.P.; Prince, S.J.D.; Warrell, J.; Mohammed, U.; Jones, G. Superpixel lattices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
56. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef]
57. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. *Slic Superpixels*; EPFL Technical Report; EPFL: Lausanne, Switzerland, 2010.
58. Zitnick, C.L.; Kang, S.B. Stereo for Image-Based Rendering using Image Over-Segmentation. *Int. J. Comput. Vis.* **2007**, *75*, 49–65. [CrossRef]
59. Ren, C.Y.; Reid, I. *gSLIC: A Real-Time Implementation of SLIC Superpixel Segmentation*; University of Oxford: Oxford, UK, 2011.
60. Gastal, E.S.; Oliveira, M.M. Domain transform for edge-aware image and video processing. *ACM Trans. Graph.* **2011**, *30*, 69. [CrossRef]
61. Ren, C.-X.; Dai, D.-Q.; Yan, H. Robust classification using L 2,1-norm based regression model. *Pattern Recognit.* **2012**, *45*, 2708–2718. [CrossRef]
62. Nie, F.; Huang, H.; Cai, X.; Ding, C.H. Efficient and robust feature selection via joint L 2, 1-norms minimization. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 6–9 December 2010; pp. 1813–1821.
63. Lin, Z.; Chen, M.; Ma, Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv* **2010**, arXiv:1009.5055.

64. Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 171–184. [CrossRef] [PubMed]
65. Cai, D.; He, X.; Han, J. Semi-supervised discriminant analysis. In Proceedings of the IEEE 11th International Confece on IEEE Computer Vision (ICCV), Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–7.
66. Yan, S.; Wang, H. Semi-supervised Learning by Sparse Representation. In Proceedings of the 2009 SIAM International Confece on Data Mining, Sparks, NV, USA, 30 April–2 May 2009; pp. 792–801.
67. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Ma, Y. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 210–227. [CrossRef]
68. Zu, B.; Xia, K.; Pan, Y.; Niu, W. A Novel Graph Constructor for Semisupervised Discriminant Analysis: Combined Low-Rank and-Nearest Neighbor Graph. *Comput. Intell. Neurosci.* **2017**, *2017*, 9290230. [CrossRef] [PubMed]
69. Peterson, L. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]
70. Baumgardner, M.F.; Biehl, L.L.; Landgrebe, D.A. 220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3, 2015. Available online: https://purr.purdue.edu/publications/1947/1 (accessed on 18 January 2018).
71. Congalton, R.G. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sens. Environ.* **1991**, *37*, 35–46. [CrossRef]
72. Richards, J.A.; Richards, J. *Remote Sensing Digital Image Analysis*; Springer: Berlin, Germany, 1999; Volume 3.
73. Thompson, W.D.; Walter, S.D. A reappraisal of the kappa coefficient. *J. Clin. Epidemiol.* **1988**, *41*, 949–958. [CrossRef]
74. Gwet, K. Kappa statistic is not satisfactory for assessing the extent of agreement between raters. *Stat. Methods Inter-Rater Reliab. Assess.* **2002**, *1*, 1–6.