

Article

# A Novel Analysis Dictionary Learning Model Based Hyperspectral Image Classification Method

Wei Wei <sup>1,2,\*</sup> , Mengting Ma <sup>1</sup>, Cong Wang <sup>1</sup>, Lei Zhang <sup>3,\*</sup>, Peng Zhang <sup>1</sup> and Yanning Zhang <sup>1,2</sup>

<sup>1</sup> School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China; mtma@mail.nwpu.edu.cn (M.M.); wangcong12@mail.nwpu.edu.cn (C.W.); zh0036ng@nwpu.edu.cn (P.Z.); ynzhang@nwpu.edu.cn (Y.Z.)

<sup>2</sup> The National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, Xi'an 710072, China

<sup>3</sup> School of Computer Science, The University of Adelaide, Adelaide 5000, Australia

\* Correspondence: weiweinwpu@nwpu.edu.cn (W.W.); nwpuzhanglei@gmail.com (L.Z.); Tel.: +86-137-7253-8134 (W.W.)

Received: 31 December 2018; Accepted: 14 February 2019; Published: 15 February 2019



**Abstract:** Supervised hyperspectral image (HSI) classification has been acknowledged as one of the fundamental tasks of hyperspectral data analysis. Witnessing the success of analysis dictionary learning (ADL)-based method in recent years, we propose an ADL-based supervised HSI classification method in this paper. In the proposed method, the dictionary is modeled considering both the characteristics within the spectrum and among the spectra. Specifically, to reduce the influence of strong nonlinearity within each spectrum on classification, we divide the spectrum into some segments, and based on this we propose HSI classification strategy. To preserve the relationships among spectra, similarities among pixels are introduced as constraints. Experimental results on several benchmark hyperspectral datasets demonstrate the effectiveness of the proposed method for HSI classification.

**Keywords:** dictionary learning; sparse representation; hyperspectral image classification; supervised method

## 1. Introduction

Hyperspectral imaging is a technology which simultaneously captures hundreds of images from a broad spectral range. The spectral information provides the hyperspectral image (HSI) with the ability to accurately analyze the image, which makes HSI widely applied in lots of remote sensing related tasks such as classification, anomaly detection, etc. [1–6].

Supervised HSI classification has been acknowledged as one of the fundamental tasks of HSI analysis [7–10], which aims to assign each pixel a pre-defined class label. It is commonly realized that supervised HSI classification method consists of a classifier and a feature extraction method. The classifier defines a strategy to identify the class labels of the test data. For example, by selecting  $k$  training samples which have the closest distance to the test sample,  $k$ -nearest neighbor ( $k$ -NN) method [11] assigns the test sample a label which dominates the selected  $k$  training samples. Support vector machine (SVM) [12,13] looks for a decision surface that linearly separates samples into two groups with a maximum margin. In addition, some advanced classifiers are proposed for HSI classification [14–18].

Feature extraction [19–21], in contrast to the classifier, is used to convert the spectrum of the pixel into a new representation space, where the generated features can be more discriminative than the spectrum. An ideal feature extraction method can generate features discriminative enough, for which

the classifier is unimportant, i.e., simple classifiers such as  $k$ -NN or SVM can also lead to a satisfied classification result with an ideal feature extraction method. Thus, researchers pay attention to this goal, and propose different feature extraction methods from different perspective [22,23], such as principal component analysis method [19], and sparse representation-based method [24], etc. Considering sparse representation has demonstrated its robustness and effectiveness for HSI classification [24–27], we focus on sparse representation-based method, and aim to propose a more effective method.

HSI data itself is not a sparse data. When we apply sparse representation method on HSI data, we need to convert HSI into a sparse data first, which is accomplished by introducing extra dictionary. According to the way in which dictionary is generated, sparse representation can be roughly divided into synthesis dictionary model-based methods [28] and analysis dictionary model-based ones [29].

For synthesis dictionary model-based methods, the dictionary  $\mathbf{D}$  and the sparse representation  $\mathbf{Y}$  is learned via

$$\begin{aligned} \min_{\mathbf{D}, \mathbf{Y}} \|\mathbf{X} - \mathbf{D}\mathbf{Y}\|_{\mathbf{F}}^2 \\ \text{s.t. } \mathbf{D} \in D \\ \|y_i\|_0 \leq T_0, i = 1, 2, \dots, n \end{aligned} \quad (1)$$

$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in R^{m_1 \times n}$  denotes a set of pixels, which includes  $n$  pixels and  $\mathbf{x}_i \in R^{m_1}$ .  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in R^{m_2 \times n}$  represents a set of  $m_2$ -dimensional sparse coefficients generated from  $\mathbf{X}$ .  $D$  is a set of constraints on  $\mathbf{D}$ .  $T_0$  controls the sparsity level of  $\mathbf{Y}$ . Some synthesis dictionary model-based methods are proposed. Sparse representation-based classification (SRC) [30] method directly uses the training samples as the dictionary. Label consistent  $k$ -singular value decomposition (LC-KSVD) algorithm [31,32] learns the dictionary as well as the sparse representation via KSVD method. To promote the discriminability of the generated sparse representation, fisher discrimination dictionary learning (FDDL) [33] is proposed by introducing an extra discriminative term. In addition, dictionary learning with structured incoherence (DL SI) method [34] promotes the discriminability by encouraging dictionaries associated with different classes.

Different from the synthesis dictionary model, the analysis dictionary model (ADL) is a newly proposed dictionary learning model, which is a dual model of the synthesis dictionary model. It models dictionary and sparse code as in [29]

$$\begin{aligned} \min_{\Omega, \mathbf{Y}} \|\mathbf{Y} - \Omega\mathbf{X}\|_{\mathbf{F}}^2 \\ \text{s.t. } \Omega \in \mathcal{W} \\ \|y_i\|_0 \leq T_0, i = 1, 2, \dots, n \end{aligned} \quad (2)$$

where  $\mathcal{W}$  is a set of constraints on the dictionary  $\Omega$ . Based on Formula (2), a discriminative analysis dictionary learning (DADL) [35] method was proposed specifically for classification. Though analysis dictionary model shows its power and efficiency for feature representation compared with synthesis dictionary model, to the best of our knowledge, it has not been used for HSI classification before, which drives us to propose a HSI classification method based on analysis dictionary model.

A new HSI oriented ADL model is proposed in this paper, which fully uses the characteristic of HSI data. First, to reduce the influence of nonlinearity within each spectrum on classification, we divide the spectrum the sensor captured into some segments. Second, we build analysis dictionary model for each segment, where the relationship of spectra is exploited to boost the discriminability of the generated codebook. Then, a voting strategy is used to obtain the final classification result. The main ideas and contributions are summarized as follows.

- (1) We introduce analysis dictionary model for supervised HSI classification, which is the first time analysis dictionary model used for HSI classification.
- (2) We propose an analysis dictionary model-based HSI classification framework. By modeling the characteristics of HSI within spectrum and among spectra, the proposed discriminative analysis dictionary model can generate better features for HSI classification.

- (3) Experimental results demonstrate the effectiveness of the proposed method for HSI classification, compared with other dictionary learning-based methods.

The remainder of this paper is structured as follows. Section 2 describes the proposed analysis dictionary model-based HSI classification method. Experimental results and analysis are provided in Section 3. Section 4 discusses the proposed method and Section 5 concludes the paper.

## 2. The Proposed Method

Denote a 3D HSI cube the sensor captured as  $\mathcal{H} \in \mathbb{R}^{r \times c \times m}$ , where  $r$  is row number,  $c$  is column number and  $m$  is band number. We extract all labeled pixels from  $\mathcal{H}$  and aggregate them as a set  $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n] \in \mathbb{R}^{m \times n}$ , where  $n$  is the pixel number. In the following, we give the framework of the proposed method first. We then introduce the details of the proposed method.

### 2.1. The Framework of the Proposed Method

We use the characteristics of HSI data to model a new HSI oriented ADL model in this paper. The entire flowchart is shown in Figure 1. Given an HSI, we divide the high-dimensional spectrum  $\mathbf{h}_i \in \mathbb{R}^m$  the sensor captured into multiple segments to reduce the influence of nonlinearity within each spectrum on classification. Second, we build analysis dictionary model for each segment, where the relationships among the spectra are exploited to boost the discriminability of the generated codebook. Then, a voting strategy is used to obtain the final classification results.

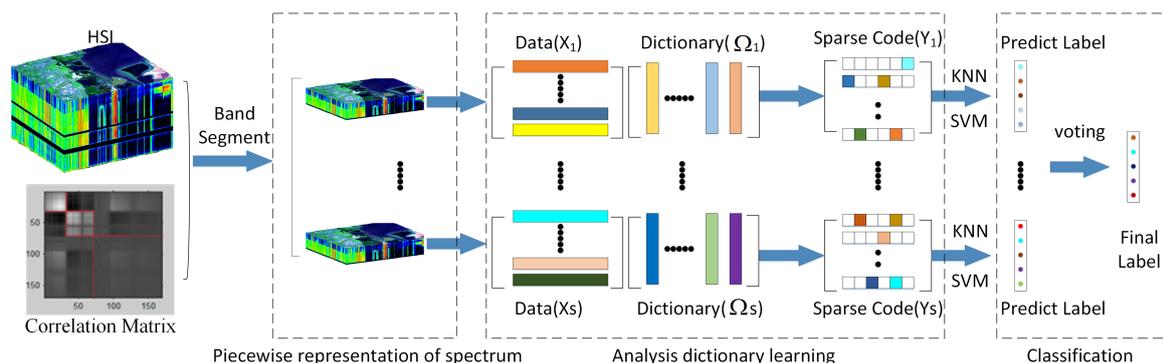


Figure 1. The proposed architecture.

### 2.2. Piecewise Representation of Spectrum

It is commonly realized that the difficulty of classification comes from class ambiguity, i.e., the sample variations come from within-class maybe larger than that from between-class. For HSI data, lots of factors will lead to class ambiguity, such as the nonlinearity of spectrum, pixel difference caused by different imaging conditions, etc. In this subsection, we pay our attention to nonlinearity of spectrum first.

Due to the nonlinearity of spectrum, directly model analysis dictionary on the entire spectrum  $\mathbf{h}_i \in \mathbb{R}^m$  the sensor captured is not a good choice, which also can be seen from the experimental results. Considering that piecewise linear representation [36] is a common strategy to deal with nonlinearity, we divide the high-dimensional spectrum  $\mathbf{h}_i$  into multiple segments first to address this problem. Then, we apply the analysis dictionary model for each segment independently.

Different methods can be used to divide the spectrum into segments. Considering correlation within spectrum shows obvious block-diagonal structure, it is used to segment the spectrum in this paper [37]. Specifically, given  $\mathbf{H}$ , we calculate the correlation matrix on spectral domain (i.e., the row direction of the matrix) as

$$Cor(i, j) = \frac{Cov(i, j)}{\sqrt{Cov(i, i) Cov(j, j)}}, \tag{3}$$

where  $Cor(i, j)$  is the correlation coefficient between the  $i$ -th band and the  $j$ -th band of  $\mathbf{H}$ . In Equation (3),  $Cov$  is the covariance matrix of  $\mathbf{H}$  and is calculated by

$$Cov = E \left[ (\mathbf{H} - E(\mathbf{H})) (\mathbf{H} - E(\mathbf{H}))^T \right]. \tag{4}$$

In Equation (4),  $E(\cdot)$  denotes the mathematical expectation. Figure 2 illustrates Indian Pines dataset and the generated correlation matrix obtained via Equation (3). In Figure 2, white color represents strong correlation while black color represents low correlation. More brighter, more correlated. It can be seen from Figure 2 that block-diagonal structure exists in the generated correlation matrix, which justify the rationality of dividing the entire spectrum  $\mathbf{h}_i$  into segments. To simplify the representation, we use  $\mathbf{x}_i \in \mathbb{R}^{m_1}$  to denote the generated segment in the following. It is noticeable that the correlation matrix is only used as an example to separate the spectrum. Other methods [38,39] can also be introduced to divide the spectrum into segments. However, this is not the focus of this paper.

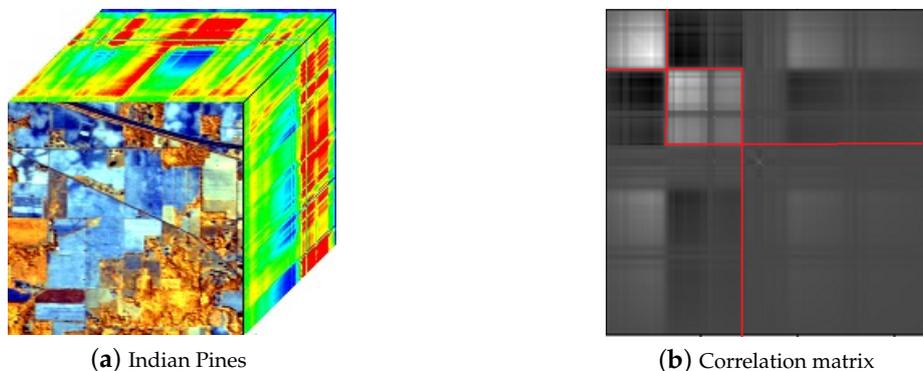


Figure 2. Indian Pines dataset (a) and the generated spectral correlation matrix (b).

### 2.3. Analysis Dictionary Learning Constrained with the Relationship of Spectra

By dividing the spectrum into segments, the nonlinearity problem of spectrum classification can be alleviated. We then construct analysis dictionary independently for each segment.

Equation (2) demonstrates a basic analysis dictionary learning method. Though it shows superiority over typical synthesis dictionary learning methods, it considers the spectrum individually without considering the relationship of spectra. However, such relationship is also an important characteristic of HSI. To take advantage of such characteristic, we propose a new analysis dictionary learning method inspired by discriminative analysis dictionary learning [35], which generate codebook with a triplet relation constraint. The constructed analysis dictionary model is given as follows.

$$\begin{aligned} \min_{\Omega, \mathbf{Y}} \sum_{i=1}^n dist(\mathbf{y}_i, \Omega \mathbf{x}_i) + \lambda_1 \sum_{i=1}^n dist(\mathbf{y}_i, \mathbf{z}_i) - \lambda_2 \sum_{i=1}^n \sum_{u=1}^n \sum_{v=1}^n \mathbf{T}'_i(u, v) [dist(\mathbf{y}_i, \mathbf{y}_u) - dist(\mathbf{y}_i, \mathbf{y}_v)] \\ + \lambda \sum_{i=1}^n \sum_{j=1}^n \mathbf{S}_{i,j} dist(\mathbf{y}_i, \mathbf{y}_j) \tag{5} \\ \text{s.t. } \Omega \in \mathcal{W} \\ \|\mathbf{y}_i\|_0 \leq T_0, \|\mathbf{y}_j\|_0 \leq T_0, \|\mathbf{y}_u\|_0 \leq T_0, \|\mathbf{y}_v\|_0 \leq T_0, \\ i = 1, 2, \dots, n, j = 1, 2, \dots, n, u = 1, 2, \dots, n, v = 1, 2, \dots, n \end{aligned}$$

In Formula (5),  $dist(\cdot)$  represents a kind of measure.  $\mathbf{z}_i$  is the target code, which can be label of spectrum  $\mathbf{h}_i$  or other equivalent representation of the label.  $\lambda_1, \lambda_2$  and  $\lambda$  are weighting coefficients which control the relative importance of different constraints. The minimization problem consists of the following four terms.

(1) The first term is the fidelity term. Minimizing it can guarantee the obtained sparse coefficient matrix  $\mathbf{Y}$  and the dictionary  $\Omega$  will reconstruct segments  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ .

(2) The second term is the discriminability promoting term [35], with which the label information  $\mathbf{z}_i$  can be introduced to generate discriminative sparse code  $\mathbf{y}_i$ . Minimizing the second term can enforce segments from the same category to have similar sparse codes.

(3) The third term is the triplet relation preserving term [35,40], which aims to preserve the local triplet topological structure of  $\mathbf{X}$  in the generated sparse representation  $\mathbf{Y}$ , i.e.,  $dist(\mathbf{y}_i, \mathbf{y}_u) \leq dist(\mathbf{y}_i, \mathbf{y}_v)$  if  $dist(\mathbf{x}_i, \mathbf{x}_u) \leq dist(\mathbf{x}_i, \mathbf{x}_v)$ . Ideal local topological structure preserving is to maximize  $\sum_{i=1}^n \sum_{u=1}^n \sum_{v=1}^n \mathbf{T}'_i(u, v) [dist(\mathbf{y}_i, \mathbf{y}_u) - dist(\mathbf{y}_i, \mathbf{y}_v)]$ , which equals to minimize  $-\sum_{i=1}^n \sum_{u=1}^n \sum_{v=1}^n \mathbf{T}'_i(u, v) [dist(\mathbf{y}_i, \mathbf{y}_u) - dist(\mathbf{y}_i, \mathbf{y}_v)]$  in Formula (5).  $\mathbf{T}'_i(u, v)$  is a supervised measure [35] which is defined as

$$\mathbf{T}'_i(u, v) = \begin{cases} -\mathbf{T}_i(u, v) \text{sign}[\mathbf{T}_i(u, v)] & , \mathbf{z}_i = \mathbf{z}_u \neq \mathbf{z}_v \\ \mathbf{T}_i(u, v) \text{sign}[\mathbf{T}_i(u, v)] & , \mathbf{z}_i = \mathbf{z}_v \neq \mathbf{z}_u \\ \mathbf{T}_i(u, v) & , \text{otherwise} \end{cases} \quad (6)$$

$\mathbf{T}_i(u, v)$  is the element in the  $u$ -th row and  $v$ -th column of matrix  $\mathbf{T}_i$ , which is calculated by  $dist(\mathbf{x}_i, \mathbf{x}_u) - dist(\mathbf{x}_i, \mathbf{x}_v)$ . The sign function  $\text{sign}(\cdot)$  is defined as

$$\text{sign}(a) = \begin{cases} -1, a < 0 \\ 0, a = 0 \\ +1, a > 0 \end{cases} \quad (7)$$

(4) The fourth term is a weighted sparsity preserving term, which guarantees the generated sparse representations similar enough if their corresponding segments are similar.  $\mathbf{S}_{i, j}$  measures the similarity between segments, which is defined as

$$\mathbf{S}_{i, j} = \frac{1}{1 + e^{(\text{SAD}(\mathbf{x}_i, \mathbf{x}_j))}}, \quad \text{SAD}(\mathbf{x}_i, \mathbf{x}_j) = \cos^{-1} \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \cdot \|\mathbf{x}_j\|_2} \quad (8)$$

It is noticeable that the third and fourth terms constrain the generated sparse representation from local structure perspective and pixel-pair perspective, which are mutual complemented. The effectiveness combing these two terms can be seen from the experimental results.

If we use a weight matrix  $\mathbf{W} \in \mathbb{R}^{n \times n}$  to replace  $\mathbf{T}'_i(u, v)$ , the local topological structure preserving term can be reformulated [35] as

$$\max_{\mathbf{Y}} \sum_{i=1}^n \sum_{u=1}^n \sum_{v=1}^n \mathbf{T}'_i(u, v) [dist(\mathbf{y}_i, \mathbf{y}_u) - dist(\mathbf{y}_i, \mathbf{y}_v)] = \min_{\mathbf{Y}} \sum_{i=1}^n \sum_{j=1}^n \mathbf{W}_{ij} dist(\mathbf{y}_i, \mathbf{y}_j), \quad (9)$$

where  $\mathbf{W}_{ij} = \sum_{k=1}^n \mathbf{T}'_i(k, j)$ . Then Equation (5) evolves to

$$\begin{aligned} \min_{\Omega, \mathbf{Y}} & \sum_{i=1}^n dist(\mathbf{y}_i, \Omega \mathbf{x}_i) + \lambda_1 \sum_{i=1}^n dist(\mathbf{y}_i, \mathbf{z}_i) + \lambda_2 \sum_{i=1}^n \sum_{j=1}^n \mathbf{W}_{i,j} dist(\mathbf{y}_i, \mathbf{y}_j) + \lambda \sum_{i=1}^n \sum_{j=1}^n \mathbf{S}_{i,j} dist(\mathbf{y}_i, \mathbf{y}_j) \\ \text{s.t. } & \Omega \in \mathcal{W} \\ & \|\mathbf{y}_i\|_0 \leq T_0, \|\mathbf{y}_j\|_0 \leq T_0, i = 1, 2, \dots, n, j = 1, 2, \dots, n \end{aligned} \quad (10)$$

By merging the last two terms in Equation (10), we obtain

$$\begin{aligned} \min_{\Omega, \mathbf{Y}} \sum_{i=1}^n \text{dist}(\mathbf{y}_i, \Omega \mathbf{x}_i) + \lambda_1 \sum_{i=1}^n \text{dist}(\mathbf{y}_i, \mathbf{z}_i) + \lambda_2 \sum_{i=1}^n \sum_{j=1}^n (\mathbf{W}_{i,j} + \rho \mathbf{S}_{i,j}) \text{dist}(\mathbf{y}_i, \mathbf{y}_j) \\ \text{s.t. } \Omega \in \mathcal{W} \\ \|\mathbf{y}_i\|_0 \leq T_0, \|\mathbf{y}_j\|_0 \leq T_0, i = 1, 2, \dots, n, j = 1, 2, \dots, n \end{aligned} \quad (11)$$

where  $\rho = \frac{\lambda_2}{\lambda}$ . Considering correntropy induced metric (CIM) [35,41] is a robust metric, it is adopted as the distance measure  $\text{dist}(\cdot)$  in this paper and calculate the distance between two given data  $\mathbf{y}_i$  and  $\mathbf{y}_j$  as

$$\text{dist}(\mathbf{y}_i, \mathbf{y}_j) = \left[ 1 - \exp\left(-\|\mathbf{y}_i - \mathbf{y}_j\|_2^2 / \sigma^2\right) \right]^{\frac{1}{2}} \quad (12)$$

By optimizing Equation (11), we can obtain the dictionary as well as the sparse representation generated from each segment, with which we can predict the classification result for each segment. However, Equation (11) is a non-convex problem, which is hard to be optimized directly. Instead, a half-quadratic technique proposed in [35] is introduced to optimize Equation (11) in this paper. Specifically, by introducing auxiliary matrices  $\mathbf{P}, \mathbf{Q}, \mathbf{R} \in R^{n \times n}$  into the optimization problem [35,42], Equation (11) can be solved by iteratively optimize  $\Omega, \mathbf{Y}$  and  $\mathbf{P}, \mathbf{Q}, \mathbf{R}$  until convergence. In the following, we only give the updating equation for these variables. We refer the readers to see [35] for the details of the optimization process.

**Step 1:** Fixing  $\mathbf{Y}, \mathbf{P}, \mathbf{Q}, \mathbf{R}$ , we update dictionary  $\Omega$  by

$$\Omega = \mathbf{Y} \mathbf{P}^t \mathbf{X}^T \left[ \mathbf{X} (\mathbf{P}^t + \lambda_2 \mathbf{L}^t) \mathbf{X}^T + \lambda_3 \mathbf{I} \right]^{-1}, \quad (13)$$

where  $t$  is the iteration number,  $\lambda_3$  is a Lagrange multiplier for  $\Omega$ , and  $\mathbf{L}$  is the Laplacian matrix of  $\mathbf{W} + \rho \mathbf{S}$ .

**Step 2:** Fixing  $\Omega, \mathbf{P}, \mathbf{Q}, \mathbf{R}$ , we update  $\mathbf{Y}$  via

$$\begin{aligned} \min_{\mathbf{y}_i} \left\| \mathbf{y}_i - \frac{\mathbf{P}_{ii}^t \Omega \mathbf{x}_i + \lambda_1 \mathbf{Q}_{ii}^t \mathbf{z}_i}{\mathbf{P}_{ii}^t + \lambda_1 \mathbf{Q}_{ii}^t} \right\|_2^2 \\ \text{s.t. } \|\mathbf{y}_i\|_0 \leq T_0 \end{aligned} \quad (14)$$

which can be solved easily by applying hard thresholding operation.

**Step 3:** Fixing  $\Omega$  and  $\mathbf{Y}$ , the auxiliary matrices  $\mathbf{P}, \mathbf{Q}, \mathbf{R}$  are updated via

$$\begin{aligned} \mathbf{P}_{ii}^{t+1} &= \exp\left(-\frac{\|\mathbf{y}_i^{t+1} - \Omega^{t+1} \mathbf{x}_i\|_2^2}{\sigma^2}\right), \\ \mathbf{Q}_{ii}^{t+1} &= \exp\left(-\frac{\|\mathbf{y}_i^{t+1} - \mathbf{z}_i\|_2^2}{\sigma^2}\right), \\ \mathbf{R}_{ij}^{t+1} &= \exp\left(-\frac{\|\Omega^{t+1} \mathbf{x}_i - \Omega^{t+1} \mathbf{x}_j\|_2^2}{\sigma^2}\right) \end{aligned} \quad (15)$$

#### 2.4. Classification via Different Segments

Once we obtain the sparse representation of each segment, i.e.,  $\mathbf{y}_i$ , we then use it to predict the class label for each segment. To discriminate the class label of the entire spectrum (i.e., pixel), we denote the label of the segment as seg-label in this paper. Any kind of classifier can be adopted to predict the seg-label for each segment. Considering that the proposed method aims to generate discriminative feature, simple classifiers including  $k$ -NN and SVM are adopted only in this paper.

Suppose we divide the entire spectrum of one pixel into  $S$  segments, we obtain  $S$  seg-labels with the adopted classifier. Denote these seg-labels as  $l_1, l_2, \dots, l_S$ , where  $l_i$  is the classification result from the  $i$ -th segment, we then predict the class label  $l_{final}$  for the pixel based on

$$l_{final} = \text{vote}(l_1, l_2, \dots, l_S). \quad (16)$$

$\text{vote}(\cdot)$  is a voting function. It selects the class that appears most frequently for the test pixel.

In this paper, we divide the spectrum into three segments for simplification. We adopt a simple voting strategy. If at least two seg-labels are same, we assign pixel the same class with the one dominate the seg-labels. Otherwise, the seg-labels are different for three segments. In this case, among three seg-labels, we randomly assign a seg-label to the class of pixel.

### 3. Experiments

We conduct experiments on HSI datasets to demonstrate the effectiveness of the proposed method. In the following, we first introduce the HSI datasets we used in the experiments. We then compare the proposed method with some state-of-the-art dictionary-based methods. Finally, we discuss the performance of the proposed method varied with different settings for HSI classification.

#### 3.1. Dataset Description

Three benchmark HSI datasets including Indian Pines dataset, Pavia University (PaviaU) dataset and Salinas Scene dataset are adopted to verify the proposed method [43,44].

**Indian Pines Dataset:** Indian Pines dataset was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in north-western Indiana, USA. The spectral range of Indian Pines is from 400 to 2450 nm. We remove 20 water absorption bands and use the remaining 200 bands for experiments. The imaged scene has  $145 \times 145$  pixels, among which 10,249 pixels are labeled. Class number of Indian Pines dataset is 16.

**PaviaU Dataset:** PaviaU dataset was acquired in Pavia University, Italy, by the Reflective Optics System Imaging Spectrometer. The spatial resolution of PaviaU dataset is 1.3 m, while the spectral range is from 430 to 860 nm. After removing 12 water absorption bands, we keep 103 bands from the original 115 bands for experiment. The imaged scene has  $610 \times 340$  pixels, among which 42,776 pixels are labeled. Class number of PaviaU dataset is 9.

**Salinas Scene Dataset:** The Salinas scene dataset was collected in Salinas Valley, California, which has a continuous spectral coverage from 400 to 2450 nm. There are  $512 \times 217$  pixels, among which 54,129 pixels were labeled and used for the experiment. After removing the water absorption bands, we keep the remaining 204 bands in the experiments. Class number of Salinas dataset is 16.

#### 3.2. Comparison Methods and Experimental Setup

We denote the proposed method as **Ours** in this paper. Since the proposed method is a dictionary learning-based HSI classification method, we mainly compare the proposed method with the existing dictionary learning-based methods. To further testify the performance of the proposed method, we compare the proposed method with a state-of-the-art deep learning-based method, i.e., 3D convolutional neural network (**3D-CNN**) [17], and the method based on the spectrum  $\mathbf{h}_i$  without feature extraction, which is denoted as **Ori** in this paper. In addition, since both piecewise representation and spectra relationship contribute to the final classification result for the proposed method, we implement two special versions of **Ours**, termed **Ours-Seg** and **Ours-Sim**, to verify the influence of these two parts on classification. **Ours-Seg** only considers the piecewise representation of spectrum whereas **Ours-Sim** only exploits the relationship of spectra for classification.

The dictionary learning-based methods we compared including sparse representation-based classification (**SRC**) [30], dictionary learning with structured incoherence (**DLSI**) [34], label consistent  $k$ -singular value decomposition algorithm (**LC-KSVD**) [31], fisher judgement dictionary learning

method (**FDDL**) [33], and discriminative analysis dictionary learning (**DADL**). **SRC**, **DLSI**, **LC-KSVD** and **FDDL** are synthesis dictionary model-based methods, whereas **DADL** and **Ours** are analysis dictionary model-based ones. In **SRC**, segmented spectrum are chosen as the dictionary directly, while learned dictionaries are used for **DLSI**, **LC-KSVD**, **FDDL**, **DADL** and **Ours**.

We normalize the HSI data into the range of 0 to 1 via a min-max normalization method. Except **3D-CNN**, which is an end-to-end classification method without dividing feature extraction and classifier, both  $k$ -NN and SVM are adopted for all other methods in the experiments to testify whether the proposed method is applicable to different classifiers. All codes of the comparing methods are implemented by the authors with tuned parameters for best performance. For the proposed method,  $\lambda_1, \lambda_2$  and  $\rho$  are optimized by cross-validation, which are set to 1, 1 and 0.05, respectively.

For all datasets, we empirically set the number of segments as 3 in the experiments. For Indian Pine dataset, the generated three segments are bands 1–30, bands 30–75, and bands 75–200. For PaviaU dataset, the generated three segments are bands 1–73, bands 73–75 and bands 75–103. While for Salinas dataset, the generated three segments are bands 1–40, bands 40–80 and bands 80–204.

Overall accuracy (OA), which defines the ratio of correctly labeled samples to all test samples, is adopted to measure HSI classification results.

### 3.3. Comparison with Other Methods

In this section, two experiments are conducted. First, we choose 20% samples from each class as the training set, based on which we then predict the class label of the test pixel for all methods. Second, we compare the performance of all methods with different amount of training samples.

#### Experimental Results with 20% Training Samples

The number of training and test samples for each dataset is given in Table 1, where 20% pixels are randomly sampled from all labeled data for training. Tables 2–7 report the average classification results for all methods across 10 rounds of different sampling, from which we can obtain the following conclusions.

(1) Compared with the synthesis dictionary model-based methods, analysis dictionary model-based methods including **DADL**, **Ours**, **Ours-Seg** and **Ours-Sim** can obtain higher classification results, which demonstrate the effectiveness of analysis dictionary model-based methods for HSI classification.

(2) Compared with  $k$ -NN, the SVM classifier can obtain better classification results with the same feature, since  $k$ -NN is a simple classifier without training while SVM tune its parameters with training data. **Ours** with  $k$ -NN classifier has better classification results, compared with all synthesis dictionary model-based methods with SVM classifier. For example, on Indian Pines dataset, the classification accuracy of **Ours** is 87.98% when using  $k$ -NN classifier, whereas the highest classification of synthesis dictionary model-based methods (i.e., **FDDL**) is only 72.98% even given SVM classifier.

(3) Compared with **DADL** which only uses the local triplet topology, the classification performance of the proposed method increases a lot. For example, the classification accuracy for **Ours** and **DADL** with  $k$ -NN classifier is 87.98% and 72.5%, respectively. The improvement of **Ours** over **DADL** comes from the fact that we simultaneously model the piecewise representation and the pixel similarity. The conclusion can also be seen from the experimental results of **Ours-Seg**, **Ours-Sim**, **Ours** and **DADL**. By comparing **Ours-Seg** and **DADL**, we can find that the classification results can be increased when we divide the spectrum into segments. By comparing **Ours-Sim** and **DADL**, we can find that the classification results also can be improved when we model pixel similarity into dictionary learning. Though **Ours-Seg** and **Ours-Sim** can obtain better classification results compared with **DADL**, they still inferior to **Ours** regarding HSI classification ability, which demonstrates that both piecewise representation and spectra relationship is important for the proposed method.

(4) Compared with **Ori** which is based on the spectrum directly, **Ours** has better classification results, which demonstrates the effectiveness of the proposed method. More importantly, the classification performance of **Ours** is more stable on all datasets, compared with **Ori**. For example,

with  $k$ -NN classifier, though the classification accuracy of **Ours** (86.29%) closes to that of **Ours** (88.38%) on Salinas dataset, there is a large difference between **Ours** (65.08%) and **Ours** (87.98%) on Indian Pines dataset.

(5) Compared with **3D-CNN**, the accuracy of **Ours** is lower when using together with  $k$ -NN classifier. However, when using this together with the SVM classifier, **Ours** can obtain better classification results compared with **3D-CNN**. This is because  $k$ -NN is a simple classifier without training while SVM and **3D-CNN** tune their parameters with training data. Thus, the performance of  $k$ -NN inferences to SVM and **3D-CNN**. In addition, since **3D-CNN** has large amount of parameters, its performance relies on large amount of training data. However, when given small amount of (e.g., 20%) training data as the propose method demands, **3D-CNN** cannot be well trained, with which the classification accuracy of **3D-CNN** inferences to **Ours** with SVM classifier.

**Table 1.** Training and test numbers for three datasets (Indian Pines, PaviaU, Salinas) used in this paper.

No.	Indiana Pines			PaviaU			Salinas		
	Class Name	Train	Test	Class Name	Train	Test	Class Name	Train	Test
1	Asphalt	1327	5304	Alfalfa	9	37	Brocoli_1	402	1607
2	Meadows	3730	14,919	Corn-notill	286	1142	Brocoli_2	745	2981
3	Gravel	420	1679	Corn-mintill	166	664	Fallow	395	1581
4	Trees	613	2451	Corn	47	190	Fallow_plow	279	1115
5	Sheets	269	1076	Grass-pasture	97	386	Fallow_smooth	536	2142
6	Bare Soil	1006	4023	Grass-trees	146	584	Stubble	792	3167
7	Bitumen	266	1064	Grass-pasture-mowed	6	22	Celery	716	2863
8	Bricks	735	2947	Hay-windrowed	96	382	Grapes	2254	9017
9	Shadows	189	758	Oats	4	16	Soil_vinyard	1241	4962
10				Soybean-notill	193	779	Corn_weeds	656	2622
11				Soybean-minill	491	1964	Lettuce_4wk	214	854
12				Soybean-clean	119	474	Lettuce_5wk	385	1542
13				Wheats	41	164	Lettuce_6wk	183	733
14				Woods	252	1013	Lettuce_7wk	214	856
15				Bulidings-Grass	77	309	Vinyard_un.	1454	5814
16				Stone-Steel-Towers	19	74	Vinyard_ve.	361	1446
Sum		8555	34,221		2049	8200		10,827	43,302

**Table 2.** Classification accuracy (%) of different methods on the Indian Pines dataset via  $k$ -NN classifier. The highest accuracy in each row is boldfaced.

No.	SRC	DLSI	LC-KSVD	FDDL	DADL	Ori	3D-CNN	Ours_Seg	Ours_Sim	Ours
1	23.16	22.22	19.66	14.52	24.70	35.48	<b>70.15</b>	51.41	30.22	61.26
2	82.50	90.96	93.86	<b>96.93</b>	88.25	53.17	96.12	93.19	92.94	95.28
3	92.92	90.96	93.86	94.13	96.31	57.61	93.19	95.91	<b>96.92</b>	96.12
4	64.76	68.70	61.08	66.12	84.40	83.78	94.11	86.10	73.24	<b>94.95</b>
5	72.47	94.52	94.33	85.21	93.12	90.45	96.12	<b>97.18</b>	92.60	95.97
6	95.84	<b>99.59</b>	95.89	94.10	94.02	93.77	97.13	97.81	97.11	96.96
7	17.04	14.20	13.30	14.87	22.18	<b>84.61</b>	60.14	43.10	60.15	57.13
8	85.92	88.02	90.67	92.10	90.19	<b>99.64</b>	90.17	97.28	97.45	95.37
9	31.10	30.42	28.06	30.18	14.17	20.00	<b>80.12</b>	32.27	16.37	51.00
10	86.21	90.74	86.17	88.03	92.10	74.35	90.12	96.22	95.10	<b>96.07</b>
11	79.23	79.71	74.54	77.20	84.11	51.49	<b>96.36</b>	94.21	86.79	96.24
12	96.67	95.95	96.50	75.24	97.62	52.92	<b>98.43</b>	97.30	97.20	96.23
13	68.95	62.89	55.40	73.94	72.12	80.00	<b>93.67</b>	84.16	70.72	90.11
14	81.34	85.30	85.30	88.54	71.92	81.97	<b>93.17</b>	86.86	69.72	91.51
15	83.10	80.85	81.48	86.17	86.22	45.16	<b>98.36</b>	88.99	88.24	97.64
16	39.89	40.08	32.17	35.59	52.46	83.33	<b>98.31</b>	68.10	51.21	87.11
OA	57.15	60.20	55.04	71.55	72.5	65.08	<b>90.75</b>	86.56	74.06	87.98

**Table 3.** Classification accuracy (%) of different methods on the Indian Pines dataset via SVM classifier. The highest accuracy in each row is boldfaced.

No.	SRC	DLSI	LC-KSVD	FDDL	DADL	Ori	3D-CNN	Ours_Seg	Ours_Sim	Ours
1	20.26	21.32	18.46	15.54	25.70	<b>80.66</b>	70.15	53.49	32.62	62.16
2	84.10	89.26	92.16	97.83	89.35	78.75	96.12	94.89	90.90	<b>97.48</b>
3	93.61	87.26	92.46	97.83	95.20	82.70	93.19	<b>98.96</b>	98.90	98.52
4	65.83	70.27	60.98	67.32	86.40	94.59	94.11	89.10	75.24	<b>95.95</b>
5	73.67	93.51	95.33	89.61	94.52	94.70	96.12	<b>99.38</b>	93.60	97.97
6	96.44	98.19	93.89	97.40	97.12	96.04	97.13	<b>99.86</b>	98.91	98.36
7	13.54	19.40	15.30	15.97	18.18	61.54	<b>60.14</b>	45.13	20.60	52.83
8	88.52	85.12	87.27	92.28	91.39	98.56	90.17	<b>99.38</b>	98.15	96.37
9	28.00	35.32	29.06	33.98	12.27	<b>100.00</b>	80.12	30.77	15.87	50.00
10	89.81	87.34	86.17	86.23	93.20	82.90	90.12	97.42	92.90	<b>99.07</b>
11	78.53	81.71	73.14	78.70	85.21	72.73	96.36	95.23	87.99	<b>97.84</b>
12	98.17	91.95	93.50	76.74	98.67	91.35	98.43	99.50	98.50	<b>99.83</b>
13	61.75	68.89	54.20	74.54	71.92	<b>100.00</b>	93.67	86.86	69.72	91.51
14	82.14	83.37	82.13	88.54	71.92	90.42	<b>93.17</b>	86.86	69.72	91.51
15	82.80	83.15	81.28	88.12	87.92	82.80	98.31	90.39	89.14	<b>99.74</b>
16	36.80	45.08	36.27	8.59	50.46	<b>89.74</b>	87.23	69.00	50.81	86.91
OA	65.22	67.35	60.46	72.98	74.92	82.72	90.75	92.14	77.97	<b>94.86</b>

**Table 4.** Classification accuracy (%) of different methods on the PaviaU dataset via *k*-NN classifier. The highest accuracy in each row is boldfaced.

No.	SRC	DLSI	LC-KSVD	FDDL	DADL	Ori	3D-CNN	Ours_Seg	Ours_Sim	Ours
1	95.29	95.99	95.23	95.11	97.02	71.50	<b>98.32</b>	97.04	97.91	97.56
2	87.79	89.53	87.15	89.07	90.01	76.39	<b>96.02</b>	92.47	92.31	95.65
3	82.86	86.20	82.67	83.99	85.29	78.79	93.18	88.94	87.42	<b>93.29</b>
4	92.06	93.24	89.38	91.10	92.90	94.80	95.12	94.77	94.48	<b>97.39</b>
5	69.87	74.02	69.37	71.77	75.69	<b>99.13</b>	86.79	80.49	78.20	85.45
6	99.80	98.98	98.43	98.65	99.60	75.15	97.17	99.56	<b>99.92</b>	99.56
7	69.56	73.27	71.23	72.40	76.00	<b>91.50</b>	88.12	79.64	78.24	86.64
8	96.64	94.92	94.21	96.36	96.92	78.46	<b>98.27</b>	96.59	96.96	97.12
9	62.43	63.77	59.33	63.04	64.81	<b>99.87</b>	85.89	72.90	70.56	81.21
OA	79.23	81.37	78.04	80.25	81.51	78.53	<b>94.16</b>	88.40	84.12	90.52

**Table 5.** Classification accuracy (%) of different methods on the PaviaU dataset via SVM classifier. The highest accuracy in each row is boldfaced.

No.	SRC	DLSI	LC-KSVD	FDDL	DADL	Ori	3D-CNN	Ours_Seg	Ours_Sim	Ours
1	96.09	92.19	92.13	91.91	96.12	83.38	98.32	96.24	98.91	<b>98.96</b>
2	88.39	87.23	90.35	92.27	92.21	93.25	96.02	90.17	93.41	<b>96.15</b>
3	83.66	93.70	84.27	85.19	86.89	84.04	93.18	90.14	90.72	<b>94.69</b>
4	90.06	90.64	82.18	93.40	91.40	<b>96.44</b>	95.12	95.17	95.28	95.09
5	74.27	82.42	73.87	78.17	78.19	<b>99.39</b>	86.79	82.99	80.20	88.15
6	95.80	95.18	94.73	95.61	97.20	92.13	97.17	96.96	<b>98.12</b>	97.24
7	76.16	79.97	79.53	76.10	79.20	<b>94.86</b>	88.12	80.84	79.74	89.14
8	94.64	91.92	90.21	93.16	95.12	87.24	98.27	97.29	95.46	<b>98.92</b>
9	67.43	70.77	69.33	73.04	70.81	<b>87.19</b>	85.89	75.78	73.26	86.11
OA	88.60	90.20	85.34	89.50	90.37	91.19	94.16	92.76	93.80	<b>95.21</b>

**Table 6.** Classification accuracy (%) of different methods on the Salinas dataset via  $k$ -NN classifier. The highest accuracy in each row is boldfaced.

No.	SRC	DLSI	LC-KSVD	FDDL	DADL	Ori	3D-CNN	Ours_Seg	Ours_Sim	Ours
1	91.79	99.00	98.63	98.20	98.38	97.90	99.75	<b>100.00</b>	99.75	<b>100.00</b>
2	99.06	99.63	99.26	99.53	99.53	99.46	<b>100.00</b>	99.90	99.77	99.93
3	99.24	99.43	98.48	98.99	97.03	98.99	<b>99.94</b>	98.61	98.23	98.55
4	99.46	98.48	97.49	98.30	98.21	99.75	99.55	99.82	99.46	<b>99.91</b>
5	99.07	98.55	98.04	98.41	96.36	96.21	99.11	99.40	99.21	<b>99.44</b>
6	99.84	99.62	99.27	99.53	99.72	99.63	99.84	<b>100.00</b>	99.87	<b>100.00</b>
7	99.69	99.58	99.20	99.48	99.51	98.85	<b>100.00</b>	<b>100.00</b>	99.86	<b>100.00</b>
8	<b>96.13</b>	66.16	66.04	66.13	76.08	64.48	65.63	84.56	84.52	84.57
9	<b>100.00</b>	99.62	99.40	99.56	99.52	96.64	99.86	99.64	99.56	99.66
10	77.27	99.21	98.77	99.09	99.21	90.25	99.68	99.25	99.09	<b>99.29</b>
11	97.42	98.36	97.07	98.01	96.84	94.24	<b>99.77</b>	92.51	92.04	92.62
12	15.24	99.35	98.64	99.16	98.51	99.94	<b>100.00</b>	99.94	<b>100.00</b>	<b>100.00</b>
13	98.36	95.08	94.82	95.91	97.82	96.65	97.95	99.59	99.68	<b>99.73</b>
14	98.25	98.48	97.20	98.13	96.73	93.45	<b>99.88</b>	99.42	99.95	99.53
15	11.01	67.34	67.15	67.29	66.53	69.45	67.54	70.33	70.26	<b>70.61</b>
16	78.91	98.69	97.86	98.41	98.55	98.13	99.38	<b>100.00</b>	99.72	<b>100.00</b>
OA	72.27	81.67	71.95	79.17	80.28	86.29	<b>93.98</b>	87.97	83.92	88.38

**Table 7.** Classification accuracy (%) of different methods on the Salinas dataset via SVM classifier. The highest accuracy in each row is boldfaced.

No.	SRC	DLSI	LC-KSVD	FDDL	DADL	Ori	3D-CNN	Ours_Seg	Ours_Sim	Ours
1	98.50	99.44	98.63	99.26	98.94	99.39	99.75	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
2	99.29	99.87	99.43	99.76	99.83	99.71	<b>100.00</b>	99.97	99.97	<b>100.00</b>
3	99.68	99.87	98.92	99.43	97.60	99.27	<b>99.94</b>	99.81	98.61	99.87
4	<b>100.00</b>	99.10	98.03	98.92	99.01	99.50	99.55	99.73	<b>100.00</b>	<b>100.00</b>
5	99.39	98.88	98.27	98.74	96.78	96.85	99.11	99.44	99.49	<b>99.63</b>
6	<b>100.00</b>	99.84	99.43	99.75	<b>100.00</b>	99.73	99.84	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
7	99.93	99.83	99.37	99.72	99.83	99.62	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
8	<b>96.21</b>	66.24	66.10	66.21	76.18	82.38	65.63	86.02	87.91	84.62
9	99.78	99.76	99.50	99.70	99.70	98.23	99.86	<b>99.90</b>	99.68	99.74
10	78.55	99.48	98.97	99.37	99.56	93.40	99.68	<b>99.84</b>	99.33	99.44
11	98.24	99.18	97.66	98.83	97.89	98.15	<b>99.77</b>	95.08	92.74	93.09
12	15.70	99.81	98.96	99.61	99.09	99.77	<b>100.00</b>	98.51	<b>100.00</b>	<b>100.00</b>
13	99.32	97.27	95.50	96.86	99.05	99.44	97.95	97.81	99.86	<b>100.00</b>
14	99.07	99.30	97.78	98.95	97.78	98.74	99.88	96.73	99.56	<b>100.00</b>
15	11.13	67.46	67.23	67.30	66.68	71.28	67.54	75.13	70.43	<b>70.61</b>
16	79.39	99.10	98.20	98.89	99.17	99.07	99.38	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
OA	79.48	88.10	76.23	86.17	89.80	91.20	93.98	92.59	90.45	<b>94.91</b>

Figures 3–8 illustrate the classification maps, where (a) represents the ground truth and (b)–(k) represent the results from different methods. In the classification map, we use a unique color to represent each category. From these figures, we can see that the proposed method with SVM classifier obtains more accurate and smoother results compared with the competing methods.

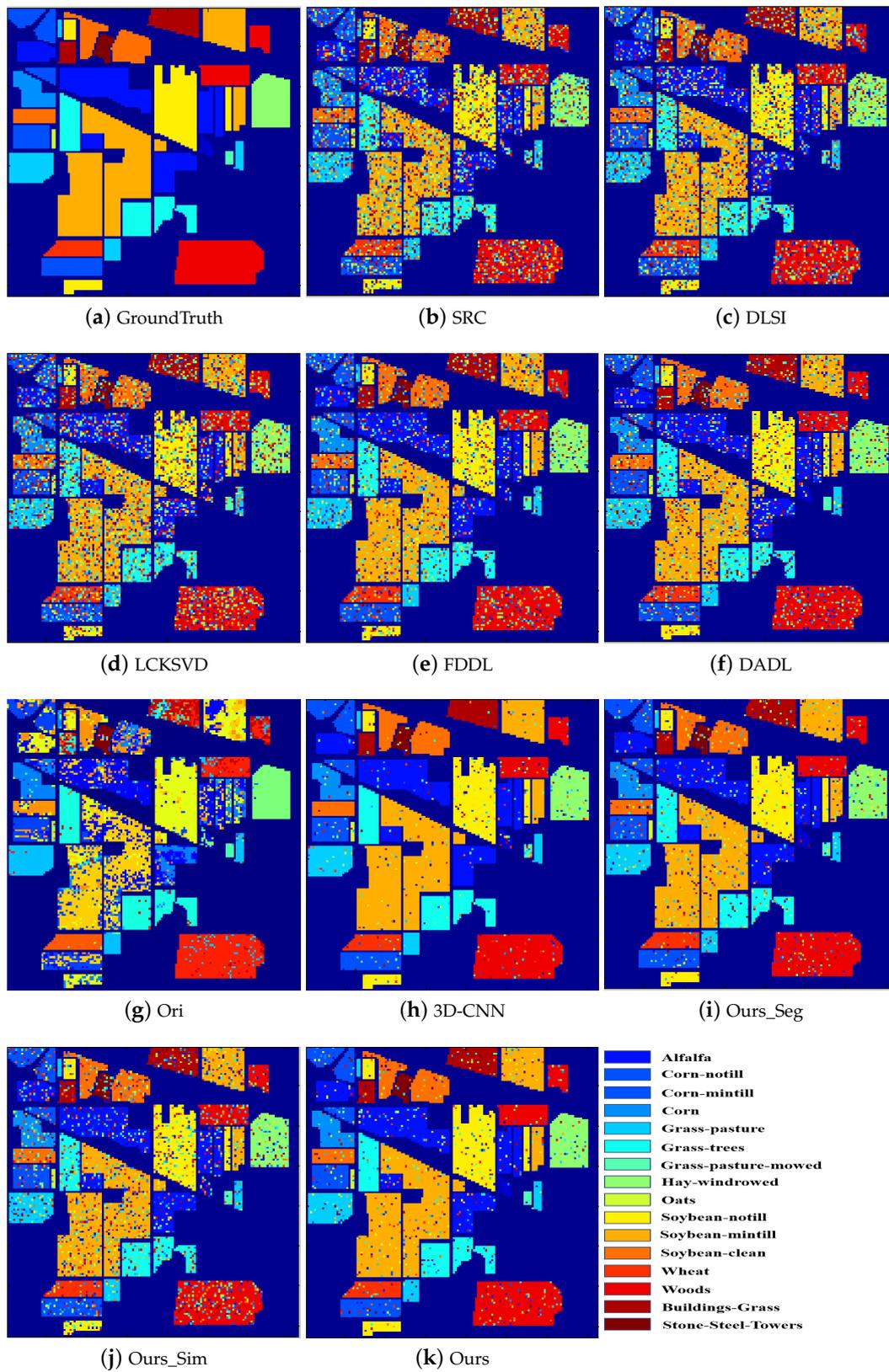


Figure 3. Classification maps of different methods on the Indian Pines dataset via  $k$ -NN classifier (a–k).

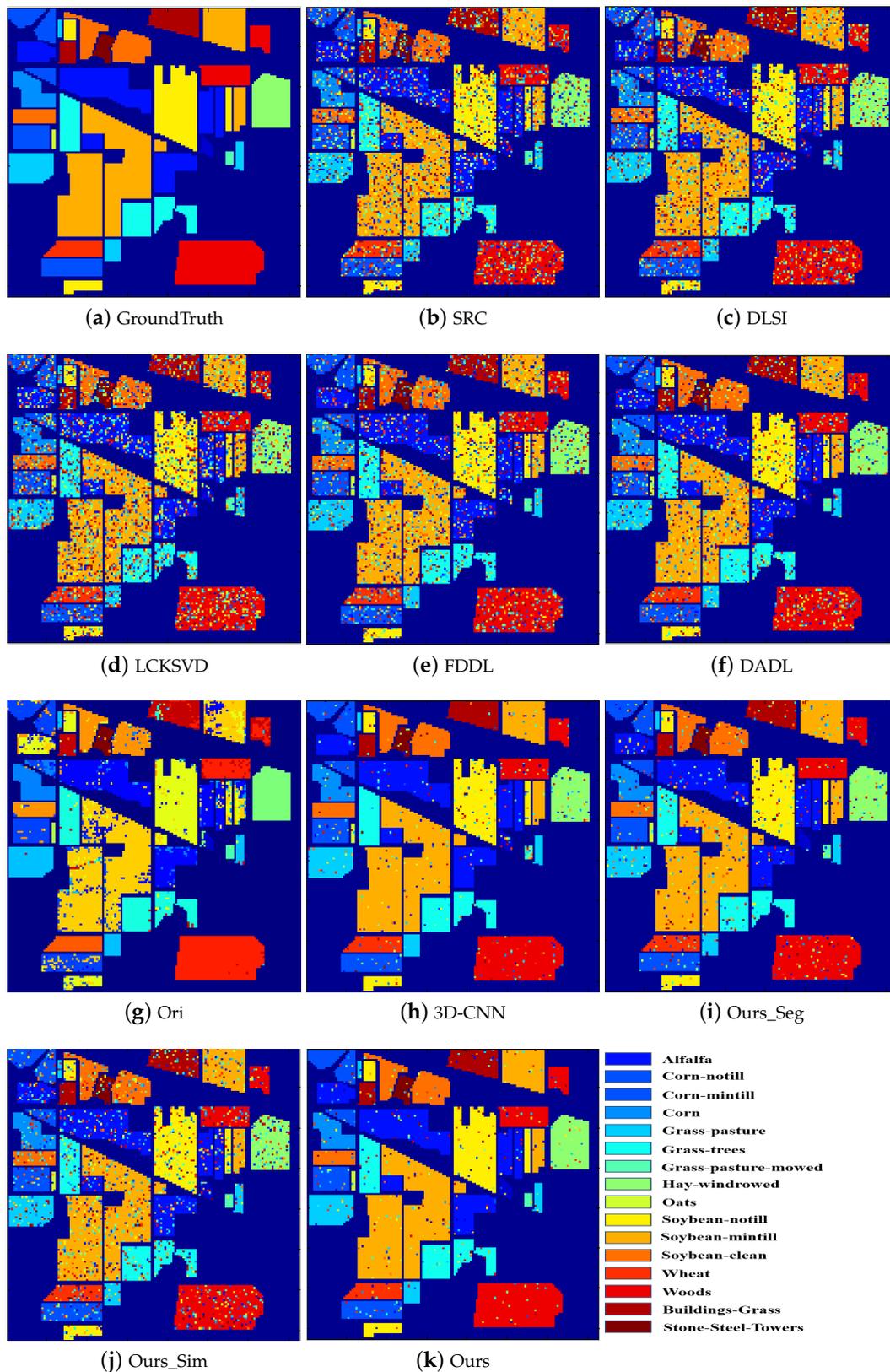
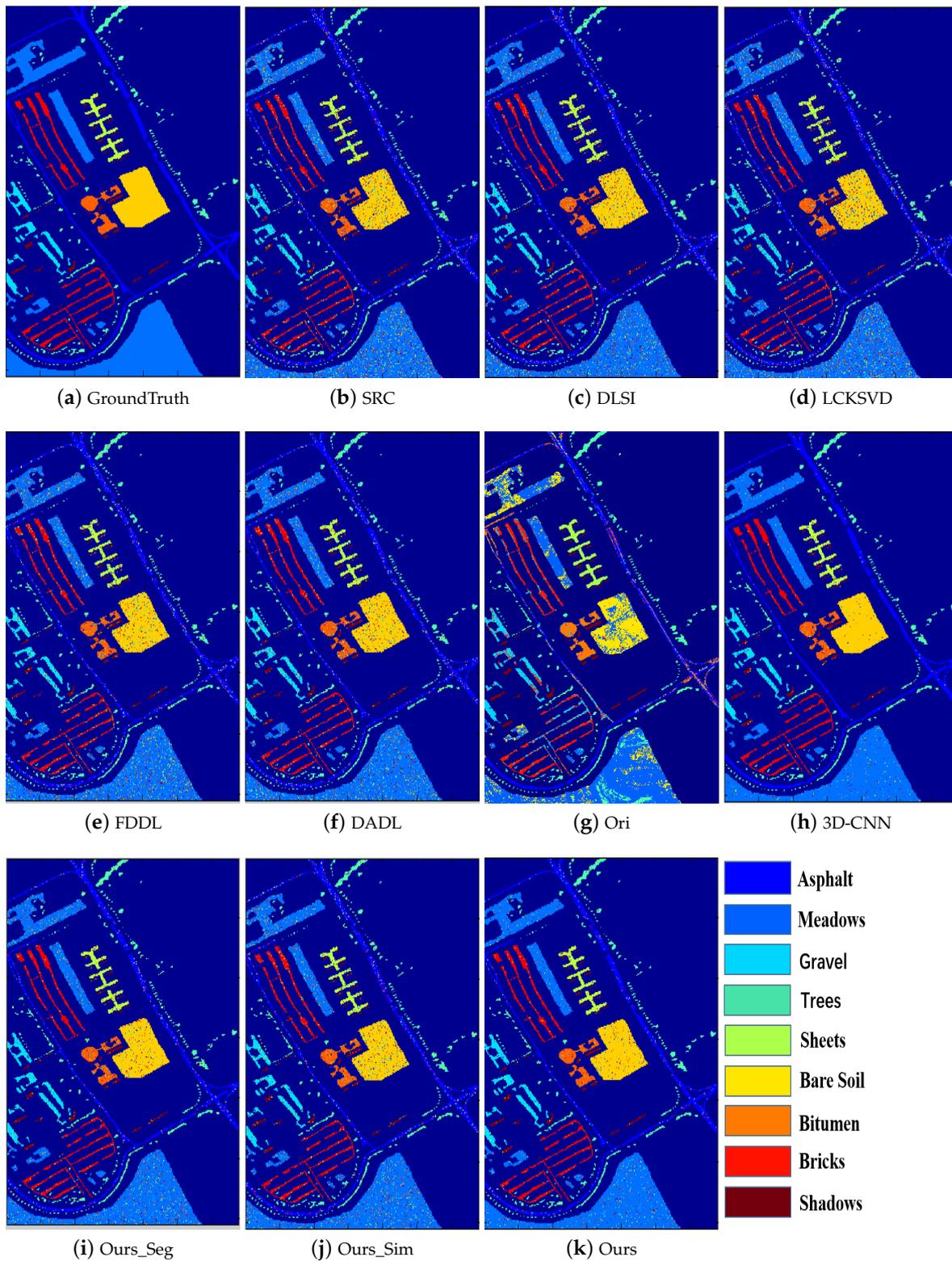
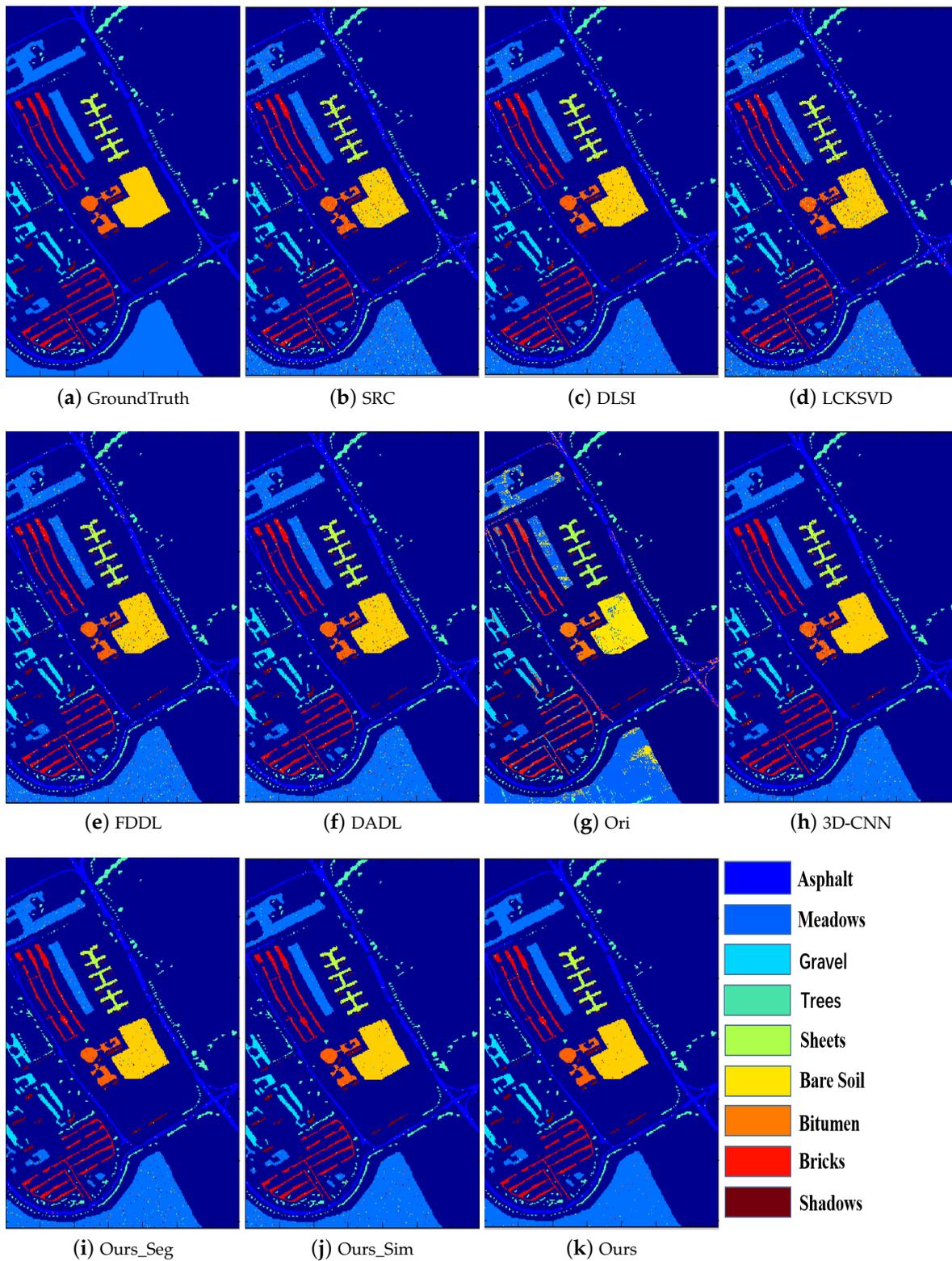


Figure 4. Classification maps of different methods on the Indian Pines dataset via SVM classifier (a–k).



**Figure 5.** Classification maps of different methods on the PaviaU dataset via  $k$ -NN classifier (a–k).



**Figure 6.** Classification maps of different methods on the PaviaU dataset via SVM classifier (a–k).

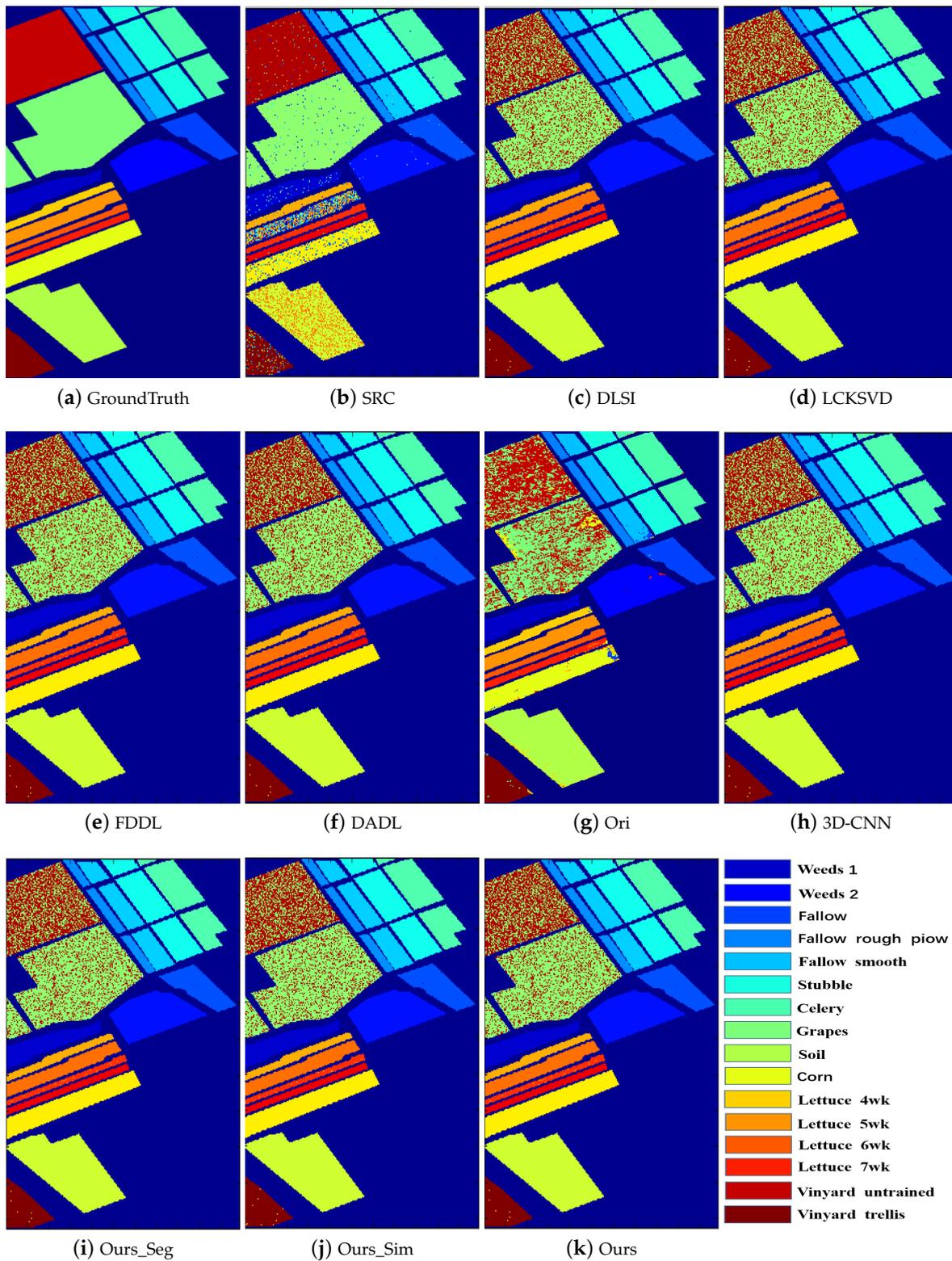
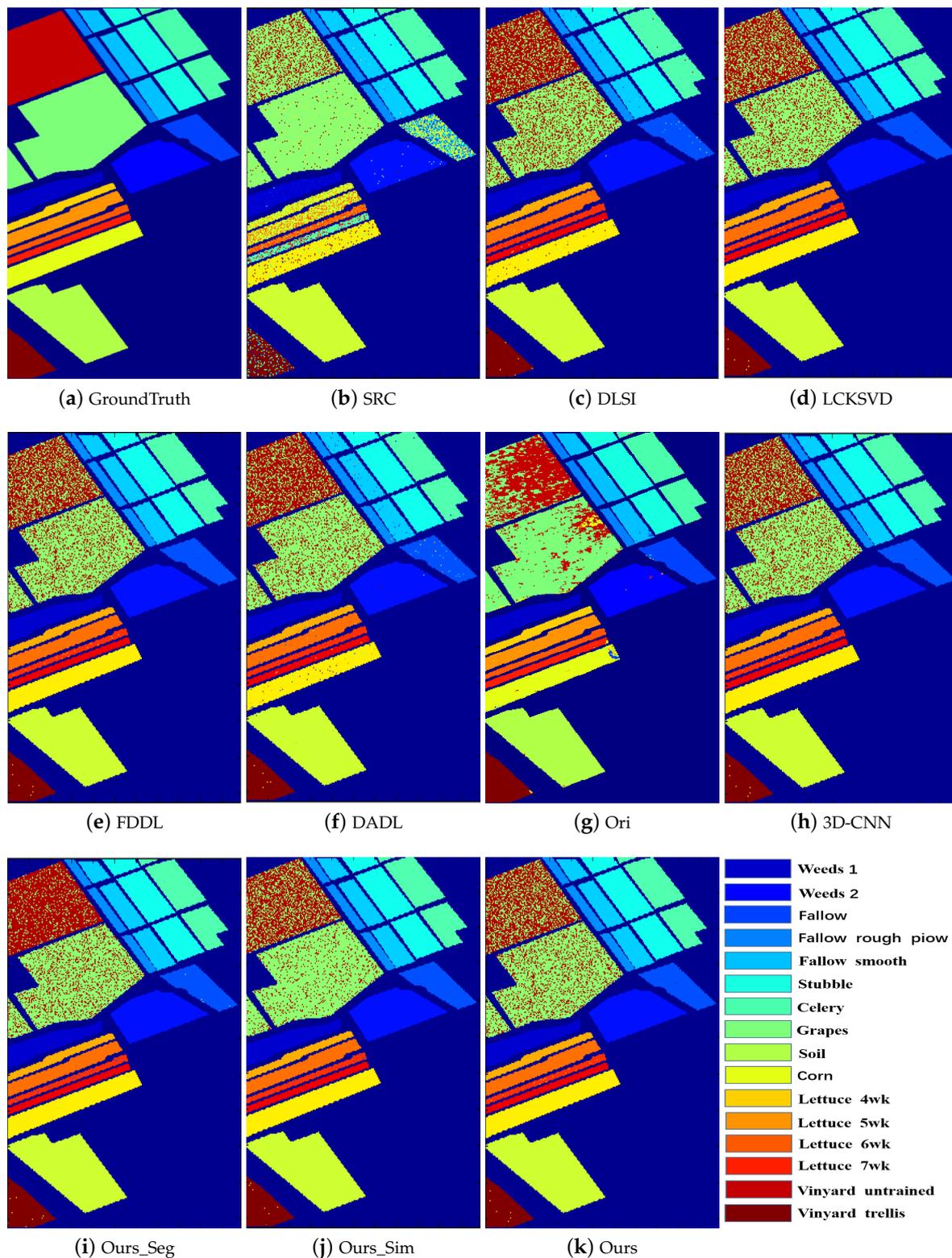


Figure 7. Classification maps of different methods on the Salinas dataset via  $k$ -NN classifier (a–k).

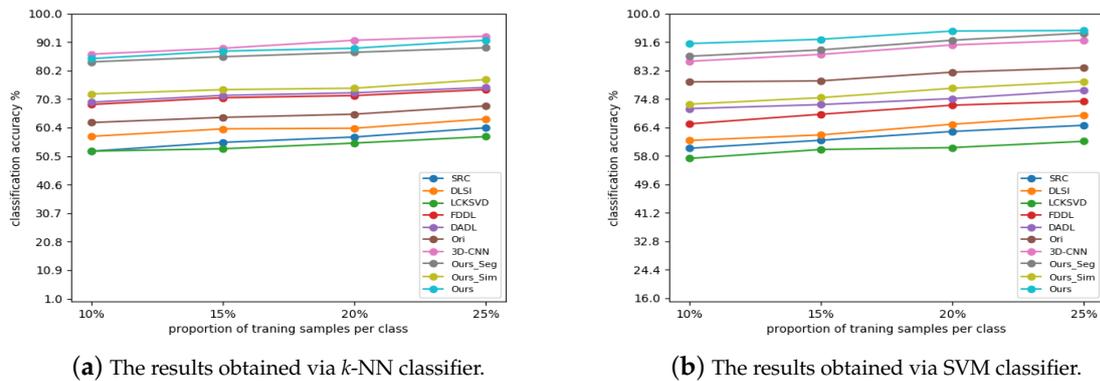


**Figure 8.** Classification maps of different methods on the Salinas dataset via SVM classifier (a–k).

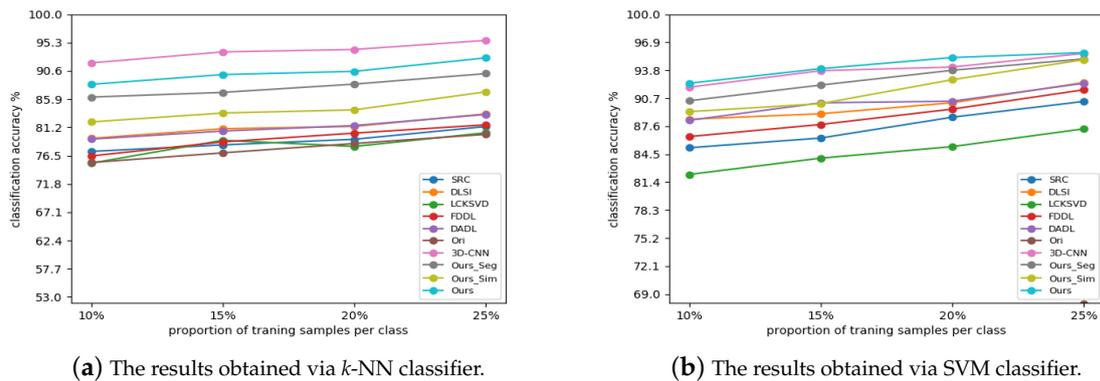
#### 3.4. Experimental Results with Different Small Amount of Training Data

The classification results varied with the different small amount of training data are shown in Figures 9–11, where the training data is varied from 10% to 25%. From the experimental results, we can see the classification results of the proposed methods increase when more samples are introduced for

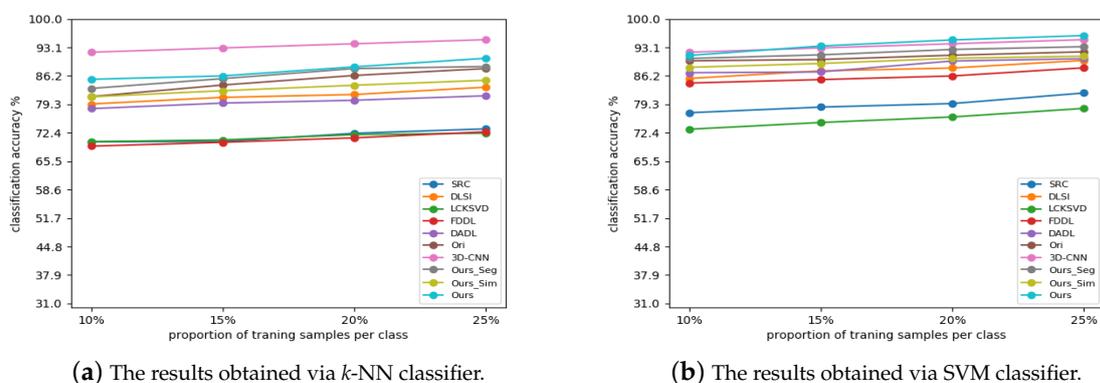
training, which is natural since classifier can be well trained with more training samples. Nevertheless, the proposed method outperforms all competing methods stably when using together with SVM classifier, and only inferior to 3D-CNN when using together with  $k$ -NN classifier since  $k$ -NN is a classification method without training. The experimental results are inconsistent with that in Section 3.3. From the above results, we can conclude that the proposed method is effective for HSI classification.



**Figure 9.** Classification performance with different numbers of training samples on Indian Pines dataset via  $k$ -NN (a) and SVM (b).



**Figure 10.** Classification performance with different numbers of training samples on PaviaU dataset via  $k$ -NN (a) and SVM (b).



**Figure 11.** Classification performance with different numbers of training samples on Salinas dataset via  $k$ -NN (a) and SVM (b).

#### 4. Discussion

In the above experiments, we divide the entire spectrum into three segments, and then adopt a voting strategy to generate the final classification result. To testify the effectiveness of the adopted dividing and voting strategies, we compare it with the classification results directly obtained from each segment and the result from the entire spectrum. In the following, we use **Seg-vote/Ours**, **Seg1**, **Seg2**, **Seg3** and **Entire** to denote the classification result from the dividing and voting strategy, the first segment, the second segment, the third segment and the entire spectrum, respectively. The experimental results are given in Table 8. We can observe that dividing and voting strategy can obtain better HSI classification results, compared with the segment or the entire spectrum-based method.

**Table 8.** Classification accuracy(%) on three datasets with different segments used.

Classifier	PaviaU				
	Seg1	Seg2	Seg3	Seg-vote	Entire
<i>k</i> -NN	72.17	80.74	84.35	90.52	84.12
SVM	82.67	91.34	92.89	95.21	93.80
Classifier	Indian Pines				
	Seg1	Seg2	Seg3	Seg-vote	Entire
<i>k</i> -NN	69.23	72.70	73.08	87.98	74.06
SVM	70.15	76.50	77.12	94.86	77.97
Classifier	Sanlinas				
	Seg1	Seg2	Seg3	Seg-vote	Entire
<i>k</i> -NN	74.56	79.76	84.07	88.38	83.92
SVM	83.67	87.6	89.17	94.91	90.45

#### 5. Conclusions

In the paper, we present a novel analysis dictionary learning model-based hyperspectral image classification method. The proposed framework naturally considers both the characteristics within the spectrum and among the spectra. By dividing the spectrum into several segments, the influence of strong nonlinearity with spectrum can be alleviated. In addition, the relationship among spectra can further improve the classification performance. Experimental results on three benchmark HSI datasets demonstrate the superiority of the proposed framework for HSI classification.

**Author Contributions:** W.W. and L.Z. conceived and designed the experiments; M.M. performed the experiments; W.W., P.Z., and Y.Z. analyzed the data; W.W., L.Z., M.M., and C.W. wrote the paper.

**Funding:** This research was funded by the National Natural Science Foundation of China (No. 61671385, 61571354, 61571362), the Natural Science Basis Research Plan in Shaanxi Province of China (No. 2017JM6021, 2018JM6015)

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

#### References

1. Ghamisi, P.; Yokoya, N.; Li, J.; Liao, W.; Liu, S.; Plaza, J.; Rasti, B.; Plaza, A. Advances in Hyperspectral Image and Signal Processing: A Comprehensive Overview of the State of the Art. *IEEE Geosci. Remote Sens. Mag.* **2018**, *5*, 37–78. [[CrossRef](#)]
2. Wei, W.; Zhang, L.; Jiao, Y.; Tian, C.; Wang, C.; Zhang, Y. Intra-Cluster Structured Low-Rank Matrix Analysis Method for Hyperspectral Denoising. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 866–880.
3. 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**, *PP*, 1–19. [[CrossRef](#)]
4. Guerra, R.; Barrios, Y.; Diaz, M.; Santos, L.; Lopez, S.; Sarmiento, R. A New Algorithm for the On-Board Compression of Hyperspectral Images. *Remote Sens.* **2018**, *10*, 428. [[CrossRef](#)]

5. Zhang, L.; Wei, W.; Bai, C.; Gao, Y.; Zhang, Y. Exploiting Clustering Manifold Structure for Hyperspectral Imagery Super-Resolution. *IEEE Trans. Image Process.* **2018**, *27*, 5969–5982. [[CrossRef](#)] [[PubMed](#)]
6. Fauvel, M.; Tarabalka, Y.; Benediktsson, J.A.; Chanussot, J.; Tilton, J.C. Advances in Spectral-Spatial Classification of Hyperspectral Images. *Proc. IEEE* **2013**, *101*, 652–675. [[CrossRef](#)]
7. Camps-Valls, G.; Bruzzone, L. Kernel-based methods for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 1351–1362. [[CrossRef](#)]
8. Wang, Q.; Lin, J.; Yuan, Y. Salient Band Selection for Hyperspectral Image Classification via Manifold Ranking. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1279–1289. [[CrossRef](#)]
9. Rajadell, O.; García-Sevilla, P.; Pla, F. Spectral-Spatial Pixel Characterization Using Gabor Filters for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 860–864. [[CrossRef](#)]
10. Xiang, X.; Li, J.; Li, S. Multiview Intensity-Based Active Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *PP*, 1–12.
11. Blanzieri, E.; Melgani, F. Nearest Neighbor Classification of Remote Sensing Images with the Maximal Margin Principle. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 1804–1811. [[CrossRef](#)]
12. Xue, Z.; Du, P.; Su, H. Harmonic Analysis for Hyperspectral Image Classification Integrated with PSO Optimized SVM. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2014**, *7*, 2131–2146. [[CrossRef](#)]
13. Sharma, S.; Buddhiraju, K.M. Spatial-spectral ant colony optimization for hyperspectral image classification. *Int. J. Remote Sens.* **2018**, *39*, 2702–2717. [[CrossRef](#)]
14. Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral Image Classification Using Deep Pixel-Pair Features. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 844–853. [[CrossRef](#)]
15. Othman, E.; Bazi, Y.; Alajlan, N.; Alhichri, H.; Melgani, F. Using convolutional features and a sparse autoencoder for land-use scene classification. *Int. J. Remote Sens.* **2016**, *37*, 2149–2167. [[CrossRef](#)]
16. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [[CrossRef](#)]
17. Li, Y.; Zhang, H.; Shen, Q. Spectral-Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network. *Remote Sens.* **2017**, *9*, 67. [[CrossRef](#)]
18. Fang, B.; Li, Y.; Zhang, H.; Chan, J.W. Hyperspectral Images Classification Based on Dense Convolutional Networks with Spectral-Wise Attention Mechanism. *Remote Sens.* **2019**, *11*, 159. [[CrossRef](#)]
19. Zabalza, J.; Ren, J.; Yang, M.; Zhang, Y.; Wang, J.; Marshall, S.; Han, J. Novel Folded-PCA for improved feature extraction and data reduction with hyperspectral imaging and SAR in remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 112–122. [[CrossRef](#)]
20. Mura, M.D.; Villa, A.; Benediktsson, J.A.; Chanussot, J.; Bruzzone, L. Classification of Hyperspectral Images by Using Extended Morphological Attribute Profiles and Independent Component Analysis. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 542–546. [[CrossRef](#)]
21. Nielsen, A.A. Kernel Maximum Autocorrelation Factor and Minimum Noise Fraction Transformations. *IEEE Trans. Image Process.* **2011**, *20*, 612–624. [[CrossRef](#)] [[PubMed](#)]
22. Majdar, R.S.; Ghassemian, H. A probabilistic SVM approach for hyperspectral image classification using spectral and texture features. *Int. J. Remote Sens.* **2017**, *38*, 4265–4284. [[CrossRef](#)]
23. Medjahed, S.A.; Saadi, T.A.; Benyettou, A.; Ouali, M. Gray Wolf Optimizer for hyperspectral band selection. *Appl. Soft Comput.* **2016**, *40*, 178–186. [[CrossRef](#)]
24. Zhang, L.; Wei, W.; Zhang, Y.; Shen, C.; Hengel, A.V.D.; Shi, Q. Dictionary Learning for Promoting Structured Sparsity in Hyperspectral Compressive Sensing. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7223–7235. [[CrossRef](#)]
25. Gao, Q.; Lim, S.; Jia, X. Improved Joint Sparse Models for Hyperspectral Image Classification Based on a Novel Neighbour Selection Strategy. *Remote Sens.* **2018**, *10*, 905. [[CrossRef](#)]
26. He, Z.; Wang, Y.; Hu, J. Joint Sparse and Low-Rank Multitask Learning with Laplacian-Like Regularization for Hyperspectral Classification. *Remote Sens.* **2018**, *10*, 322. . [[CrossRef](#)]
27. Zhang, L.; Wei, W.; Zhang, Y.; Shen, C.; van den Hengel, A.; Shi, Q. Cluster Sparsity Field: An Internal Hyperspectral Imagery Prior for Reconstruction. *Int. J. Comput. Vis.* **2018**, *126*, 797–821. [[CrossRef](#)]
28. Rubinstein, R.; Bruckstein, A.M.; Elad, M. Dictionaries for Sparse Representation Modeling. *Proc. IEEE* **2010**, *98*, 1045–1057. [[CrossRef](#)]

29. Zhang, S.; Zhang, M.; He, R.; Sun, Z. Transform-invariant dictionary learning for face recognition. In Proceedings of the IEEE International Conference on Image Processing, Paris, France, 27–30 October 2015; pp. 348–352.
30. Wright, J.; Yang, A.Y.; Sastry, S.S.; Ma, Y. Robust Face Recognition via Sparse Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 210–227. [[CrossRef](#)]
31. Jiang, Z.; Lin, Z.; Davis, L.S. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* **2011**, *42*, 1697–1704.
32. Kviatkovsky, I.; Gabel, M.; Rivlin, E.; Shimshoni, I. On the Equivalence of the LC-KSVD and the D-KSVD Algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 411–416. [[CrossRef](#)] [[PubMed](#)]
33. Yang, M.; Zhang, L.; Feng, X.; Zhang, D. Fisher Discrimination Dictionary Learning for sparse representation. *Proc. IEEE Int. Conf. Comput. Vis.* **2011**, *24*, 543–550.
34. Ramirez, I.; Sprechmann, P.; Sapiro, G. Classification and clustering via dictionary learning with structured incoherence and shared features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
35. Guo, J.; Guo, Y.; Kong, X.; Zhang, M.; He, R. Discriminative Analysis Dictionary Learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
36. Kiani, V.; Harati, A.; Vahedian, A. Planelets—A Piecewise Linear Fractional Model for Preserving Scene Geometry in Intra-Coding of Indoor Depth Images. *IEEE Trans. Image Process.* **2017**, *26*, 590–602. [[CrossRef](#)] [[PubMed](#)]
37. Wang, C.; Zhang, L.; Wei, W.; Zhang, Y. When Low Rank Representation Based Hyperspectral Imagery classification Meets Segmented Stacked Denoising Auto Encoder Based Spatial-Spectral Feature. *Remote Sens.* **2017**, *10*, 284. [[CrossRef](#)]
38. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 881–892. [[CrossRef](#)]
39. Xiang, X.; Li, J.; Wu, C.; Plaza, A. Regional clustering-based spatial preprocessing for hyperspectral unmixing. *Remote Sens. Environ.* **2018**, *204*, 333–346.
40. Luo, D.; Ding, C.H.Q.; Nie, F.; Huang, H. Cauchy Graph Embedding. In Proceedings of the International Conference on International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011; pp. 553–560.
41. Ran, H.; Hu, B.-G.; Zheng, W.-S.; Kong, X.-W. Robust principal component analysis based on maximum correntropy criterion. *IEEE Trans. Image Process.* **2011**, *20*, 1485–1494. [[CrossRef](#)] [[PubMed](#)]
42. Nikolova, M.; Ng, M. *Analysis of Half-Quadratic Minimization Methods for Signal and Image Recovery*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2005; pp. 937–966.
43. Zhang, L.; Zhang, Y.; Yan, H.; Gao, Y.; Wei, W. Salient Object Detection in Hyperspectral Imagery using Multi-scale Spectral-Spatial Gradient. *Neurocomputing* **2018**, *291*, 215–225. [[CrossRef](#)]
44. Camps-Valls, G.; Gomez-Chova, L.; Munoz-Mari, J.; Vila-Frances, J.; Calpe-Maravilla, J. Composite kernels for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2006**, *3*, 93–97. [[CrossRef](#)]

