Collaborative Superpixelwised PCA for Hyperspectral Image Classification

Chao Yao, Junrui Gu, Zehua Guo, *Senior Member, IEEE*, Miao Ma, Qingrui Guo, Gong Cheng, *Member, IEEE*.

***Abstract*—Extracting spectral-spatial features from Hyperspectral imagery (HSI) has been proven to be efficient for classification tasks. A recently developed superpixelwised PCA (SuperPCA), which has shown its promising performance, is a prominent technique in spectral-spatial feature extraction. However, we have discovered that SuperPCA may lead to an intra-class dispersion problem, which can result in a decrease in classification accuracy. In this paper, a novel method called Collaborative Superpixelwised PCA (CSPCA) is proposed to address this issue. The main idea behind CSPCA is to collaboratively learn the projections for each superpixel. Specifically, CSPCA first employs a superpixel segmentation technique to generate superpixels. Next, the mean vectors of samples within each superpixel are utilized to model the manifold structure of the data. Then, a novel objective function is formulated, which aims to simultaneously preserve the obtained manifold structure between superpixels and the structure within each superpixel. To optimize the objective function, the Manopt toolbox is employed in the proposed method. The effectiveness of the proposed approach is validated through experimental evaluations conducted on five HSI data sets.**

***Index Terms*—Dimension reduction, Hyperspectral imagery (HSI) classification, spectral-spatial feature learning, superpixelwised PCA.**

I. Introduction

Hyperspectral imaging has gained significant attention in remote sensing applications, such as land use monitoring [1] [2], urban planning [3] [4], due to its strong capability in capturing rich spectral information. Hyperspectral image (HSI) provides a high-dimensional spectral signature for each pixel, enabling detailed analysis and discrimination of different materials and land cover types. However, the high dimensionality of hyperspectral data brings challenges in terms of computational complexity and the curse of dimensionality [5].

To address these challenges, dimensionality reduction (DR) methods have been widely employed to reduce the data dimension in HSI. The DR methods aim to extract a lower-dimensional representation of the data that retains the relevant spectral information. Band selection (BS) [6] [7] [8] and feature extraction (FE) [9] [10] are two effective approaches for DR in HSI. BS techniques aim to select a subset of the most informative spectral bands from the original high-dimensional hyperspectral data, while FE techniques map the original high-dimensional hyperspectral data onto a lower-dimensional feature space. In general, BS techniques can preserve the physical information of the spectral data but always exhibit lower performance compared to FE ones. Thus, we focus on FE methods for the HSI classification task in this work.

Many FE techniques have been applied for HSI classification tasks in recent decades. Farrell and Mersereau [11] demonstrated the effectiveness of Principal Component Analysis (PCA) in detecting targets in HSI, while Bandos et al. [12] applied Linear Discriminant Analysis (LDA) for HSI classification. In order to capture the nonlinear structure of the data, manifold learning [13] [14] and kernel-based methods [15] [16] have been proposed to extract more discriminative features. To explore the spectral information for visualization and classification, Kang et al. [17] converted a 1-D spectrum into a 2-D image recently. More recently, the deep learning approaches [18] [19] [17] have been adopted to enhance the feature representations. While these techniques primarily utilize the spectral information of the data to determine class labels, they do not directly incorporate the spatial information of HSI, which has been proven to be effective for classification.

To incorporate the spatial information into FE techniques, various approaches have been proposed, which can be generally categorized into spatial filtering [20] [21], tensorial representation [22] [23] [24] and superpixel segmentation [25] [26] [27]. Spatial filtering considers the correlations between neighboring pixels to incorporate spatial information. Tensorial representation treats the HSI data as a tensor rather than a vector, directly integrating spatial information into the dimension reduction process. Superpixel segmentation divides the HSI data into groups by jointly considering the texture, color, and sample location information. This allows FE techniques to be applied directly to each individual group. Among these ways, the spatial filtering methods are typically applied as a preprocessing step to enhance the spatial information in the data. Tensorial representation methods, while effective, can be computationally expensive due to the increased complexity of tensor operations. Therefore, this study focuses on superpixel segmentation-based techniques. By grouping pixels into superpixels, these methods enable the direct application of FE techniques on each group, facilitating the extraction of spatially enhanced features for HSI classification tasks.

SuperPCA [25] is a representative work for superpixel

This research was supported in part by the Fundamental Research Funds for Central Universities under Grant 1301032207; and in part by the Regional Innovation Guidance Project of Shaanxi under Grant 2022QFY0105, and in part of the National Natural Science Foundation of China under Grant 62377031. Manuscript received XXX; revised XXX. (Corresponding author: Gong Cheng.)

Chao Yao, Junrui Gu, and Miao Ma are with the School of Computer Science, Shaanxi Normal University, Xi'an 710119, China.

Zehua Guo is with the School of Automation, Beijing Institute of Technology, Bejing 100081, China.

Qingrui Guo is with State Grid Xinjiang Electric Power Research Institute, Urumqi 830000, China.

Gong Cheng is with the School of Automation, Northwestern Polytechnical University, Xi'an 710129, China. (Email: gcheng@nwpu.edu.cn)

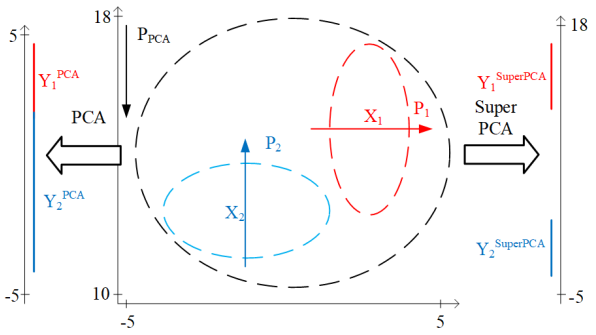


Fig. 1. An example illustrating why dispersion occurs in SuperPCA. $\mathbf{X}_1$ and $\mathbf{X}_2$ represent the distributions of samples from the same category but located in different superpixels. The right side shows the distribution of these samples in 1D space after SuperPCA, while the left side illustrates the distribution after PCA.

segmentation-based methods. Compared with other typical spectral-spatial feature extraction techniques, SuperPCA has demonstrated superior performance. However, we have discovered that SuperPCA may destroy the intra-class structure of the data, resulting in dispersion between samples from the same category but located in different superpixels in the low-dimensional feature space. To illustrate this dispersion problem, an example is provided in Fig. 1. The objective of SuperPCA is to find different projections for samples originating from different superpixels. However, if these samples exist in different regions within the distribution, the resulting projections would magnify the differences between them. As illustrated in Fig. 1, the distributions of samples from two different superpixels are represented in red and blue, respectively. The principal vectors, $\mathbf{P}_1$ and $\mathbf{P}_2$, exhibit significant differences, which leads to the samples being mapped to distant locations in the 1D projection. Specifically, the $\mathbf{X}_1$ is mapped along the X-axis, which ranges in $[-5, 5]$, while $\mathbf{X}_2$ is mapped along the Y-axis, with a range of $[10, 18]$. Consequently, after SuperPCA, the data are distanced from each other in the 1D space. Fig. 2 further demonstrates the dispersion problem of SuperPCA using real-world data. In Fig. 2a, the raw data of *Corn* class in **Indian Pines** data set is visualized using tSNE [28]. Different colors and shapes denote samples located in various superpixels. It is evident that the samples belonging to *Corn* class are closely distributed both in Figs. 2a and 2b, which represent the distributions of the original samples and the samples after PCA transformation, respectively. However, a noticeable dispersion between samples projected by SuperPCA from different superpixels can be observed in Fig. 2c. The dispersion problem in SuperPCA increases the complexity of classification, as it can lead to overly intricate boundaries and a higher risk of overfitting, particularly in high-dimensional spaces like HSI. Previous studies [29] [30] have shown that reducing intra-class variance can significantly improve class separability and enhance classification accuracy.

To tackle the dispersion problem in SuperPCA, we propose a new spectral-spatial FE method named Collaborate Superpixelwised PCA (CSPCA) in this paper. CSPCA aims to consider the relations between samples from different superpixels when finding the projections for each superpixel. Specifically, we first utilize PCA to identify the first principal of HSI, and employ Entropy Rate based Segmentation (ERS) [31] to generate superpixels. Then, the arithmetic centers of each superpixel are used to model the manifold structure of the data. Later, a novel object function that preserves both the obtained manifold structure between superpixels and the data's structure within each superpixel is proposed. To optimize this object function, an appropriate toolbox is employed in our proposed method. Experimental results on five HSI data sets, which are Indian Pines, the University of Pavia, Salinas, WHU-Hi-LongKou, and WHU-Hi-HanChuan, validate the effectiveness of the proposed method.

In summary, the contributions of this paper are highlighted as follows:

1) We identify the dispersion problem in SuperPCA and introduce CSPCA to effectively solve the problem. The proposed CSPCA jointly considers the projections for each superpixel, thereby preserving the intra-class structure of the data.
2) A novel objective function is introduced, where the manifold structure is leveraged to maintain the intra-class relation of data from different superpixels, and PCA is employed to maintain the structure of data from the same superpixel simultaneously. To optimize the proposed objective function, the Manopt toolbox is employed, ensuring efficient and effective optimization procedures.

The rest of this paper is organized as follows. Section II reviews the related work briefly. Section III introduces the details of CSPCA and the optimization methods. Experimental results are discussed in Section IV, and the conclusion of the paper is presented in Section V.

II. Related Works

In this section, we provide an overview of FE algorithms for HSI classification. These works are categorized into spectral-based methods, superpixel-based methods, and deep learning-based methods. It is important to highlight that in this paper, uppercase boldface symbols, denoted as $\mathbf{G}$ and $\mathbf{P}$, signify matrices. Lowercase boldface symbols, such as $\mathbf{x}_i$, represent vectors. Additionally, regular typed symbols are used for values, such as w_{ij} and η.

A. Spectral-based methods

HSI consists of hundreds of contiguous spectral bands, where each pixel is represented as a feature vector containing hundreds of entries. However, the high dimensionality of HSI data presents challenges in classification tasks, including the curse of dimensionality, computational complexity, and redundancy in the spectral information. To address these problems, feature extraction methods in machine learning are adopted in HSI classification tasks.

PCA is a classical method in machine learning, which has been widely used in HSI processing [32] [33] and classification [11] [34] tasks. Given a data matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{d \times n}$, where $\mathbf{x}_i$ is the ith sample with d dimensionality, and

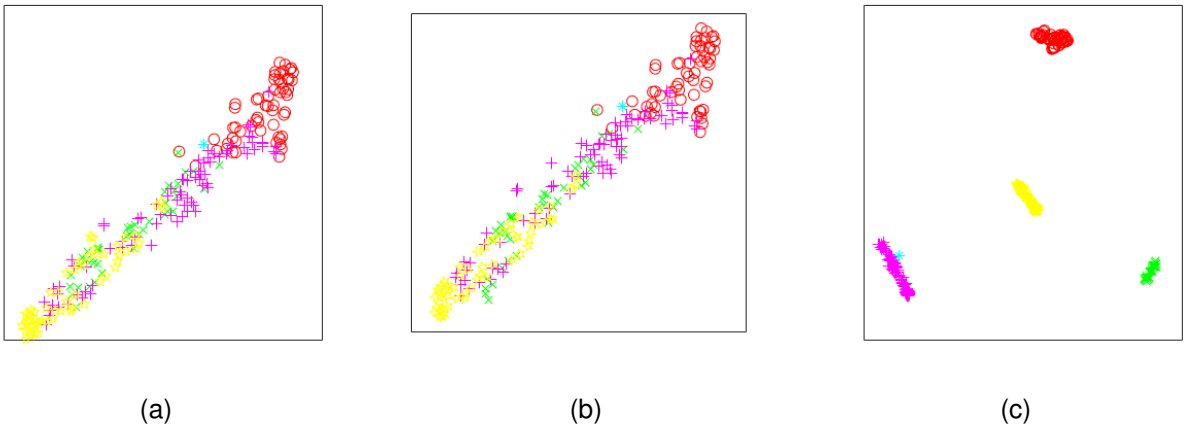


Fig. 2. An example of dispersion problem. The samples of class *Corn* in **Indian Pines** data set are processed by tSNE for 2D illustration. (a) shows the raw data. (b) displays the data after PCA. (c) presents the data after SuperPCA. Different colors and shapes represent the samples located in different superpixels.

n is the number of samples. PCA aims to find a projection matrix $\mathbf{P} \in \mathbb{R}^{d\times l}$ by

$$\mathbf{P}^* = \arg\max_{\mathbf{P}} \mathrm{Tr}(\mathbf{P}^T\mathbf{S}\mathbf{P}), \tag{1}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix, $\mathbf{S} = \frac{1}{n}\sum(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$ represents the covariance matrix of $\mathbf{X}$, and $\boldsymbol{\mu} = \frac{1}{n}\sum \mathbf{x}_i$ is the mean vector of the data. The problem in Eq.(1) can be solved by selecting the eigenvectors corresponding to the l largest eigenvalues of $\mathbf{S}$.

Using the optimal matrix $\mathbf{P}$, PCA transforms each sample $\mathbf{x}_i$ to $\mathbf{y}_i$ by $\mathbf{y}_i = \mathbf{P}^T\mathbf{x}_i$. When $d < l$, PCA can effectively reduce the dimensionality of the data. It is important to note that PCA assumes the data follows a Gaussian distribution, allowing the principal components to capture the main information of the data. However, in the case of HSI, the data usually do not adhere to a Gaussian distribution as they originate from different classes [35].

To address this problem, manifold learning methods, which enforce that pixels with similar signatures remain close to each other in the low-dimensional space, have been proposed for HSI classification. For instance, Wang et al. [36] employed Locality Preserving Projections (LPP), which is a classical manifold learning method, to find a low-dimensional embedding, which can preserve the geometric structure of the data. Duan et al. [37] proposed a semisupervised FE technique that first measures the distance between samples on a manifold, and then constructs hypergraphs to effectively incorporate both labeled and unlabeled training data. Jiang et al. [38] introduced a new method by incorporating Laplacian regularization and local enhancement into collaborative representation projection, which allows for better representation of the manifold structure and improves the performance of HSI classification tasks.

The spectral-based methods have made significant progress HSI feature extraction over the past decades. However, these methods often treat the HSI data, originally structured as a 3D cube, by flattening it into a 2D matrix to derive the low-dimensional representation. This procedure largely overlooks the critical spatial information presented in the image, where neighboring pixels often belong to the same category, thus missing valuable contextual information. A recent work in [17] applied sequence data processing algorithms on the 1D spectrum to derive 2D features for classification and visualization. This approach marked a new direction in the development of the spectral-based FE method for HSI.

B. Superpixel-based methods

Superpixel segmentation techniques always serve as a preprocessing step to incorporate spatial information into the spectral-based FE methods. The superpixel segmentation techniques first divide an HSI into multiple homogeneous regions, the FE methods can then be applied to each homogeneous region to extract low-dimensional representation. By integrating spatial information into spectral FE techniques, superpixel-based methods can achieve improved performance in HSI classification tasks.

SuperPCA is a representative superpixel-based technique for HSI classification tasks, where PCA is applied to each superpixel region to extract the local features. By integrating spatial and spectral information, SuperPCA outperforms the spectral-based FE methods. To further improve its classification performance, several extensions have been developed, such as GSuperPCA [39], RSuperPCA [39], SuperULDA [40], SuperPCA-DA [41], and BAMS [27] were proposed. GSuperPCA concatenates global features extracted from the entire HSI with local features via SuperPCA, thus combining global and local information. RSuperPCA enhances SuperPCA by reconstructing pixels within the same superpixel block to reduce noise. SuperULDA tackles the issue of mixed objects and strong noise by applying LDA to the samples with pseudo-labels to extract more discriminative local features. SuperPCA-DA integrates data augmentation techniques with SuperPCA, generating diverse new samples. BAMS adaptively determines the optimal number of superpixels band-by-band, extracting optimal joint spatial and spectral features. These methods collectively demonstrate advancements in the utilization of superpixel-based approaches for HSI classification.

C. Deep learning-based methods

In recent years, deep learning methods have been widely applied to HSI classification tasks due to their superior capability for automatic feature extraction. Among these methods, Convolutional Neural Networks (CNNs) have become one of the most popular architectures, leveraging both spectral and spatial information for improved classification accuracy. For instance, 2D CNNs [42] [43] treat HSI as typical multi-channel 2D images and focus on spatial features, while 3D CNNs [44] [45] directly process the HSI cube, capturing spectral-spatial features. However, these deep learning models are typically trained in an end-to-end manner, requiring labeled data for supervised learning. To address the lack of labeled data in unsupervised settings, autoencoder-based architectures have been explored, as demonstrated in the works of Zhang et al. [46] and Zhang et al. [47], enabling unsupervised deep model training for HSI classification.

Despite the significant advantages of deep learning methods in HSI classification, one major challenge remains: the high computational complexity and memory requirements not only during training but also at the inference stage. Many deep models, particularly complex architectures like 3D CNNs and hybrid models, rely on resource-intensive operations, and often require specialized hardware such as GPUs, which increases both the time and cost of implementation, making such approaches less feasible for resource-constrained environments, such as edge computing, onboard satellite systems, or low-power devices. Given these constraints, in this work, we focus on traditional machine learning FE methods. Approaches like SuperPCA and CSPCA require significantly less computational power during inference, as feature transformation involves only simple matrix operations. These methods are thus more practical for scenarios demanding lightweight and efficient solutions, providing an alternative to deep learning-based methods in contexts where computational resources are limited.

III. Collaborative superpixelwised PCA

The details of the proposed method will be presented in this section, which can be summarized into three main steps. Firstly, the samples are assigned to different regions through superpixel segmentation. Secondly, the relationship between samples from different superpixels is modeled, and manifold learning techniques are employed to capture this relationship. Finally, PCA projections are collaboratively learned for each superpixel. Fig. 3 shows an overall framework of the proposed method.

A. Superpixel segmentation

Graph-based methods [31] and gradient-based methods [48] represent two typical types of superpixel segmentation algorithms. Compared with the gradient-based ones, the graph-based methods have demonstrated superior performance and higher efficiency. Therefore, we adopt Entropy Rate Segmentation (ERS), a representative graph-based method, for superpixel segmentation in our approach.

Given a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ denotes the set of vertices representing the pixels in the image, and $\mathbf{E}$ represents the set of edges measuring the similarities between vertices. ERS selects a subset of edges $\mathbf{A} \subseteq \mathbf{E}$ to construct a resulting graph $\mathcal{G}^* = (\mathbf{V}, \mathbf{A})$ that consists of exactly N subgraphs, given a predefined number of superpixels N. The objective function of ERS is formulated as

$$\mathbf{A}^* = \arg\max_{\mathbf{A}} \mathrm{Tr}(H(\mathbf{A}) + \alpha B(\mathbf{A})), s.t. \mathbf{A} \subseteq \mathbf{E}, \quad (2)$$

where $H(\mathbf{A})$ represents the entropy rate term, which encourages generating homogeneous and compact regions, and $B(\mathbf{A})$ is used to enforce region sizes to be similar. A greedy heuristic algorithm is employed to find the optimal subset $\mathbf{A}^*$.

ERS is a powerful technique for segmenting greyscale images into superpixel regions with a given number of superpixels. When applied to HSI, however, the high dimensionality of the data presents a challenge. To effectively apply ERS in this context, PCA is first employed to reduce the multi-channel HSI into a single-channel image in our scheme. This preprocessing step enables the subsequent application of ERS for generating desired superpixels.

B. Model the relationship between superpixels

The samples within each superpixel are probably from the same category, making the mean vector of the samples a meaningful representation of the formulated. To model the structure of these mean vectors, we adopt the criterion used in Locality Linear Embedding (LLE) [49], which can be indicated as follows:

$$\mathbf{w}_i = \arg\min_{\mathbf{w}_i} \parallel \boldsymbol{\mu}_i - \sum_j w_{ij} \boldsymbol{\mu}_j \parallel^2, s.t. \sum_j w_{ij} = 1, \quad (3)$$

where $\boldsymbol{\mu}_i$ is the mean vector of samples within the i-th superpixel, and the reconstruction weights w_{ij} are constrained to 0 if $\mathbf{x}_j$ is not in the $K-$nearest neighborhood of $\mathbf{x}_i$. The optimal $\mathbf{w}_i$ for Eq.(3) can be solved by $\mathbf{M}\mathbf{w}_i = \mathbf{1}$, where $\mathbf{M} = \sum_j (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T$.

By iteratively applying Eq.(3) to each sample, an $N \times N$ matrix $\mathbf{G}^T = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N]$ is obtained, which represents the manifold structure of the data. It is important to note that each row in $\mathbf{G}$ corresponds to the reconstruction weights of each sample.

To preserve the manifold structure in Eq.(3) in subspace, the following formulation can be employed:

$$\mathbf{P} = \arg\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \sum_i \parallel \tilde{\boldsymbol{\mu}}_i - \sum_j w_{ij} \tilde{\boldsymbol{\mu}}_j \parallel^2 . \quad (4)$$

where $\tilde{\boldsymbol{\mu}}_i = \mathbf{P}^T \boldsymbol{\mu}_i$ and $\tilde{\boldsymbol{\mu}}_j = \mathbf{P}^T \boldsymbol{\mu}_j$ are the mean vectors in the low-dimensional subspace, the constraint $\mathbf{P}^T\mathbf{P} = \mathbf{I}$ is enforced to ensure that the projection matrix $\mathbf{P}$ is orthogonal and avoids trivial solutions in the optimization process.

C. Collaboratively learned PCA projections

SuperPCA finds the projection matrix for each superpixel by minimizing the reconstruction error, which can be formulated as follows:

$$\mathcal{L}(\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N) = \sum_i \sum_j \parallel \mathbf{P}_i \mathbf{P}_i^T \mathbf{x}_{ij} - \mathbf{x}_{ij} \parallel^2, \quad (5)$$

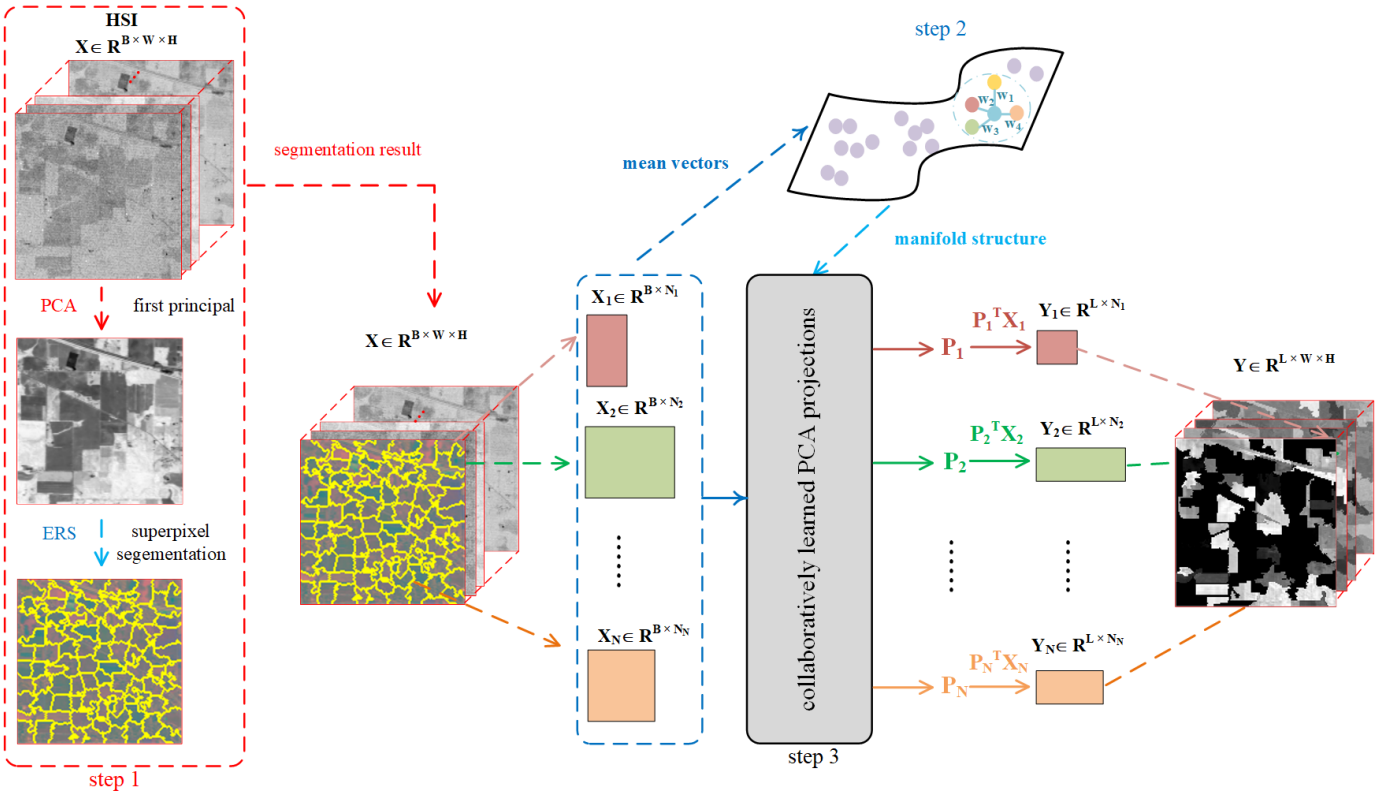


Fig. 3. Outline of the proposed CSPCA.

where $\mathbf{P}_i$ is the projection matrix for the i-th superpixel, and $\mathbf{x}_{ij}$ is the j-th sample within the i-th superpixel. Because $\mathbf{P}_i$ is only related to the samples within the i-th superpixel, the optimal set $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N\}$ can be obtained by solving a set of PCA problems, individually. It should be noted that PCA is known to effectively preserve the structure of given data, as has been proven in prior studies.

To address the dispersion problem in SuperPCA, the manifold structure is employed to ensure that low-dimensional representations are similar when their corresponding high-dimensional data exhibit similarities. Since different superpixels have different projection matrices, $\tilde{\boldsymbol{\mu}}_i = \mathbf{P}_i^T \boldsymbol{\mu}_i$ holds in Eq.(4). By combining Eqs.(4) and (5), a new objective is formulated as follows:

$$\begin{aligned}&\mathcal{L}(\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N) = \\ &\sum_i (\| \mathbf{P}_i^T \boldsymbol{\mu}_i - \sum_j w_{ij} \mathbf{P}_j^T \boldsymbol{\mu}_j \|^2 + \eta \sum_j \| \mathbf{P}_i \mathbf{P}_i^T \mathbf{x}_{ij} - \mathbf{x}_{ij} \|^2), \\ &s.t.\ \mathbf{P}_i^T \mathbf{P}_i = \mathbf{I},\end{aligned} \tag{6}$$

where the first term maintains the structure of the data between superpixels, while the second term preserves the structure of data located in the same superpixel, η is the weight to balance the two terms.

D. Optimization method

To find the optimal $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N\}$ for each superpixel, the coordinate descent method is adopted in our method. Specifically, the projection matrix $\mathbf{P}_s$ for the s-th superpixel is optimized while keeping the other $N-1$ matrices fixed, using the following formulation:

$$\begin{aligned}&\mathbf{P}_s^* = \arg\min\ \mathcal{L}(\mathbf{P}_1, \ldots, \mathbf{P}_s, \ldots \mathbf{P}_N), \\ &s.t.\ \mathbf{P}_s^T \mathbf{P}_s = \mathbf{I},\ s = 1, \ldots, N.\end{aligned} \tag{7}$$

To solve the problem in Eq.(7), by combining Eq.(6), we present a proposition as follows.

Proposition 3.1: Eq.(7) is equivalent to the equation as follows:

$$\begin{aligned}&\mathcal{L}(\mathbf{P}_1, \ldots, \mathbf{P}_s \ldots \mathbf{P}_N) \\ &= \mathbf{Tr}(\mathbf{P}_s^T \mathbf{A}_s \mathbf{P}_s) + 2\mathbf{Tr}(\mathbf{P}_s^T \mathbf{B}_s),\ s.t.\ \mathbf{P}_s^T \mathbf{P}_s = \mathbf{I},\end{aligned} \tag{8}$$

where $\mathbf{A}_s = \boldsymbol{\mu}_s \boldsymbol{\mu}_s^T - \eta \mathbf{X}_s \mathbf{X}_s^T$, $\mathbf{B}_s = \sum_{i,j} w_{is} w_{ij} \boldsymbol{\mu}_s \boldsymbol{\mu}_j^T \mathbf{P}_j - \sum_i (w_{is} + w_{si}) \boldsymbol{\mu}_s \boldsymbol{\mu}_i^T \mathbf{P}_i$.

Proof.

$$\begin{aligned}
&\sum_i(\| \mathbf{P}_i^T\boldsymbol{\mu}_i - \sum_j w_{ij}\mathbf{P}_j^T\boldsymbol{\mu}_j \|^2 + \eta\sum_j \| \mathbf{P}_i\mathbf{P}_i^T\mathbf{x}_{ij} - \mathbf{x}_{ij} \|^2)\\
&= \sum_i((\mathbf{P}_i^T\boldsymbol{\mu_i} - \sum_j w_{ij}\mathbf{P}_j^T\boldsymbol{\mu}_j)^T(\mathbf{P}_i^T\boldsymbol{\mu}_i - \sum_j w_{ij}\mathbf{P}_j^T\boldsymbol{\mu}_j)\\
&+ \sum_j(\mathbf{P}_i\mathbf{P}_i^T\mathbf{x}_{ij} - \mathbf{x}_{ij})^T(\mathbf{P}_i\mathbf{P}_i^T\mathbf{x}_{ij} - \mathbf{x}_{ij}))\\
&\overset{\mathbf{P}_s^T\mathbf{P}_s=\mathbf{I}}{=} \sum_i \boldsymbol{\mu}_i^T\mathbf{P}_i\mathbf{P}_i^T\boldsymbol{\mu}_i + \sum_{i,j,k} w_{ij}w_{ik}\boldsymbol{\mu}_j^T\mathbf{P}_j\mathbf{P}_k^T\boldsymbol{\mu}_k\\
&- 2\sum_{i,j} w_{ij}\boldsymbol{\mu}_i^T\mathbf{P}_i\mathbf{P}_j^T\boldsymbol{\mu}_j + \eta\sum_{i,j}(\mathbf{x}_{ij}^T\mathbf{x}_{ij} - 2\mathbf{x}_{ij}^T\mathbf{P}_i\mathbf{P}_j^T\mathbf{x}_{ij})\\
&= \sum_i \mathbf{Tr}(\mathbf{P}_i^T(\boldsymbol{\mu}_i\boldsymbol{\mu}_i^T - \eta\mathbf{X}_i\mathbf{X}_i^T)\mathbf{P}_i) - 2\sum_{i.k} w_{ij}\mathbf{Tr}(\mathbf{P}_j^T\boldsymbol{\mu}_j\boldsymbol{\mu}_i^T\\
&\mathbf{P}_i) + \sum_{i,j,k} w_{ij}w_{ik}\mathbf{Tr}(\mathbf{P}_j^T\boldsymbol{\mu}_j\boldsymbol{\mu}_k^T\mathbf{P}_k) + \sum_i \mathbf{Tr}(\mathbf{X}_i\mathbf{X}_i^T).
\end{aligned}$$

To solve the projection matrix $\mathbf{P}_s$, the terms that are not related to $\mathbf{P}_s$ are eliminated. Then the above equation becomes

$$\begin{aligned}
&\mathbf{Tr}(\mathbf{P}_s^T(\boldsymbol{\mu}_s\boldsymbol{\mu}_s^T - \eta\mathbf{X}_s\mathbf{X}_s^T)\mathbf{P}_s) - 2\sum_j w_{sj}\mathbf{Tr}(\mathbf{P}_j^T\boldsymbol{\mu}_j\boldsymbol{\mu}_s^T\mathbf{P}_s))\\
&-2\sum_i w_{is}\mathbf{Tr}(\mathbf{P}_s^T\boldsymbol{\mu}_s\boldsymbol{\mu}_i^T\mathbf{P}_i)) + \sum_{i,k} w_{is}w_{ik}\mathbf{Tr}(\mathbf{P}_s^T\boldsymbol{\mu}_s\boldsymbol{\mu}_k^T\mathbf{P}_k)\\
&+\sum_{i,j} w_{is}w_{ik}\mathbf{Tr}(\mathbf{P}_j^T\boldsymbol{\mu}_j\boldsymbol{\mu}_s^T\mathbf{P}_s)\\
&=\mathbf{Tr}(\mathbf{P}_s^T\mathbf{A}_s\mathbf{P}_s) + 2\mathbf{Tr}(\mathbf{P}_s^T\mathbf{B}_s).
\end{aligned}$$

In Eq.(8), the matrix $\mathbf{A}_s$ captures the relationships between samples within the s-th superpixel, while the matrix $\mathbf{B}_s$ describes the relationship between samples across different superpixels. To further elaborate, the first term in $\mathbf{B}_s$ incorporates the second-order links within the graph $\mathcal{G}$, where both $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_j$ are in the neighborhoods of $\boldsymbol{\mu}_i$. The second term in $\mathbf{B}_s$ considers the first-order links in the graph, with $\boldsymbol{\mu}_s$ located in the neighborhoods of $\boldsymbol{\mu}_i$ or vice versa. In this way, the proposed CSPCA can incorporate the relations of samples between superpixels and solve the dispersion problem in SuperPCA.

Minimizing the problem in Eq.(8) is challenging because the solution is restricted to a Stiefel manifold, which makes the problem not only non-convex but also numerically expensive [50]. In order to address this issue, the solver *Stiefel manifold* in Manopt[1], which is a toolbox specifically designed for optimization on manifolds, is adopted in our method. The *Stiefel manifold* solver requires the derivative of Eq.(8), given by:

$$\frac{\partial\mathcal{L}(\mathbf{P}_1,\ldots,\mathbf{P}_s,\ldots,\mathbf{P}_N)}{\partial\mathbf{P}_s} = 2\mathbf{A}_s\mathbf{P}_s + 2\mathbf{B}_s. \tag{9}$$

The optimization procedure of the proposed CSPCA is outlined in Algorithm 1.

[1]https://www.manopt.org/reference/manopt/manifolds/stiefel/stiefelfactory.html

Algorithm 1 Optimization procedures of CSPCA

Input: An HSI $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$, where $\mathbf{X}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{iN_i}\}$ denotes the samples from the i-th superpixel, and N_i is number of samples within this superpixel. The mean vectors set $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_N\}$, the balance weight η, the number of iterations T, the weighting matrix $\mathbf{G}$.

Output: The projection matrices $\{\mathbf{P}_1^*, \mathbf{P}_2^*, \ldots, \mathbf{P}_N^*\}$.

```
Initial the projection matrices {P_1^(0), P_2^(0), ..., P_N^(0)};
for t = 0 to T do
    for s = 1 to N do
        Calculate the loss A_s and B_s in Eq.(7), the derivative
        ∂L(P_s)/∂P_s in Eq.(9) using {P_i^(t)}, i = 1, ..., s−1, s+1, ..., N;
        Calculate the optimal P_s^(t) by the solver Stiefel manifold;
    end for
end for
return {P_1^(T), P_2^(T), ..., P_N^(T)}.
```

IV. Experiments

A. Data sets

To evaluate the proposed CSPCA, experiments are performed on several public HSI data sets, which are Indian Pines (IP), the University of Pavia (PU), Salinas (Sa), WHU-Hi-LongKou (LK) [51] [52], and WHU-Hi-HanChuan (HC) [51] [52]. The detailed information of these data sets is as follows.

1) Indian Pines: The Indian Pines data set was collected by the AVIRIS sensor in 1992. It contains 145×145 pixels with 224 spectral bands. 24 bands are removed due to their low signal-to-noise properties. In this way, each pixel is with 200 bands in our experiments. This data set consists of 10249 labeled samples from 16 different classes, the number of samples in each class is listed in TABLE I. Fig. 4a presents the false color map of Indian Pines. The false color map of PU is presented in Fig. 4a.
2) University of Pavia: This data set was collected by the ROSIS sensor, which recorded a scene in Pavia, northern in Italy. It contains 610×340 pixels and 103 spectral bands after removing the noise and water-absorption bands. This data set has 42776 samples from 9 classes. The number of samples in each class is also presented in TABLE I. The false color map of PU is presented in Fig. 4b.
3) Salinas: This data set was acquired using the AVIRIS sensor in Salinas Valley, USA. It comprises 610×340 pixels with 224 spectral bands. After eliminating 20 noisy and water bands, 204 bands are used in our experiments. The data set includes 53129 samples from 16 different classes. The number of samples in each class is also outlined in TABLE I. The false color map of SA is presented in Fig. 4c.
4) WHU-Hi-LongKou: This data set was acquired using an 8-mm focal length Headwall Nano-Hyperspec sensor in Longkou Town, Hubei Province, China. The camera

TABLE I
NUMBER OF SAMPLES IN THE INDIAN PINES, UNIVERSITY OF PAVIA, SALINAS, WHU-HI-LONGKOU, AND WHU-HI-HANCHUAN.

	Indian Pines		University of Pavia		Salinas		WHU-Hi-LongKou		WHU-Hi-HanChuan	
	Class name	Numbers	Class name	Numbers	Class name	Numbers	Class name	Numbers	Class name	Numbers
c1	Alfalfa	46	Asphalt	6631	Broccoli green weeds1	2009	Corn	34511	Strawberry	44735
c2	Corn-notill	1428	Meadows	18649	Broccoli green weeds2	3726	Cotton	8374	Cowpea	22753
c3	Corn-mintill	830	Gravel	2099	Fallow	1976	Sesame	3031	Soybean	10287
c4	Corn	237	Tress	3064	Fallow rough plow	1394	Broad-leaf soybean	63212	Sorghum	5353
c5	Grass-pasture	483	Mental sheets	1345	Fallow smooth	2678	Narrow-leaf soybean	4151	Water spinach	1200
c6	Grass-tress	730	Bare soil	5029	Stubble	3959	Rice	11854	Watermelon	4533
c7	Grass-pasture-mowed	28	Bitumen	1330	Celery	3579	Water	67056	Greens	5903
c8	Hay-windrowed	478	Bricks	3682	Grapes untrained	11271	Roads and houses	7124	Trees	17978
c9	Oats	20	shadow	947	Soil vineyard develop	6203	Mixed weed	5229	Grass	9469
c10	Soybean-nottill	972			Corn senesced green seed	3278			Red roof	10516
c11	Soybean-mintill	2455			Lettuce romaine 4wk	1068			Gray roof	16911
c12	Soybean-clean	593			Lettuce romaine 5wk	1927			Plastic	3679
c13	Wheat	205			Lettuce romaine 6wk	916			Bare soil	9116
c14	Woods	1265			Lettuce romaine 7wk	1070			Road	18560
c15	Buildings-grass-trees-dirves	386			Vineyard untrained	7268			Bright object	1136
c16	Stone-steel-towers	93			Vineyard vertical trellis	1807			Water	75401
	Total number	10249	Total number	42776	Total number	54129	Total number	204542	Total number	257580

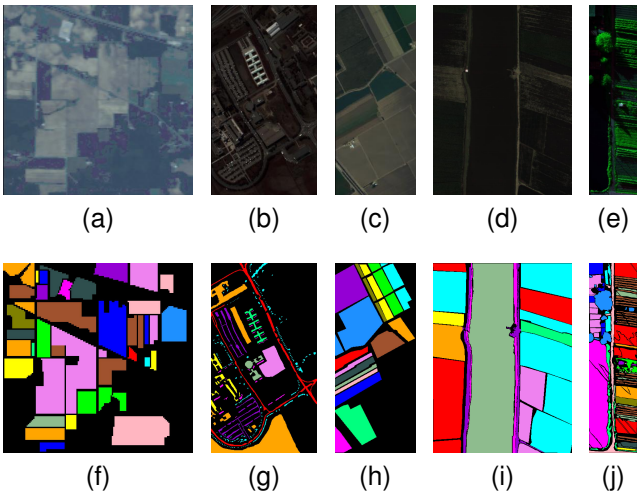


Fig. 4. The false color images and ground truth maps. The false color images of (a) IP. (b) PU. (c) SA. (d) LK. (e) HC. The ground truth maps of (f) IP. (g) PU. (h) SA. (i) LK. (j) HC.

was equipped on a DJI Matrice 600 Pro UAV platform. This data set contains 550×400 pixels with 270 spectral bands from 400 to 1000 nm. There are 204542 labeled samples from 9 different classes, the number of samples in each class is listed in TABLE I. The false color map of LK is illustrated in Fig. 4d.

5) WHU-Hi-HanChuan: This data set was acquired in Hanchuan, Hubei Province, China, with a 17-mm focal length Headwall Nano-Hyperspec sensor equipped on a Leica Aibot X6 UAV platform. There are 1217×303 pixels in the image, and each pixel has 274 spectral bands. There are 257580 labeled samples from 16 different classes, the number of samples in each class is also presented in TABLE I. The false color map of HC is illustrated in Fig. 4e.

B. Experimental setup

In this experiment, several well-known FE techniques, including several methods only using spectral information, such as PCA, LPP, LRCRP [38], and TDWR [17], as well as methods that incorporate spatial-spectral information through superpixel segmentation, such as tensorSSA [22], SuperPCA [25], Reconstruction SuperPCA (RSuperPCA) [39], Concatenated SuperPCA (CSuperPCA) [39], SuperULDA [40], SuperPCA-DA [41], and BAMS [27]. Additionally, we establish the performance of features without FE as a baseline, referred to as raw features in the experiments. The original HSIs are used without any further preprocessing, and the local filtered procedure in S3-ULDA is also eliminated. The implementations of PCA, LPP, SuperPCA, RSuperPCA, CSuperPCA, LRCRP, tensorSSA, TWDR, SuperULDA, SuperPCA-DA, and BAMS can be accessed from the authors' websites, respectively. To determine the category of each sample, we first apply the samples in HSI to the FE methods. Subsequently, we use a Support Vector Machine (SVM) to classify the samples based on the low-dimensional representation obtained from FE techniques. All experiments are conducted using Matlab R2019b, ensuring a consistent and standardized evaluation framework across the compared methods.

To test the proposed method, the overall accuracy (OA), average accuracy (AA), and kappa are used to evaluate the performance of different FE techniques. The evaluation is conducted on 10 random splits sets in [25], with separate training and testing sets.

To evaluate the performance of our proposed method, two experiments are conducted on a desktop computer equipped with a 2.5GHz Core i5-12400F CPU and 32GB RAM. In the first experiment, we focus on investigating the optimal parameters in CSPCA. Based on the results of the first experiment, we aim to assess the effectiveness of CSPCA compared to SuperPCA and other common FE techniques in the second experiment.

C. Parameters tuning

In the proposed CSPCA, several parameters need to be predefined when applying the method to HSI data. These include the number of superpixels, denoted by N, which controls the granularity of superpixel segmentation. The number of nearest neighbors in LLE, denoted by K, influences the construction of the local neighbors. The balancing weight, denoted by η, adjusts the trade-off between the local and global information

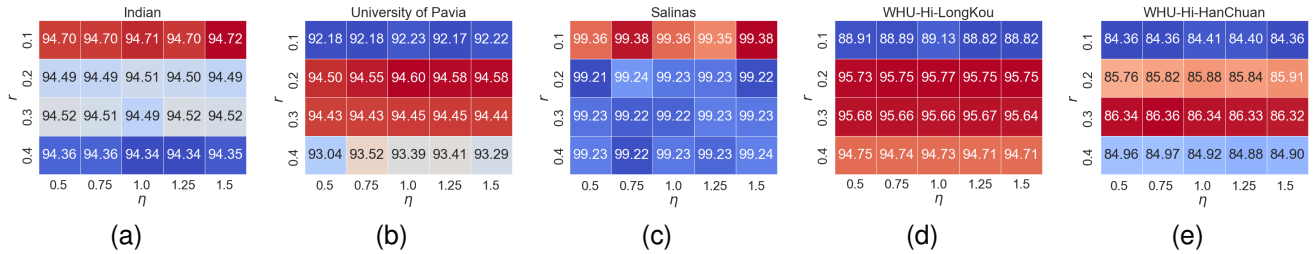


Fig. 5. The OAs with different parameters. (a) illustrates the OAs obtained with different values for r and η on Indian Pines data set. Similarly, (b), (c), (d) and (e) illustrate the OAs with the same setting on the University of Pavia, Salinas, WHU-Hi-LongKou and WHU-Hi-HanChuan data sets, respectively.

in CSPCA; The dimensionality of the low-dimensional space, denoted by l, determines the size of the extracted features.

For the balancing weight η and the number of nearest neighbors K are involved in Eq. 6, we first perform a joint search for optimal values across the five data sets. Following the setting of SuperPCA, the dimensionality of the subspace is fixed at 30, with 100 superpixels for IP and SA, and 20 for the other three data sets. 30 samples from each category are chosen to train the SVM. For η, its value is varied within the range of $[0.5, 0.75, 1, 1.25, 1.5]$. Additionally, since the number of nearest neighbors K is inherently related to the number of superpixels N, the ratio $r = \frac{K}{N}$ is evaluated by testing values in the range of $[0.1, 0.2, 0.3, 0.4]$. The experimental results are illustrated in Fig. 5, providing insights into the influence of different values of η and r on the performance of CSPCA.

Fig. 5 reveals that the the ratio of r is the main factor influencing the performance of CSPCA. On the IP data sets, the OA of CSPCA achievs best when $r = 0.1$ and then decreases with increasing r. It is important to note that the number of superpixels N is fixed at 100 in IP, and a higher value of r means a larger K, which could potentially lead to confusion between samples from different categories. A similar phenomenon can be observed in the SA data set, where a N is also set to 100. For the PU and LK data sets, the OA achieves best when $r = 0.2$, this may be attributed to the fact that too few samples cannot efficiently capture the geometrical relationship of the data. Different from PU and LK, the OA achieves best when $r = 0.3$ on the HC data set, this may be because there are more samples in HC, and a small K cannot capture the relationship of the data.

Based on the results from the above experiments, we empirically set the balance weight η to 1 for all five data sets. Additionally, we empirically determine the ratio r to be 0.1 for data sets with a large number of superpixels, 0.2 for data sets with a moderate number of superpixels, and 0.3 for data sets with very few superpixels. Consequently, the number of superpixels is small. Thus, the number of nearest neighbors K is set to 10 for IP and SA data sets, and 4 for the other three data sets.

We then evaluate the impact of dimensionality of the reduced feature, which is denoted as l, and several experiments are performed. In these experiments, K is set according to the previously determined values, and we fix $\eta = 1$. The subspace dimension l is varied from 5 to 95 with an interval of 10, and the corresponding overall classification accuracies

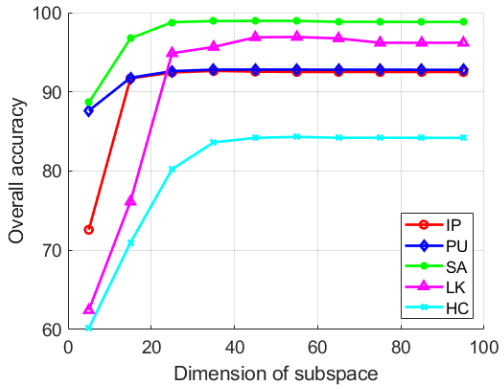


Fig. 6. The OAs with different reduced feature dimensions in Indian Pines, University of Pavia, Salinas, WHU-Hi-LongKou, and WHU-Hi-HanChuan.

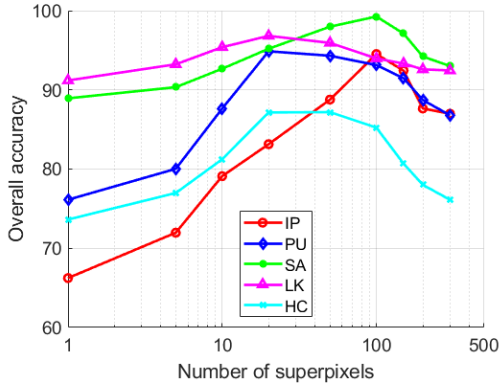


Fig. 7. The OAs with different number of superpixels in Indian Pines, University of Pavia, Salinas, WHU-Hi-LongKou, and WHU-Hi-HanChuan.

are evaluated using 20 training samples per class with SVM.

The experimental results are illustrated in Fig. 6, and several insights can be observed. For example, concerning the dimensionality l, the best OA of 92.94% is attained at $l = 35$ for IP. Conversely, the worst OA of 72.61% is noted at $l = 5$. Comparable patterns are observed for PU, with OA ranging from 88.01% to 93.11%, and for SA, where OA fluctuates between 89.09% and 99.03%.

It is apparent that when $l = 5$, the classification accuracies are consistently low on all data sets, which aligns with common intuition. A limited number of features hinders the capacity to convey adequate discriminative information.

Nevertheless, with increasing values of l, the OAs exhibit a steady improvement. Eventually, a point is reached where the growth of OAs becomes marginal, suggesting that the available discriminative information is already effectively utilized. Further increasing l would escalate the complexity and computational demands of the classifier without substantial performance enhancements.

Finally, we conduct several experiments to analyze the impact of the number of superpixels, which is denoted by N. In these experiments, the balancing weight η is fixed to 1, and the dimensionality l is set to 30. Based on the results from the previous experiments, we set $K = \lfloor N \times 0.3 \rfloor$ when $N < 10$, and $K = \lfloor N \times 0.2 \rfloor$ when $10 \leq N < 50$, and $K = \lfloor N \times 0.1 \rfloor$ when $N \geq 50$, where $\lfloor \cdot \rfloor$ denotes the floor function. we evaluate the corresponding overall classification accuracies using 30 training samples per class with SVM as N ranges in $[1, 5, 10, 20, 50, 100, 150, 200, 300]$.

The experimental results are shown in Fig. 7. From the results, it can be observed that the classification accuracies first increase and then decrease as N increases. However, the number of superpixels has varying effects on different data sets. For instance, when $N = 100$, the OAs are highest on the IP and SA data sets. In contrast, the best OAs on PU and LK data sets are achieved when $N = 20$, while on the HC data set, the optimal OA occurs when $N = 50$, though the accuracy remains close to that obtained when $N = 20$.

D. Comparison with other FE techniques

We select $Tr = 5, 10, 20, 30$ samples from each class for training the SVM, with 5 samples from each class for validation, and the remaining samples are used for testing. For classes with an insufficient number of samples, we select a maximum of half of the total samples from them. Specifically, the SVM uses the Radius Basis Function (RBF) kernel, and the optimal γ in RBF is determined through cross-validation, with values searched from $[0.01, 0.1, 1, 5, 10, 15, 20, 30, 40, 50, 100, 200, 300, 400, 500]$. Based on the experimental results in Section III.C, the number of superpixels N is set to 100 for both IP and SA data sets, and set to 20 for PU, LK, and HC data sets. In RSuperPCA, the number of nearest spatial neighbors is set to 15 , as recommended by [39]. In CSPCA, the number of nearest neighbors in Eq.(3) is set to 10 for IP and SA, and 4 for PU and LK, and 6 for HC, and the balance weight $\eta = 1$ for all the data sets.

The average OA, AA, and Kappa of all algorithms on five public data sets are reported in TABLE II when the dimensionality of the subspace is set to 30. It should be noted that PCA is further applied to the features obtained by TWDR, ULDA, and BAMS, for their dimensionalities are larger than 30. The dimensionality of the feature obtained by SuperPCA-DA is equal to the dimensionality of the raw feature. To clearly show the best result on each data set, we highlight them in bold. From the experimental results, several conclusions can be observed as follows.

1. The performance of CSPCA outperforms SuperPCA in all tested scenarios, implying that the new objective function proposed in Eq.(6) effectively addresses the intra-class difference problem caused by the SuperPCA. To validate our findings, we choose a random split from IP data set and visually present the classification results in Fig. 8. When comparing Fig. 8h with Fig. 8n, it is evident that CSPCA significantly improves the classification accuracy of samples in *Soybean-mintill*. Specifically, a blue box highlights a region where some samples are misclassified. This is primarily due to SuperPCA finding the subspace based solely on the samples within the superpixel, leading to the distribution of samples in subspace overlaps with samples from other categories. By considering the relationship between superpixels, CSPCA achieves a tighter distribution of samples from the same category, resulting in an improvement in Fig. 8n. We also provide extra experiments results using random splits from PU, SA, LK, and HC data sets in the Supplementary Material to further observe the visual performances of the compared methods.

2. Compared to other spatial-spectral FE algorithms, CSPCA demonstrates superior classification accuracies in most cases. Several key observations can be drawn from the results. Firstly, RSuperPCA achieves better performance on SA when number of training samples is 5. This improvement can be attributed to the transmission of label information between superpixels during the reconstruction operation, which is useful for the samples within superpixels lacking labeled data. Secondly, the BAMS outperforms CSPCA in several cases under the same setting due to its multiscale determination scheme, and the proposed CSPCA only chooses one superpixel segmentation result to extract the spatial-spectral feature. Thirdly, it is surprising that the CSuperPCA, which merges global features directly with local features obtained from SuperPCA, demonstrates competitive performance in the experiments. We imply that CSPCA performs worse than RSuperPCA, and BAMS in this case because the number of samples in SA is large, and the number of superpixels is also significant. When the training samples are very limited, insufficient training information is fed into SVM, leading to suboptimal performances for CSPCA. The experiments highlight the utility of reconstruction and concatenation operations in HSI tasks, which can be integrated into CSPCA to further enhance its performance.

3. The performances of spectral FE algorithms are notably surpassed by spatial-spectral approaches. Among the spectral FE algorithms, manifold-based methods such as LRCRP, LPP, and TWDR, always have superior classification accuracies. This can be attributed to the capability of the manifold-based FE algorithms to preserve the local structure of the data in the subspace, as opposed to traditional FE algorithms like PCA, which focus on capturing the global information of the data in the subspace. Interestingly, the performances of raw feature and PCA outperform manifold-based and several spatial-spectral FE algorithms on SA with 5 training samples per class. This phenomenon is primarily due to the small proportion of training samples, meanwhile the number of superpixels is large. With a limited number of training samples, even if the local data structure is preserved, the classification boundary might not be accurately delineated. Moreover, the

TABLE II
CLASSIFICATION PERFORMANCES OF THE 13 METHODS ON INDIAN PINES, UNIVERSITY OF PAVIA, SALINAS, WHU-HI-LONGKOU, AND WHU-HI-HANCHUAN. THE **T.N.s/C** REPRESENTS THE NUMBER OF TRAINING SAMPLES FROM EACH CLASS.

data set	T.N.s/C	metric	raw	PCA	LPP	LRCRP	TWDR	tensor SSA	Super PCA	RSuper PCA	CSuper PCA	Super ULDA	Super PCA-DA	BAMS	CSPCA
IP	5	OA(%)	44.53	44.72	47.05	49.32	45.03	46.95	63.49	53.23	64.29	68.76	71.09	70.61	**71.15**
		AA(%)	58.16	57.97	61.39	58.64	63.17	62.32	78.11	69.19	78.27	80.29	80.59	80.47	**80.66**
		kappa	0.3864	0.3901	0.4143	0.4287	0.3900	0.4042	0.5869	0.4785	0.5974	0.6433	0.6712	0.6697	**0.6722**
	10	OA(%)	54.99	55.02	56.69	57.79	54.10	62.78	82.67	69.58	83.12	83.15	83.09	82.68	**83.24**
		AA(%)	68.39	67.85	69.01	65.82	68.88	75.12	89.15	80.98	89.65	89.68	**89.76**	88.82	89.51
		kappa	0.5072	0.5013	0.5181	0.5251	0.4901	0.5822	0.8014	0.6593	0.8086	0.8092	**0.8100**	0.8053	0.8092
	20	OA(%)	63.78	62.13	64.96	64.68	66.33	73.34	92.39	80.88	92.33	92.11	91.38	90.21	**92.39**
		AA(%)	74.97	73.19	74.02	73.23	76.12	83.23	**95.01**	88.69	94.98	94.89	94.43	93.44	94.99
		kappa	0.5942	0.5769	0.6113	0.6034	0.6229	0.6992	0.9116	0.7825	0.9138	0.9090	0.9029	0.8901	**0.9142**
	30	OA(%)	68.75	66.23	69.69	69.08	73.37	80.01	94.45	85.570	**94.71**	93.22	93.89	93.15	94.55
		AA(%)	78.44	76.40	78.71	79.69	81.18	87.64	96.28	91.23	96.25	96.19	96.09	95.55	**96.38**
		kappa	0.6489	0.6205	0.6592	0.6517	0.6988	0.7718	0.9312	0.8359	**0.9388**	0.9345	0.9309	0.9248	0.9379
PU	5	OA(%)	64.50	65.19	68.12	67.07	65.12	65.66	68.27	66.39	76.69	72.69	65.19	74.71	**77.21**
		AA(%)	67.49	68.19	69.87	49.11	72.75	72.21	72.12	73.23	80.29	77.37	68.34	**81.59**	77.29
		kappa	0.5413	0.5522	0.5765	0.3279	0.5601	0.5632	0.5958	0.5698	0.7011	0.6523	0.5533	0.6812	**0.7088**
	10	OA(%)	70.05	69.12	69.38	71.04	73.12	77.35	83.14	81.29	89.13	84.00	70.25	84.09	**89.89**
		AA(%)	75.57	75.61	75.89	75.11	75.42	82.34	85.22	85.51	**90.13**	87.39	75.89	89.18	89.33
		kappa	0.6174	0.6152	0.6151	0.6249	0.6441	0.7108	0.7842	0.7579	0.8539	0.7938	0.6211	0.7975	**0.8652**
	20	OA(%)	74.52	75.17	74.51	75.79	74.65	85.62	87.04	86.70	90.95	87.48	75.28	90.65	**92.40**
		AA(%)	78.88	79.68	79.32	79.50	80.13	89.46	89.53	89.26	91.28	89.83	79.96	**93.88**	93.00
		kappa	0.6863	0.6805	0.6700	0.6789	0.6906	0.8005	0.8370	0.8161	0.8882	0.8382	0.6960	0.8924	**0.8982**
	30	OA(%)	76.04	74.22	75.96	77.09	75.50	87.34	90.39	86.20	93.30	90.36	77.43	92.71	**93.83**
		AA(%)	81.19	80.90	81.13	80.37	79.85	91.63	91.83	90.93	93.69	91.55	81.71	95.01	**93.84**
		kappa	0.6847	0.6900	0.6956	0.6960	0.6807	0.8451	0.8740	0.8391	0.9054	0.8742	0.7057	0.9094	**0.9178**
SA	5	OA(%)	81.30	81.35	79.20	78.17	78.15	81.91	76.38	79.87	78.43	82.02	84.05	**88.15**	78.36
		AA(%)	84.95	87.78	86.48	85.84	86.15	86.61	81.17	83.91	84.10	85.07	89.42	**91.22**	82.67
		kappa	0.7884	0.7968	0.7717	0.7571	0.7697	0.7898	0.7463	0.7916	0.7657	0.8081	0.8257	**0.8870**	0.7550
	10	OA(%)	85.01	84.01	82.96	84.83	84.88	86.63	92.68	92.87	93.17	92.65	91.15	91.65	**94.09**
		AA(%)	88.81	88.99	89.12	89.81	90.04	91.16	93.84	94.07	94.15	93.71	93.80	92.89	**94.72**
		kappa	0.8356	0.8298	0.8132	0.8383	0.8423	0.8530	0.9149	0.9354	0.9104	0.9339	0.9019	0.9221	**0.9359**
	20	OA(%)	87.46	87.41	87.77	86.44	86.87	89.88	97.14	98.24	97.28	97.89	95.13	98.15	**98.26**
		OA(%)	92.56	92.35	92.03	92.60	93.60	92.68	97.88	98.36	97.60	96.98	95.15	98.47	**98.68**
		kappa	0.8637	0.8515	0.8632	0.8496	0.8693	0.8844	0.9590	0.9820	0.9730	0.9745	0.9710	0.9630	**0.9842**
	20	OA(%)	88.29	87.25	89.39	87.93	90.46	91.46	97.17	98.64	97.18	98.52	97.51	99.09	**99.24**
		AA(%)	93.22	92.19	92.96	93.22	94.40	95.21	98.40	98.26	98.42	97.61	97.48	97.55	**98.99**
		kappa	0.8673	0.8658	0.8780	0.8729	0.8817	0.8976	0.9825	0.9657	0.9719	0.9599	0.9744	0.9866	**0.9902**
LK	5	OA(%)	78.81	79.78	78.90	69.87	78.58	83.04	78.19	84.09	83.51	75.56	80.30	**85.56**	84.39
		AA(%)	67.88	68.31	66.92	60.11	66.07	70.42	81.13	75.70	84.10	65.16	69.64	**87.13**	84.57
		kappa	0.7368	0.7379	0.7333	0.6090	0.7083	0.7821	0.7301	0.8062	0.8058	0.6870	0.7501	**0.8311**	0.8307
	10	OA(%)	85.61	86.01	86.21	70.65	85.65	88.07	87.80	89.89	90.33	82.74	87.77	91.86	**92.44**
		AA(%)	81.43	80.72	83.73	65.68	81.48	81.72	88.80	87.49	90.04	71.91	84.10	91.07	**92.14**
		kappa	0.8169	0.8139	0.8154	0.6478	0.8296	0.8459	0.8628	0.8615	0.8762	0.7745	0.8564	0.9101	**0.9123**
	20	OA(%)	89.22	89.65	91.34	78.74	89.02	89.82	92.93	92.65	93.50	87.37	91.93	94.09	**94.68**
		AA(%)	86.81	87.90	89.85	75.30	86.85	88.75	91.88	89.96	93.49	78.65	92.35	92.30	**93.48**
		kappa	0.8604	0.8631	0.8850	0.7362	0.8726	0.8830	0.9118	0.9146	0.9183	0.8307	0.9102	0.9285	**0.9344**
	30	OA(%)	90.83	90.83	91.96	81.11	92.20	92.43	93.62	94.08	92.64	88.43	94.53	94.97	**96.05**
		AA(%)	89.59	89.50	91.87	77.86	91.33	91.26	94.37	92.97	94.63	83.73	94.66	94.51	**95.47**
		kappa	0.8857	0.8714	0.9025	0.7580	0.8961	0.8980	0.9182	0.9082	0.9155	0.8571	0.9144	0.9468	**0.9508**
HC	5	OA(%)	58.62	57.88	57.79	52.30	57.08	57.79	61.92	61.51	62.68	62.28	59.76	58.18	**64.22**
		AA(%)	49.41	48.87	48.61	42.90	48.95	47.30	**64.59**	53.18	63.06	59.95	52.92	56.01	63.57
		kappa	0.5266	0.5321	0.5267	0.4685	0.5184	0.5005	0.5730	0.5620	0.5934	0.5755	0.5376	0.5704	**0.6008**
	10	OA(%)	63.30	64.20	65.22	61.05	69.04	67.49	73.78	70.67	74.97	69.20	67.07	70.07	**75.69**
		AA(%)	57.78	56.16	57.12	52.08	61.74	73.89	61.87	63.86	74.59	72.45	63.35	71.78	**76.61**
		kappa	0.5915	0.6100	0.5922	0.5645	0.6467	0.7011	0.6412	0.6693	0.7111	0.7125	0.6293	0.6521	**0.7179**
	20	OA(%)	71.41	70.44	69.79	70.73	73.90	77.42	80.41	77.23	82.85	81.15	79.27	79.92	**83.55**
		AA(%)	64.77	64.06	62.94	63.82	70.25	72.76	80.63	72.10	80.76	81.40	78.36	80.99	**83.28**
		kappa	0.6701	0.6574	0.6559	0.6679	0.6971	0.7316	0.7642	0.7362	0.7969	0.7647	0.7636	0.7700	**0.8079**
	30	OA(%)	72.74	73.28	74.02	73.62	78.97	81.51	82.62	80.51	86.11	81.50	80.90	84.40	**85.88**
		AA(%)	67.90	67.83	68.19	67.16	76.87	83.40	77.27	76.26	85.40	83.68	78.60	83.04	**85.97**
		kappa	0.7034	0.6797	0.6976	0.6960	0.7832	0.7892	0.7986	0.8427	0.8367	0.8165	0.7959	0.8344	**0.8459**

large number of superpixels results in only a few superpixels containing labeled samples. Consequently, raw features and PCA, which do not compromise the global information of the data, yield superior performance in such scenarios.

V. DISCUSSION

The main contribution of this work is the development of the CSPCA algorithm, which effectively addresses the dispersion problem observed in SuperPCA. The dispersion problem refers to the phenomenon where samples from the same category but located in different superpixels are scattered far from each other in the feature space, as shown in Fig. 2. This issue is particularly problematic for classification tasks because it increases intra-class variance, leading to more complex decision boundaries and higher risks of overfitting.

TABLE III
CLASSIFICATION ACCURACIES FOR EACH CLASS USING 20 TRAINING SAMPLES ON INDIAN PINES.

	raw	PCA	LPP	LRCRP	TWDR	tensor SSA	Super PCA	RSuper PCA	CSuper PCA	Super ULDA	Super PCA-DA	BAMS	CSPCA
c1	89.23	87.31	92.31	94.23	84.62	79.62	100.00	100.00	100.00	100.00	100.00	95.38	100.00
c2	45.68	39.67	41.97	53.09	44.67	53.15	85.09	85.09	84.61	83.76	86.61	79.34	85.20
c3	58.74	59.81	56.67	67.32	60.86	52.42	90.99	90.99	91.46	90.38	91.49	88.47	90.69
c4	73.00	70.83	65.44	78.11	63.13	86.96	93.69	93.69	94.52	94.38	93.18	89.95	94.01
c5	83.78	83.37	92.87	87.04	74.73	75.72	93.28	93.28	93.17	94.97	92.94	91.43	93.26
c6	88.31	87.04	92.39	91.41	82.54	75.00	98.27	98.27	98.56	97.87	98.08	99.59	97.96
c7	92.86	92.14	78.57	95.71	85.71	89.29	97.14	97.14	97.14	97.14	96.43	97.14	97.14
c8	87.42	86.35	88.86	96.51	91.48	91.55	99.61	99.61	99.61	99.67	99.61	99.19	99.61
c9	80.00	71.00	60.00	100.00	70.00	84.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
c10	63.31	62.71	74.26	73.49	64.71	60.97	87.71	87.71	87.63	87.39	87.46	88.28	87.73
c11	51.00	50.26	53.51	64.41	53.59	56.10	92.64	92.64	93.47	90.71	87.49	88.30	93.31
c12	61.41	58.94	64.40	72.25	70.33	68.94	92.58	92.58	92.57	93.63	92.15	93.96	92.53
c13	96.70	91.43	94.59	98.59	92.43	94.59	99.46	99.46	99.46	99.46	99.46	99.46	99.46
c14	76.68	74.79	76.14	87.72	79.20	84.10	95.71	95.71	95.73	98.10	98.35	95.90	95.73
c15	57.10	56.42	54.92	77.49	64.21	75.66	95.93	95.93	95.60	96.07	93.74	95.57	95.63
c16	95.34	95.89	97.26	97.12	93.15	93.84	98.49	98.49	98.36	97.67	97.67	97.81	98.49
OA	63.81	62.16	65.36	73.56	64.71	66.51	92.45	79.20	92.63	92.13	91.59	90.40	92.57
AA	75.04	73.25	74.01	83.41	73.46	76.37	95.04	87.84	95.12	95.08	94.67	93.74	95.05
kappa	0.5945	0.5764	0.6120	0.7011	0.6036	0.6231	0.9136	0.7650	0.9157	0.9101	0.9042	0.8907	0.9149

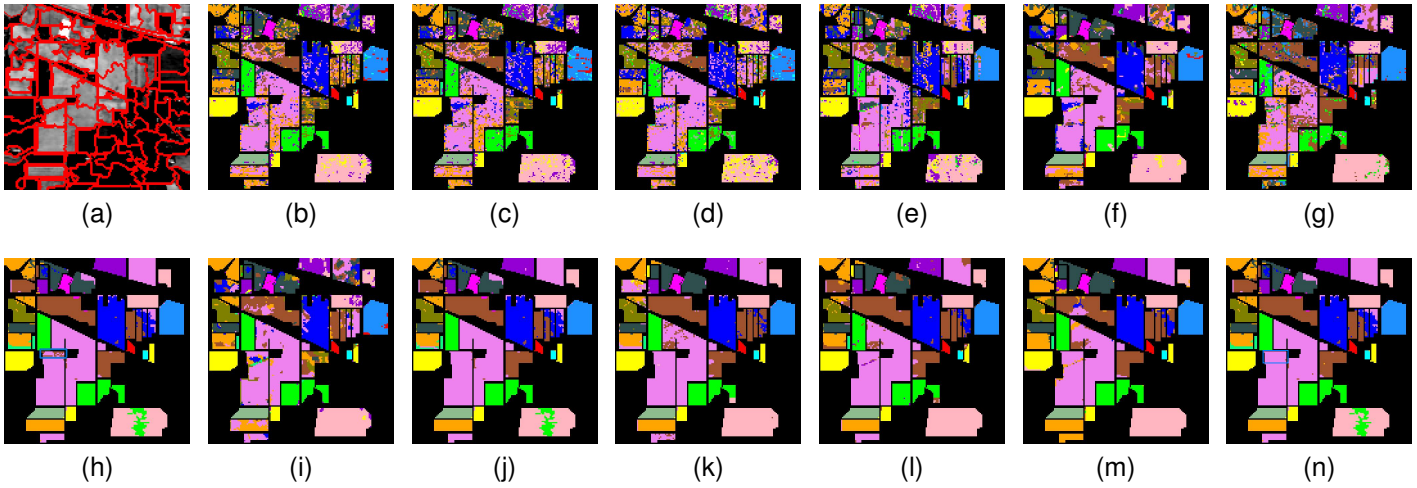


Fig. 8. Classification maps are from different methods on the Indian Pines data set. (a) The superpixels result by ERS. (b) Raw feature(67.89%). (c) PCA(65.36%). (d) LPP(66.12%). (e) LRCRP(65.11%). (f) tensorSSA(76.42%). (g) TWDR(69.70). (h) SuperPCA(94.80%). (i) RSuperPCA(83.34%). (j) GSuperPCA(94.91%). (k) SuperULDA (93.50%). (l) SuperPCA-DA (92.16%). (m) BAMS(90.33%). (n) CSPCA (95.02%).

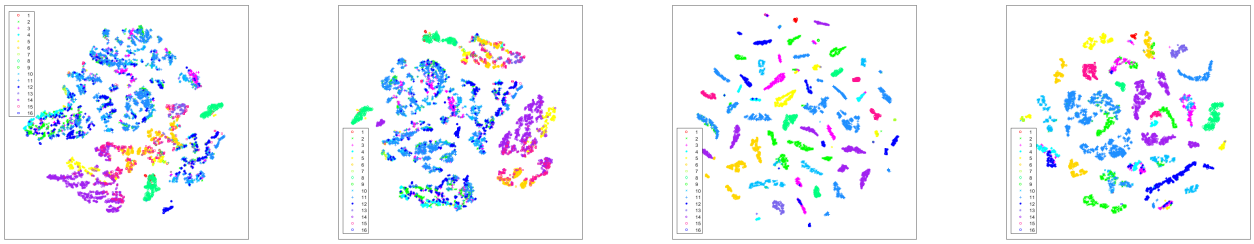
Prior studies [29] [30] have demonstrated that reducing the intra-class variance can improve classification performance. In contrast, PCA does not exhibit this issue. This raises a common question: why does SuperPCA still outperform PCA in HSI classification tasks? To explore this, we use t-SNE to visualize the distribution of all the samples in the IP data set, as shown in Fig. 9.

It can be observed that the raw features and PCA-extracted features of different classes are not well-separated, while SuperPCA and CSPCA produce more distinct class separations, despite the presence of the dispersion problem in SuperPCA. When applying SVM for classification, kernel methods can alleviate the dispersion issue, but struggle with classes that exhibit strong overlap. CSPCA resolves this issue more effectively by reducing intra-class dispersion while maintaining inter-class separability. This helps explain why CSPCA performs best among the three methods, and why SuperPCA still outperforms PCA in these tasks.

To delve deeper into the reason, PCA searches for a single subspace $\mathbf{P}$ to extract features, which often makes it difficult or impossible to find a subspace where all features are separable. On the other hand, SuperPCA and CSPCA identify a set of subspaces tailored for each group of features, resulting in better separability of the extracted features. Furthermore, this work demonstrates that reducing intra-class dispersion can benefit the HSI classification tasks.

VI. CONCLUSION

In this paper, we have identified the dispersion problem in SuperPCA and introduced CSPCA, a new FE technique, to address this problem. The core concept of CSPCA is to incorporate the relations between samples from different superpixels when finding the projection for each superpixel. We have formulated a novel objective function combining the



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

Fig. 9. The distributions of (a) raw features, (b) features extracted by PCA, (c) features extracted by SuperPCA, and (d) features extracted by CSPCA in IP data set are shown. Different colors represent different categories in these figures.

global information between superpixels and local information within each superpixel. Experiments on five HSI data sets have validated the effectiveness of the proposed method.

However, CSPCA underperforms compared to several state-of-the-art FE methods in certain scenarios. Furthermore, determining the optimal number of superpixels, which can affect its performance, is also a limitation of CSPCA. In future work, we aim to enhance CSPCA's performance. First, reconstructing each sample from its neighbors, a technique shown to effectively reduce noise in HSI classification could serve as a preprocessing step to improve CSPCA. Additionally, incorporating multilevel superpixel segmentation [25] into the CSPCA may further enhance its performance. Moreover, we plan to explore a method for automatically determining the number of superpixels, which could optimize the superpixel selection process and improve the overall effectiveness of CSPCA.

REFERENCES

[1] X. Zheng, H. Cui, and X. Lu, "Multiple source domain adaptation for multiple object tracking in satellite video," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–11, 2023.

[2] L. Ren, D. Hong, L. Gao, X. Sun, M. Huang, and J. Chanussot, "Orthogonal subspace unmixing to address spectral variability for hyperspectral image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–13, 2023.

[3] D. Hong, B. Zhang, H. Li, Y. Li, J. Yao, C. Li, M. Werner, J. Chanussot, A. Zipf, and X. X. Zhu, "Cross-city matters: A multimodal remote sensing benchmark dataset for cross-city semantic segmentation using high-resolution domain adaptation networks," *Remote Sensing of Environment*, vol. 299, p. 113856, 2023.

[4] X. Zheng, H. Cui, C. Xu, and X. Lu, "Dual teacher: A semisupervised cotraining framework for cross-domain ship detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023.

[5] F. Cao, Z. Yang, X. Hong, Y. Cheng, Y. Huang, and J. Lv, "Supervised dimensionality reduction of hyperspectral imagery via local and global sparse representation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3860–3874, 2021.

[6] C.-I. Chang, L.-C. Lee, B. Xue, M. Song, and J. Chen, "Channel capacity approach to hyperspectral band subset selection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 10, pp. 4630–4644, 2017.

[7] Y. Zhou, Q. Yao, S. Huo, and X. Li, "Hyperspectral band selection with iterative graph auto-encoder," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[8] Q. Du and H. Yang, "Similarity-based unsupervised band selection for hyperspectral image analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 4, pp. 564–568, 2008.

[9] B. Kumar, O. Dikshit, A. Gupta, and M. K. Singh, "Feature extraction for hyperspectral image classification: A review," *International Journal of Remote Sensing*, vol. 41, no. 16, pp. 6248–6287, 2020.

[10] S. Lorenz, P. Ghamisi, M. Kirsch, R. Jackisch, B. Rasti, and R. Gloaguen, "Feature extraction for hyperspectral mineral domain mapping: A test of conventional and innovative methods," *Remote Sensing of Environment*, vol. 252, p. 112129, 2021.

[11] M. D. Farrell and R. M. Mersereau, "On the impact of PCA dimension reduction for hyperspectral detection of difficult targets," *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 2, pp. 192–195, 2005.

[12] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 862–873, 2009.

[13] Z. Wang, B. Du, Q. Shi, and W. Tu, "Domain adaptation with discriminative distribution and manifold embedding for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 7, pp. 1155–1159, 2019.

[14] G. Shi, H. Huang, Z. Li, and Y. Duan, "Multi-manifold locality graph preserving analysis for hyperspectral image classification," *Neurocomputing*, vol. 388, pp. 45–59, 2020.

[15] H. Su, B. Zhao, Q. Du, and P. Du, "Kernel collaborative representation with local correlation features for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1230–1241, 2018.

[16] J. Xia, N. Falco, J. A. Benediktsson, P. Du, and J. Chanussot, "Hyperspectral image classification with rotation random forest via KPCA," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 4, pp. 1601–1609, 2017.

[17] X. Kang, Y. Zhu, P. Duan, and S. Li, "Two dimensional spectral representation," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[18] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[19] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 279–317, 2019.

[20] X. Cao, B. Ji, Y. Ji, L. Wang, and L. Jiao, "Hyperspectral image classification based on filtering: a comparative study," *Journal of Applied Remote Sensing*, vol. 11, no. 3, p. 035007, 2017.

[21] X. Zhang, X. Jiang, J. Jiang, Y. Zhang, X. Liu, and Z. Cai, "Spectral–spatial and superpixelwise PCA for unsupervised feature extraction of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–10, 2021.

[22] H. Fu, G. Sun, A. Zhang, B. Shao, J. Ren, and X. Jia, "Tensor singular spectrum analysis for 3-d feature extraction in hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–14, 2023.

[23] C. Li, B. Rasti, X. Tang, P. Duan, J. Li, and Y. Peng, "Channel-layer-oriented lightweight spectral-spatial network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[24] L. Qian, X. Wang, M. Hong, H. Wang, and Y. Zhang, "An efficient and adaptive reconstructive homogeneous block-based local tensor robust PCA for feature extraction of hyperspectral images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.

[25] J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, and L. Wang, "SuperPCA: A superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4581–4593, 2018.

[26] H. Liu, W. Li, X.-G. Xia, M. Zhang, and R. Tao, "Superpixelwise collaborative-representation graph embedding for unsupervised dimension reduction in hyperspectral imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4684–4698, 2021.

[27] J. Li, H. Sheng, M. Xu, S. Liu, and Z. Zeng, "BAMS-FE: Band-by-band adaptive multiscale superpixel feature extraction for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–15, 2023.

[28] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[29] Z. Zhang, C. Luo, H. Wu, Y. Chen, N. Wang, and C. Song, "From individual to whole: reducing intra-class variance by feature aggregation," *International Journal of Computer Vision*, vol. 130, no. 3, pp. 800–819, 2022.

[30] Q. Zhao, S. Jia, and Y. Li, "Hyperspectral remote sensing image classification based on tighter random projection with minimal intra-class variance algorithm," *Pattern Recognition*, vol. 111, p. 107635, 2021.

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

[32] G. Luo, G. Chen, L. Tian, K. Qin, and S.-E. Qian, "Minimum noise fraction versus principal component analysis as a preprocessing step for hyperspectral imagery denoising," *Canadian Journal of Remote Sensing*, vol. 42, no. 2, pp. 106–116, 2016.

[33] X. Shen, W. Bao, and K. Qu, "Subspace-based preprocessing module for fast hyperspectral endmember selection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3386–3402, 2021.

[34] X. Kang, X. Xiang, S. Li, and J. A. Benediktsson, "Pca-based edge-preserving features for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7140–7151, 2017.

[35] A. Cheriyadat and L. M. Bruce, "Why principal component analysis is not an appropriate feature extraction method for hyperspectral data," in *IGARSS 2003*, vol. 6. IEEE, 2003, pp. 3420–3422.

[36] Z. Wang and B. He, "Locality perserving projections algorithm for hyperspectral image dimensionality reduction," in *19th International Conference on Geoinformatics*. IEEE, 2011, pp. 1–4.

[37] Y. Duan, H. Huang, and T. Wang, "Semisupervised feature extraction of hyperspectral image using nonlinear geodesic sparse hypergraphs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

[38] X. Jiang, L. Xiong, Q. Yan, Y. Zhang, X. Liu, and Z. Cai, "Unsupervised dimensionality reduction for hyperspectral imagery via Laplacian regularized collaborative representation projection," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[39] X. Zhang, X. Jiang, J. Jiang, Y. Zhang, X. Liu, and Z. Cai, "Spectral–spatial and superpixelwise PCA for unsupervised feature extraction of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–10, 2022.

[40] P. Lu, X. Jiang, Y. Zhang, X. Liu, Z. Cai, J. Jiang, and A. Plaza, "Spectral-spatial and superpixelwise unsupervised linear discriminant analysis for feature extraction and classification of hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[41] S. Gao, X. Jiang, Y. Zhang, X. Liu, Q. Xiong, and Z. Cai, "Superpixelwise PCA based data augmentation for hyperspectral image classification," *Multimedia Tools and Applications*, pp. 1–21, 2024.

[42] C. Yu, R. Han, M. Song, C. Liu, and C.-I. Chang, "A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial–spectral fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 2485–2501, 2020.

[43] X. Ma, W. Wang, W. Li, J. Wang, G. Ren, P. Ren, and B. Liu, "An ultra-lightweight hybrid CNN based on redundancy removal for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[44] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847–858, 2017.

[45] M. Esmaeili, D. Abbasi-Moghadam, A. Sharifi, A. Tariq, and Q. Li, "Resmorcnn model: Hyperspectral images classification using residual-injection morphological features & 3d-cnn layers," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023.

[46] S. Zhang, M. Xu, J. Zhou, and S. Jia, "Unsupervised spatial-spectral cnn-based feature learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022.

[47] M. Zhang, M. Gong, H. He, and S. Zhu, "Symmetric all convolutional neural-network-based unsupervised feature extraction for hyperspectral images classification," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 2981–2993, 2020.

[48] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[49] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[50] B. Jiang and Y.-H. Dai, "A framework of constraint preserving update schemes for optimization on stiefel manifold," *Mathematical Programming*, vol. 153, no. 2, pp. 535–575, 2015.

[51] Y. Zhong, X. Wang, Y. Xu, S. Wang, T. Jia, X. Hu, J. Zhao, L. Wei, and L. Zhang, "Mini-UAV-borne hyperspectral remote sensing: From observation and processing to applications," *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 4, pp. 46–62, 2018.

[52] Y. Zhong, X. Hu, C. Luo, X. Wang, J. Zhao, and L. Zhang, "WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF," *Remote Sensing of Environment*, vol. 250, p. 112012, 2020.