Hyperspectral Image Classification in the Presence of Noisy Labels

Junjun Jiang, Jiayi Ma, Zheng Wang, Chen Chen, and Xianming Liu

Abstract*—Label information plays an important role in supervised hyperspectral image classification problem. However, current classification methods all ignore an important and inevitable problem—labels may be corrupted and collecting clean labels for training samples is difficult, and often impractical. Therefore, how to learn from the database with noisy labels is a problem of great practical importance. In this paper, we study the influence of label noise on hyperspectral image classification, and develop a random label propagation algorithm (RLPA) to cleanse the label noise. The key idea of RLPA is to exploit knowledge (*e.g.*, the superpixel based spectral-spatial constraints) from the observed hyperspectral images and apply it to the process of label propagation. Specifically, RLPA first constructs a spectral-spatial probability transfer matrix (SSPTM) that simultaneously considers the spectral similarity and superpixel based spatial information. It then randomly chooses some training samples as "clean" samples and sets the rest as unlabeled samples, and propagates the label information from the "clean" samples to the rest unlabeled samples with the SSPTM. By repeating the random assignment (of "clean" labeled samples and unlabeled samples) and propagation, we can obtain multiple labels for each training sample. Therefore, the final propagated label can be calculated by a majority vote algorithm. Experimental studies show that RLPA can reduce the level of noisy label and demonstrates the advantages of our proposed method over four major classifiers with a significant margin—the gains in terms of the average OA, AA, Kappa are impressive, *e.g.*, 9.18%, 9.58%, and 0.1043. The Matlab source code is available at https://github.com/junjun-jiang/RLPA.

***Index Terms*—Hyperspectral image classification, noisy label, label propagation, superpixel segmentation.**

I. Introduction

DUe to the rapid development and proliferation of hyperspectral remote sensing technology, hundreds of narrow spectral wavelengths for each image pixel can be easily acquired by space borne or airborne sensors, such as AVIRIS, HyMap, HYDICE, and Hyperion. This detailed spectral reflectance signature makes accurately discriminating materials of interest possible [1], [2], [3]. Because of the numerous demands in ecological science, ecology management, precision agriculture, and military applications, a large number of hyperspectral image classification algorithms have appeared on the scene [4], [5], [6], [7] by exploiting the spectral similarity and spectral-spatial feature [8], [9], [10], [11]. These methods can be divided into two categories: supervised and unsupervised. The former is generally based on clustering first and then manually determining the classes. Through incorporating the label information, these supervised methods leverage powerful machine learning algorithms to train a decision rule to predict the labels of the testing pixels. In this paper, we mainly focus on the supervised hyperspectral image classification techniques.

In the past decade, the remote sensing community has introduced intensive works to establish an accurate hyperspectral image classifier. A number of supervised hyperspectral image classification methods have been proposed, such as Bayesian models [12], neural networks [13], random forest [14], [15], support vector machine (SVM) [16], sparse representation classification [17], [18], extreme learning machine (ELM) [19], [20], and their variants [21]. Benefiting from elaborately established hyperspectral image databases, these well-trained classifiers have achieved remarkably good results in terms of classification accuracy.

However, actual hyperspectral image data inevitably contain considerable noise [22]: feature noise and label noise. To deal with the feature noise, which is caused by limited light in individual bands, and atmospheric and instrumental factors, many spectral feature noise robust approaches have been proposed [23], [24], [25], [26]. Despite being pervasive, label noise has received less attention than feature noise due to the following reasons: (i) When the information provided to an expert is very limited or the land cover is highly complex, *e.g.*, low inter-class and high intra-class variabilities, it is very easy to cause mislabeling. (ii) The low-cost, easy-to-get automatic labeling systems or inexperienced personnel assessments are less reliable [27]. (iii) If multiple experts label the same image at the same time, the labeling results may be inconsistent between different experts [28]. (iv) Information loss (due to data encoding and decoding and data dissemination) will also cause label noise.

Recently, the classification problem in the presence of label noise is becoming increasingly important and many label noise robust classification algorithms have been proposed [29], [30], [31], [32]. These methods be divided into two major categories: label noise-tolerant classification and label noise cleansing methods.The former adopts the strategies of bagging and boosting, or decision tree based ensemble techniques, while the latter aims to filter the label noise by exploiting

J. Jiang and X. Liu are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China. E-mail: jiangjunjun@hit.edu.cn; csxm@hit.edu.cn.

J. Ma is with the Electronic Information School, Wuhan University, Wuhan 430072, China. E-mail: jyma2010@gmail.com.

Z. Wang is with the Digital Content and Media Sciences Research Division, National Institute of Informatics, Tokyo 101-8430, Japan. E-mail: wangz@nii.ac.jp.

C. Chen is with the Center for Research in Computer Vision, University of Central Florida (UCF), Orlando, FL 32816-2365 USA. E-Mail: chenchen870713@gmail.com.

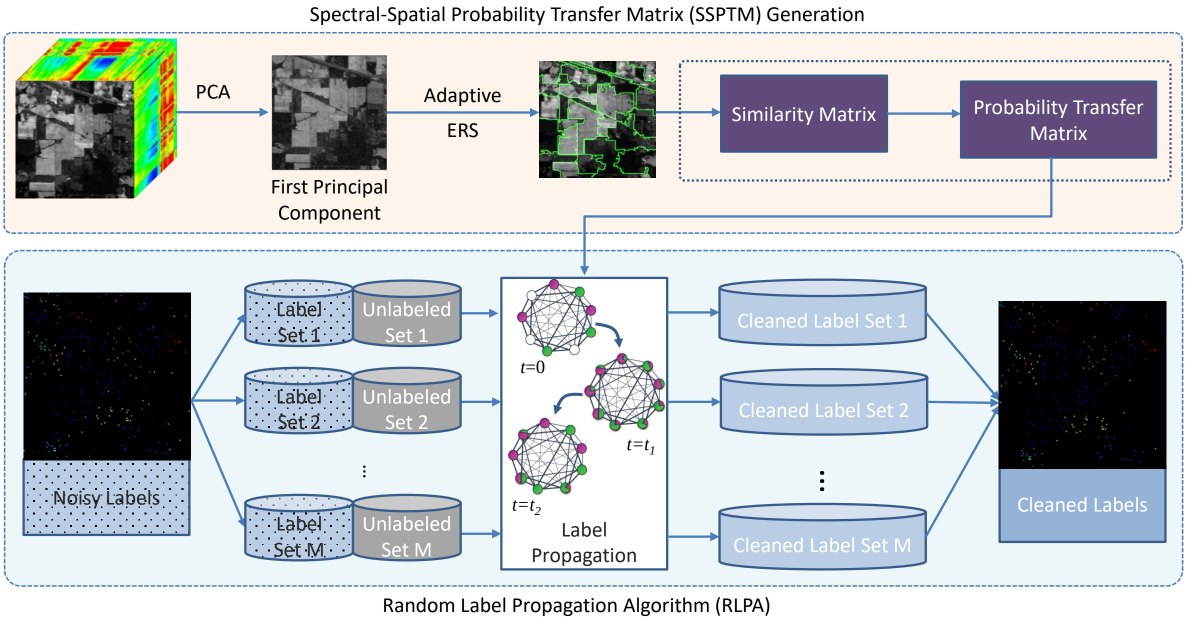


Fig. 1. Schematic diagram of the proposed random label propagation algorithm based label noise cleansing process. The up dashed block demonstrates the procedure of SSPTM generation, while the bottom dashed block demonstrates the main steps of the random label propagation algorithm.

the prior knowledge of the training samples. For more details about the general classification problem with label noise, interested reader is referred to [33] and the references therein. Generally speaking, the label noise-tolerant classification model is often designed for a specific classifier, so that the algorithm lacks universality. In contrast, as a pre-processing method, the label noise cleansing method is more general and can be used for any classifier, including the above-mentioned noisy label robust classification model. Therefore, this study will focus on the more universal noisy label cleansing approach.

Although considerable literature deals with the general image classification, there is very little research work on the classification of hyperspectral images under noisy labels [22], [34]. However, in the actual classification of hyperspectral images, this is a more urgent and unavoidable problem. As reported by Pelletier *et al.*'s study [22], the noisy labels will also mislead the training procedure of the hyperspectral image classification algorithm and severely decrease the classification accuracy of land cover. Nevertheless, there is still relatively little work specifically developed for hyperspectral image classification when encountered with label noise. Therefore, hyperspectral image classification in the presence of noisy labels is a problem that requires a solution.

In this paper, we propose to exploit the spectral-spatial constraints based knowledge to guide the cleansing of noisy labels under the label propagation framework. In particular, we develop a random label propagation algorithm (RLPA). As shown in Fig. 1, it includes two steps: (i) spectral-spatial probability transfer matrix (SSPTM) generation and (ii) random label propagation. At the first step, considering that spatial information is very important for the similarity measurement of different pixels [9], [35], [10], [36], we propose a novel affinity graph construction method which simultaneously considers the spectral similarity and the superpixel segmentation based spatial constraint. The SSPTM can be generated through the constructed affinity graph. In the second step, we randomly divide the training database to a labeled subset (with "clean" labels) and an unlabeled subset (without labels), and then perform the label propagation procedure on the affinity graph to propagate the label information from labeled subset to the unlabeled subset. Since the process of random assignment (of clean labeled samples and unlabeled samples) and propagation can be executed multiple times, the unlabeled subset will receive the multiple propagated labels. Through fusing the multiple labels of many label propagation steps with a majority vote algorithm (MVA), we can thus cleanse the label information. The philosophy behind this is that the samples with real labels dominate all training classes, and we can gradually propagate the clean label information to the entire dataset by random splitting and propagation. The proposed method is tested on three real hyperspectral image databases, namely the Indian Pines, University of Pavia, and Salinas Scene, and compared to some existing approaches using overall accuracy (OA) metric. It is shown that the proposed method outperforms these methods in terms of OA and visual classification map.

The main contributions of this article can be summarized as follows:

- We provide an effective solution for hyperspectral image classification in the presence of noisy labels. It is very general and can be seamlessly applied to the current classifiers.
- By exploiting the hyperspectral image prior, *i.e.*, the superpixel based spectral-spatial constraints, we propose a novel probability transfer matrix generation method, which can ensure label information of the same class propagate to each other, and prevent the label propagation of samples from different classes.
- The proposed RLPA method is very effective in cleansing

the label noise. Through the preprocess of RLPA, it can greatly improve the performance of the original classifiers, especially when the label noise level is very large.

This paper is organized as follows: In Section II, we present the problem setup. Section III shows the influence of label noise on the hyperspectral image classification performance. In Section IV, the details of the proposed RLPA method are given. Simulations and experiments are presented in Section V, and Section VI concludes this paper.

II. Problem Formulation

In this section, we formalize the foundational definitions and setup of the noisy label hyperspectral image classification problem. A hyperspectral image cube consists of hundreds of nearly contiguous spectral bands, with high (5-10 nm) spectral resolution, from the visible to infrared spectrum for each image pixel. Given some labeled pixels in a hyperspectral image, the task of hyperspectral image classification is to predict the labels of unseen pixels. Specifically, let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2 \cdots, \mathbf{x}_N\} \in \mathbb{R}^D$ denote a database of pixels in a D dimensional input spectral space, and $\mathcal{Y} = \{1, 2, \cdots, C\}$ denote a label set. The class labels of $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ are denoted as $\{y_1, y_2, \cdots, y_N\}$. Mathematically, we use a matrix $\mathbf{Y} \in \mathbb{R}^{N \times C}$ to represent the label, where $\mathbf{Y}_{ij} = 1$ if $\mathbf{x}_i$ is labeled as j. In order to model the label noise process, we additionally introduce another variable $\tilde{\mathbf{Y}} \in \mathbb{R}^{N \times C}$ that is used to denote the noise observed label. Let ρ denotes the label noise level (also called error rate or noise rate [37]) specifying the probability of one label being flipped to another, and thus ρ_{jk} can be mathematically formalized as:

$$\rho_{jk} = P(\tilde{\mathbf{Y}}_{ik} = 1 | \mathbf{Y}_{ij} = 1), \forall j \neq k, \text{ and } j, k \in \{1, 2, \cdots, C\}. \quad (1)$$

For example, when $\rho = 0.3$, it means that for a pixel $\mathbf{x}_i$, whose label is j, there is a 30% probability to be labeled as the other class k ($k \neq j$). To help make sense of this, we give the pseudo-codes of the noisy label generation process in Algorithm 1. $\texttt{size}(\mathbf{X})$ is a function that returns the sizes of each dimension of array $\mathbf{X}$, $\texttt{rand}(N)$ is a function that returns a random scalar drawn from the standard uniform distribution on the open interval $(0, 1)$, $\texttt{find}(\mathbf{X})$ is a function that locates all nonzero elements of an array $\mathbf{X}$, and $\texttt{randperm}(N)$ is a function that returns a row vector containing a random permutation of the integers from 1 to N inclusive.

In this paper, our main task it to predict the label of an unseen pixel $\mathbf{x}_t$, with the training data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ and the noisy label matrix $\tilde{\mathbf{Y}}$.

III. Influence of Label Noise on Hyperspectral Image Classification

In this section, we examine the influence of label noise on the hyperspectral image classification problem. As shown in Fig. 2, we demonstrate the impact of label noise on four different classifiers: neighbor nearest (NN), support vector machines (SVM), random forest (RF), and extreme learning machine (ELM). The noise level changes from 0 to 0.9 at an interval of 0.1. In Fig. 2, we report the average OA over ten runs (more details about the experimental settings can be found in Section V) as a function of the noise level. Noisy label based algorithm (NLA) represents the classification with the noisy labels. From these results, we can draw the following four conclusions:

Algorithm 1 Noisy label generation.

```
Input: The clean label matrix $\mathbf{Y}$ and the level of label noise $\rho$.
Output: The noisy label matrix $\tilde{\mathbf{Y}}$.
$[N, C] = \texttt{size}(\mathbf{Y})$;
$\tilde{\mathbf{Y}} = \mathbf{Y}$;
$\mathbf{k} = \texttt{rand}(N, 1)$;
for $i$ = 1 to $N$ do
    if $\mathbf{k}(i) \leq \rho$ then
        $p = \texttt{find}(\mathbf{Y}_{i,:} = 1)$;
        $\mathbf{r} = \texttt{randperm}(C)$;
        $\mathbf{r}(p) = [\,]$;          \\ [ ] is the null set.
        $\tilde{\mathbf{Y}}_{\mathbf{r}(1),:} = 1$;
    end if
end for
```

1) With the increase of the label noise level, the performance of all classification methods is gradually declining. Meanwhile, we also notice that the impact of label noise is not identical for all classifiers. Among these four classifiers, RF and ELM are relatively robust to label noise. When the label noise level is not large, these two classifiers can obtain better performance. In contrast, NN and SVM are much more sensitive to the label noise level. The poor results of NN and SVM can be attributed to their reliance on nearest samples and support vectors.

2) The University of Pavia and Salinas Scene databases have the same number of training samples (*e.g.*, 50)[1], but the decline rate of OA on the University of Pavia is significantly faster than that of Salinas Scene database. This is mainly because that the number of classes in the Salinas database is larger than that of the University of Pavia database ($C = 16$ *vs.* $C = 9$). With the same label noise level and same number of training samples, the more the classes are, the greater the probability of choosing the correct samples is[2]. This point is illustrated by Fig. 3. When the noise is not very large, *e.g.*, $\rho \leq 0.7$, the samples with true labels can often dominate. In this case, a good classifier can also get satisfactory performance.

3) We also show the ideal case that we know the noisy label samples and remove these training samples to obtain a noiseless training subset. From the comparisons (please refer to the same colors in each subfigure), we observe that there

[1] In this analysis, we only paid attention to these two databases in order to avoid the impact of different numbers of training sample. For the Indian Pines database, we select 10% samples for each class and the number of training samples is not the same as that in the two other databases.

[2] In this situation, for each class (after adding label noise), although the ratio of samples with corrected labels to samples with incorrectly labeled sample is $\frac{1-\rho}{\rho}$, this will reduce to $\frac{1-\rho}{\rho/(C-1)}$ when we consider the ratio of samples with corrected labels to samples labeled another class. For example, when $\rho = 0.5$, $C = 16$, the ratio of samples with corrected labels to samples labeled another class is 15:1.

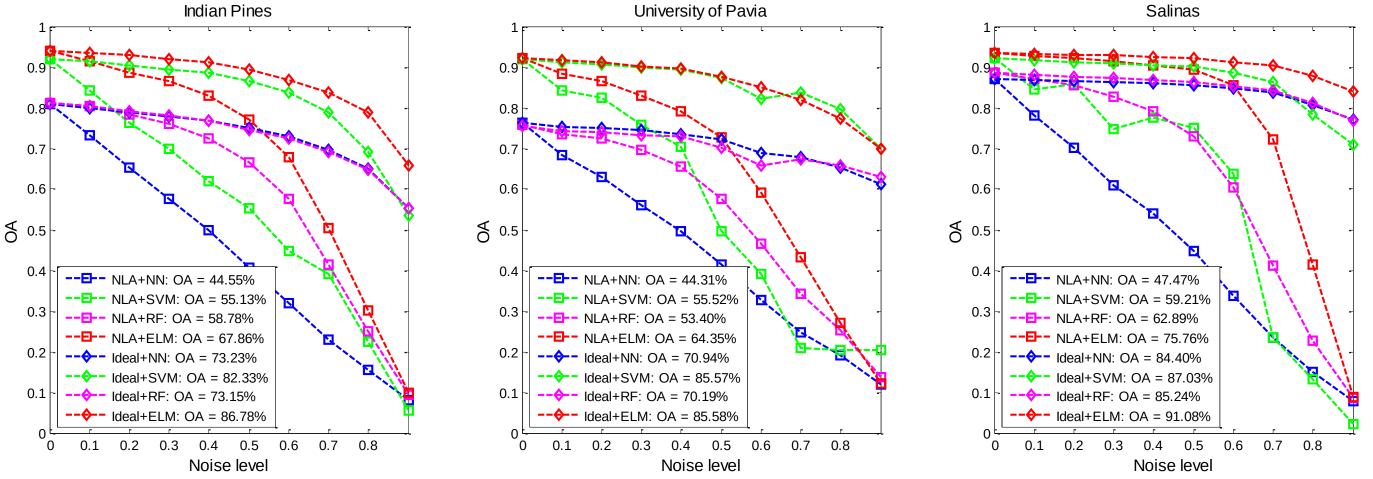


Fig. 2. Influence of the label noise on the performance (in term of OA of different classifiers) on the Indian Pines, University of Pavia, and Salinas Scene databases.

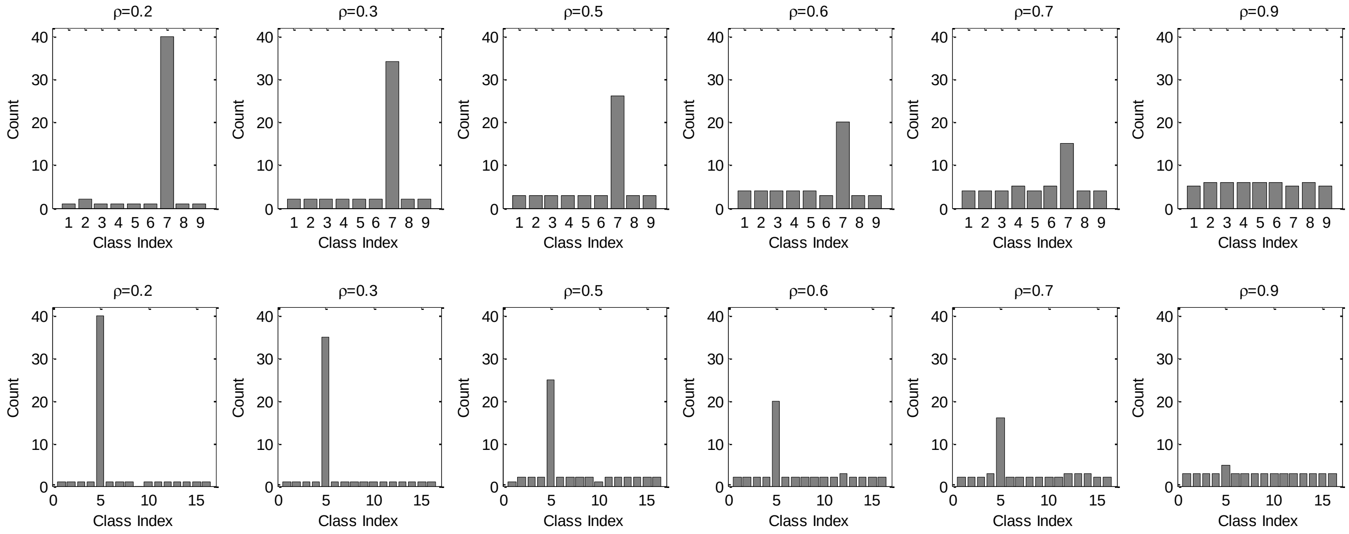


Fig. 3. The distribution of a correct class at different levels of label noise ρ. First row: the distribution of samples with true label 7 under different label noise ρ for the University of Pavia database which has nine classes. Second row: the distribution of samples with true label 5 under different levels of label noise ρ for the Salinas Scene database which has 16 classes.

is considerable room of improvement for the strategy of label noise cleansing-based algorithms. This also demonstrates the importance of preprocessing based on label noise cleansing.

IV. Proposed Method

A. Overview of the Framework

To handle the label noise, there are two main kinds of methods. The first class is to design a specific classifier that is robust to the presence of label noise, while the other obvious and tempting method is to improve the label quality of training samples. Since the latter is intuitive and can be applied to any of the subsequent classifiers, in this paper we mainly focus on how to improve and cleanse the labels. The main steps are illustrated by Fig. 4. Firstly, the prior knowledge (*e.g.*, neighborhood relationship or topology) is extracted from the training set and used to regularize the filter of label noise. Based on the cleaned labels, we can expect an intermediate classification result.

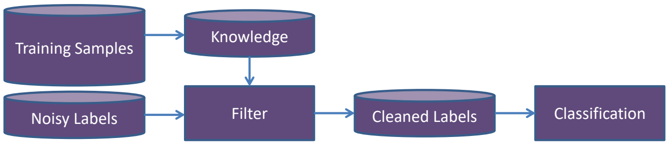


Fig. 4. The typical procedure of labels cleansing based mthod for hyperspectral image classification in the presence of label noise.

The core idea of the proposed label cleansing approach is to randomly remove the labels of some selected samples, and then apply the label propagation algorithm to predict the labels of these selected (unlabeled) samples according to a predefined SSPTM. The philosophy behind this method is that the samples with correct labels account for the majority, therefore, we can gradually propagate the clean label information to the entire samples by random splitting and propagation. This is reasonable because when the samples with wrong labels account for the majority, we cannot obtain the clean label for the samples anyway. As we know, traditional label propagation methods are sensitive to noise. This is mainly because when the label contains noise and there is no extra prior information, it is very hard for these traditional methods to construct a reasonable probability transfer matrix. The label noise can not only be removed, but is likely to be spread. Though our method is also label propagation based, we can take full advantage of the priori knowledge of hyperspectral images, *i.e.*, the superpixel based spectral-spatial constraint, to construct the SSPTM, which is the key to this label propagation based algorithm. Based on the constructed SSPTM, we can ensure that samples with same classes can be propagated to each other with a high probability, and samples with different classes cannot be propagated.

Fig. 1 illustrates the schematic diagram of the proposed method. In the following, we will first introduce how to generate the probability transfer matrix with both the spectral and spatial constraints. Then we present the random label propagation approach.

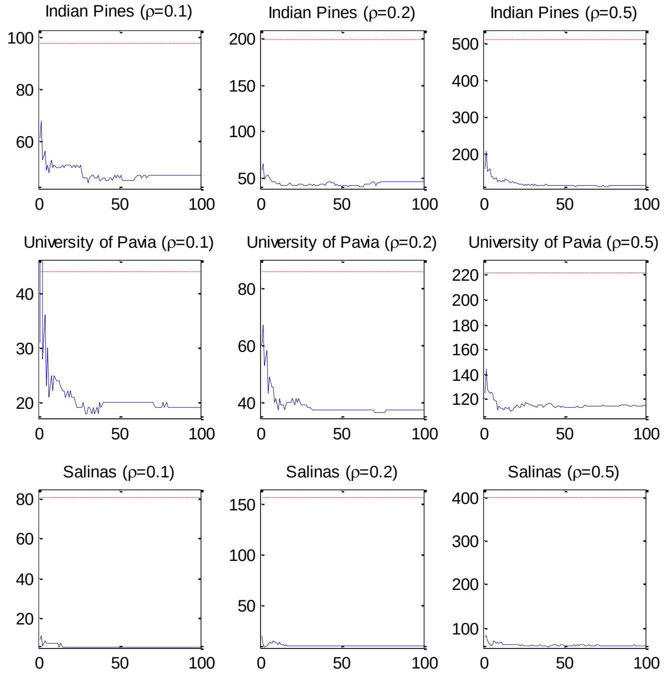


Fig. 5. Plots of the number of noisy label samples (N.N.L.S.) according to the iterations of the RLPA (blue line) under three different noise levels ($\rho = 0.1, 0.2, 0.5$). We also show the initial number (red dashed line) of noisy label samples for comparison.

B. *Construction of Spectral-Spatial Affinity Graph*

The definition of the edge weights between neighbors is the key problem in constructing an affinity graph. To measure the similarity between pixels in a hyperspectral image, the simplest way is to calculate the spectral difference through Euclidean distance, spectral angle mapper (SAM), spectral correlation mapper (SCM), or spectral information measure (SIM). However, these measurements all ignore the rich spatial information contained in a hyperspectral image, and the spectral similarity is often inaccurate due to low inter-class and high intra-class variabilities.

Our goal is to propagate label information only among samples with the same category. However, the spectral similarity based affinity graph cannot prevent label propagation of similar samples with different classes. In this paper, we propose a spectral-spatial similarity measurement approach. The basic assumption of our method is that the hyperspectral image has many homogeneous regions and pixels from one homogeneous region are more likely to be the same class. Therefore, when defining the edge weights of the affinity graph, the spectral similarity as well as the spatial constraint is taken into account at the same time.

1) Generation of Homogeneous Regions: As in many superpixel segmentation based hyperspectral image classification and restoration methods [38], [39], [40], [41], we adopt entropy rate superpixel segmentation (ESR) [42] due to its promising performance in both efficiency and efficacy. Other state-of-the-art methods such as simple linear iterative clustering (SLIC) [43] can also be used to replace the ERS. Specially, we first obtain the first principal component (through principal component analysis (PCA) [44]) of hyperspectral images, I_f, capturing the major information of hyperspectral images. This further reduces the computational cost for superpixel segmentation. It should be noted that other state-of-the-art methods such as [45] can also be equally used to replace the PCA. Then, we perform ESR on I_f to obtain the superpixel segmentation,

$$I_f = \bigcup_{k}^{T} \mathscr{X}_k, \;\; s.t. \; \mathscr{X}_k \cap \mathscr{X}_g = \emptyset, \; (k \neq g), \tag{2}$$

where T denotes the number of superpixels, and $\mathscr{X}_k$ is the k-th superpixel. The setting of T is an open and challenging problem, and is usually set experimentally. Following [46], we also introduce an adaptive parameter setting scheme to determine the value of T by exploiting the texture information. Specifically, the Laplacian of Gaussian (LoG) operator [47] is applied to detect the image structure of the first principal component of hyperspectral images. Then we can measure the texture complexity of hyperspectral images based on the detected edge image. The more complex the texture of hyperspectral images, the larger the number of superpixels, and vice versa. Therefore, we define the number of superpixel as follows:

$$T = T_{base} \frac{N_f}{N_I}, \tag{3}$$

where N_f denotes the number of nonzero elements in the detected edge image, N_I is the size of I_f, *i.e.*, the total number of pixels in I_f, and T_{base} is a fixed number for all hyperspectral images. In this way, the number of superpixels T is set adaptively, based on the spatial characteristics of different hyperspectral images. In all our experiments, we set $T_{base} = 2000$.

2) Construction of Spectral-Spatial Regularized Probabilistic Transition Matrix: Based on the segmentation result, we can construct the affinity graph by putting an edge between pixels within a homogeneous region and letting the edge weights between pixels from different homogeneous region be zero:

$$\mathbf{W}_{ij} = \begin{cases} \exp\left(-\frac{sim(\mathbf{x}_i,\mathbf{x}_j)^2}{2\sigma^2}\right), & \mathbf{x}_i, \mathbf{x}_j \in \mathscr{X}_k, \\ 0, & \mathbf{x}_i \in \mathscr{X}_k \text{ and } \mathbf{x}_j \in \mathscr{X}_g. \end{cases} \tag{4}$$

Here, $sim(\mathbf{x}_i, \mathbf{x}_j)$ denotes the spectral similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$. In this paper, we use the Euclidean distance to measure their similarity,

$$sim(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||_2, \tag{5}$$

where $\|\cdot\|$ is the l_2 norm of a vector. In Eq. (4), the variance σ is calculated region adaptively through the mean variance of all pixels in each homogeneous region:

$$\sigma = \left(\frac{1}{|\mathscr{X}_k|} \sum_{\mathbf{x}_i,\mathbf{x}_j \in \mathscr{X}_k} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \right)^{0.5}, \tag{6}$$

where $|\cdot|$ is the cardinality operator.

Upon acquiring the spectral-spatial regularized affinity graph, the label information can be propagated between nodes through the connected edges. The larger the weight between two nodes, the easier it becomes to travel. Therefore, we can define a probability transition matrix $\mathbf{T}$:

$$\mathbf{T}_{ij} = P(j \to i) = \frac{\mathbf{W}_{ij}}{\sum_{k=1}^{N} \mathbf{W}_{kj}}, \tag{7}$$

where $\mathbf{T}_{ij}$ can be seen as the probability to jump from node j to node i.

C. Random Label Propagation through Spectral-Spatial Neighborhoods

It is a very challenging problem to cleanse the label noise from the original label space. However, as a hyperspectral image, we can exploit the availableinformation about the spectral-spatial knowledge to guide the labeling of adjacent pixels. Specifically, to cleanse the noise of labels, we propose a RLPA based method. We randomly select some noisy training samples as "clean' labeled samples and set the remaining samples as unlabeled samples. The label propagation algorithm is then used to propagate the information from the "clean" labeled samples to the unlabeled samples.

Concretely, we divide the training database $\mathcal{X}$ to a labeled subset $\mathcal{X}_L = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l\}$, whose label matrix is denoted as $\tilde{\mathbf{Y}}_L = \tilde{\mathbf{Y}}(:, 1 : l) \in \mathbb{R}^{l \times C}$, and an unlabeled subset $\mathcal{X}_U = \{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \cdots, \mathbf{x}_N\}$, whose labels are discarded. l is the number of training samples that are selected for building up the "clean" labeled subset, $l = \texttt{round}(N * \eta)$. Here, η denotes the "clean" sample proportion in the total training samples, and $\texttt{round}(a)$ is a function that rounds the elements of a to the nearest integers. It should be noted that we set the first l pixels as the labeled subset, and the rest as the unlabeled subset for the convenience of expression. In our experiments, these two subsets are randomly selected from the training database $\mathcal{X}$. Now, our task is to predict the labels $\tilde{\mathbf{Y}}_U$ of unlabeled pixels $\mathcal{X}_U$, based on the graph constructed by the superpixel based spectral-spatial affinity graph.

Algorithm 2 Random label propagation algorithm (RLPA) based label noise cleansing.

```
Input: Training samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, and the corresponding labels $\{y_1, y_2, \cdots, y_N\}$, parameters $\eta$ and $\alpha$.
Output: The cleaned labels $\{y_1^*, y_2^*, \cdots, y_N^*\}$.
for $s$ = 1 to $S$ do
    $\texttt{rand}(`seed`, s)$;
    $\mathbf{k} = \texttt{randperm}(N)$;
    $l = \texttt{round}(N * \eta)$;
    $\tilde{\mathbf{Y}}_L^{(s)} = \tilde{\mathbf{Y}}(:, \mathbf{k}(1 : l)) \in \mathbb{R}^{l \times C}$;
    $\tilde{\mathbf{Y}}_U^{(s)} = 0$;
    $\tilde{\mathbf{Y}}_{LU}^{(s)} = [\tilde{\mathbf{Y}}_L^{(s)}; \tilde{\mathbf{Y}}_U^{(s)}]$;
    $\tilde{\mathbf{F}}^{*(s)} = (1 - \alpha)(\mathbf{I} - \mathbf{T})^{-1}\tilde{\mathbf{Y}}_{LU}^{(s)}$;
    for $i$ = 1 to $N$ do
        $y_i^{(s)} = \arg\max_j \mathbf{F}_{ij}^{*(s)}$
    end for
end for
for $i$ = 1 to $N$ do
    $y_i^* = MVA(\{y_i^{(1)}, y_i^{(2)}, \cdots, y_i^{(s)}\})$
end for
```

In the same manner as the label propagation algorithm (LPA) [48], in this paper we present to iteratively propagate the labels of the labeled subset $\tilde{\mathbf{Y}}_L$ to the remaining unlabeled subset $\mathcal{X}_U$ based on the spectral-spatial affinity graph. Let $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2 \cdots, \mathbf{f}_N] \in \mathbb{R}^{N \times C}$ be the predicted label. At each propagation step, we expect that each pixel absorbs a fraction of label information from its neighbors within the homogeneous region on the spectral-spatial constraint graph, and retains some label information of its initial label. Therefore, the label of $\mathbf{x}_i$ at time $t+1$ becomes,

$$\mathbf{f}_i^{t+1} = \alpha \sum_{\mathbf{x}_i,\mathbf{x}_j \in \mathscr{X}_k} \mathbf{T}_{ij}\mathbf{f}_j^t + (1 - \alpha)\tilde{\mathbf{y}}_{LU,i}, \tag{8}$$

where $0 < \alpha < 1$ is a parameter that balancing the contribution between the current label information and the label information received from its neighbors, and $\mathbf{y}_i^{LU}$ is the i-th column of $\tilde{\mathbf{Y}}_{LU} = [\tilde{\mathbf{Y}}_L; \tilde{\mathbf{Y}}_U]$. It is worth noting that we set the initial labels of these unlabeled samples as $\tilde{\mathbf{Y}}_U = \mathbf{0}$.

Mathematically, Eq. (8) can be also rewritten as follows,

$$\mathbf{F}^{t+1} = \alpha \mathbf{T}\mathbf{F}^t + (1 - \alpha)\tilde{\mathbf{Y}}_{LU}. \tag{9}$$

Following [49], we learn that Eq. (9) can be converged to an optimal solution:

$$\mathbf{F}^* = \lim_{t \to \infty} \mathbf{F}^t = (1 - \alpha)(\mathbf{I} - \mathbf{T})^{-1}\tilde{\mathbf{Y}}_{LU}. \tag{10}$$

$\mathbf{F}^*$ can be seen as a function that assigns labels for each pixel,

$$y_i = \arg\max_j \mathbf{F}_{ij}^* \tag{11}$$

Since the initial label and unlabeled samples are generated randomly, We can repeat the above process of random assignment (of "clean" labeled samples and unlabeled samples) and propagation, and obtain multiple labels for each training sample. In particular, we can get different label matrices $\tilde{\mathbf{Y}}_{LU}^{(s)}$

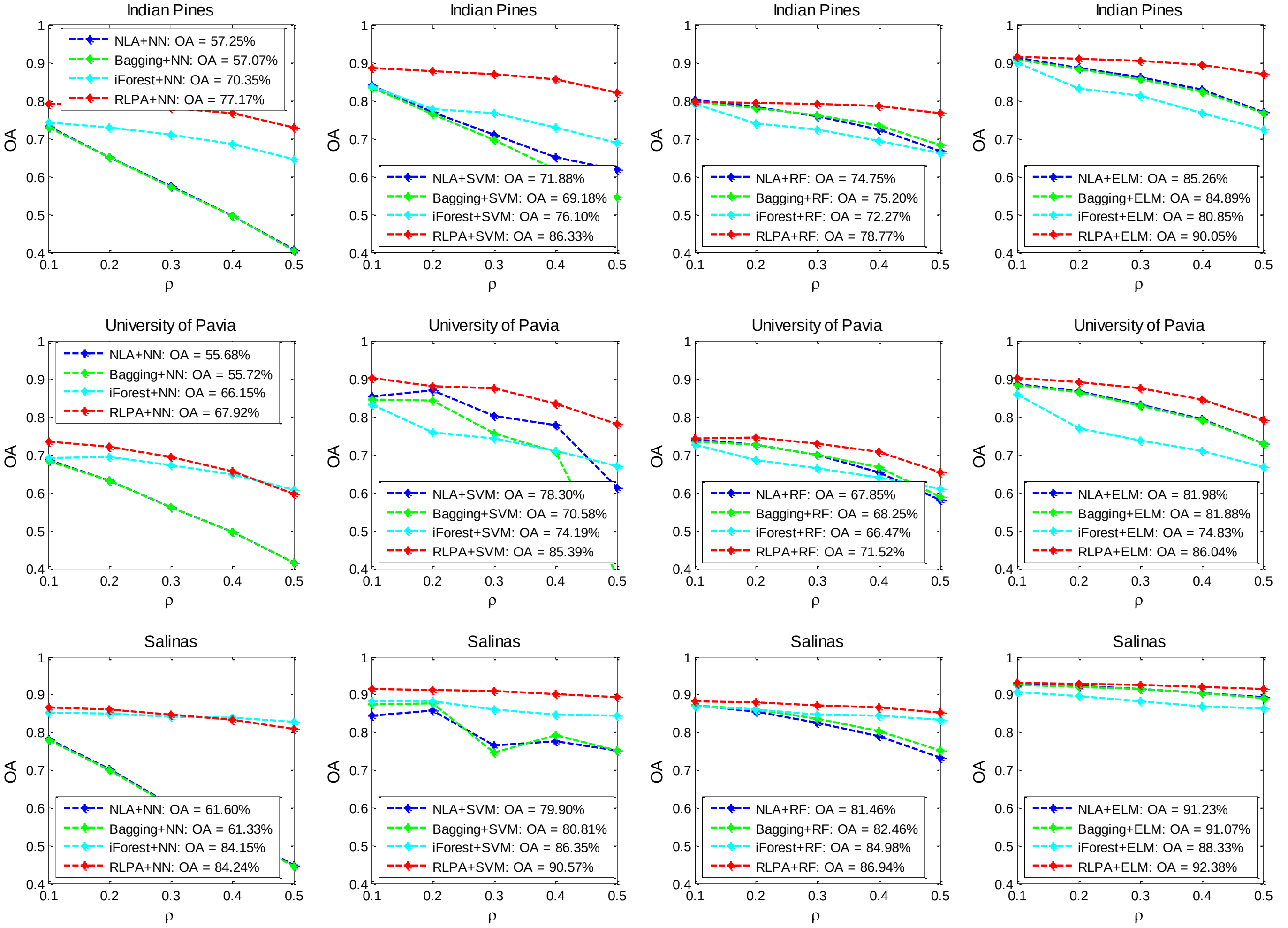


Fig. 6. The quantitative classification results in term of OA of four different methods (NLA, Bagging, iForest, and RLPA) with four different classifiers (NN, SVM, RF, and ELM) on the Indian Pines (the first row), University of Pavia (the second row), and Salinas Scene (the third row). The average OAs of four different methods on three databases with four different classifiers are: NLA (OA = 73.93%), Bagging (OA = 73.20%), iForest (OA = 77.09%), and RLPA (OA = 83.11%).

at the s th round, $s = 1, 2, ..., S$. Here, S is the total number in iterations. We can then calculate the label assignment matrix $\mathbf{F}^{*(1)}, \mathbf{F}^{*(2)}, \cdots, \mathbf{F}^{*(S)}$ according to Eq. (10). Thus, we obtain S labels for $\mathbf{x}_i$, $y^{(1)}, y^{(2)}, \cdots, y^{(S)}$. The final propagated label can be calculated by MVA [50].

Because we fully considered the spatial information of hyperspectral images in the process of propagation, we can expect that these propagated label results are better than the original noisy labels in the sense of the proportion that noisy label samples is decreasing (as the number of iterations increase). We illustrate this point in Fig. 5, which plots the number of noisy label samples according to the iterations of the proposed RLPA under three different noise levels. With the increase of iteration, the number of noisy label samples becomes less and less. The red dashed line shows the initial number of noisy label samples. Obviously, after a certain number of iterations, the number of noisy label samples is significantly reduced. In our experiments, we fix the value of S to 100.

Algorithm 2 shows the entire process of our proposed RLPA based label cleansing method. MVA represents the majority vote algorithm that returns the majority of a sequence of elements.

V. Experiments

In this section, we describe how we set up the experiments. Firstly, we introduce the three hyperspectral image databases used in our experiments. Then, we show the comparison of our results with four other methods. Subsequently, we demonstrate the effectiveness of our proposed SSPTM. Finally, we assess the influence of parameter settings. We intend to release our codes to the research community to reproduce our experimental results and learn more details of our proposed method from them.

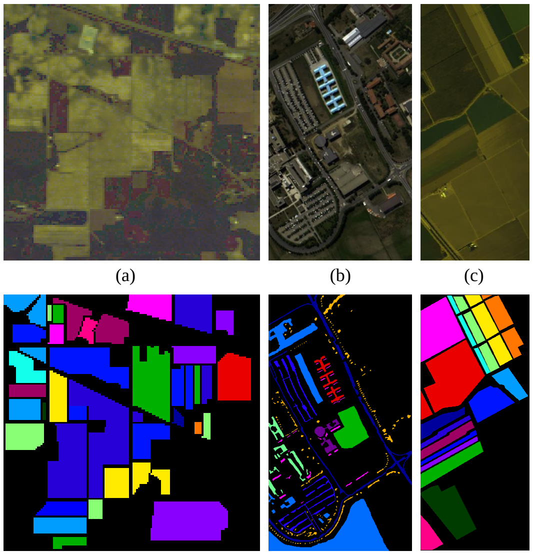
A. Database

In order to evaluate the proposed RLPA method, we use three publicly available hyperspectral image databases[3].

[3] http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

TABLE I
NUMBER OF SAMPLES IN THE INDIAN PINES, UNIVERSITY OF PAVIA, AND SALINAS SCENE IMAGES. THE BACKGROUND COLOR IS USED TO DISTINGUISH DIFFERENT CLASSES.

Indian Pines		University of Pavia		Salinas Scene	
Class Names	Numbers	Class Names	Numbers	Class Names	Numbers
Alfalfa	46	Asphalt	6631	Brocoli_green_weeds_1	2009
Corn-notill	1428	Bare soil	18649	Brocoli_green_weeds_2	3726
Corn-mintill	830	Bitumen	2099	Fallow	1976
Corn	237	Bricks	3064	Fallow_rough_plow	1394
Grass-pasture	483	Gravel	1345	Fallow_smooth	2678
Grass-trees	730	Meadows	5029	Stubble	3959
Grass-pasture-mowed	28	Metal sheets	1330	Celery	3579
Hay-windrowed	478	Shadows	3682	Grapes_untrained	11271
Oats	20	Trees	947	Soil_vinyard_develop	6203
Soybean-notill	972			Corn_senesced_green_weeds	3278
Soybean-mintill	2455			Lettuce_romaine_4wk	1068
Soybean-clean	593			Lettuce_romaine_5wk	1927
Wheat	205			Lettuce_romaine_6wk	916
Woods	1265			Lettuce_romaine_7wk	1070
Buildings-Grass-Trees-Drives	386			Vinyard_untrained	7268
Stone-Steel-Towers	93			Vinyard_vertical_trellis	1807
Total Number	10249	Total Number	42776	Total Number	54129


(a) (b) (c)

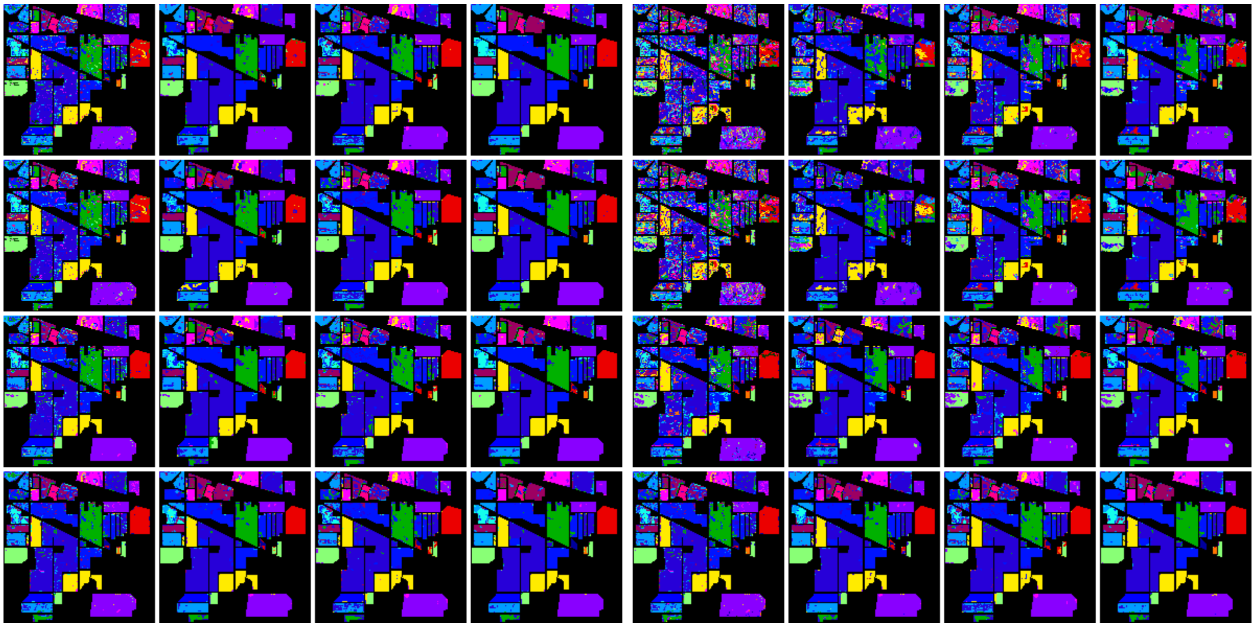
(d) (e) (f)

Fig. 7. The RGB composite images and ground reference information of three hyperspectral image databases: (a) Indian Pines, (b) University of Pavia, and (c) Salinas Scene.

1) The first hyperspectral image database is the *Indian Pine*, covering the agricultural fields with regular geometry, was acquired by the AVIRIS sensor in June 1992. The scene is 145×145 pixels with 20 m spatial resolution and 220 bands in the 0.4-2.45 m region. In this paper, 20 low SNR bands are removed and a total of 200 bands are used for classification. This database contains 16 different land-cover types, and approximately 10,249 labeled pixels are from the ground-truth map. Fig. 7 (a) shows an infrared color composite image and Fig. 7 (d) is the ground reference data.
2) The second hyperspectral image database is the *University of Pavia*, covering an urban area with some buildings and large meadows, which contains a spatial coverage of 610×340 pixels and is collected by the ROSIS sensor under the HySens project managed by DLR (the German Aerospace Agency). It generates 115 spectral bands, of which 12 noisy and water-bands are removed. It has a spectral coverage from 0.43-0.86 μm and a spatial resolution of 1.3 m. Approximately 42,776 labeled pixels with nine classes are from the ground truth map, details of which are provided in Table I. Fig. 7 (b) shows an infrared color composite image and Fig. 7 (e) is the ground reference data.
3) The third hyperspectral image database is the *Salinas Scene*, capturing an area over Salinas Valley, CA, USA, was collected by the 224-band AVIRIS sensor over Salinas Valley, California. It generates 512×217 pixels and 204 bands over 0.4-2.5 μm with spatial resolution of 3.7 m, of which 20 water absorption bands are removed before classification. In this image, there are approximately 54,129 labeled pixels with 16 classes sampled from the ground truth map, details of which are provided in Table I. Fig. 7 (c) shows an infrared color composite image and Fig. 7 (f) is the ground reference data.

For the three databases, the training and testing samples are randomly selected from the available ground truth maps. The class-specific numbers of labeled samples are shown in Table I. For the Indian Pines database, 10% of the samples are randomly selected for training, and the rest is used testing. As for the other databases, *i.e.*, University of Pavia and Salinas Scene, we randomly choose 50 samples from each class to build the training set, leaving the remaining samples form the testing set.

As discussed previously, we add random noise to the labels of training samples with the level of ρ. In other words, each label in the training set will flip to another with the probability of ρ. In our experiments, we only show the comparison results of different methods with $\rho \leq 0.5$. That is, given a labeled training database, we assume that more than half of the labels are correct, because that the label information is provided by an expert and the labels are not random. Therefore, there are reasons to make such an assumption. Specifically, in our experiments we test typical cases where $\rho = 0.1, 0.2, 0.3, 0.4, 0.5$.



(a) $\rho = 0.1$

(b) $\rho = 0.5$

Fig. 8. The classification maps of four different methods (each row represents different methods) with four different classifiers (each column represents different classifiers) on the Indian Pines database when (a) $\rho = 0.1$ and (b) $\rho = 0.5$. From the first row to the last row: LNA, Bagging, iForest, and RLPA, from the first column to the last column: NN, SVM, RF, and ELM. Please zoom in on the electronic version to see a more obvious contrast.

B. Result Comparison

To demonstrate the effectiveness of the proposed method, we test our proposed framework with four widely used classifiers in the field of hyperspectral image calcification, which are nearest neighborhood (NN) [51], support vector machine (SVM) [16], random forest [14], [15], and extreme learning machine (ELM) [19], [20]. Since there is no specific noisy label classification algorithm for hyperspectral images, we carefully design and adjust some label noise robust general classification methods to adapt our framework. In particular, the four comparison methods used in our experiments are the following:

- Noisy label based algorithm (*NLA*): we directly use the training samples and their corresponding noisy labels to train the classification models using the above-mentioned four classifiers.
- Bagging-based classification (*Bagging*) [52]: the approach of [52] first produces different training subsets by resampling (70% of training samples are selected each time), and then fuses the classification results of different training subsets.
- isolation Forest (*iForest*) [53]: this is an anomaly detection algorithm, and we apply it to detect the noisy label samples. In particular, in the training phase, it constructs many isolation trees using sub-samples of the given training samples. In the evaluation phase, the isolation trees can be used to calculate the score for each sample to determine the anomaly points. Finally, these samples will be removed when their anomaly scores exceeds the predefined threshold.
- *RLPA*: the proposed random label propagation based label noise cleansing method operates by repeating the random assignment and label propagation, and fusing the label information by different iterations.

NLA can be seen as a baseline, Bagging-based method [52] is a classification ensemble strategy that has been proven to be robust to label noise [54]. iForest [53] can be regarded as a label cleansing processing as our proposed method in the sense that the goals of these methods are to remove the samples with noisy labels. In our experiments, we carefully tuned the parameters of the four classifiers to achieve the best performance under different comparison methods. Specifically, set all parameters to a larger range, and the reported results of different comparison methods with different classifiers are the best when setting appropriate values for the parameters.

Generally speaking, the OA, average accuracy (AA), and the Kappa coefficient can be used to measure the performance of different classification results. In Table II, Table III, and Table IV, we report the OA, AA, and Kappa scores of four different methods with four different classifiers on the Indian Pines, University of Pavia, and Salinas Scene databases, resepctively. The average OA, AA, and Kappa of LNA, Bagging, iForest, and RLPA for all cases are reported at Table V. To make the comparison more intuitive, we plot their OA performance in Fig. 6[4]. In the legend of each subfigure, we also give the average OA of all five noise levels of different methods. From these results, we can draw the following conclusions:

- When compared with using the original training samples with label noise (*i.e.*, the NLA method), the Bagging method cannot boost the performance. This indicates that re-sampling the training samples cannot improve the performance of the algorithm in the presence of noisy

[4] Since these three measurements of OA, AA, and Kappa are consistent with each other, we only plot the results in terms of OA in all our experiments

TABLE II
OA, AA, AND KAPPA PERFORMANCE OF FOUR DIFFERENT METHODS WITH FOUR DIFFERENT CLASSIFIERS ON THE INDIAN PINES DATABASE.

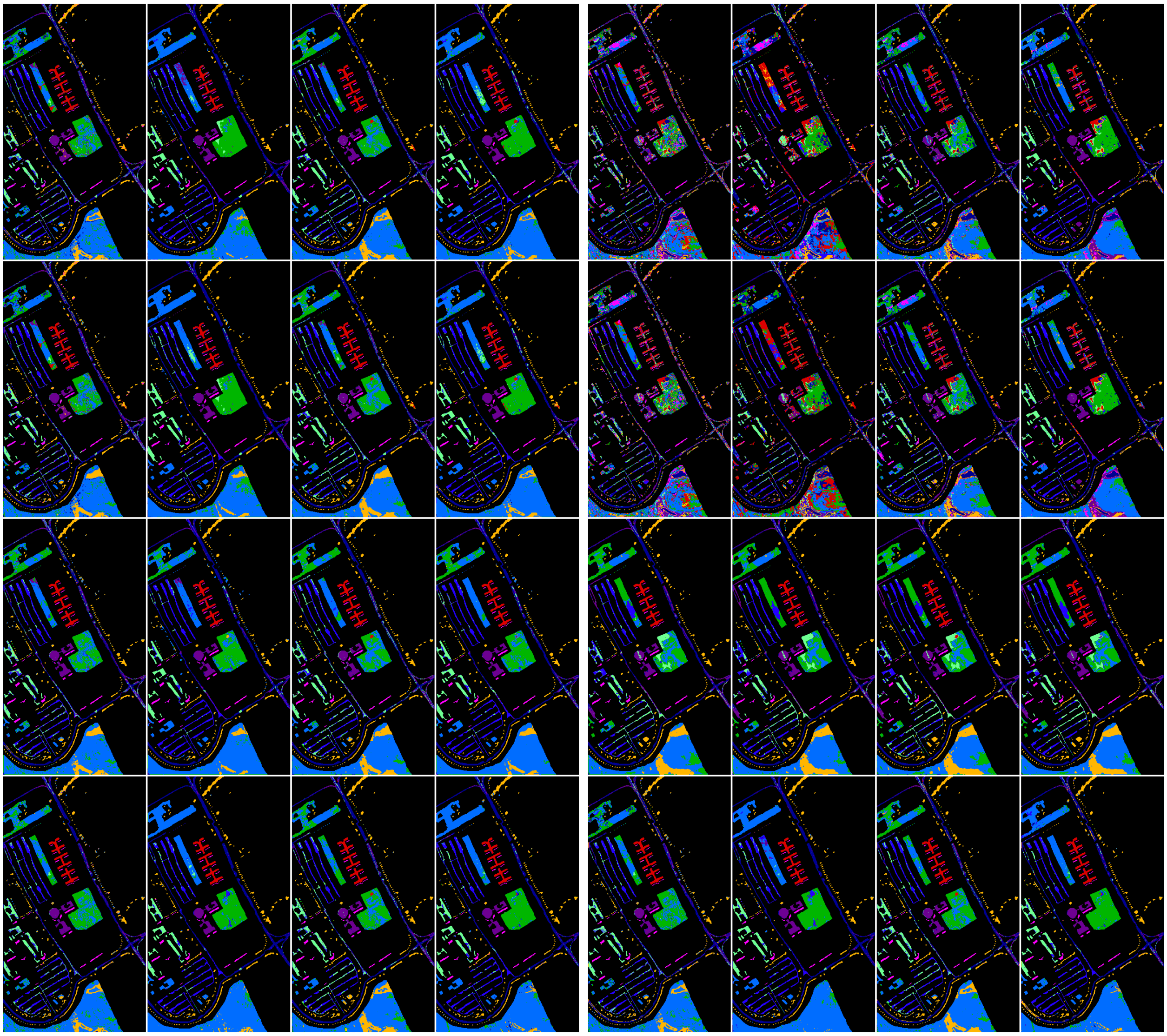
ρ	*Classifier*	OA [%]				AA [%]				Kappa			
		NLA	Bagging	iForest	RLPA	NLA	Bagging	iForest	RLPA	NLA	Bagging	iForest	RLPA
0.1	NN	73.24	73.01	74.34	79.30	69.55	69.32	69.50	72.63	0.6966	0.6941	0.7085	0.7635
	SVM	84.21	83.56	83.73	88.60	61.37	61.17	64.28	74.56	0.8176	0.8101	0.8122	0.8695
	RF	80.29	79.64	79.11	79.72	68.37	67.09	66.68	66.26	0.7734	0.7656	0.7597	0.7665
	ELM	91.35	90.84	90.11	91.66	84.92	84.26	82.53	83.31	0.9012	0.8954	0.8869	0.9047
0.2	NN	65.24	65.04	73.08	78.88	62.73	62.49	66.14	72.32	0.6086	0.6063	0.6925	0.7587
	SVM	77.16	76.42	77.96	88.01	54.93	51.11	62.78	74.67	0.7340	0.7269	0.7444	0.8627
	RF	78.41	78.09	74.03	79.46	67.37	66.45	61.26	66.12	0.7518	0.7476	0.7006	0.7635
	ELM	88.65	88.48	83.30	91.31	82.26	82.06	73.44	83.23	0.8704	0.8684	0.8082	0.9006
0.3	NN	57.46	57.21	70.95	78.02	54.28	53.98	61.75	70.31	0.5247	0.5219	0.6688	0.7491
	SVM	71.10	69.68	76.82	87.03	46.58	41.34	58.29	71.18	0.6603	0.6462	0.7315	0.8514
	RF	75.95	76.28	72.54	79.13	64.40	64.09	58.52	65.55	0.7243	0.7276	0.6837	0.7598
	ELM	86.41	85.82	81.51	90.59	77.28	76.66	68.72	80.94	0.8447	0.8378	0.7878	0.8925
0.4	NN	49.76	49.57	68.71	76.73	47.82	47.53	60.12	69.66	0.4421	0.4400	0.6426	0.7348
	SVM	65.04	61.72	73.01	85.86	39.00	32.61	57.25	71.21	0.5851	0.5468	0.6867	0.8381
	RF	72.43	73.52	69.59	78.79	61.06	61.56	56.91	65.61	0.6844	0.6965	0.6494	0.7562
	ELM	82.93	82.49	76.91	89.62	72.53	71.83	64.61	80.63	0.8050	0.8000	0.7345	0.8815
0.5	NN	40.56	40.51	64.64	72.91	40.06	39.89	55.39	65.31	0.3458	0.3454	0.5967	0.6923
	SVM	61.91	54.50	68.96	82.16	35.80	26.58	53.36	64.02	0.5469	0.4538	0.6388	0.7952
	RF	66.68	68.46	66.08	76.75	57.04	58.29	53.39	63.47	0.6212	0.6403	0.6096	0.7329
	ELM	76.94	76.82	72.43	87.07	67.74	67.54	58.94	76.69	0.7373	0.7359	0.6826	0.8522

TABLE III
OA, AA, AND KAPPA PERFORMANCE OF FOUR DIFFERENT METHODS WITH FOUR DIFFERENT CLASSIFIERS ON THE UNIVERSITY OF PAVIA DATABASE.

ρ	*Classifier*	OA [%]				AA [%]				Kappa			
		NLA	Bagging	iForest	RLPA	NLA	Bagging	iForest	RLPA	NLA	Bagging	iForest	RLPA
0.1	NN	73.24	73.01	74.34	79.30	69.55	69.32	69.50	72.63	0.6044	0.6019	0.6137	0.6628
	SVM	84.21	83.56	83.73	88.60	61.37	61.17	64.28	74.56	0.8120	0.8014	0.7839	0.8718
	RF	80.29	79.64	79.11	79.72	68.37	67.09	66.68	66.26	0.6702	0.6657	0.6540	0.6755
	ELM	91.35	90.84	90.11	91.66	84.92	84.26	82.53	83.31	0.8494	0.8472	0.8174	0.8700
0.2	NN	65.24	65.04	73.08	78.88	62.73	62.49	66.14	72.32	0.5401	0.5409	0.6165	0.6461
	SVM	77.16	76.42	77.96	88.01	54.93	51.11	62.78	74.67	0.8312	0.7958	0.6968	0.8431
	RF	78.41	78.09	74.03	79.46	67.37	66.45	61.26	66.12	0.6528	0.6543	0.6117	0.6782
	ELM	88.65	88.48	83.30	91.31	82.26	82.06	73.44	83.23	0.8259	0.8218	0.7102	0.8589
0.3	NN	57.46	57.21	70.95	78.02	54.28	53.98	61.75	70.31	0.4606	0.4614	0.5919	0.6146
	SVM	71.10	69.68	76.82	87.03	46.58	41.34	58.29	71.18	0.7383	0.6780	0.6744	0.8376
	RF	75.95	76.28	72.54	79.13	64.40	64.09	58.52	65.55	0.6184	0.6191	0.5839	0.6597
	ELM	86.41	85.82	81.51	90.59	77.28	76.66	68.72	80.94	0.7806	0.7796	0.6699	0.8366
0.4	NN	49.76	49.57	68.71	76.73	47.82	47.53	60.12	69.66	0.3907	0.3896	0.5626	0.5716
	SVM	65.04	61.72	73.01	85.86	39.00	32.61	57.25	71.21	0.7132	0.6165	0.6375	0.7892
	RF	72.43	73.52	69.59	78.79	61.06	61.56	56.91	65.61	0.5675	0.5816	0.5551	0.6350
	ELM	82.93	82.49	76.91	89.62	72.53	71.83	64.61	80.63	0.7335	0.7319	0.6358	0.8019
0.5	NN	40.56	40.51	64.64	72.91	40.06	39.89	55.39	65.31	0.3051	0.3054	0.5215	0.5054
	SVM	61.91	54.50	68.96	82.16	35.80	26.58	53.36	64.02	0.5309	0.2953	0.5939	0.7209
	RF	66.68	68.46	66.08	76.75	57.04	58.29	53.39	63.47	0.4850	0.4963	0.5265	0.5714
	ELM	76.94	76.82	72.43	87.07	67.74	67.54	58.94	76.69	0.6570	0.6590	0.5917	0.7349

TABLE IV
OA, AA, AND KAPPA PERFORMANCE OF FOUR DIFFERENT METHODS WITH FOUR DIFFERENT CLASSIFIERS ON THE SALINAS SCENE DATABASE.

ρ	*Classifier*	OA [%]				AA [%]				Kappa			
		NLA	Bagging	iForest	RLPA	NLA	Bagging	iForest	RLPA	NLA	Bagging	iForest	RLPA
0.1	NN	78.07	77.94	85.10	86.45	83.95	83.94	91.72	93.01	0.7579	0.7564	0.8350	0.8497
	SVM	84.44	87.42	88.22	91.43	91.74	93.09	93.14	95.45	0.8272	0.8598	0.8694	0.9047
	RF	86.97	87.14	86.71	88.09	92.18	92.43	92.33	93.27	0.8553	0.8572	0.8526	0.8677
	ELM	92.69	92.57	90.54	92.96	96.31	96.24	95.20	96.58	0.9186	0.9173	0.8949	0.9216
0.2	NN	70.22	70.01	84.93	85.89	75.12	74.96	91.33	92.76	0.6721	0.6698	0.8331	0.8436
	SVM	85.87	87.58	88.24	91.13	91.30	92.59	93.16	95.20	0.8415	0.8614	0.8694	0.9013
	RF	85.54	86.06	85.98	87.82	90.54	90.95	91.66	93.12	0.8395	0.8453	0.8445	0.8648
	ELM	92.27	92.10	89.63	92.82	95.92	95.82	94.59	96.49	0.9139	0.9121	0.8848	0.9201
0.3	NN	60.85	60.60	84.08	84.79	65.44	65.21	90.26	92.14	0.5710	0.5685	0.8237	0.8318
	SVM	76.62	74.73	85.99	90.91	89.47	83.08	91.48	95.12	0.7437	0.7214	0.8445	0.8989
	RF	82.59	83.61	84.69	87.12	87.52	88.61	90.34	92.78	0.8070	0.8182	0.8301	0.8571
	ELM	91.34	91.33	88.22	92.56	95.07	95.10	93.52	96.31	0.9036	0.9035	0.8692	0.9172
0.4	NN	53.99	53.54	83.72	83.27	57.83	57.42	89.91	91.28	0.4958	0.4911	0.8197	0.8150
	SVM	77.52	79.24	84.79	90.08	85.98	85.48	90.52	94.37	0.7525	0.7692	0.8313	0.8897
	RF	79.03	80.42	84.27	86.47	83.62	85.28	90.01	92.54	0.7675	0.7829	0.8256	0.8500
	ELM	90.44	90.33	86.89	92.03	94.17	94.15	92.69	96.10	0.8936	0.8924	0.8545	0.9114
0.5	NN	44.86	44.55	82.89	80.79	47.17	47.00	88.53	89.22	0.3978	0.3945	0.8104	0.7879
	SVM	75.03	75.08	84.52	89.28	75.97	73.22	89.54	94.45	0.7206	0.7206	0.8281	0.8808
	RF	73.19	75.05	83.25	85.20	77.06	79.53	88.62	91.50	0.7034	0.7238	0.8143	0.8360
	ELM	89.39	89.03	86.39	91.55	93.26	93.10	91.70	95.67	0.8819	0.8779	0.8490	0.9061



(a) $\rho = 0.1$ (b) $\rho = 0.5$

Fig. 9. The classification maps of four different method (each row represents different methods) with four different classifiers (each column represents different classifiers) on the University of Pavia database when (a) $\rho = 0.1$ and (b) $\rho = 0.5$. From the first row to the last row: LNA, Bagging, iForest, and RLPA, from the first column to the last column: NN, SVM, RF, and ELM.

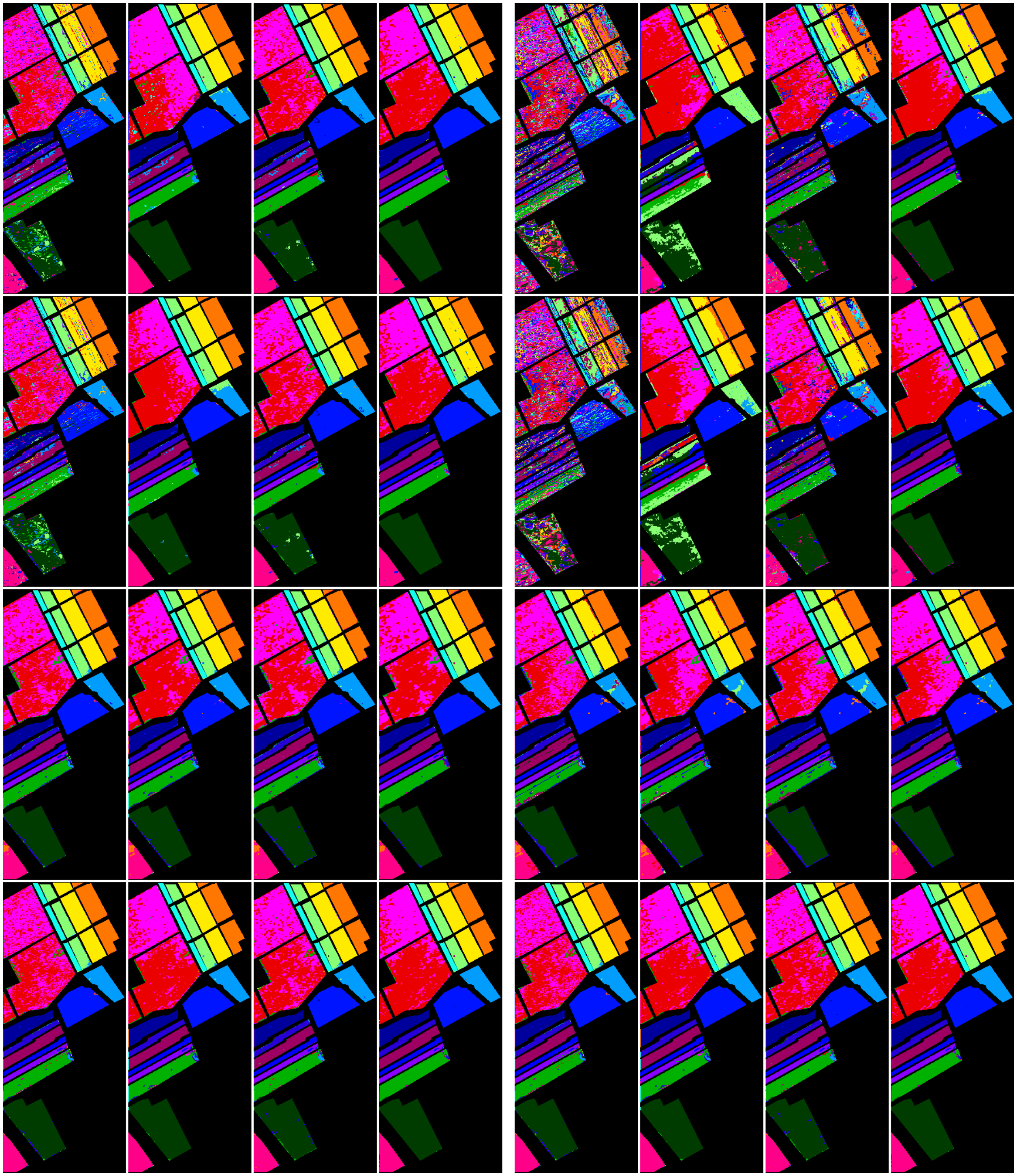
TABLE V
THE AVERAGE PERFORMANCE IN TERMS OF OA, AA, AND KAPPA OF NLA, BAGGING, IFOREST, AND RLPA.

Methods	OA [%]	AA [%]	Kappa
NLA	73.93	73.26	0.6951
Bagging	73.20	71.94	0.6865
iForest	77.09	78.04	0.7293
RLPA	83.11	82.84	0.7994

labels. Moreover, the strategy of re-sampling will result in decreasing the total amount of training samples, so that the classification performance may also be degraded, *e.g.*, the performance of Bagging is even worse than NLA.

- The performance of iForest (the cyan lines) is classifier and database dependent. Specifically, it performs well on the NN for all three databases, but it may be even worse than the NLA and Bagging methods. From the average result, iForest can gain more than three percentages when compare to NLA. It should be noted that as an anomaly detection algorithm, iForest has a bottleneck that it can only detect the noise samples but cannot cleanse its label.
- The proposed RLPA method (the red lines) can obtain better performance (especially when the noise level is large) than all comparison methods in almost all situations. The improvement also depends on the classifier, *e.g.*, the gain of RLPA over NLA can reach 10% for the NN and SVM classifiers and will reduce to 3% for the RF and ELM classifiers. Nevertheless, the gains in term of the average OA, AA, Kappa of our proposed RLPA method over the NLA are still very impressive, *e.g.*, 9.18%, 9.58%, and 0.1043.

To further demonstrate the classification results of different methods, in Fig. 8, Fig. 9, and Fig. 10, we show the visual results in term of the classification map on two noise levels ($\rho = 0.1$ and $\rho = 0.5$) for the three databases. For each

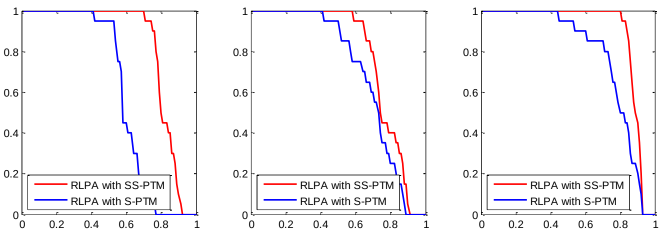


(a) $\rho = 0.1$

(b) $\rho = 0.5$

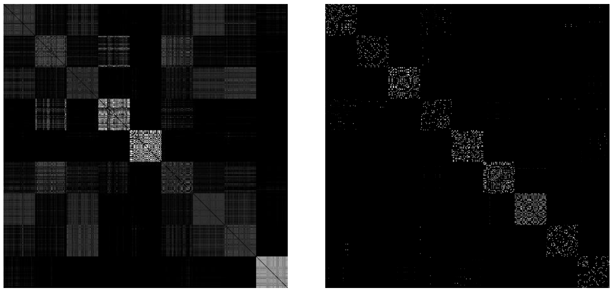
Fig. 10. The classification maps of four different methods (each row represents different methods) with four different classifiers (each column represents different classifiers) on the Salinas Scene database when (a) $\rho = 0.1$ and (b) $\rho = 0.5$. From the first row to the last row: LNA, Bagging, iForest, and RLPA, from the first column to the last column: NN, SVM, RF, and ELM.

subfigure, each row represents different methods and each column represents different classifiers. Specifically, from the first row to the last row: LNA, Bagging, iForest, and RLPA, from the first column to the last column: NN, SVM, RF, and ELM. When compared with LNA, Bagging, and iForest, the proposed RLPA with ELM classifier achieves the best performance. However, the classification maps of RLPA may result in a salt-and-pepper effect especially in the smooth regions, whose pixels should be the same class. This is mainly because that the RLPA is essentially a pixel-wise method, and the neighbor pixels may produce inconsistent classification results. To alleviate this problem, the approach of incorporating spatial constraint to fuse the classification result of RLPA can be expected to obtain satisfying results.



(a) Indian Pines (b) University of Pavia (c) Salinas

Fig. 11. Classification accuracy statistics using RLPA with/without spatial constraint on the three databases. The horizontal axis represents the OA scores, while the vertical axis marks the percentage of larger than the score marked on the horizontal axis.



(a) S-PTM (b) SS-PTM

Fig. 12. Visualizations of the probability transfer matrices of (a) S-PTM and (b) SS-PTM. Note that we rescale the intensity values of the matrix for observation.

C. Effectiveness of SSPTM

To verify the effectiveness of the proposed spectral-spatial probability transform matrix generation method, we compare it to the baseline that the similarity between two pixels in only calculated by their spectral difference. To compare the results of spectral-spatial probability transform matrix (SS-PTM) based method and spectral probability transform matrix based method (S-PTM), in Fig. 11, we report the statistical curves of OA scores of four comparison methods with four classifiers, *i.e.*, a vector containing 20 elements, whose values are the OA of different situations. It shows a considerable quantitative advantage of SS-PTM compared to S-PTM.

To further analysis the effectiveness of introducing the spatial constraint, in Fig. 12 we show the probability transform matrices with/without a spatial constraint. The two matrices are generated on the University of Pavia database, in which includes 9 classes and 50 training samples per class. From the results, we observe that SS-PTM is a sparse and highly diagonalization matrix, and S-PTM is a dense and non-diagonal matrix. That is to say, SS-PTM does make sense for recovering the hidden structure of data and guarantees the label propagation only within the same class. In contrast to S-PTM, which has many edges between samples with different labels (please refer to the non-diagonal blocks), it may wrongly propagate the label information.

D. Parameter Analysis

From the framework of RLPA, we learn that there are two parameters determining the performance of the proposed method: (i) the parameter η denoting the "clean" sample proportion in the total training samples, and (ii) the parameter α used to balance the contribution between the current label information and the label information received from its neighbors. In our study, we empirically set their values by grid search. Fig. 13 shows the influence of these two parameters on the classification performance in term of OA. It should be noted that we only give the average results of RLPA on the three databases with ELM classifier under $\rho = 0.3$. In fact, we can obtain similar conclusions under other situations. From the results, we observe that too small values of η or α may be inappropriate. This indicates that "clean" labeled samples play an important role in label propagation. If too few "clean" labeled samples are selected (η is small), the label information will be insufficient for the subsequent effective label propagation process. At the same time, as the value of η becomes larger, the performance also starts to deteriorate. This is mainly because that too large value of η will make the label propagation meaningless in the sense that very few samples need to absorb label information from its neighbors. In our experiments, we fix η to 0.7. Similarly, the value of α cannot be set too large or too small. A too small value of α implies that the final labels completely determined by the selected "clean" labeled samples. At the same time, a too large value of α will make it very difficult to absorb label information from the labeled samples. In our experiments, we fix α to 0.9.

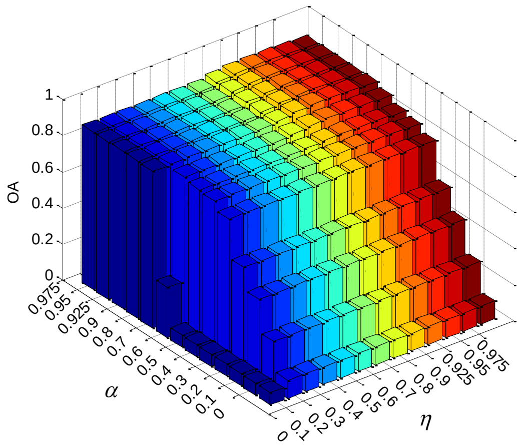


Fig. 13. The classification result in term of average OA on the three databases with ELM classifier under $\rho = 0.3$ according to different α and η, whose values vary from 0.1 to 0.975.

VI. Conclusion And Future Work

In this paper we study a very important but pervasive problem in practice—hyperspectral image classification in the presence of noisy labels. The existing classifiers assume, without exception, that the label of a sample is completely clean. However, due to the lack of information, the subjectivity of human judgment or human mistakes, label noise inevitably exists in the generated hyperspectral image data. Such noisy labels will mislead the classifier training and severely decrease the classification performance. Therefore, in this paper we develop a label noise cleansing algorithm based on the random label propagation algorithm (RLPA). RLPA can incorporate the spectral-spatial prior to guide the propagation process of label information. Extensive experiments on three public databases are presented to verify the effectiveness of our proposed approach, and the experimental results demonstrate much improvement over the approach of directly using the noisy samples.

In this paper, we simply use random noise to generate noisy labels. For all classes, they have the same percentage of samples with label noise. However, in real conditions label noise may be sample-dependent, class-dependent, or even adversarial. For example, when the mislabeled pixels come from the edge of the region or are similar to one another, such noise will be more difficult to handle. Therefore, how to deal with real label noise will be our future work.

References

[1] A. J. Brown, "Spectral curve fitting for automatic hyperspectral data analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1601–1608, 2006.

[2] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2013.

[3] J. Ma, J. Jiang, H. Zhou, J. Zhao, and X. Guo, "Guided locality preserving feature matching for remote sensing image registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4435–4447, 2018.

[4] X. Jia, B.-C. Kuo, and M. M. Crawford, "Feature mining for hyperspectral image classification," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 676–697, 2013.

[5] A. J. Brown, B. Sutter, and S. Dunagan, "The marte vnir imaging spectrometer experiment: Design and analysis," *Astrobiology*, vol. 8, no. 5, pp. 1001–1011, 2008.

[6] A. J. Brown, S. J. Hook, A. M. Baldridge, J. K. Crowley, N. T. Bridges, B. J. Thomson, G. M. Marion, C. R. de Souza Filho, and J. L. Bishop, "Hydrothermal formation of clay-carbonate alteration assemblages in the nili fossae region of mars," *Earth and Planetary Science Letters*, vol. 297, no. 1-2, pp. 174–182, 2010.

[7] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, 2018.

[8] J. Jiang, C. Chen, Y. Yu, X. Jiang, and J. Ma, "Spatial-aware collaborative representation for hyperspectral remote sensing image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 404–408, 2017.

[9] R. Ji, Y. Gao, R. Hong, Q. Liu, D. Tao, and X. Li, "Spectral-spatial constraint hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 3, pp. 1811–1824, 2014.

[10] X. Kang, S. Li, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, 2014.

[11] F. Tong, H. Tong, J. Jiang, and Y. Zhang, "Multiscale union regions adaptive sparse representation for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 9, 2017.

[12] D. A. Landgrebe, *Signal theory methods in multispectral remote sensing*. John Wiley & Sons, 2005, vol. 29.

[13] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2271–2282, 2010.

[14] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, 2005.

[15] J. Xia, P. Du, X. He, and J. Chanussot, "Hyperspectral remote sensing image classification based on rotation forest," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 1, pp. 239–243, 2014.

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

[17] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, 2011.

[18] Y. Gao, J. Ma, and A. L. Yuille, "Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2545–2560, 2017.

[19] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, 2015.

[20] A. Samat, P. Du, S. Liu, J. Li, and L. Cheng, "E^2LMs: Ensemble extreme learning machines for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1060–1069, 2014.

[21] B. Waske, S. van der Linden, J. A. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, 2010.

[22] C. Pelletier, S. Valero, J. Inglada, N. Champion, C. Marais Sicre, and G. Dedieu, "Effect of training class label noise on classification performances for land cover mapping with satellite image time series," *Remote Sensing*, vol. 9, no. 2, p. 173, 2017.

[23] H. Othman and S.-E. Qian, "Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 397–408, 2006.

[24] S. Prasad, W. Li, J. E. Fowler, and L. M. Bruce, "Information fusion in the redundant-wavelet-transform domain for noise-robust hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 9, pp. 3474–3486, 2012.

[25] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising employing a spectral–spatial adaptive total variation model," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3660–3677, 2012.

[26] C. Li, Y. Ma, J. Huang, X. Mei, and J. Ma, "Hyperspectral image denoising using the robust low-rank tensor recovery," *JOSA A*, vol. 32, no. 9, pp. 1604–1612, 2015.

[27] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2008, pp. 254–263.

[28] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 00, no. Apr, pp. 1297–1322, 2010.

[29] D. Angluin and P. Laird, "Learning from noisy examples," *Machine Learning*, vol. 2, no. 4, pp. 343–370, 1988.

[30] N. D. Lawrence and B. Schölkopf, "Estimating a kernel fisher discriminant in the presence of label noise," in *ICML*, vol. 1. Citeseer, 2001, pp. 306–313.

[31] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Advances in neural information processing systems*, 2013, pp. 1196–1204.

[32] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 38, no. 3, pp. 447–461, 2016.

[33] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2014.

[34] F. Condessa, J. Bioucas-Dias, and J. Kovačević, "Supervised hyperspectral image classification with rejection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 6, pp. 2321–2332, 2016.

[35] H. Pu, Z. Chen, B. Wang, and G. M. Jiang, "A novel spatial-spectral similarity measure for dimensionality reduction and classification of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7008–7022, Nov 2014.

[36] X. Zheng, Y. Yuan, and X. Lu, "Dimensionality reduction by spatial–spectral preservation in selected bands," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5185–5197, 2017.

[37] A. T. Kalai and R. A. Servedio, "Boosting in the presence of noise," *Journal of Computer and System Sciences*, vol. 71, no. 3, pp. 266–290, 2005.

[38] J. Li, H. Zhang, and L. Zhang, "Efficient superpixel-level multitask joint sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5338–5351, 2015.

[39] J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, and L. Wang, "SuperPCA: A superpixelwise principal component analysis approach for unsupervised feature extraction of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4581–4593, 2018.

[40] S. Zhang, S. Li, W. Fu, and L. Fang, "Multiscale superpixel-based sparse representation for hyperspectral image classification," *Remote Sensing*, vol. 9, no. 2, p. 139, 2017.

[41] F. Fan, Y. Ma, C. Li, X. Mei, J. Huang, and J. Ma, “Hyperspectral image denoising with superpixel segmentation and low-rank representation,” *Information Sciences*, vol. 397, pp. 48–68, 2017.

[42] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, “Entropy rate superpixel segmentation,” in *CVPR*. IEEE, 2011, pp. 2097–2104.

[43] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, p. 2274, 2012.

[44] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[45] Q. Wang, J. Lin, and Y. Yuan, “Salient band selection for hyperspectral image classification via manifold ranking,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1279–1289, June 2016.

[46] L. Fang, N. He, S. Li, P. Ghamisi, and J. A. Benediktsson, “Extinction profiles fusion for hyperspectral images classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1803–1815, 2018.

[47] J. Canny, “A computational approach to edge detection,” in *Readings in Computer Vision*. Elsevier, 1987, pp. 184–203.

[48] R. Kothari and V. Jain, “Learning from labeled and unlabeled data,” in *International Joint Conference on Neural Networks*, 2002, pp. 2803–2808.

[49] X. Zhu and Z. Ghahramani, “Learning from labeled and unlabeled data with label propagation,” 2002.

[50] Y. Freund, “Boosting a weak learning algorithm by majority,” *Information and computation*, vol. 121, no. 2, pp. 256–285, 1995.

[51] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[52] J. Abellán and A. R. Masegosa, “Bagging schemes on the presence of class noise in classification,” *Expert Systems with Applications*, vol. 39, no. 8, pp. 6827–6837, 2012.

[53] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation-based anomaly detection,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, p. 3, 2012.

[54] A. Krieger, C. Long, and A. Wyner, “Boosting noisy data,” in *ICML*, 2001, pp. 274–281.