remote sensing

MDPI

Article

Label Noise Cleansing with Sparse Graph for Hyperspectral Image Classification

Qingming Leng [1], Haiou Yang [2,†] and Junjun Jiang [3,*]

[1] School of Information Science and Technology, Jiujiang University, Jiujiang 332005, China; qingming_leng@jju.edu.cn
[2] School of Tourism and Territorial Resources, Jiujiang University, Jiujiang 332005, China; yanghaiou2007@jju.edu.cn
[3] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
* Correspondence: jiangjunjun@hit.edu.cn
† Haiou Yang is the co-first author.

Received: 20 April 2019; Accepted: 6 May 2019; Published: 10 May 2019

Abstract: In a real hyperspectral image classification task, label noise inevitably exists in training samples. To deal with label noise, current methods assume that noise obeys the Gaussian distribution, which is not the real case in practice, because in most cases, we are more likely to misclassify training samples at the boundaries between different classes. In this paper, we propose a spectral–spatial sparse graph-based adaptive label propagation (SALP) algorithm to address a more practical case, where the label information is contaminated by random noise and boundary noise. Specifically, the SALP mainly includes two steps: First, a spectral–spatial sparse graph is constructed to depict the contextual correlations between pixels within the same superpixel homogeneous region, which are generated by superpixel image segmentation, and then a transfer matrix is produced to describe the transition probability between pixels. Second, after randomly splitting training pixels into "clean" and "polluted," we iteratively propagate the label information from "clean" to "polluted" based on the transfer matrix, and the relabeling strategy for each pixel is adaptively adjusted along with its spatial position in the corresponding homogeneous region. Experimental results on two standard hyperspectral image datasets show that the proposed SALP over four major classifiers can significantly decrease the influence of noisy labels, and our method achieves better performance compared with the baselines.

Keywords: hyperspectral image classification; label noise cleansing; spectral–spatial sparse graph; adaptive label propagation

1. Introduction

Hyperspectral image (HSI), which includes numerous contiguous bands of spectral information of the observation, have been easily and cheaply acquired by various remote sensing sensors in recent years. HSI classification [1,2], which aims at the high-precision and pixel-level categorization of land use/land over types, has been significantly promoted for many practical applications [3,4], such as ecological monitoring, precision agriculture, and mineral exploration. As a result of the enduring and increasing demand of classification accuracy, the supervised paradigm of HSI classification benefiting from powerful knowledge of manual labels, is always one of the main topics of interest in the remote sensing community [5,6]. In the past decades, a number of effective HSI classification methods have been proposed based on multifarious supervised models, e.g., Bayesian model [7], neural networks [8], support vector machine (SVM) [9], sparse representation learning [10], collaborative representation learning [11,12], extreme learning machine (ELM) [13,14], and generative adversarial network [15,16]. They can generally be well-trained and provide satisfying results.

Commonly speaking, the availability of supervised models highly depends on the validity of manual labels. However, some unpredictable mislabeled samples (namely, label noise) naturally exist in real-world hyperspectral datasets when human labelers are involved, and the noise-polluted pixels would evidently decrease the performance of supervised classifiers [17]. To deal with the unavoidable label noise problem, researchers have proposed many useful methods [18–24] that mainly fall into three categories [25]: label noise-robust classification, label noise-tolerant classification, and label noise cleansing. Label noise-robust classification focuses on studying discriminative classifiers that are insensitive to label noise, label noise-tolerant classification is concerned with modeling the label noise simultaneously learning with the classifiers. Both approaches can be regarded as a style called "learning with the label noise," which is always designed for specific classification models with sophisticated learning algorithms and thus has less generalization and scalability. By contrast, label noise cleansing is a preprocessing step, and it works on learning a universal noise filter to improve the label quality of training data for any classifiers. Therefore, this study focuses on the label noise cleansing style that is more general for HSI classification with noisy labels.

HSI classification in the presence of label noise is an exigent and challenging issue. However, few label cleansing methods are found when even considering the literature of generic image classification [26–28]. For example, Pelletier et al. [17] evaluated the influence of land cover label noise on classification performances. They demonstrated that the accuracies of two classic models, namely, SVM and random forest (RF), were sensitive to mislabeled training data. Unfortunately, they did not provide a solution to handle the label noise. Jiang et al. [29] proposed a random label propagation algorithm (RLPA), which can be the only relevant effort. They first constructed a probability transfer matrix to depict the affinity of hyperspectral pixels and then relabeled the training pixels by exploiting the label information from neighbors via random label propagation. The performance of several typical classifiers trained on relabeled samples had significant improvements over that of classifiers trained on initial samples with label noise. This approach assumed that noise obeys the Gaussian distribution, and noisy labels were generated by employing only random noise. However, the random noise cannot simulate the actual situation, because in most cases, the labelers probably misclassify hyperspectral pixels at the boundaries between different classes. Two kinds of label noise are more likely to take place during the practical labeling procedure [25]: random label noise and boundary label noise. On the one hand, hyperspectral pixels are so subtle that labelers may mislabel the samples unintentionally, which can increase the probability and randomness of noisy labels. On the other hand, diverse land use/land over types are often spatially staggered, and the labels of boundary pixels between adjacent surface types are harder to determine than those of other pixels. The neighborhood affinity is probably not robust when many noisy labels gather around the borders. Thus, a universal solution is needed to cope with the two types of label noise.

In this study, we propose a spectral–spatial sparse graph-based adaptive label propagation (SALP) algorithm to clean random label noise and boundary label noise for HSI classification. The framework shown in Figure 1 mainly includes two steps: First, we construct a spectral–spatial sparse graph to depict the affinity between pixels. Both spectral information and spatial constraint are important for calculating the similarity of different pixels [30–33], and sparse graph that explicitly considers the influence of data noise is insensitive to the gathered noise [34–39]. Therefore, a spectral–spatial sparse graph is adopted to describe the contextual correlations between pixels within same superpixel homogeneous region that are obtained by entropy rate superpixel segmentation (ESR) [40], and then a transfer matrix is produced to describe the transition probability between pixels. Second, we employ an adaptive label propagation algorithm to relabel the training samples. The spectral–spatial superpixel homogeneous region contains some useful prior knowledge that pixels with similar land use/land over types are often adjacent, and the central pixels in a superpixel homogeneous region are usually farther from the true border between different surface types more than other pixels (refer to the mapping results in Figure 1). We can adaptively relabel the pixels based on their spatial position in the corresponding homogeneous region. Hence, after randomly splitting training pixels into "clean" and “polluted,” an adaptive label propagation process is utilized to iteratively propagate the label information from “clean” to “polluted” based on the transfer matrix. The centered “polluted” pixels are relabeled by exploiting the label information from neighborhood “clean” samples, whereas the labels of other “polluted” pixels are adjusted by sparse “clean” pixels that locate in the same superpixel homogeneous region. Finally, the label propagation is carried out several times, and the final labels for each pixels are calculated based on a majority vote algorithm (MVA) [41]. The proposed SALP is tested on two real HSI databases, namely, Indian Pines and Salinas Scene, and our performance over four typical classifiers can significantly decrease the influence of noisy labels and exceed the baselines.

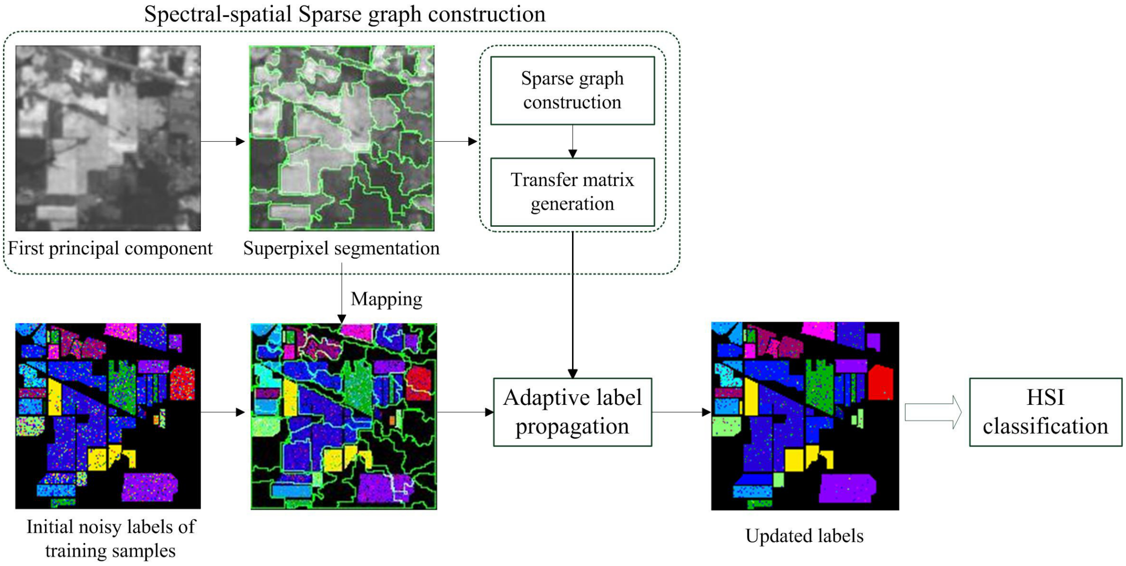

Spectral-spatial Sparse graph construction
First principal component
Superpixel segmentation
Sparse graph construction
Transfer matrix generation
Mapping
Initial noisy labels of training samples
Adaptive label propagation
Updated labels
HSI classification

Figure 1. The framework of the proposed SALP algorithm , which mainly includes spectral–spatial sparse graph construction and adaptive label propagation.

The main contributions of this study can be summarized as the following three aspects:

(1) We carefully analyzed and examined the core issue and the influence of label noise for HSI classification, which is an useful guideline to present a label noise-polluted classification work.

(2) A novel label noise cleansing method, namely, SALP algorithm, is proposed to deal with random label noise and boundary label noise.

(3) Experimental results on two publicly practical datasets show that the proposed SALP over four major classifiers can obviously reduce the impact of noisy labels, and its performance can surpass the baselines in terms of overall accuracy (OA), average accuracy (AA), Kappa coefficient, and visual classification map.

This paper is organized as follows: Section 2 formulates and discusses the problem statement; Section 3 presents the methodology of the proposed SALP; Section 4 draws the experimental results and discussions; and Section 5 concludes this article.

2. Problem Statement

This section first presents the mathematical definitions of HSI classification in the presence of label noise, and then quantitatively discusses the impact of random label noise and boundary label noise.

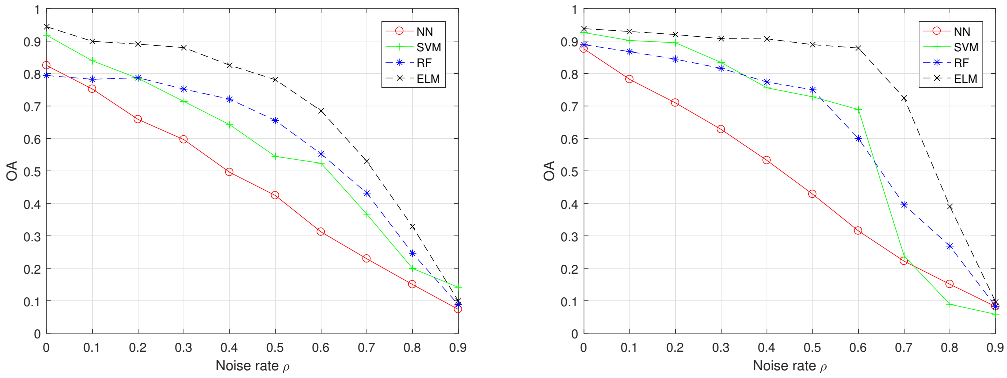
Given a HSI composed of hundreds of contiguous spectral bands, some pixels with labels are regarded as training data, and the other pixels without labels are treated as testing data. When the HSI classification meets label noise, there are a certain level of noisy labels, including random label noise and boundary label noise, exist in the training pixels. To deal with the noisy labels, label noise cleansing is to relabel the initial labels to ensure that the performance of supervised classifiers using relabeled training data is better than that of using initial data.

Specifically, a set of training pixels in a D dimensional spectral feature space are denoted as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}, \mathbf{x}_i \in \mathbb{R}^D$, i is an integer in the interval $[1, N]$. $\gamma = \{1, 2, \cdots, C\}$ depicts a label set, the class labels of $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ are denoted as $\{y_1, y_2, \cdots, y_N\}, y_j \in \gamma$, j is an integer in the interval $[1, C]$, and a matrix $\mathbf{Y} \in \mathbb{R}^{D\times C}$ is employed to denote the ground truth labels, where $\mathbf{Y}_{ij} = 1$ if j is the label of $\mathbf{x}_i$. To model the label noise process, another matrix $\widetilde{\mathbf{Y}} \in \mathbb{R}^{D\times C}$ is adopted to represent the labels with the noise of training pixels, and our goal is to forecast the label of $\mathbf{x}_l \in \mathbf{X}$ with the matrix $\widetilde{\mathbf{Y}}$ in certain noise rate [42] ρ. And ρ_{jk} that is the probability of one sample being marked by incorrect labels can be indicated as:

$$\begin{aligned} \rho_{jk} = P\left(\widetilde{\mathbf{Y}}_{ik} = 1 | \mathbf{Y}_{ij} = 1\right), j, k \in \{1, 2, \cdots, C\}, \\ \forall j \neq k, i \in \{1, 2, \cdots, N\}. \end{aligned} \tag{1}$$

The influence of random label noise on HSI classification was examined in Jiang's study [29], and results showed that random label noise can degenerate the performance of supervised classifiers. To further analyze the integrated impact of random label noise and boundary label noise, we test the noisy label-based algorithm (NLA), in which training samples and their corresponding noisy labels are directly used to train the classifiers. Four typical classifiers are utilized, namely, neighbor nearest (NN), SVM, RF, and ELM. Two standard hyperspectral datasets, i.e., the Indian Pines and Salinas, are adopted, and ground truth samples are randomly splited into training set and testing set. The label noise uses "both" setting that is a fusion of random label noise and boundary label noise. The noise rate is set as $\rho = \{0, 0.1, 0.2, \cdots, 0.9\}$ with an interval of 0.1. The OA is employed to measure the classification performance. More details about the sample setting, the label noise setting, and other experimental setup can be found in Section 4.2.

As shown in Figure 2, three phenomena occur: First, all classifiers have good performance when learning the classifiers without label noise, whereas their performance declines quickly along with the increase of ρ. Second, some classifiers may have good results at low noise rate. For example, ELM that pertains to single-hidden layer feed-forward neural network, can randomly generates the input weights and compute the output weights with a least squares solution. It is more computationally efficient meanwhile usually obtain similar or better generalization performance than traditional neural networks and SVM [13,43]. However, all classifiers are still sensitive to label noise especially when ρ is high. Third, the performance of all classifiers is poor at a high noise rate and converges to a similarly low value. Above all, the random label noise and boundary label noise can seriously and negatively affect the availability of supervised classifiers. Thus, studying a general and effective label noise cleansing method for HSI classification is urgent.



(**a**) Indian Pine dataset.

(**b**) Salinas dataset.

Figure 2. The OA of NLA over four typical classifiers at different noise rate on two standard hyperspectral datasets. The results of NN, SVM, RF and ELM are labeled in red "o", green "+", blue "*", and black "x".

3. Proposed Method

3.1. Overview of the Proposed Method

Label noise cleansing is performed to relabel the initial labeled training samples rather than the unlabeled samples adopted for unsupervised models [44,45]. Thus, the use of original labels is the key to the cleansing methodology. Although the process of pixel-level labeling is difficult and complicated, most labels are still credible because they are always generated by experts. Therefore, a intuitional idea is to relabel a pixel based on the majority of label information from its correlational pixels (e.g., neighbors [29]). The neighborhood affinity may be useful for cleansing the random label noise that is randomly distributed. However, different surface types are often so spatially staggered that the labels of boundary pixels are easily mislabeled, and the neighborhood affinity is probably less discriminative to eliminate boundary label noise.

To solve the problem, this study proposes SALP algorithm, and its core idea can be described as follows: First, the sparse graph that explicitly considers the influence of data noise is insensitive to the gathered label noise, and its sparsity is datum-adaptive instead of manually determining the size of neighborhood [46–48]. Second, spatial information is important for measuring the similarity of different pixels. A superpixel homogeneous region produced by superpixel image segmentation may provide a useful spatial constraint for graph construction [49,50]. Third, the centered pixels in a superpixel homogeneous region are often farther from the true border between different surface types more than other pixels. A good strategy is to adaptively adjust the label propagation for each pixel along with its spatial position in the corresponding homogeneous region. Therefore, we utilize a spectral–spatial sparse graph to depict the affinity between pixels within the same superpixel homogeneous region, and then iteratively relabel the training samples by exploiting the graph through an adaptive label propagation algorithm.

3.2. Spectral-Spatial Sparse Graph Construction

The core issue of graph construction is weight measurement between nodes, and a straightforward way to achieve this is to compute the spectral similarity between pixels. However, constructing a full graph for all pixels is complicated, and the commonly used similarities based on Euclidean distance and spectral angle mapper, etc., are not robust enough to handle the low inter-class and high intra-class spectral difference [51,52]. We notice that spatial information is helpful for differentiating the spectral features [43]. Thus, the initial HSI is segmented into many non-overlapping superpixel homogeneous regions, and then a sparse graph is generated based on the sparse spectral similarity with spatial constraint. The graphical illustration of spectral–spatial sparse graph is shown in Figure 3.

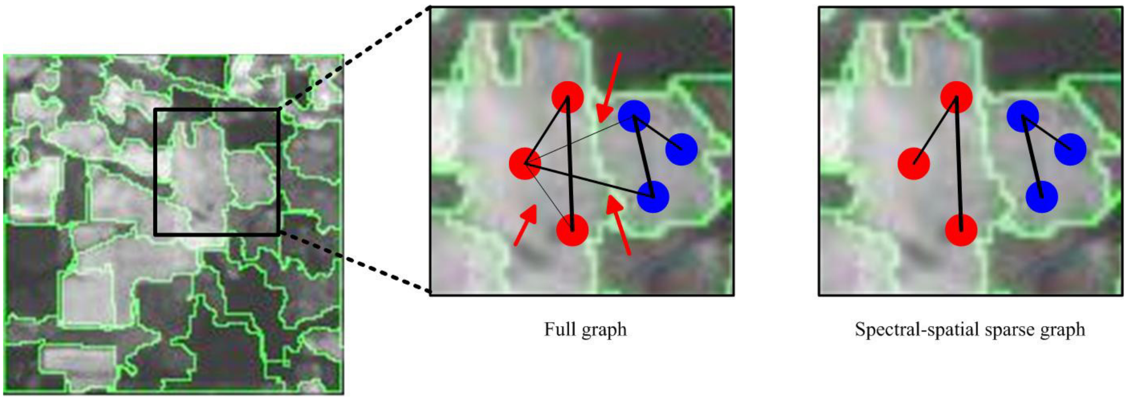


Figure 3. The graphical illustration of spectral–spatial sparse graph. In a full graph, vertexes are densely connected with others. In a spectral–spatial sparse graph, vertexes are sparsely linked with those pixels that are located in the same homogeneous regions. The links between different homogeneous regions and some weak links marked with red arrows are removed from the full graph.

Firstly, following the superpixel segmentation procedure [29], I_f is the first principal component of an HSI that contains hundreds of contiguous spectral bands, and it can be acquired through principal component analysis [53] so as to reduce the computational complexity. Then, a set of superpixel homogeneous regions $S = \{s_1, s_2, \cdots, s_T\}$ is produced via ESR,

$$\begin{gathered} I_f = \bigcup_{k}^{T} s_k, \quad s.t.\ s_k \cap s_g = \emptyset, \\ \forall k, g \in \{1, 2, \cdots, T\},\ k \neq g, \end{gathered} \tag{2}$$

Secondly, a sparse graph is adopted to depict the contextual correlations between hyperspectral pixels. Instead of constructing a graph in the whole image, we employs a spatial constraint, in which the sparse graph is generated for the pixels in the same superpixel homogeneous region. Specifically, the training data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}, \mathbf{x}_i \in \mathbb{R}^D$ are coded sparsely by ℓ_0-norm optimization:

$$\arg\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{\alpha}_i\|_0,\ s.t.\ \mathbf{x}_i = \mathbf{B}_i \boldsymbol{\alpha}_i, \tag{3}$$

where $\|\bullet\|_0$ is the ℓ_0-norm of a vector that counts the number of nonzero elements. $\mathbf{B}_i = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N, \mathbf{I}\} \in \mathbb{R}^{D\times(D+N-1)}$, $\boldsymbol{\alpha}_i \in \mathbb{R}^{D+N-1}$. ALthough sparsity optimization is a nonconvex NP-hard problem, if the solution of $\boldsymbol{\alpha}_i$ is sparse enough [54], Equation (3) can be relaxed by optimizing the following convex function:

$$\arg\min_{\boldsymbol{\alpha}_i} \|\mathbf{x}_i - \mathbf{B}_i \boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1, \tag{4}$$

where $\|\bullet\|_1$ is the ℓ_1-norm, which sums the absolute value of the vector, and can be solved as a linear program. λ is used to weight the importance of minimizing $\boldsymbol{\alpha}_i$.

Set sparse graph $G = \{\mathbf{X}, \mathbf{W}\}$ with the $\mathbf{X}$ as the vertices set and the $\mathbf{W}$ as the weight matrix, and the edge weights $\mathbf{W}_{ij}$ from $\mathbf{x}_i$ to $\mathbf{x}_j$ can be denoted by:

$$\mathbf{W}_{ij} = \begin{cases} {\boldsymbol{\alpha}_i}^j, & \mathbf{x}_i, \mathbf{x}_j \in s_k, \\ 0, & \mathbf{x}_i \in s_k, \mathbf{x}_j \in s_g, \end{cases} \tag{5}$$

where ${\boldsymbol{\alpha}_i}^j$ is the j-th coefficient of $\boldsymbol{\alpha}_i$. Generally, a large $\mathbf{W}_{ij}$ corresponds to the greater effect of $\mathbf{x}_i$ on $\mathbf{x}_j$ in the label propagation procedure. Notably, correlations may also exist between $\mathbf{x}_i$ and other pixels, and the overall affinity between associated pixels should be examined to calculate the transfer matrix $\mathbf{T}_{ij}$, which indicates the probability of label information traveling from $\mathbf{x}_i$ to $\mathbf{x}_j$ as follows:

$$\mathbf{T}_{ij} = P\left(i \to j\right) = \frac{\mathbf{W}_{ij}}{\sum_{k=1}^{N} \mathbf{W}_{ik}}. \tag{6}$$

3.3. Adaptive Label Propagation

Label propagation is performed to deliver the label information from the correctly labeled pixels to the incorrect ones [55]. Although the ground truth is unknown, we can randomly divide the training data several times and then fuse all the results of label propagation based on the MVA. In each time, some training pixels that are randomly set as “clean” would preserve the labels, and the others are treated as “polluted” pixels that are unlabeled. To propagate the labels of “clean” samples, a common strategy that the “polluted” pixels absorb the label information from their neighbors [29] based on the transfer matrix $\mathbf{T}$, is sometimes useful to cope with the random label noise. However, boundary label noise is intensively located around the borders, and neighborhood affinity may bring about wrong labels for boundary pixels. Therefore, we update the label propagation adaptively for each pixel along with its spatial position in the corresponding homogeneous region.

Compared with the centered pixels in a superpixel homogeneous region, the pixels close to the homogeneous region border are more likely to be near the true boundary between different surface types. Thus, the labels of centered "polluted" pixels are altered based on the information from their neighborhood "clean" samples, whereas other "polluted" pixels are relabeled by exploiting the label information from sparse "clean" pixels that locate in the same superpixel homogeneous region. Specifically, a set of training pixels $\mathbf{X}$ is stochastically split into two parts: the "clean" subset $\mathbf{X}_L = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l\}$ with its label matrix $\widetilde{\mathbf{Y}}_L = \widetilde{\mathbf{Y}}(:, 1:l) \in \mathbb{R}^{l \times C}$, l is the number of labeled pixels that is decided by the noise rate, $l = round\,(N \times \rho)$; the "polluted" subset $\mathbf{X}_U = \{\mathbf{x}_{l+1}, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ without labels. Then, the objective of adaptive label propagation is to iteratively forecast the label matrix $\widetilde{\mathbf{Y}}_U$ of $\mathbf{X}_U$ by exploiting the transfer matrix $\mathbf{T}$.

Assuming that the label prediction matrix of all the training pixels is $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N\} \in \mathbb{R}^{N \times C}$, the label of $\mathbf{x}_i$ at time $t+1$ is

$$\mathbf{f}_i^{t+1} = \begin{cases} \theta \sum\limits_{\mathbf{x}_i \in s_k} \sum\limits_{\mathbf{x}_j \in s_k^{i,neigh}} \mathbf{T}_{ji}\mathbf{f}_j^t + (1-\theta)\,\widetilde{\mathbf{Y}}_{LU}^i, & \text{if } \|\mathbf{x}_i - s_k^{mean}\| < \sigma_k, \\ \theta \sum\limits_{\mathbf{x}_i, \mathbf{x}_j \in s_k} \mathbf{T}_{ji}\mathbf{f}_j^t + (1-\theta)\,\widetilde{\mathbf{Y}}_{LU}^i, & others, \end{cases} \tag{7}$$

where $\widetilde{\mathbf{Y}}_{LU}^i$ is the i-th column of $\widetilde{\mathbf{Y}}_{LU} = \left[\widetilde{\mathbf{Y}}_L; \widetilde{\mathbf{Y}}_U\right]$. θ is to balance the influence between the present label and other label information acquired from the referential pixels. We fix θ to 0.9 in all the experiments. $\mathbf{f}_j^t$ is the references label $\mathbf{f}_j$ at time t, the references $s_k^{i,neigh}$ are the neighbors of $\mathbf{x}_i$ when $\mathbf{x}_i$ is close to the mean of the superpixel homogeneous region, s_k^{mean}. Otherwise, $\mathbf{f}_j^t$ can be the labels of all other pixels within s_k. Parameter σ_k is the variance of s_k. Following the optimization [56], Equation (7) can converge to:

$$\begin{aligned} \mathbf{F}^* &= \lim_{t \to \infty} \mathbf{F}^t \\ &= \lim \theta \mathbf{T}\mathbf{F}^{t-1} + (1-\theta)\,\widetilde{\mathbf{Y}}_{LU} \\ &= (1-\theta)\,(\mathbf{I} - \mathbf{T})^{-1}\widetilde{\mathbf{Y}}_{LU}, \end{aligned} \tag{8}$$

and the cleaned label of x_i can be denoted as:

$$y_i^* = \arg\max_j \mathbf{F}_{ji}^*. \tag{9}$$

To reduce the error of random selection, the above process would be repeated for many times, and the final propagated label can be calculated by MVA. The pseudocodes of the proposed SALP is overall described as Algorithm 1.

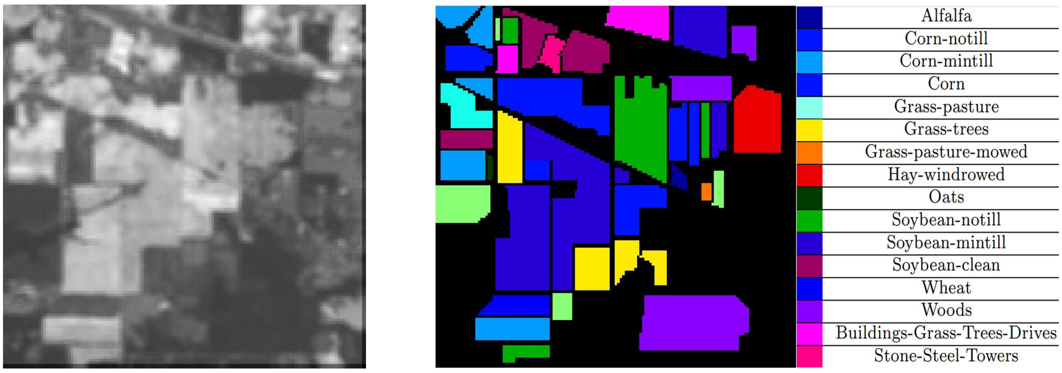
Algorithm 1 The proposed SALP algorithm.

```
Input: A hyperspectral image $I_f$; The training pixels $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ with their labels $\{y_1, y_2, \cdots, y_N\}$; Parameters $\rho$ and $\theta$.
Output: The cleaned label $\{y_1^*, y_2^*, \cdots, y_N^*\}$.
  Generate the superpixel homogeneous region set $S$ of $I_f$ by Equation (2);
  Construct the sparse graph by Equation (5);
  Calculate the transfer matrix $\boldsymbol{T}$ by Equation (6);
  Update the label of $\boldsymbol{x}_i$ at time $t+1$ based on $\boldsymbol{T}$ by Equation (7);
  Acquire the cleaned label of $\boldsymbol{x}_i$ by Equation (9);
```

4. Results and Discussions

4.1. Datasets

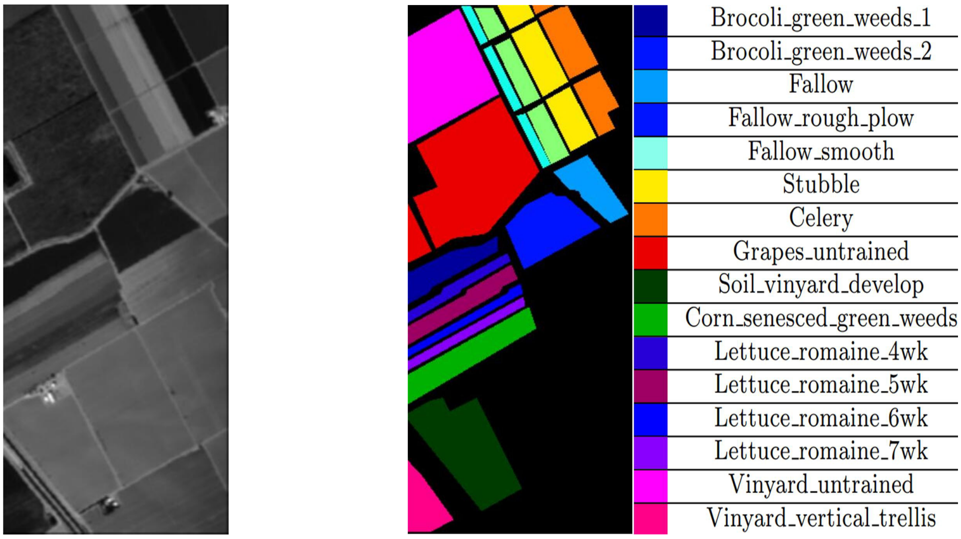
Two publicly standard HSI datasets are adopted to evaluate our method. The *Indian Pines* was captured by the AVIRIS sensor in northwestern Indiana, the scene of which includes two-thirds agriculture, and one-third forest or other natural perennial vegetation. The dataset consists of 145 × 145 pixels with 20 m per pixel and 220 bands in the wavelength range of 0.4–2.45 m. This study utilizes 200 bands for classification after removing 24 water absorption bands [15,29,57], and 10,249 labeled pixels with 16 different land cover types from the ground truth map. The gray image and the false-color composite image of the corresponding ground truth map are demonstrated in Figure 4.



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

Figure 4. (**a**) The gray image and (**b**) the corresponding ground truth map of Indian dataset.

The *Salinas* was collected by the 224-band AVIRIS sensor over Salinas Valley, California. Similar to the Indian Pines dataset, 20 water absorption bands were removed in this scene. This study employs the remaining 204 bands that are over 0.4–2.5 µm of 512 × 217 pixels with a spatial resolution of 3.7 m, and 54,129 labeled pixels with 16 classes sampled from the ground truth map. The gray image and the false-color composite image of the corresponding ground truth map are demonstrated in Figure 5.



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

Figure 5. (**a**) The gray image and (**b**) the corresponding ground truth map of Salinas dataset.

4.2. Experimental Setup

This section introduces the setup of our experiments, which mainly contains data setting, label noise generation, classification baselines, and evaluation metrics.

For the Indian Pines and Salinas datasets, pixels from the ground truth map are randomly divided into the training and testing samples. Specifically, on the Indian Pines dataset, 10 percent of pixels are randomly selected as training samples, and the others are regarded as testing samples. On the Salinas dataset, 50 training pixels from each class are randomly chosen for training, and the rest is for testing.

To comprehensively evaluate our method, we generate two kinds of label noise setting on training data through the procedure below. The first setting is random label noise (abbr. "random"). Although we focus on the more practical issue that some noisy labels may gather around the borders between different surface classes, and some of them distribute arbitrarily, existing methods always employ only random label noise. Therefore, hyperspectral pixels from the training data are randomly mislabeled at certain rate in the "random" setting. The second setting is the fusion of random label noise and boundary label noise (abbr. "both"). The boundary pixels are manually sought out in the ground truth label matrix, their k-nearest neighbors (KNN) are mislabeled to some extent, then the random noisy labels are acquired from the other pixels. The number of boundary noisy labels is same as that of random noisy labels in the "both" setting. The noise rate is set as $\rho = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. We do not show the comparison results with $\rho > 0.5$, because the labels of HSIs are provided by experts, and most of labels are always correct although a large amount of label noise probably exists in the practical applications.

Label noise cleansing is a process of HSI classification. We use four typical classifiers, namely, NN, SVM, RF, and ELM, to verify our method. In other words, we can evaluate the effectiveness of SALP based on the result of comparing the performance of classifier learning with the initial training data and the relabeled one. Three commonly-used metrics, i.e., the OA, AA, and the Kappa coefficient are used to measure the performance [57,58].

4.3. Results Comparison and Analysis with the "Random" Setting

Existing methods usually adopt only the random label noise. Thus, this section tests our method with the "random" setting in which label noise distributes randomly in the whole HSI. Three baselines of label noise cleansing are compared, which include NLA, isolation Forest (iForest) [59], and RLPA [29]. NLA is a basic baseline in which pixels with the corresponding noisy labels are directly used to train the classifiers. iForest is an anomaly detection algorithm that can be used to detect the noisy labels through the steps below: Many isolation trees are produced based on the sub-samples of training samples. Afterwards, the anomaly score for each sample is computed by analyzing the isolation trees, and those samples with scores that surpass the predefined threshold would be removed. RLPA is the state-of-the-art of label noise cleansing, and it is the first label cleansing solution to deal with the random noise label for HSI classification. The OA, AA, and Kappa coefficient of classification results over four typical classifiers on the Indian Pines and the Salinas datasets are shown in Tables 1 and 2.

Table 1. Comparisons of baselines and SALP over four typical classifiers with the "random" Setting on the Indian Pines dataset. The best results are bolded.

ρ	Classifier	OA [%]				AA [%]				Kappa Coefficient			
		NLA	iForest	RLPA	SALP	NLA	iForest	RLPA	SALP	NLA	iForest	RLPA	SALP
0.1	NN	73.24	74.34	79.30	**80.49**	69.55	69.50	72.63	**72.90**	0.6966	0.7085	0.7635	**0.777**
	SVM	84.21	83.73	**88.60**	86.75	61.37	64.28	74.56	**75.44**	0.8176	0.8122	**0.8695**	0.8480
	RF	**80.29**	79.11	79.72	78.69	**68.37**	66.68	66.26	65.85	**0.7734**	0.7597	0.7665	0.7538
	ELM	**91.35**	90.11	91.66	90.07	**84.92**	82.53	83.31	81.88	0.9012	0.8869	**0.9047**	0.8844
0.2	NN	65.24	73.08	78.88	**80.46**	62.73	66.14	72.32	**73.17**	0.6086	0.6925	0.7587	**0.7766**
	SVM	77.16	77.96	**88.01**	86.85	54.93	62.78	74.67	**75.63**	0.7340	0.7444	**0.8627**	0.8491
	RF	78.41	74.03	**79.46**	78.96	**67.37**	61.26	66.12	66.64	0.7518	0.7006	**0.7635**	0.7571
	ELM	88.65	83.30	**91.31**	90.03	82.26	73.44	**83.23**	82.50	0.8704	0.8082	**0.9006**	0.8860

Table 1. *Cont.*

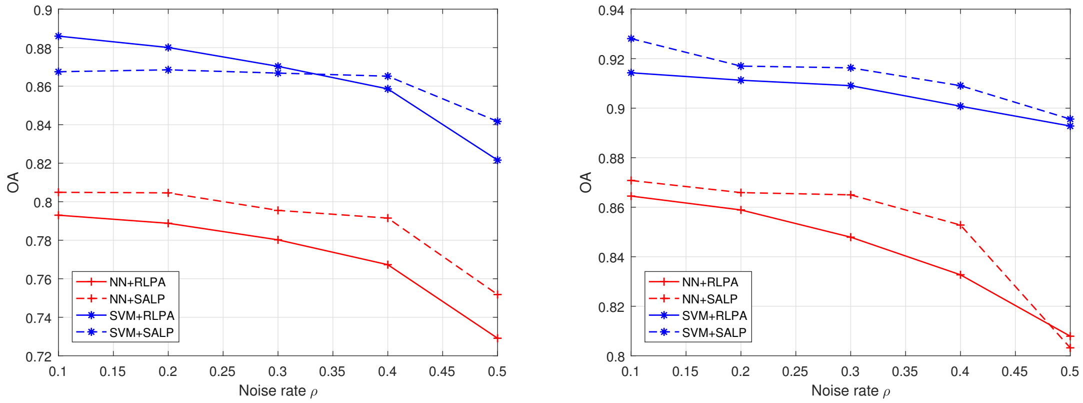
ρ	Classifier	OA [%]				AA [%]				Kappa Coefficient			
		NLA	**iForest**	**RLPA**	**SALP**	**NLA**	**iForest**	**RLPA**	**SALP**	**NLA**	**iForest**	**RLPA**	**SALP**
0.3	NN	57.46	70.95	78.02	**79.55**	54.28	61.75	70.31	**71.74**	0.5247	0.6688	0.7491	**0.7667**
	SVM	71.10	76.82	**87.03**	86.68	46.58	58.29	71.18	**74.23**	0.6603	0.7315	0.8514	**0.8595**
	RF	75.95	72.54	79.13	**79.27**	64.40	58.52	65.55	**66.39**	0.7243	0.6837	0.7598	**0.7609**
	ELM	86.41	81.51	**90.59**	89.70	77.28	68.72	80.94	**81.67**	0.8447	0.7878	**0.8925**	0.8822
0.4	NN	49.76	68.71	76.73	**79.15**	47.82	60.12	69.66	**70.49**	0.4421	0.6426	0.7348	**0.7622**
	SVM	65.04	73.01	85.86	**86.52**	39.00	57.25	71.21	**72.62**	0.5851	0.6867	0.8381	**0.8533**
	RF	72.43	69.59	**78.79**	78.64	61.06	56.91	65.61	**64.78**	0.6844	0.6494	**0.7562**	0.7538
	ELM	82.93	76.91	89.62	**90.27**	72.53	64.61	80.63	**81.46**	0.8050	0.7345	0.8815	**0.8880**
0.5	NN	40.56	64.64	72.91	**75.18**	40.06	55.39	65.31	**65.69**	0.3458	0.5967	0.6923	**0.7205**
	SVM	61.91	68.90	82.16	**84.17**	35.80	53.36	64.02	**65.86**	0.5469	0.6388	0.7952	**0.8241**
	RF	66.68	66.08	**76.75**	76.39	57.04	53.39	63.47	**63.55**	0.6212	0.6096	0.7329	**0.7292**
	ELM	76.94	72.43	87.07	**87.11**	67.74	58.94	76.69	**77.40**	0.7373	0.6826	0.8522	**0.8540**

Table 2. Comparisons of baselines and SALP over four typical classifiers with the "random" Setting on the Salinas dataset. The best results are bolded.

ρ	Classifier	OA [%]				AA [%]				Kappa Coefficient			
		NLA	**iForest**	**RLPA**	**SALP**	**NLA**	**iForest**	**RLPA**	**SALP**	**NLA**	**iForest**	**RLPA**	**SALP**
0.1	NN	78.07	85.10	86.45	**87.08**	83.95	91.72	93.01	**93.51**	0.7579	0.8350	0.8497	**0.8566**
	SVM	84.44	88.22	91.43	**92.81**	91.74	93.14	95.45	**95.93**	0.8272	0.8694	0.9047	**0.9200**
	RF	86.97	86.71	88.09	**90.10**	92.18	92.33	93.27	**94.52**	0.8553	0.8526	0.8677	**0.8901**
	ELM	92.69	90.54	92.96	**93.85**	96.31	95.20	96.58	**96.84**	0.9186	0.8949	0.9216	**0.9315**
0.2	NN	70.22	84.93	85.89	**86.59**	75.12	91.33	92.76	**93.34**	0.6721	0.8331	0.8436	**0.8504**
	SVM	85.87	88.24	91.13	**91.70**	91.30	93.16	95.20	**95.70**	0.8415	0.8694	0.9013	**0.9079**
	RF	85.54	85.98	87.82	**89.15**	90.54	91.66	93.12	**94.22**	0.8395	0.8445	0.8648	**0.8775**
	ELM	92.27	89.63	92.82	**93.41**	95.92	94.59	96.49	**96.80**	0.9139	0.8848	0.9201	**0.9267**
0.3	NN	60.85	84.08	84.79	**86.5**	65.44	90.26	92.14	**93.36**	0.5710	0.8237	0.8318	**0.8512**
	SVM	76.62	85.99	90.91	**91.63**	89.47	91.48	95.12	**95.63**	0.7437	0.8445	0.8989	**0.9069**
	RF	82.59	84.69	87.12	**88.95**	87.52	90.34	92.78	**94.21**	0.8070	0.8301	0.8571	**0.8796**
	ELM	91.34	88.22	92.56	**93.38**	95.07	93.52	96.31	**96.65**	0.9036	0.8692	0.9172	**0.9263**
0.4	NN	53.99	83.72	83.27	**85.28**	57.83	89.91	91.28	**92.65**	0.4958	0.8197	0.8150	**0.8369**
	SVM	77.52	84.79	90.08	**90.91**	85.98	90.52	94.37	**95.34**	0.7525	0.8313	0.8897	**0.9066**
	RF	79.03	84.27	86.47	**88.88**	83.62	90.01	92.54	**94.31**	0.7675	0.8256	0.8500	**0.8767**
	ELM	90.44	86.89	92.03	**93.17**	94.17	92.69	96.10	**96.62**	0.8936	0.8545	0.9114	**0.9241**
0.5	NN	44.86	82.89	**80.79**	80.32	47.17	88.53	89.22	**89.49**	0.3978	**0.8104**	0.7879	0.783
	SVM	75.03	84.52	89.28	**89.56**	75.97	89.54	**94.45**	93.85	0.7206	0.8281	0.8808	**0.8842**
	RF	73.19	83.25	**85.20**	85.10	77.06	88.62	91.50	**91.81**	0.7034	0.8143	**0.8360**	0.8352
	ELM	89.39	86.39	91.55	**91.99**	93.26	91.70	**95.67**	95.43	0.8819	0.8490	0.9061	**0.9109**

On the basis of the results, three observations can be summarized as follows: (1) Compared with the NLA that directly learns with noisy data, SALP can significantly decrease the influence of random label noise. Our OA, AA, and Kappa coefficient outperform those of NLA at most noise rates, especially when high rates of label noise are present. For example, the improvements of OA at $\rho = 0.5$ are 34.62%, 22.26%, and 9.71%, those of AA are 25.63%, 30.06%, 6.51%, and 9.66%, and those of Kappa coefficient are 37.47%, 16.36%, 13.18%, 2.9% for NN, SVM, RF, and ELM, respectively on the Indian Pines dataset, and greater improvements exist on the Salinas dataset. (2) Compared with the baselines of label noise cleansing, i.e., iForest and RLPA, SALP achieves better performance in most cases. The OA and Kappa coefficient of RLPA is slightly higher than that of SALP at low noise rates on the Indian Pines dataset, because the neighborhood constraint of RLPA still works well when the arbitrary noisy labels are almost dispersively distributed. However, our results exceed those of RLPA on Indian Pines at high noise rates, thereby probably inducing more agminated noisy labels even though a "random" setting exists. Moreover, the AA of SALP is evidently better than that of RLPA on both Indian Pines and Salinas dataset, and almost all of our results surpass baselines on the Salinas dataset. (3) Compared with ELM and RF that pertain to the label-robust learning paradigm and can be less influenced by noisy labels, NN and SVM are more sensitive to label noise. However, the performance degradations of NN and SVM based on

SALP are usually less than those based on other baselines along with the increase in noise rate, it further demonstrates the effectiveness of our method for cleansing the label noise. The OA trend of SALP and RLPA (the best baseline) is shown in Figure 6. As can be seen, our method has a smoother curve and stable performance than RLPA on two datasets.



(**a**) Indian Pines dataset.

(**b**) Salinas dataset.

Figure 6. OA trend of RLPA and SALP over NN and SVM with the "random" Setting. The proposed SALP is marked by imaginary line, and the RLPA is marked by solid line. The results of NN is labeled in red "+", and that of SVM is labeled in blue "*".

4.4. Results, Comparison and Analysis with the "both" Settings

In this section, we evaluate the effectiveness of SALP with the "both" setting, in which training samples are polluted by random label noise and boundary label noise, which is a more practical case than the "random" setting. However, few studies employ boundary label noise. Thus, we carefully choose RLPA, which is the only label cleansing method for HSI classification and the NLA as baselines. The OA, AA, and Kappa coefficient of classification results over four typical classifiers on two hyperspectral datasets are shown in Tables 3 and 4.

Table 3. Comparisons of baselines and SALP over four typical classifiers with the "both" Setting on the Indian Pines dataset. The best results are bolded.

ρ	Classifier	OA [%]			AA [%]			Kappa Coefficient		
		NLA	RLPA	SALP	NLA	RLPA	SALP	NLA	RLPA	SALP
0.1	NN	75.23	79.48	**79.99**	67.80	72.14	**72.88**	0.7071	0.7513	**0.7537**
	SVM	83.93	85.12	**85.74**	62.57	76.38	**77.42**	0.8150	**0.8627**	0.8362
	RF	78.24	78.90	**79.73**	62.71	66.06	**66.99**	0.7498	0.7568	**0.7664**
	ELM	89.94	92.02	**92.43**	81.42	82.63	**83.91**	0.8850	0.9089	**0.9135**
0.2	NN	65.86	78.45	**78.53**	63.31	71.81	**71.91**	0.6169	0.7566	**0.7714**
	SVM	78.51	84.85	**87.82**	55.69	73.41	**76.45**	0.7539	0.8481	**0.8602**
	RF	**78.72**	77.65	76.73	62.06	**65.37**	64.70	**0.7560**	0.7422	0.7314
	ELM	89.06	**91.93**	91.35	80.48	81.22	**82.46**	0.8752	**0.9077**	0.9009
0.3	NN	59.63	77.55	**77.93**	56.04	71.27	**71.56**	0.5368	0.7446	**0.7485**
	SVM	71.41	84.03	**86.89**	45.48	72.47	**73.45**	0.6693	0.8395	**0.8495**
	RF	75.20	77.98	**78.69**	63.96	64.15	**64.92**	0.7149	0.7475	**0.7557**
	ELM	87.99	**91.60**	90.45	79.40	**82.41**	82.13	0.8625	**0.8939**	0.8909
0.4	NN	49.58	77.46	**79.36**	44.97	71.97	**72.10**	0.4417	0.7439	**0.7553**
	SVM	64.24	83.18	**85.45**	36.40	64.37	**67.35**	0.5783	0.8129	**0.8325**
	RF	72.11	78.62	**78.45**	58.67	**65.93**	64.98	0.6823	**0.7544**	0.7524
	ELM	82.57	**91.06**	90.16	69.94	81.73	**82.04**	0.8010	**0.8922**	0.8875

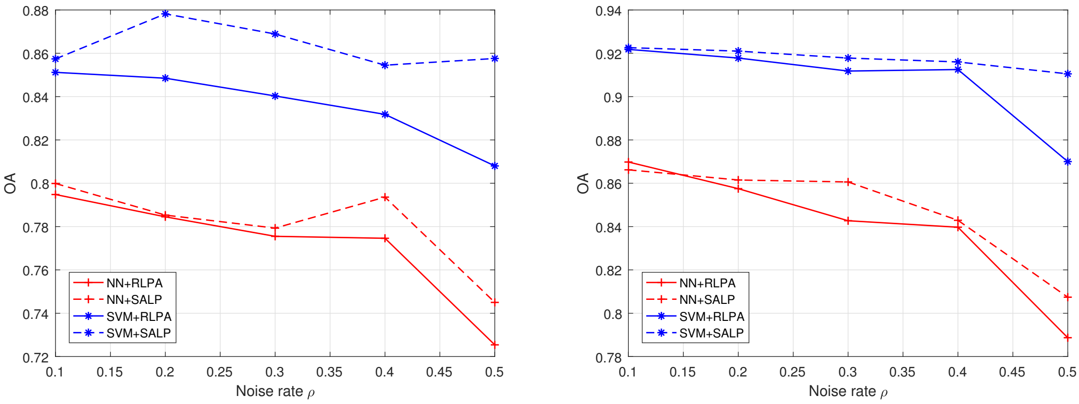
Table 3. *Cont.*

ρ	Classifier	OA [%]			AA [%]			Kappa Coefficient		
		NLA	**RLPA**	**SALP**	**NLA**	**RLPA**	**SALP**	**NLA**	**RLPA**	**SALP**
0.5	NN	42.42	72.54	**74.50**	42.18	66.20	**68.17**	0.3665	0.6892	**0.7117**
	SVM	54.47	80.80	**85.76**	28.55	61.92	**64.45**	0.4615	0.7898	**0.8313**
	RF	65.53	78.05	**78.79**	53.61	62.89	**63.03**	0.6061	0.7468	**0.7490**
	ELM	78.10	89.98	**90.74**	67.51	81.26	**81.78**	0.7505	0.8856	**0.8881**

Table 4. Comparisons of baselines and SALP over four typical classifiers with the "both" Setting on the Salinas dataset. The best results are bolded.

ρ	Classifier	OA [%]			AA [%]			Kappa Coefficient		
		NLA	**RLPA**	**SALP**	**NLA**	**RLPA**	**SALP**	**NLA**	**RLPA**	**SALP**
0.1	NN	78.20	**86.98**	86.62	84.43	**93.63**	93.52	0.7602	0.8528	**0.8556**
	SVM	90.19	92.18	**92.26**	94.09	95.85	**95.97**	0.8906	0.9082	**0.9087**
	RF	86.75	87.65	**87.65**	92.23	93.25	**93.25**	0.8640	0.8632	**0.8632**
	ELM	92.96	93.86	**93.95**	96.68	**96.99**	96.97	0.9281	0.9309	**0.9310**
0.2	NN	70.96	85.75	**86.15**	76.77	93.01	**93.34**	0.6803	0.8368	**0.8416**
	SVM	89.49	91.78	**92.10**	91.79	**95.87**	95.78	0.8831	0.9014	**0.9121**
	RF	84.44	86.48	**87.00**	90.37	**92.93**	92.86	0.8272	0.8503	**0.8514**
	ELM	92.02	93.79	**93.79**	95.67	96.12	**96.38**	0.9113	0.9191	**0.9237**
0.3	NN	62.78	84.27	**86.06**	66.43	92.16	**92.88**	0.5911	0.8287	**0.844**
	SVM	83.4	91.18	**91.78**	88.29	94.98	**95.32**	0.8136	0.9021	**0.9065**
	RF	81.60	86.11	**86.59**	88.21	92.29	**93.14**	0.7967	0.8544	**0.8605**
	ELM	90.76	**93.83**	93.44	93.96	96.10	**96.50**	0.8970	0.9116	**0.9267**
0.4	NN	53.23	83.97	**84.29**	58.89	91.21	**92.77**	0.4891	0.8080	**0.8365**
	SVM	75.60	91.25	**91.60**	83.71	94.31	**95.40**	0.7301	**0.9027**	0.9004
	RF	77.40	85.99	**86.26**	82.90	91.95	**92.99**	0.7501	0.8450	**0.8477**
	ELM	90.70	**93.68**	93.27	93.47	96.08	**96.26**	0.8989	0.9144	**0.9247**
0.5	NN	42.82	78.87	**80.74**	46.64	87.47	**88.93**	0.3788	0.7521	**0.7680**
	SVM	72.84	87.00	**91.05**	79.77	92.66	**94.03**	0.6986	0.8563	**0.8704**
	RF	75.00	**86.58**	86.24	77.21	91.05	**92.16**	0.7222	0.8313	**0.8569**
	ELM	88.90	**93.59**	93.16	93.16	**95.71**	95.51	0.8767	0.9073	**0.9100**

Four conclusions can be drawn based on the results: (1) The proposed SALP achieves evidently better results than NLA in terms of OA, AA, and Kappa coefficient, which means that our method can effectively clean the fusion of random label noise and boundary label noise. (2) The performance of the proposed SALP algorithm surpasses that of RLPA, which is the state-of-the-art of label noise cleansing for HSI classification in most situations, especially when high noise rates exist, and the improvements of results with the "both" setting are more obvious compared with those with the "random" setting, it states that our SALP is useful and suitable to deal with both random label noise and boundary label noise. (3) For RF and ELM that are insensitive to label noise, SALP can perserve their robustness. For more sensitive NN and SVM, our method can slow down the degradation of classification performance along with the increase in ρ. The OA trend of SALP and RLPA is compared in Figure 7, and the OA curves of our method are steadier those of RLPA accompanied by the rise in the noise rate. (4) To further present the visualization results, the classification maps on two noise levels ($\rho = 0.1$ and $\rho = 0.5$) are shown in Figures 8 and 9 for the Indian Pines, and in Figure 10 for the Salinas dataset. The classification maps of SALP result in improved labels especially for the boundary pixels between different surface types and achieve a smoother effect for the hyperspectral pixels in some tiny surface classes, e.g., the northeast of Indian Pines.



(**a**) Indian Pines dataset.

(**b**) Salinas dataset.

Figure 7. OA trend of RLPA and SALP over NN and SVM with the "both" setting. The proposed SALP is marked by imaginary line, and the RLPA is marked by solid line. The results of NN is labeled in red "+", and that of SVM is labeled in blue "*".

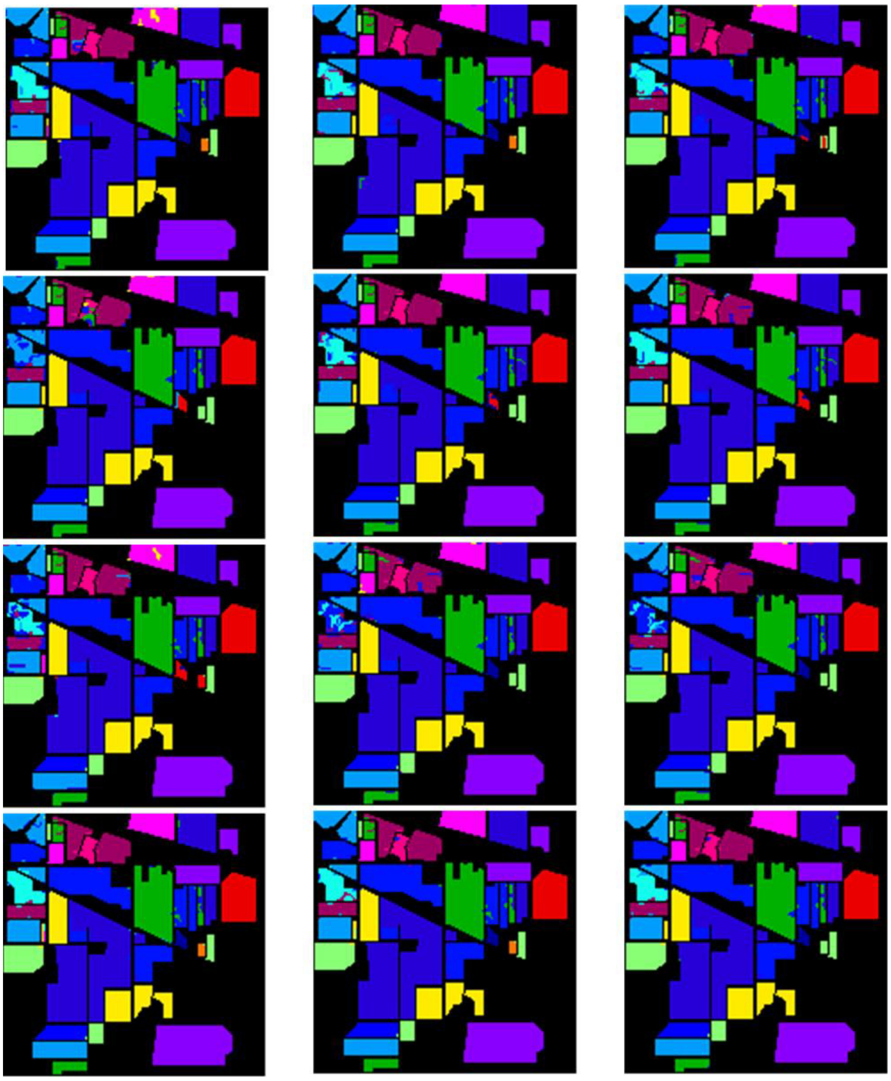


Figure 8. The classification maps of baselines and SALP over four classifiers on the Indian Pines dataset when ρ = 0.1. From the top row down to the last row: NN, SVM, RF, and ELM, from the first column to the last column: NLA, RLPA, and SALP. Please zoom in to see the details.

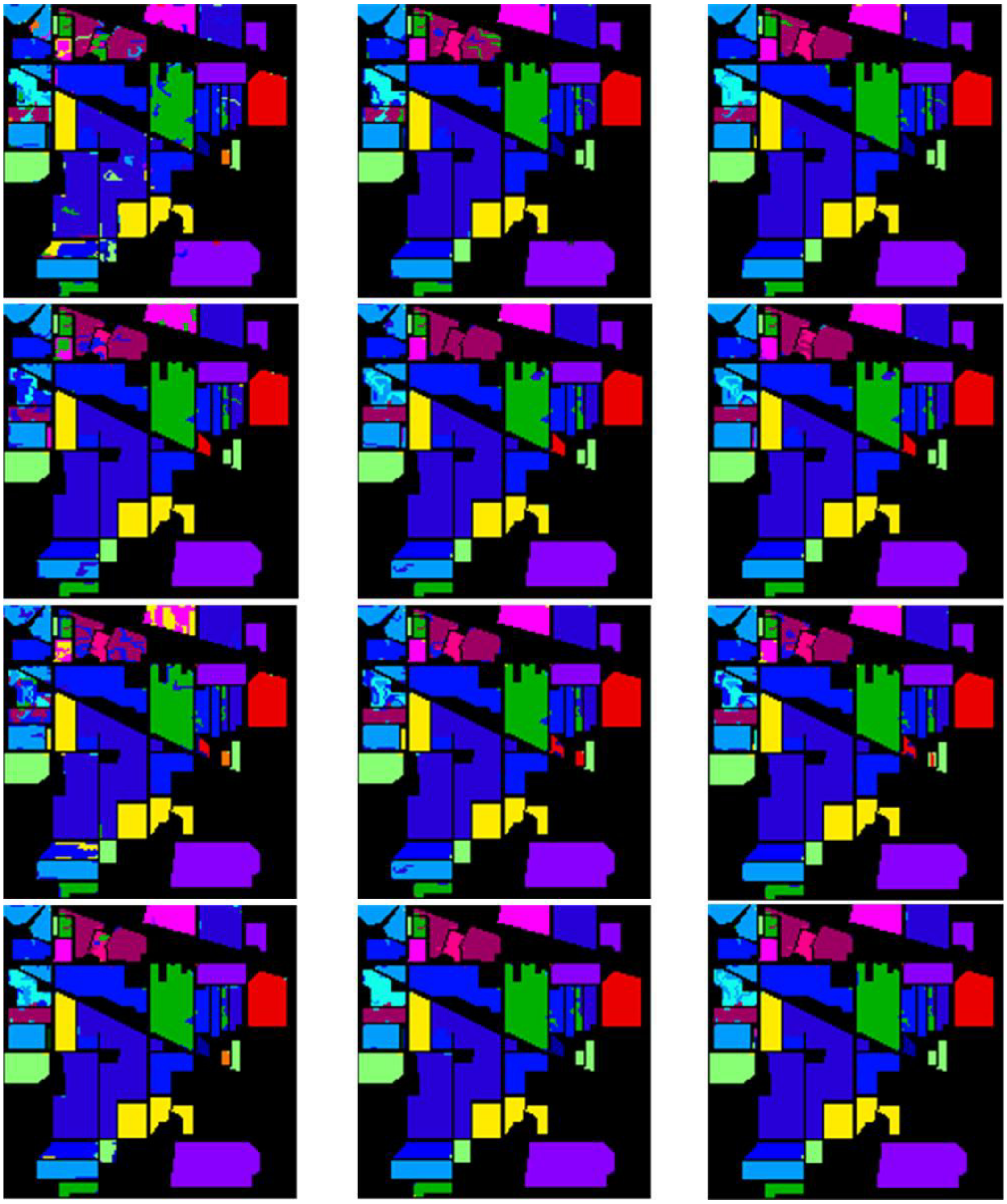
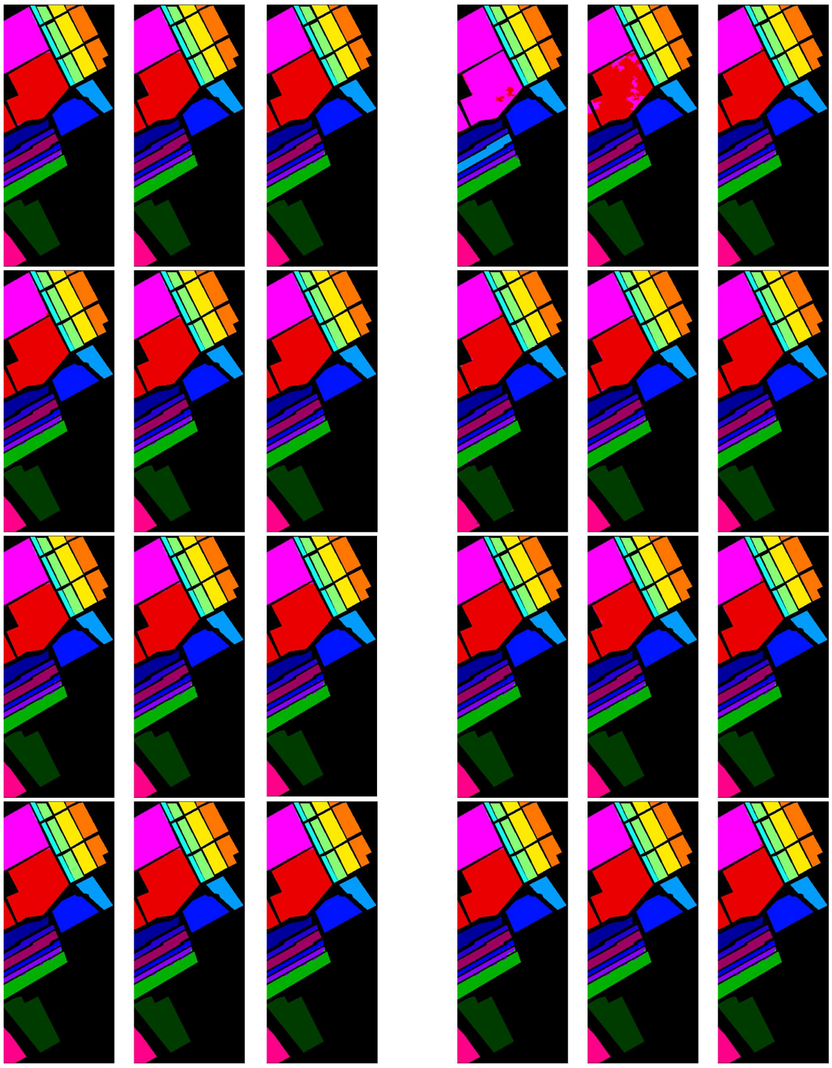


Figure 9. The classification maps of baselines and SALP over four classifiers on the Indian Pines dataset when ρ = 0.5. From the top row down to the last row: NN, SVM, RF, and ELM, from the first column to the last column: NLA, RLPA, and SALP. Please zoom in to see the details.



(a) $\rho = 0.1$

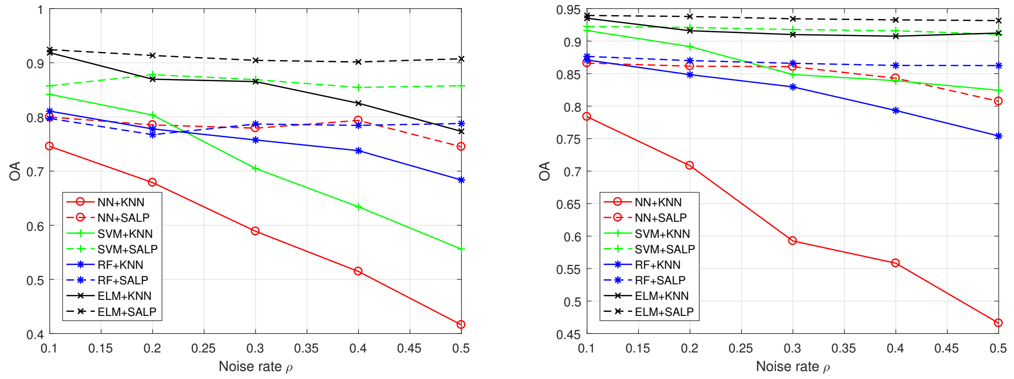
(b) $\rho = 0.5$

Figure 10. The classification maps of baselines and SALP over four classifiers on the Salinas dataset when (**a**) ρ = 0.1 and (**b**) ρ = 0.5. From the top row down to the last row: NN, SVM, RF, and ELM, from the first column to the last column: NLA, RLPA, and SALP. Please zoom in to see the details.

4.5. Further Discussion

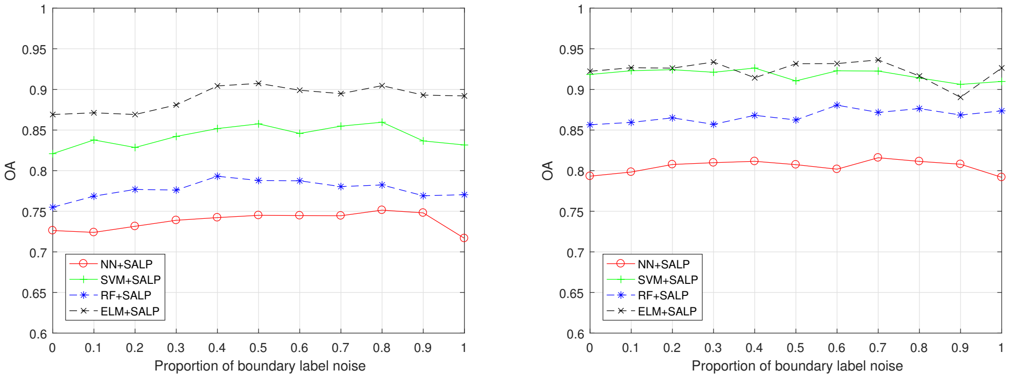
Compared with KNN graph. Figure 3 shows the graphical difference between full graph and spectral–spatial sparse graph. To present a quantitative comparison, we test a commonly-used full graph that is constructed based on KNN, and the results in the "both" setting are shown in Figure 11. We observe that the performace of SALP is significantly better than that of KNN graph at any noise rates. Moreover, the results of KNN graph regress rapidly as the noise rate grows, especially when the classifiers adopt NN and SVM. The robustness demonstrates the effectiveness of SALP, and KNN graph is sensitive to label noise for HSI classification.

Influence of boundary label noise on SALP. The proportion of random label noise and boundary label noise is equal in the previous "both" setting. In this section, we evaluate the OA trend of SALP when the proportion is altered. As shown in Figure 12, two phenomenon can be observed: First, the curves with different proportions of boundary label noise on two datasets are steady, which further proves the superiority of SALP. Second, there exists an increasing trend of OA until the proportion of boundary label noise is more than 0.8, especially on the Indina Pines dataset. On one hand, it demonstrates that SALP is usually suitable for coping with boundary label noise. On the other hand, Indian Pines has more small surface patches than Salinas, and noisy labels probably submerge these patches if there are too much boundary label noise, which may reduce the effectiveness of SALP.



(**a**) Indian Pines dataset. (**b**) Salinas dataset.

Figure 11. OA of KNN graph and SALP with the "both" Setting. The proposed SALP is marked by imaginary line, and KNN graph is marked by solid line. The results of NN, SVM, RF and ELM are labeled in red "o", green "+", blue "*", and black "x".



(**a**) Indian Pines dataset. (**b**) Salinas dataset.

Figure 12. OA trend of SALP with different proportion of boundary label noise in the "both" setting. The results of NN, SVM, RF and ELM are labeled in red "o", green "+", blue "*", and black "x".

5. Conclusions

Label noise is an unavoidable and urgent issue for HSI classification. Compared with the state-of-the-art that employs only the random label noise, this study proposes a SALP algorithm to handle a more practical noise condition in which the real HSI classification task is more likely to involve random label noise and boundary label noise. The SALP first constructs a robust sparse graph based on the spectral similarity between hyperspectral pixels with spatial constraint, and then adaptively propagates the label information from "clean" samples to "polluted" samples by exploiting the graph. Three major conclusions can be drawn based on extensive experimental results on two standard public datasets: First, our method always achieves better performance than baselines in terms of OA, AA and Kappa coefficient with either "random" setting or "both" setting, thereby indicating that SALP can effectively clean random label noise and boundary label noise. Second, SALP can slow down the degradation of classification performance along with the increase in label noise rate for those label noise-sensitive classifiers, e.g., NN and SVM. Third, the performance of our method is usually steady and sometimes improved when the proportion of boundary label noise is increased in the "both" setting, thus demonstrating that SALP can handle the gathered boundary label noise.

Author Contributions: Conceptualization, Q.L. and J.J.; Methodology, Q.L., H.Y. and J.J.; Software, Q.L. and H.Y.; Validation, Q.L. and H.Y.; Formal analysis, Q.L. and H.Y.; Investigation, Q.L. and H.Y.; Resources, Q.L.; Data curation, H.Y.; Writing—original draft preparation, Q.L.; Writing—review and editing, Q.L., H.Y. and J.J.; Visualization, H.Y.; Supervision, J.J.; Project administration, J.J.; Funding acquisition, Q.L. and J.J.

Funding: The research was supported by the National Nature Science Foundation of China (61562048).

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

Abbreviations

The following abbreviations are used in this manuscript:

SALP	Spectral–spatial sparse graph based adaptive label propagation
HSI	Hyperspectral image
SVM	Support vector machines
ELM	Extreme learning machine
RLPA	Random label propagation algorithm
ESR	Entropy rate superpixel segmentation
MVA	Majority vote algorithm
OA	Overall accuracy
NLA	Noisy label based algorithm
NN	Neighbor nearest
RF	Random forest
KNN	*k*-nearest neighbors
iForest	isolation Forest

References

1. Camps-Valls, G.; Tuia, D.; Bruzzone, L.; Benediktsson, J.A. Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. *IEEE Signal Process. Mag.* **2014**, *31*, 45–54. [CrossRef]
2. Tuia, D.; Persello, C.; Bruzzone, L. Domain adaptation for the classification of remote sensing data: An overview of recent advances. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 41–57. [CrossRef]
3. Schneider, S.; Murphy, R.J.; Melkumyan, A. Evaluating the performance of a new classifier—The GP-OAD: A comparison with existing methods for classifying rock type and mineralogy from hyperspectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2014**, *98*, 145–156. [CrossRef]
4. Tiwari, K.; Arora, M.; Singh, D. An assessment of independent component analysis for detection of military targets from hyperspectral images. *Int. J. Appl. Earth Obs. Geoinf.* **2011**, *13*, 730–740. [CrossRef]
5. He, L.; Li, J.; Liu, C.; Li, S. Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1579–1597. [CrossRef]
6. Ghamisi, P.; Plaza, J.; Chen, Y.; Li, J.; Plaza, A.J. Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–32. [CrossRef]
7. Landgrebe, D.A. *Signal Theory Methods in Multispectral Remote Sensing*; John Wiley & Sons: Hoboken, NJ, USA, 2005; Volume 29.
8. Ratle, F.; Camps-Valls, G.; Weston, J. Semisupervised neural networks for efficient hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 2271–2282. [CrossRef]
9. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
10. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3973–3985. [CrossRef]
11. Jiang, J.; Chen, C.; Yu, Y.; Jiang, X.; Ma, J. newblock Spatial-aware collaborative representation for hyperspectral remote sensing image classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 404–408. [CrossRef]
12. Jiang, X.; Song, X.; Zhang, Y.; Jiang, J.; Gao, J.; Cai, Z. Laplacian regularized spatial-aware collaborative graph for discriminant analysis of hyperspectral imagery. *Remote Sens.* **2019**, *11*, 29. [CrossRef]
13. Li, W.; Chen, C.; Su, H.; Du, Q. Local binary patterns and extreme learning machine for hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3681–3693. [CrossRef]

14. Samat, A.; Du, P.; Liu, S.; Li, J.; Cheng, L. Ensemble Extreme Learning Machines for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 1060–1069. [CrossRef]
15. Zhu, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative adversarial networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5046–5063. [CrossRef]
16. Ma, J.; Yu, W.; Liang, P.; Li, C.; Jiang, J. FusionGAN: A generative adversarial network for infrared and visible image fusion. *Inf. Fusion* **2019**, *48*, 11–26. [CrossRef]
17. Pelletier, C.; Valero, S.; Inglada, J.; Champion, N.; Marais Sicre, C.; Dedieu, G. Effect of training class label noise on classification performances for land cover mapping with satellite image time series. *Remote Sens.* **2017**, *9*, 173. [CrossRef]
18. Angluin, D.; Laird, P. Learning from noisy examples. *Mach. Learn.* **1988**, *2*, 343–370. [CrossRef]
19. Lawrence, N.D.; Schölkopf, B. Estimating a kernel Fisher discriminant in the presence of label noise. In Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, Williamstown, MA, USA, 28 June–1 July 2001; Volume 1, pp. 306–313.
20. Natarajan, N.; Dhillon, I.S.; Ravikumar, P.K.; Tewari, A. Learning with noisy labels. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 1196–1204.
21. Liu, T.; Tao, D. Classification with noisy labels by importance reweighting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 447–461. [CrossRef]
22. Tu, B.; Zhang, X.; Kang, X.; Zhang, G.; Li, S. Density Peak-Based Noisy Label Detection for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1573–1584. [CrossRef]
23. Kang, X.; Duan, P.; Xiang, X.; Li, S.; Benediktsson, J.A. Detection and correction of mislabeled training samples for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5673–5686. [CrossRef]
24. Gao, Y.; Ma, J.; Yuille, A.L. Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. *IEEE Trans. Image Process.* **2017**, *26*, 2545–2560. [CrossRef] [PubMed]
25. Frénay, B.; Verleysen, M. Classification in the presence of label noise: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 845–869. [CrossRef]
26. You, Y.L.; Kaveh, M. Fourth-order partial differential equations for noise removal. *IEEE Trans. Image Process.* **2000**, *9*, 1723–1730. [CrossRef] [PubMed]
27. Zhu, Z.; You, X.; Chen, C.P.; Tao, D.; Ou, W.; Jiang, X.; Zou, J. An adaptive hybrid pattern for noise-robust texture analysis. *Pattern Recognit.* **2015**, *48*, 2592–2608. [CrossRef]
28. Condessa, F.; Bioucas-Dias, J.; Kovačević, J. Supervised hyperspectral image classification with rejection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 2321–2332. [CrossRef]
29. Jiang, J.; Ma, J.; Wang, Z.; Chen, C.; Liu, X. Hyperspectral image classification in the presence of noisy labels. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 851–865. [CrossRef]
30. Ji, R.; Gao, Y.; Hong, R.; Liu, Q.; Tao, D.; Li, X. Spectral–spatial constraint hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 1811–1824.
31. Pu, H.; Chen, Z.; Wang, B.; Jiang, G.M. A novel spatial–spectral similarity measure for dimensionality reduction and classification of hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7008–7022.
32. Kang, X.; Li, S.; Benediktsson, J.A. Spectral–spatial hyperspectral image classification with edge-preserving filtering. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 2666–2677. [CrossRef]
33. Zheng, X.; Yuan, Y.; Lu, X. Dimensionality reduction by spatial–spectral preservation in selected bands. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5185–5197. [CrossRef]
34. Bin, C.; Jianchao, Y.; Shuicheng, Y.; Yun, F.; Huang, T. Learning with l1-graph for image analysis. *IEEE Trans. Image Process.* **2010**, *19*, 858–866.
35. Gu, Y.; Feng, K. L1-graph semisupervised learning for hyperspectral image classification. In Proceedings of the 2012 IEEE International Geoscience and Remote Sensing Symposium, Munich, Germany, 22–27 July 2012; pp. 1401–1404.
36. Wang, X.; Zhang, X.; Zeng, Z.; Wu, Q.; Zhang, J. Unsupervised spectral feature selection with l1-norm graph. *Neurocomputing* **2016**, *200*, 47–54. [CrossRef]
37. Liu, L.; Chen, L.; Chen, C.P.; Tang, Y.Y. Weighted joint sparse representation for removing mixed noise in image. *IEEE Trans. Cybern.* **2017**, *47*, 600–611. [CrossRef] [PubMed]

38. Liu, L.; Chen, C.P.; You, X.; Tang, Y.Y.; Zhang, Y.; Li, S. Mixed noise removal via robust constrained sparse representation. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 2177–2189. [CrossRef]
39. 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]
40. Liu, M.Y.; Tuzel, O.; Ramalingam, S.; Chellappa, R. Entropy rate superpixel segmentation. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 2097–2104.
41. Freund, Y. Boosting a weak learning algorithm by majority. *Inf. Comput.* **1995**, *121*, 256–285. [CrossRef]
42. Kalai, A.T.; Servedio, R.A. Boosting in the presence of noise. *J. Comput. Syst. Sci.* **2005**, *71*, 266–290. [CrossRef]
43. Chen, C.; Li, W.; Su, H.; Liu, K. Spectral–spatial classification of hyperspectral image based on kernel extreme learning machine. *Remote Sens.* **2014**, *6*, 5795–5814. [CrossRef]
44. Ma, L.; Crawford, M.M.; Zhu, L.; Liu, Y. Centroid and Covariance Alignment-Based Domain Adaptation for Unsupervised Classification of Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 2305–2323. [CrossRef]
45. Jiang, J.; Ma, J.; Chen, C.; Wang, Z.; Cai, Z.; Wang, L. SuperPCA: A superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4581–4593. [CrossRef]
46. Jia, S.; Zhang, X.; Li, Q. Spectral–Spatial Hyperspectral Image Classification Using $\ell_{1/2}$ Regularized Low-Rank Representation and Sparse Representation-Based Graph Cuts. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2473–2484. [CrossRef]
47. Ma, D.; Yuan, Y.; Wang, Q. Hyperspectral anomaly detection via discriminative feature learning with multiple-dictionary sparse representation. *Remote Sens.* **2018**, *10*, 745. [CrossRef]
48. Ma, J.; Zhao, J.; Jiang, J.; Zhou, H.; Guo, X. Locality preserving matching. *Int. J. Comput. Vis.* **2019**, *127*, 512–531. [CrossRef]
49. Zhang, S.; Li, S.; Fu, W.; Fang, L. Multiscale superpixel-based sparse representation for hyperspectral image classification. *Remote Sens.* **2017**, *9*, 139. [CrossRef]
50. Li, J.; Zhang, H.; Zhang, L. Efficient superpixel-level multitask joint sparse representation for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 5338–5351.
51. Xue, Z.; Du, P.; Li, J.; Su, H. Simultaneous sparse graph embedding for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 6114–6133. [CrossRef]
52. Chen, M.; Wang, Q.; Li, X. Discriminant analysis with graph learning for hyperspectral image classification. *Remote Sens.* **2018**, *10*, 836. [CrossRef]
53. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
54. Donoho, D.L. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Commun. Pure Appl. Math.* **2006**, *59*, 797–829. [CrossRef]
55. Kothari, R.; Jain, V. Learning from labeled and unlabeled data. In Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN'02 (Cat. No. 02CH37290), Honolulu, HI, USA, 12–17 May 2002; Volume 3, pp. 2803–2808.
56. Zhu, X.; Ghahramani, Z. *Learning from Labeled and Unlabeled Data with Label Propagatio*n; Technical Report CMU-CALD-02-107; Carnegie Mellon University: Pittsburgh, PA, USA, 2002.
57. Cheng, G.; Li, Z.; Han, J.; Yao, X.; Guo, L. Exploring hierarchical convolutional features for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6712–6722. [CrossRef]
58. Qin, Y.; Bruzzone, L.; Li, B.; Ye, Y. Cross-Domain Collaborative Learning via Cluster Canonical Correlation Analysis and Random Walker for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**. [CrossRef]
59. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data (TKDD)* **2012**, *6*, 3. [CrossRef]