remote sensing

MDPI

Article

Euler Kernel Mapping for Hyperspectral Image Clustering via Self-Paced Learning

Fenggan Zhang [1,†], Hao Yan [2,†], Jianwei Zhao [1], Haojie Hu [1,*]

[1] Xi'an Research Institute of High Technology, Xi'an 710025, China
[2] School of Mathematical Sciences, Dalian University of Technology, Dalian 116038, China
* Correspondence: haojiehu705@gmail.com
† These authors contributed equally to this work.

Abstract: Clustering, as a classical unsupervised artificial intelligence technology, is commonly employed for hyperspectral image clustering tasks. However, most existing clustering methods designed for remote sensing tasks aim to solve a non-convex objective function, which can be optimized iteratively, beginning with random initializations. Consequently, during the learning phase of the clustering model, it may easily fall into bad local optimal solutions and finally hurt the clustering performance. Additionally, prevailing approaches often exhibit limitations in capturing the intricate structures inherent in hyperspectral images and are very sensitive to noise and outliers that widely exist in remote sensing data. To address these issues, we proposed a novel Euler kernel mapping for hyperspectral image clustering via self-paced learning (EKM-SPL). EKM-SPL first employs self-paced learning to learn the clustering model in a meaningful order by progressing samples from easy to complex, which can help to remove bad local optimal solutions. Secondly, a probabilistic soft weighting scheme is employed to measure complexity across the data sample, which makes the optimization process more reasonable. Thirdly, in order to more accurately characterize the intricate structure of hyperspectral images, Euler kernel mapping is used to convert the original data into a reproduced kernel Hilbert space, where the nonlinearly inseparable clusters may become linearly separable. Moreover, we innovatively integrate the coordinate descent technique into the optimization algorithm to circumvent the computational inefficiencies and information loss typically associated with conventional kernel methods. Extensive experiments conducted on classic benchmark hyperspectral image datasets illustrate the effectiveness and superiority of our proposed model.

Keywords: hyperspectral image; clustering; self-paced learning; Euler kernel mapping

check for updates

Citation: Zhang, F.; Yan, H.; Zhao, J.; Hu, H. Euler Kernel Mapping for Hyperspectral Image Clustering via Self-Paced Learning. *Remote Sens.* **2024**, *16*, 4097. https://doi.org/10.3390/rs16214097

Academic Editor: Salah Bourennane

Received: 11 September 2024
Revised: 15 October 2024
Accepted: 21 October 2024
Published: 2 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the domain of remote sensing, hyperspectral images (HSIs) organically combines traditional spatial imaging with advanced imaging techniques. Leveraging both continuous spectral data and rich spatial information obtained from hyperspectral sensors, HSI can obviously enhance the overall quality and utility of the imagery [1–3]. HSIs usually perform 3D cube imaging of the target area needed for research, gathering rich spectral data consisting of tens to hundreds of bands, after which subtle differences between various ground objects can be captured [4–8]. HSIs are widely used in many fields, including mineral exploration [9], quantitative inversion of bioinformatics [10], aviation reconnaissance [8,11], and so on. However, the interpretation of HSIs often relies on a large number of high-quality labeled samples, thus avoiding the under-fitting problem caused by insufficient training samples. In practice, sample collection and labeling are tiresome, costly, time consuming, and ineffective. Additionally, it is hard to gather training samples in some risky and complicated environments, which severely restricts the application of HSIs. Therefore, it is necessary to develop an unsupervised learning mechanism for the analysis and application of hyperspectral images.

Clustering is a classical unsupervised learning strategy that has received a lot of attention in the fields of machine learning and data mining [8,12–16]. Researchers have presented a number of hyperspectral remote sensing image clustering algorithms over the last decade [4,17,18], which contribute to HSI interpretation and information extraction [7,19,20]. These algorithms mine the structural characteristics of HSIs, distinguishing various pixels based on different similarity metrics, such as distance [8,21], correlation [22], spectral angle [23], and so on. Clustering is considered as an effective artificial intelligence (AI) method for remote sensing [24,25], especially in the absence of labeled samples, which significantly boosts the applications of HSIs. In the past dacade, clustering techniques for HSIs have been well developed and widely investigated, including centroid-based clustering, density-based clustering, graph-based clustering, bionics-based clustering, and so on. The density-based clustering for HSI methods divide pixels into clusters according to the density criterion, during which the pixels in HSIs can be clustered based on the local density and the relative distance of the pixels. For example, [26] assumed that cluster centers are surrounded by low-density pixels and are relatively distant from areas of high density in order to identify the optimal clustering center. However, the high-dimensional feature space of HSIs is relatively sparse and uniformly distributed, so the assumption of density-based clustering is not fully satisfied. Graph-based clustering models the correlation between pixels using an adjacency matrix (similarity matrix) and can be broadly categorized into (1) complete graph clustering [27,28], (2) bipartite graph-based clustering [29], and (3) abbreviated graph-based clustering [30]. However, due to the time required to construct and analyze the adjacency matrix, such methods often require a trade-off between accuracy and efficiency. Biomimicry-based clustering methods employ biological models, such as artificial neural networks, to reproduce the internal structure of HSIs [31]. Nevertheless, the intricate structure of HSIs does not adapt well to specific biological models, which limits its performance.

Centroid-based clustering is one of the most popular methods for hyperspectral remote sensing images [4]. It bases its loss function on the sum of squared Euclidean distances between individual pixels and their respective cluster centroids, assuming that all clusters have a "spherical" shape in the feature space. Following random initialization, the clustering centroids are iteratively updated according to the distance between the centroid and the samples within the cluster until the loss satisfies the tolerance requirement or the number of iterations reaches a predefined maximum value, as shown in Figure 1. One of the most well-known centroid-based clusterings is the *k*-means algorithm [13,14,32], which minimizes the partition error of Euclidean distances based on the similarity between within-cluster samples and their cluster centroids. *K*-means is widely appreciated for its simplicity and reliability, delivering good results with a low computational cost. Based on *k*-means, many improved methods have been proposed [33,34]. Fuzzy C-Means (FCM) [35,36] is proposed to consider the uncertainty of how pixels are divided during the clustering process, allowing each pixel belong to several classes at the same time. Specifically, FCM assigns each pixel a fuzzy membership in the range [0, 1], and it has achieved better results in some real-world datasets. Since then, researchers have developed many enhanced FCM models [37]. Refs. [38,39] proposed a method to capture richer spatial neighborhood information for target pixels by incorporating spatial information (FCM-S1), simplifying the calculation process through the use of average filtering results.

Although these methods have contributed to the clustering of hyperspectral remote sensing images, most existing approaches are designed to solve non-convex objective functions, which require iterative optimization starting from a random initialization (i.e., the "mountain climbing" process), until either the tolerance requirements are met or a predefined number of iterations is reached [4,14,32,40]. Specifically, the learning process will automatically stop when the partition error based on the Euclidean distance is less than the preset threshold or the number of iterations reaches a predefined maximum value. Thus, when learning the clustering model, it may easily fall into bad local optimal solutions, especially with noise and outliers, and finally hurt the clustering performance. This can be solved by repeatedly

running the algorithm with different initializations [14,33,34]; however, this strategy has a high computational cost and offers little benefit in an unsupervised environment. So far, no reasonable strategy has been proposed to help the clustering model automatically escape bad local optimal solutions.

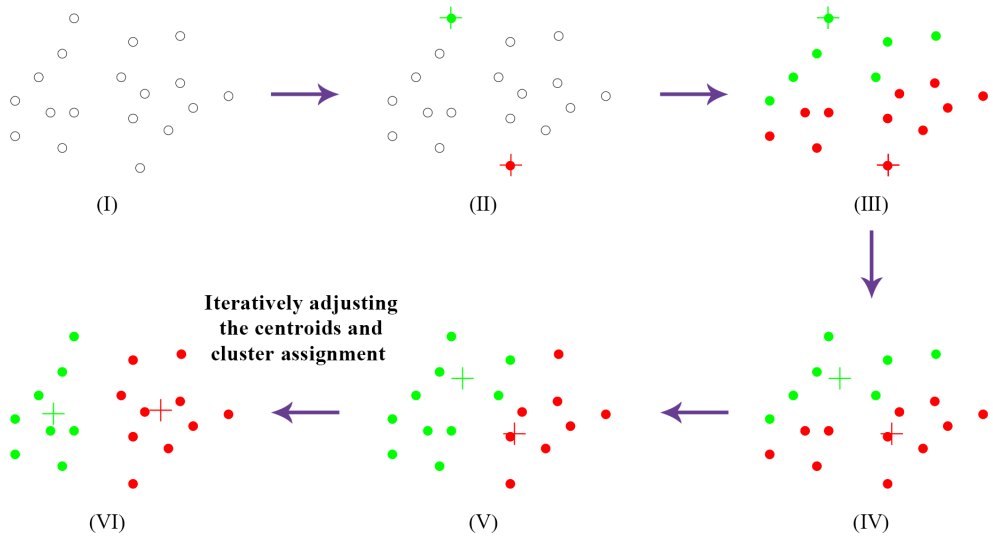


Figure 1. The overall idea of EKM-SPL.

To deal with this problem, we propose a novel Euler kernel mapping for hyperspectral images clustering via self-paced learning (EKM-SPL) to improve the clustering performance of hyperspectral images. In summary, our work provides the following contributions:

- An effective kernel mapping, Euler kernel [41] mapping, is subtly employed to induce a feature map that transforms the input data of HSIs to a complex feature space (i.e., Reproducing Kernel Hilbert Space (RKHS)). In such space, the nonlinearly inseparable clusters may become linearly separable, so we can better describe the complex and nonlinear structure of HSIs together with improved clustering performance.
- The Euler kernel mapping implicitly inherits the advantage of the Fourier feature, during which the influence of noise is minimized due to the range of the cosine function. Thus, it is less sensitive to noise and outliers, which is very useful for the clustering of remote sensing data.
- Instead of roughly dividing samples into "easy" or "complex", a probabilistic soft weighting scheme is employed to measure complexity across them, which makes the optimization process more reasonable.
- The coordinate descent method is employed to directly derive clustering labels through the kernel technique. This method is unlike previous approaches that usually comprise two stages, including iteratively refining the continuous relaxed label matrix and discretizing it through additional steps, which may result in severe information loss together with high computational complexity.
- EKM-SPL adopts the self-paced learning (SPL) and is employed for the clustering of hyperspectral remote sensing images (HSIs), through which the clustering model can be optimized in a meaningful order by progressing pixels from easy to complex. With the help of SPL, the HSIs clustering model is capable of removing bad local optimums.
- The computational complexity of our proposed method is very similar to the classic *k*-means, since the dimensionality of the Euler feature space is invariant without calculating and storing the kernel matrix, which eliminates the heavy time and memory cost of kernel trick.
- Extensive experimental results on HSI data demonstrate the effectiveness and superiority of the proposed EKM-SPL model.

The remainder of this paper is organized as follows: In Section 2, we introduce the traditional and common centroid-based clustering methods, including *k*-means and fuzzy C-means (FCM). In Section 3, we first introduce Euler kernel mapping and self-paced learning as the preliminary work. Then, we introduce the model of our proposed EKM-SPL algorithm and its optimization. In Section 4, we demonstrate the superior performance and effectiveness of our model through detailed experiments on four representative hyperspectral image datasets. Finally, we make a summary of this work and give an idea for future work.

2. Related Work

k-Means

In hyperspectral remote sensing images, *k*-means [32] minimizes the partition error based on the Euclidean distances of within-cluster pixels and their cluster centroids. The principle of *k*-means is simple and reliable, and it can achieve good results with small time cost, so it is widely welcomed by researchers.

However, *k*-means measures the similarity of spatial samples through the Euclidean distance. It bases the loss function on the sum of squared Euclidean distances between particular pixels and their associated centroids. In addition, *k*-means is generally based on the "spherical" structure assumption, which assumes that in the sample space, the cluster centroid of each cluster is located in the centroid of a "spherical" structure, and the sample points within the cluster exist in the interior of this "spherical". Thus, *k*-means cannot work well when the data distribution is nonlinearly inseparable. Due to the exceedingly complex and nonlinearly inseparable structure of HSIs, as well as the large spectral variability of HSIs, the inconsistency of the clustering results of *k*-means with the real situation frequently appears, which further limits the clustering performance.

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$ be n training samples and d be the feature dimension. *k*-means dividing samples into different groups so that the similarity between samples in the same group is higher than the similarity between samples in different groups. And the similarity between samples is measured through Euclidean distances in primitive space. Let $\boldsymbol{m}_j$ be the j-th cluster center, and $\boldsymbol{M} = [\boldsymbol{m}_1, \cdots, \boldsymbol{m}_c]$ where c is the cluster number. So, the objective function of the *k*-means clustering model can be described as

$$\min_{\boldsymbol{M},\boldsymbol{F} \in \mathrm{Ind}} \sum_{i=1}^{n} \sum_{j=1}^{c} F_{ij} \|\boldsymbol{x}_i - \boldsymbol{m}_j\|_2^2, \tag{1}$$

where $\boldsymbol{F} \in \mathrm{Ind}$ is the cluster indicator matrix. If $\boldsymbol{x}_i$ belongs to the j-th cluster, $F(i,j) = 1$; otherwise, $F(i,j) = 0$. Take the derivative of Equation (1) w.r.t $\boldsymbol{m}_j$, and set it to zero:

$$\boldsymbol{m}_j = \frac{1}{\boldsymbol{n}_j} \sum_{i=1}^{n} F_{ij} \boldsymbol{x}_i, \tag{2}$$

where $\boldsymbol{n}_j = \sum_{i=1}^{n} F_{ij}$ and $\boldsymbol{n}_j \neq 0$. Then, the problem could be solved by updating $\boldsymbol{F}$ and $\boldsymbol{M}$ alternatively until convergence:

(1) Update $\boldsymbol{M}$ with $\boldsymbol{F}$ fixed. $\boldsymbol{m}_j$ can be updated through Equation (2).

(2) Update $\boldsymbol{F}$ with $\boldsymbol{M}$ fixed. Following random initialization, the clustering centroids are iteratively updated according to the distance between the centroid and the samples with the cluster until the sum of the squared Euclidean distances of all the samples to their centroids satisfies the tolerance requirement or the number of iterations reaches a predefined maximum value. To be more detailed, the position of the 1 element in the i-th row of the cluster indicator matrix F represents the cluster that the i-th sample belongs, during which the Euclidean distance from a sample to the center of each group is calculated, and the sample is divided into the group with the smallest distance corresponding to it.

In order to solve the problem in which complex data structures with nonlinearly inseparable clusters cannot be described [42], kernel *k*-means [43] applies common ker-

nel functions like Gaussian kernel function (RBF) [40,42,44] to implicitly transform the data into higher-dimensional or even infinite-dimensional feature spaces (i.e., Reproduced Kernel Hilbert Space (RKHS)) [14,40,45]. In RKHS, complex data structures can be more easily interpreted and described, so that the nonlinearly inseparable clusters may also become linearly separable for k-means to process. Based on kernel k-means, many improved algorithms have been derived, such as multiple kernel clustering [40,46] and its extensions [47]. However, they cannot be applied to large-scale data like HSIs because they require enormous computation and memory to process kernel matrices. At the same time, the kernel trick [40,42,48] calculates the distance between data points and cluster centers with a cost quadratic with dataset size. Thus, it is such a great burden for kernel k-means to deal with the HSI data, since the time and memory cost of the kernel matrix processing and the optimization are too heavy.

3. The Proposed Method

3.1. Euler Kernel Mapping

Euler kernel mapping is a effective implementation of the kernel trick [40,49]; it is a cosine-metric-based Mercer kernel [41,50], so it can accomplish robust nonlinear clustering. Specifically, the Euler kernel mapping can inherently induce a feature mapping that transforms input data into a complex domain, maintaining the same dimensionality as the original input. Conversely, Mercer kernels (like RBF and polynomial kernels) typically effectuate nonlinear mappings that implicitly elevate data to an infinite or substantially higher dimensional space. However, the Euler kernel mapping is more intrinsic and precise than the other kernel mappings, because it can directly know the data structure in the feature space of a specific dimension without any approximation.

Let $\phi_e(\cdot) : \mathcal{R}^d \to \mathcal{C}^d$ be a kernel mapping from d-dimensional real space $\mathcal{R}^d$ into d-dimensional complex RKHS space $\mathcal{C}^d$ as follows:

$$\phi_e(\boldsymbol{x}_i) = \frac{1}{\sqrt{2}} e^{i\alpha\pi\boldsymbol{x}_i} = \frac{1}{\sqrt{2}} \begin{bmatrix} e^{i\alpha\pi\boldsymbol{x}_i(1)} \\ e^{i\alpha\pi\boldsymbol{x}_i(2)} \\ \cdots \\ e^{i\alpha\pi\boldsymbol{x}_i(d)} \end{bmatrix}. \tag{3}$$

So, let $\phi_e(x) = [\phi_e(\boldsymbol{x}_1), \cdots, \phi_e(\boldsymbol{x}_n)]$ be the mapping of $\boldsymbol{X}$, whose dimensions are finite, and the kerenl matrix $\boldsymbol{G}_e$ is defined as follows.

$$\boldsymbol{G}_e(p,q) = \langle \phi_e(\boldsymbol{x}_p), \phi_e(\boldsymbol{x}_q) \rangle = \frac{1}{2}\sum_{c=1}^{d} \cos(\alpha\pi(\boldsymbol{x}_p(c) - \boldsymbol{x}_q(c))) - i\frac{1}{2}\sum_{c=1}^{d} \sin(\alpha\pi(\boldsymbol{x}_p(c) - \boldsymbol{x}_q(c))) \tag{4}$$

where i represents the imaginary unit, and it means $\boldsymbol{G}_e$ is defined in the complex space, which is different from kernels such as RFB. In particular, the Euler kernel mapping implicitly inherits the advantage of the Fourier feature, which is period-based and closely related to the cosine function. Thus, the Euler kernel can enhance the robustness of clustering model, since it is less sensitive to noise and outliers, which is very useful for hyperspectral remote sensing images clustering.

For lifted representation in the complex feature domain that the Euler kernel mapping induces, suppose $\boldsymbol{x}_p = \boldsymbol{a}_p + i\boldsymbol{b}_p$ and $\boldsymbol{x}_q = \boldsymbol{a}_q + i\boldsymbol{b}_q$; then, the similarity in the RKHS space between $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ can be denoted as the following equation.

$$d(\boldsymbol{x}_p, \boldsymbol{x}_q) = \|\boldsymbol{a}_p - \boldsymbol{a}_q\|_2^2 + \|\boldsymbol{b}_p - \boldsymbol{b}_q\|_2^2, \tag{5}$$

In particular, when $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ are written in a form like Equation (3), then Equation (5) could be simplified as the following formulation.

$$d(\boldsymbol{x}_p, \boldsymbol{x}_q) = \sum_{c=1}^{d} (1 - \cos(\alpha\pi(\boldsymbol{x}_p(c) - \boldsymbol{x}_q(c)))) \tag{6}$$

where $\boldsymbol{x}_p(c)$ is the entry of $\boldsymbol{x}_p$. Apparently, the definition of $d(\cdot, \cdot)$ transitions the Euler kernel from the complex space to the real space, which lays a very important foundation for the application of the Euler kernel mapping. In addition, the cosine value lies in the interval $[-1, 1]$, so the measure of similarity through Equation (5) is less sensitive to noisy features than that directly calculated European distances on the raw data.

3.2. Self-Paced Euler k-Means for Hyperspectral Images Clustering

In EKM-SPL, we subtly employ the Euler kernel mapping to *k*-means, which can essentially induce a feature map that transforms the input data of HSIs to a complex feature space (i.e., Reproducing Kernel Hilbert Space (RKHS)) [14]. In RKHS, the nonlinearly inseparable clusters may become linearly separable, breaking the limitations caused by the "spherical" assumption that cannot accurately describe the extremely complex inherent structure of the HSI data. Moreover, the Euler kernel mapping implicitly inherits the advantage of the Fourier feature, which is period-based and closely related to the cosine function. Hence, the cosine function is used to calculate the distance (similarity) between each pair of data points in the RKHS space where the cosine function's wide range reduces the impact of noise, and the outliers will make the assigned cluster deviate from itself in RKHS space.

What is more, self-paced learning (SPL) is employed for hyperspectral image clustering and learns the clustering model in a meaningful order, gradually progressing from simpler to more complex sample pixels, which can help circumvent undesirable local optima. Meanwhile, instead of roughly dividing samples into "easy" or "complex", a probabilistic soft weighting scheme is employed to measure complexity across them, which makes the optimization process more reasonable. Lastly, our proposed EKM-SPL can directly calculate the distance between a data point and any cluster centroid, and it does not need to calculate and store the kernel matrix. Thus, the computational complexity of our proposed method is very similar to the classic *k*-means, which is linear to the size of the dataset, since the dimensionality of the feature space is invariant, which eliminates the heavy time and memory cost of the kernel method, making it more feasible in applications of HSIs.

Recently, self-paced learning (SPL) has demonstrated its effectiveness in evading undesired local optima and enhancing generalization outcomes [14,51]. By imitating the styles of human learning, SPL learns the model in a meaningful order, which begins with simple samples and gradually brings more complex samples [52,53]. Kumar et al. was the first to put forward the concept of self-paced learning (SPL) [52], and they proposed the basic strategy of automatically learning samples by determining whether a sample is easy or complex into a training model, which integrated the training model and sample selection process into a new unified framework. By imitating the styles of human learning, self-paced learning (SPL) learns the model in a meaningful order, which begins with simple samples and gradually brings more complex samples [52,53].

The objective of EKM-SPL can be started by minimizing the sum of the squared errors of distance (similarity), which is defined as

$$\min_{\boldsymbol{m}_j, \boldsymbol{F} \in \mathrm{Ind}} \sum_{i=1}^{n} \sum_{j=1}^{c} F_{ij} \|\phi_e(\boldsymbol{x}_i) - \boldsymbol{m}_j\|_2^2, \tag{7}$$

where $\phi_e(\cdot) : \mathcal{R}^d \to \mathcal{C}^d$ is the Euler kernel mapping from the d-dimensional real space $\mathcal{R}^d$ into the d-dimensional complex RKHS space $\mathcal{C}^d$. More details are shown as follows.

$$\phi_e(\boldsymbol{x}_i) = \frac{1}{\sqrt{2}} e^{i\alpha\pi\boldsymbol{x}_i} = \frac{1}{\sqrt{2}} \begin{bmatrix} \cos(\alpha\pi\boldsymbol{x}_i(1)) + i\sin(\alpha\pi\boldsymbol{x}_i(1)) \\ \cos(\alpha\pi\boldsymbol{x}_i(2)) + i\sin(\alpha\pi\boldsymbol{x}_i(2)) \\ \cdots \\ \cos(\alpha\pi\boldsymbol{x}_i(d)) + i\sin(\alpha\pi\boldsymbol{x}_i(d)) \end{bmatrix}, \tag{8}$$

in which $\boldsymbol{m}_j$ denotes the center of the j-th cluster. If $\phi(\boldsymbol{x}_i)$ is assigned to the j-th cluster then $F(i,j) = 1$; otherwise, $F(i,j) = 0$. Our proposed method can immediately determine the distance between a data point and any cluster centroids using Euler kernel mapping without analyzing the large-scale kernel matrix. Equation (7) can be reformulated in matrix form:

$$\min_{\boldsymbol{M},\boldsymbol{F}\in\mathrm{Ind}} \|\phi_e(\boldsymbol{X}) - \boldsymbol{M}\boldsymbol{F}^T\|_F^2, \tag{9}$$

where $\phi_e(\boldsymbol{X}) = [\phi_e(\boldsymbol{x}_1), \cdots, \phi_e(\boldsymbol{x}_n)]$ and $\boldsymbol{M} = [\boldsymbol{m}_1, \cdots, \boldsymbol{m}_c]$ denotes the clustering centroid matrix. Similar to all the centroid-based clustering methods, our proposed method is a non-convex problem with the risk of getting stuck in bad local minima, which can be alleviated by self-paced learning (SPL) as mentioned above. According to [52], there is a weighted loss term on all samples as well as a regularizer term imposed on weights in SPL. More samples are automatically added to the training from simple to complicated by gradually increasing the penalty of the regularizer, which can be further formulated as

$$\begin{aligned} &\min_{\boldsymbol{M},\boldsymbol{F}\in\mathrm{Ind},\boldsymbol{W}} \|(\phi_e(\boldsymbol{X}) - \boldsymbol{M}\boldsymbol{F}^T)\boldsymbol{W}\|_F^2 + f(\boldsymbol{W}), \\ &s.t.\ \ w_i^p \in [0,1], \forall i \in [1,n], \forall p \in [1,v], \end{aligned} \tag{10}$$

where $\boldsymbol{W} = \mathrm{diag}(\boldsymbol{w})$ denotes a diagonal matrix and $\boldsymbol{w} = [\sqrt{w_1}, \cdots, \sqrt{w_n}]^T$ comprises the weights of n samples. $f(\boldsymbol{W})$ is the regularizer determining the samples to be selected in training. The previously adopted $f(\boldsymbol{W})$ in SPL [52] can be simplified as

$$f(\boldsymbol{W}) = -\lambda \sum_{i=1}^{n} w_i, \tag{11}$$

which indicates that the optimal weight for the i-th sample can be formulated as

$$w_i = \begin{cases} 1, & \text{if } \ell_i \le \lambda, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where ℓ_i represents the reconstruction error for the i-th sample. The parameter λ serves as a threshold, distinguishing between "easy" samples with losses below it and "complex" samples with losses exceeding it. During the learning process, λ regulates the pace at which new samples are acquired, gradually decreasing through iterative optimization.

The classical regularizer in Equation (11) roughly selects samples by assigning binary weights. It is unreasonable to roughly divide samples into "easy" or "complex" due to the non-homogeneous distribution of noise. In applications, soft weighting is proven to be more effective, faithfully reflecting the importance of samples. Hence, we design a soft weighted scheme of $f(\boldsymbol{W})$, which can be formulated as

$$f(\boldsymbol{W}) = \sum_{i=1}^{n} \ln(1 + e^{-\lambda} - w_i)^{(1+e^{-\lambda}-w_i)} + \ln(w_i)^{w_i} - \lambda w_i, \tag{13}$$

which indicates that the optimal weight for the i-th sample can be formulated as

$$w_i = \frac{1 + e^{-\lambda}}{1 + e^{\ell_i - \lambda}}. \tag{14}$$

Equation (14) is a smoothed function (related to ℓ_i) regarded as an adapted logistic function, providing a probabilistic interpretation of the weights with output values in $[0, 1]$. Instead of hard separating the samples (kernels) into "easy" or "complex", as λ in Equation (12) deciding $w_i = 1$ or $w_i = 0$, λ in Equation (14) tends to assign samples the probabilities of being "easy". Moreover, as λ increases, more samples and kernels are prone to be included to train a mature model. Combining Equations (10) and (13), we obtain the final objective function, in which the complexity of the sample of HSI data is probabilistically measured, and the model is gradually trained in a meaningful order to avoid falling into bad local optimums.

3.3. Optimization

To address the problem (10), an alternative optimization strategy can be employed. This involves implementing distinct procedures for updating $\boldsymbol{F}$, $\boldsymbol{M}$ and $\boldsymbol{W}$ in an iterative manner.

Step 1: Update $\boldsymbol{M}$ with $\boldsymbol{F}$ and $\boldsymbol{W}$ fixed. The objective function of problem (10) is as follows:

$$\min_{\boldsymbol{M}} \|(\phi_e(\boldsymbol{X}) - \boldsymbol{M}\boldsymbol{F}^T)\boldsymbol{W}\|_F^2, \tag{15}$$

Take the derivative of problem (15) with respect to $\boldsymbol{M}$ setting it to zero:

$$\boldsymbol{M} = \phi_e(\boldsymbol{X})\boldsymbol{D}\boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F})^{-1} \tag{16}$$

where $\boldsymbol{D} = \boldsymbol{W}\boldsymbol{W} = \mathrm{diag}([w_1, \cdots, w_n])$.

Step 2: Update $\boldsymbol{F}$ with $\boldsymbol{M}$ and $\boldsymbol{W}$ fixed. Problem (10) can be reformulated as

$$\min_{\boldsymbol{F} \in \mathrm{Ind}} \|(\phi_e(\boldsymbol{X}) - \boldsymbol{M}\boldsymbol{F}^T)\boldsymbol{W}\|_F^2. \tag{17}$$

Then, problem (17) can be expanded and rewritten into

$$\min_{\boldsymbol{F} \in \mathrm{Ind}} \mathrm{Tr}(\boldsymbol{W}\boldsymbol{K}_e\boldsymbol{W}) - 2\mathrm{Tr}(\boldsymbol{W}\phi_e^H(\boldsymbol{X})\boldsymbol{M}\boldsymbol{F}^T\boldsymbol{W}) + \mathrm{Tr}(\boldsymbol{W}\boldsymbol{F}\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{F}^T\boldsymbol{W}) \tag{18}$$

where $\boldsymbol{K}_e = \phi_e^H(\boldsymbol{X})\phi_e(\boldsymbol{X})$. Substituting Equation (16) to problem (18), we obtain the following objective function:

$$\max_{\boldsymbol{F} \in \mathrm{Ind}} \mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{A}_e\boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F})^{-1}), \tag{19}$$

where we denote $\boldsymbol{A}_e = \boldsymbol{D}\boldsymbol{K}_e\boldsymbol{D}$. Noting that the clustering indicator matrix $\boldsymbol{F} = [\boldsymbol{f}_1; \boldsymbol{f}_2; \cdot; \boldsymbol{f}_c]$ is orthogonal among each column, where $\boldsymbol{f}_l$ denotes the l-th column of $\boldsymbol{F}$, so

$$\boldsymbol{F}^T\boldsymbol{A}_e\boldsymbol{F} = \begin{bmatrix} \boldsymbol{f}_1^T\boldsymbol{A}_e\boldsymbol{f}_1 & \boldsymbol{f}_1^T\boldsymbol{A}_e\boldsymbol{f}_2 & \cdots & \boldsymbol{f}_1^T\boldsymbol{A}_e\boldsymbol{f}_c \\ \boldsymbol{f}_2^T\boldsymbol{A}_e\boldsymbol{f}_1 & \boldsymbol{f}_2^T\boldsymbol{A}_e\boldsymbol{f}_2 & \cdots & \boldsymbol{f}_1^T\boldsymbol{A}_e\boldsymbol{f}_c \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{f}_c^T\boldsymbol{A}_e\boldsymbol{f}_1 & \boldsymbol{f}_c^T\boldsymbol{A}_e\boldsymbol{f}_2 & \cdots & \boldsymbol{f}_c^T\boldsymbol{A}_e\boldsymbol{f}_c \end{bmatrix}, \tag{20}$$

and

$$\boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F}=\begin{bmatrix} \boldsymbol{f}_1^T\boldsymbol{D}\boldsymbol{f}_1 & 0 & \cdots & 0 \\ 0 & \boldsymbol{f}_2^T\boldsymbol{D}\boldsymbol{f}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{f}_c^T\boldsymbol{D}\boldsymbol{f}_c \end{bmatrix}. \tag{21}$$

Note that the clustering indicator matrix $\boldsymbol{F}$ is orthogonal among each column, and $\boldsymbol{f}_l$ represents the l-th column of $\boldsymbol{F}$. Thus, problem (19) becomes

$$\max_{\boldsymbol{F}\in \text{ind}} \sum_{l=1}^{c} \frac{(\boldsymbol{f}_l^T\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))(\boldsymbol{f}_l^T\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))^T}{\boldsymbol{f}_l^T\boldsymbol{D}\boldsymbol{f}_l}. \tag{22}$$

Now, we tend to directly obtain the clustering indicator matrix $\boldsymbol{F}$ by using coordinate descent [54]. Before the iteration in which all variables are fixed except the i-th row being updated to the optimum, $\boldsymbol{F}$ is initialized so we can predefine the two main parts of Equation (22): $(\boldsymbol{f}_l^T\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))(\boldsymbol{f}_l^T\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))^T$ and $\boldsymbol{f}_l^T\boldsymbol{D}\boldsymbol{f}_l$. In detail, the element 1 is assumed to be located in the m-th column of the i-th row of $\boldsymbol{F}$, where $m \in \{1, \cdots, c\}$ keeps changing.

Then, we update the i-th row of $\boldsymbol{F}$, denoting $\boldsymbol{F}^{(s)}$, $s \in \{1, \ldots, c\}$ to verify c kinds of situations including $\{[1, 0, \cdots, 0], \cdots, [0, 0, \cdots, 1]\}$ as $\{\boldsymbol{F}^{(1)}, \cdots, \boldsymbol{F}^{(c)}\}$ with varying the position of 1. Thus, $\boldsymbol{F}^{(s)}$ indicates that only the s-th element in the i-th row of $\boldsymbol{F}$ is 1, whereas the rest are 0. Note that $\boldsymbol{F}^{(s)}$ and $\boldsymbol{F}$ differ only in the i-th row, so Equation (22) becomes

$$\max_{\boldsymbol{F}^{(s)}} g(\boldsymbol{F}^{(s)}) = \sum_{l=1}^{c} \frac{(\boldsymbol{f}_l^{(s)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))(\boldsymbol{f}_l^{(s)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))^T}{\boldsymbol{f}_l^{(s)T}\boldsymbol{D}\boldsymbol{f}_l^{(s)}}, \text{s.t. } \boldsymbol{F}^{(s)} \in \{\boldsymbol{F}^{(1)}, \cdots, \boldsymbol{F}^{(c)}\}, \tag{23}$$

where $\boldsymbol{f}_l^{(s)}$ represents the l-th column of $\boldsymbol{F}^{(s)}$. Thus, problem (22) becomes a search for $\boldsymbol{F}^{(s)}$ that maximizes $g(\boldsymbol{F}^{(s)})$ as the updated optimal $\boldsymbol{F}$. Now, the overall complexity is $\mathcal{O}((n^3 + 3n^2)c^2v)$, which is derived mainly from calculating each type of $\boldsymbol{F}^{(s)}$. For futher simplification, we introduce $\boldsymbol{F}^{(0)}$, in which the i-th row is all 0 and the rest of the rows are the same as the latest updated $\boldsymbol{F}$. Therefore, solving Equation (23) is equivalent to solving the following two-term subtraction:

$$\begin{aligned} \max_{\boldsymbol{F}^{(s)}} \sum_{l=1}^{c} \left(\frac{(\boldsymbol{f}_l^{(s)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))(\boldsymbol{f}_l^{(s)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))^T}{\boldsymbol{f}_l^{(s)T}\boldsymbol{D}\boldsymbol{f}_l^{(s)}} - \frac{(\boldsymbol{f}_l^{(0)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))(\boldsymbol{f}_l^{(0)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))^T}{\boldsymbol{f}_l^{(0)T}\boldsymbol{D}\boldsymbol{f}_l^{(0)}} \right), \\ \text{s.t. } \boldsymbol{F}^{(s)} \in \{\boldsymbol{F}^{(1)}, \cdots, \boldsymbol{F}^{(c)}\}, \end{aligned} \tag{24}$$

where $\boldsymbol{f}_l^{(0)}$ represents the l-th column of $\boldsymbol{F}^{(0)}$. $\boldsymbol{F}^{(s)}$ and $\boldsymbol{F}^{(0)}$ are identical except for the s-th column, so problem (24) becomes

$$\begin{aligned} \max_{\boldsymbol{F}^{(s)}} \mathcal{J}(\boldsymbol{F}^{(s)}) = \frac{(\boldsymbol{f}_s^{(s)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))(\boldsymbol{f}_s^{(s)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))^T}{\boldsymbol{f}_s^{(s)T}\boldsymbol{D}\boldsymbol{f}_s^{(s)}} - \frac{(\boldsymbol{f}_s^{(0)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))(\boldsymbol{f}_s^{(0)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))^T}{\boldsymbol{f}_s^{(0)T}\boldsymbol{D}\boldsymbol{f}_s^{(0)}}, \\ \text{s.t. } \boldsymbol{F}^{(s)} \in \{\boldsymbol{F}^{(1)}, \cdots, \boldsymbol{F}^{(c)}\}, \end{aligned} \tag{25}$$

after which the computational complexity is reduced slightly: $\mathcal{O}((n^2 + 3n)v)$ for calculating one type of $\boldsymbol{F}^{(s)}$, so the overall complexity becomes $\mathcal{O}((n^3 + 3n^2)cv)$. For futher simplification, the main parts in Equation (25), $(\boldsymbol{f}_l^{(s)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))(\boldsymbol{f}_l^{(s)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))^T$, $\boldsymbol{f}_l^{(s)T}\boldsymbol{D}\boldsymbol{f}_l^{(s)}$, $(\boldsymbol{f}_l^{(0)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))(\boldsymbol{f}_l^{(0)T}\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))^T$, $\boldsymbol{f}_l^{(0)T}\boldsymbol{D}\boldsymbol{f}_l^{(0)}$ can be calculated, respectively, with the help of $(\boldsymbol{f}_l^T\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))(\boldsymbol{f}_l^T\boldsymbol{D}\boldsymbol{\phi}_e^H(\boldsymbol{X}))^T$ and $\boldsymbol{f}_l^T\boldsymbol{D}\boldsymbol{f}_l$ in Equation (22), which can be determined once the origional $\boldsymbol{F}$ is given. It is clear that the value of $\mathcal{J}(\boldsymbol{F}^{(s)})$ depends on whether element 1 is settled in the same position of $\boldsymbol{F}^{(s)}$ and $\boldsymbol{F}$: that is, whether s equals to m. Further discussion follows.

When $s = m$, we obtain $f_s^{(s)} = f_s$, and $f_s^{(0)} = f_s - \delta$ denoting $\delta \in \mathbb{R}^n$ as a column vector with the i-th element being 1, and all the rest of the elements are zeros, so Equation (25) can be obtained by

$$(f_s^{(s)T} D\phi_e^H(X))(f_s^{(s)T} D\phi_e^H(X))^T = (f_s^T D\phi_e^H(X))(f_s^T D\phi_e^H(X))^T, \tag{26}$$

$$f_s^{(s)T} D f_s^{(s)} = f_s^T D f_s^T, \tag{27}$$

$$\begin{aligned}(f_s^{(0)T} D\phi_e^H(X))(f_s^{(0)T} D\phi_e^H(X))^T =& (f_s^T D\phi_e^H(X))(f_s^T D\phi_e^H(X))^T \\ &- 2f_s^T D\phi_e^H(X)\phi_e(X)D(:,i) + D(i,:)\phi_e^H(X)\phi_e(X)D(:,i),\end{aligned} \tag{28}$$

$$f_s^{(0)T} D f_s^{(0)} = f_s^T D f_s - 2f_s^T D(:,i) + D_p(i,i), \tag{29}$$

where $D(:,i)$ represents the i-th column of D, $D(i,:)$ represents the i-th row of D, and $D_p(i,i)$ represents the element settled in the i-th row of $D(:,i)$, so we have

$$\mathcal{J}(F^{(s)}) = \frac{(f_s^T D\phi_e^H(X))(f_s^T D\phi_e^H(X))^T}{f_s^T D f_s} - \frac{(f_s^T D\phi_e^H(X))(f_s^T D\phi_e^H(X))^T - 2f_s^T D\phi_e^H(X)\phi_e(X)D(:,i) + D(i,:)\phi_e^H(X)\phi_e(X)D(:,i)}{f_s^T D f_s - 2f_s^T D(:,i) + D_p(i,i)}.$$

When $s \neq m$, we gain $f_s^{(0)} = f_s$, and $f_s^{(s)} = f_s + \delta$, so

$$(f_s^{(0)T} D\phi_e^H(X))(f_s^{(0)T} D\phi_e^H(X))^T = (f_s^T D\phi_e^H(X))(f_s^T D\phi_e^H(X))^T, \tag{30}$$

$$f_s^{(0)T} D f_s^{(0)} = f_s^T D f_s, \tag{31}$$

$$\begin{aligned}(f_s^{(s)T} D\phi_e^H(X))(f_s^{(s)T} D\phi_e^H(X)\phi_e^H(X))^T =& (f_s^T D\phi_e^H(X))(f_s^T D\phi_e^H(X))^T \\ &+ 2f_s^T D\phi_e^H(X)\phi_e(X)D(:,i) + D(i,:)\phi_e^H(X)\phi_e(X)D(:,i),\end{aligned} \tag{32}$$

$$f_s^{(s)T} D f_s^{(s)} = f_s^T D f_s + 2f_s^T D(:,i) + D_p(i,i), \tag{33}$$

and we have

$$\mathcal{J}(F^{(s)}) = \frac{(f_s^T D\phi_e^H(X))(f_s^T D\phi_e^H(X))^T + 2f_s^T D\phi_e^H(X)\phi_e(X)D(:,i) + D(i,:)\phi_e^H(X)\phi_e(X)D(:,i)}{f_s^T D f_s + 2f_s^T D(:,i) + D_p(i,i)} - \frac{(f_s^T D\phi_e^H(X))(f_s^T D\phi_e^H(X))^T}{f_s^T D f_s}.$$

Now, we have simplified the overall computational complexity to $\mathcal{O}(ncv)$ by calculating $f_s^T D\phi_e^H(X)\phi_e(X)D(:,i)$ in Equations (28) and (32) which are both $\mathcal{O}(n)$. Moreover, $\mathcal{O}(ncv)$ can be further reduced to $\mathcal{O}(nv)$ due to the sparsity of F: there is only one element 1 in each row, and the rest are all 0, which do not participate in calculation. According to Equations (30) and (34), we gain the ideal $F^{(s)}$ and reach the optimal $\mathcal{J}(F^{(s)})$ which can be denoted as $F^{(s^*)}$. s^* is the optimal position of 1 in the i-th row, i.e., the optimal cluster indicator of the i-th data sample.

When $s^* = m$, the optimal solution is the same as the one from the lastest iteration; that is, $F^{(s^*)} = F$. If the optimal solution $F^{(s^*)} \neq F$, i.e., $s^* \neq m$, we need to update the saved $(f_l^T D\phi_e^H(X))(f_l^T D\phi_e^H(X))^T, f_l^T D f_l, \forall l \in \{m, s^*\}$ by

$$(f_m^T D\phi_e^H(X))(f_m^T D\phi_e^H(X))^T = (f_m^{(0)T} D\phi_e^H(X))(f_m^{(0)T} D\phi_e^H(X))^T, \tag{34}$$

$$f_m^T D f_m = f_m^{(0)T} D f_m^{(0)}, \tag{35}$$

$$(f_{s^*}^T D\phi_e^H(X))(f_{s^*}^T D\phi_e^H(X))^T = (f_{s^*}^{(s^*)T} D\phi_e^H(X))(f_{s^*}^{(s^*)T} D\phi_e^H(X))^T, \tag{36}$$

$$f_{s^*}^T D f_{s^*} = f_{s^*}^{(s^*)T} D f_{s^*}^{(s^*)}, \tag{37}$$

where $(f_m^{(0)T} D\phi_e^H(X))(f_{(0)^T}^T D\phi_e^H(X))^T$ and $f_m^{(0)T} D f_m^{(0)}$ have been computed by Equations (28) and (29). $(f_{s^*}^{(s^*)T} D\phi_e^H(X))(f_{s^*}^{(s^*)T} D\phi_e^H(X))^T$ and $f_{s^*}^{(s^*)T} D f_{s^*}^{(s^*)}$ have been computed by Equations (32) and (33), so no more calculation is needed. There is no need to

update the values for the other columns. We can complete updating all n rows through iteration and obtain the optimal $\boldsymbol{F}$ as indicated in Algorithm 1. If t_1 iterations are required for this step to converge, then the entire computational complexity is $O(n^2vt_1)$.

Algorithm 1 The procedure to solve problem (10)

Input: Kernels $\{\phi_e^H(\boldsymbol{X}) \in \mathbb{R}^{n\times k}\}_{p=1}^{v}$, number of clusters c.
Parameter: λ controlling the learning pace.
Output: Cluster indicator matrix: $\boldsymbol{F} \in$ Ind.

```
Initialize cluster indicator matrix F and weighting W.
while not converge do
  Update M_p through Equation (16).
  Precompute (f_l^T D φ_e^H(X))(f_l^T D φ_e^H(X))^T as well as f_l^T D f_l,
  ∀l ∈ {1,...,c} and ∀p ∈ {1,...,v}.
  while not converge do
    for i = 1...n do
      Let m be the location of 1 in the i-th row of F.
      for s = 1...c do
        Calculate J(s) by Equation (30) or Equation (34).
      end for
      s* ← arg max_s J(s).
      F(i,s*) ← 1, F(i,m) ← 0.
    end for
  end while
  Update Update W through Equation (12) or Equation (14).
end while
return F
```

Step 3: Update W with M, F fixed. Problem (10) can be reformulated as

$$\begin{aligned} &\min_{\boldsymbol{W}} \sum_{i=1}^{n} \|\phi_e(\boldsymbol{x}_i) - \boldsymbol{M}\boldsymbol{f}_i^H\|_2^2 w_i + f(w_i), \\ &s.t.\ \ w_i \in [0,1], \forall i \in [1,n], \end{aligned} \tag{38}$$

where we denote $\ell_i = \|\phi_e(\boldsymbol{x}_i) - \boldsymbol{M}\boldsymbol{f}_i^T\|_2^2$. The optimal weight for the i-th sample in the p-th kernel with the hard weighting scheme

$$w_i^p = \begin{cases} 1, & \text{if } \ell_i^p \le \lambda, \\ 0, & \text{otherwise}, \end{cases} \tag{39}$$

or with the smoothed weighting scheme

$$w_i^p = \frac{1 + e^{-\lambda}}{1 + e^{\ell_i^p - \lambda}}. \tag{40}$$

Both hard and smoothed weighting schemes can learn multi-kernel models in a meaningful order by progressing both samples and kernels from simple to complex, which aids in eliminating undesirable local optimums. We will conduct a detailed analysis of these two schemes in the experiments.

4. Experiments

The effectiveness of our proposed EKM-SPL method on the hyperspectral image datasets will be examined in this section. To serve as benchmarks, the following classic and important centroid-based clustering algorithms were chosen: k-means [32], FCM [35,36], and FCM-S1 [38]. All of the tests are carried out on a computer running MATLAB 2020a with Radeon Graphics 3.20 GHz and 32 G RAM. Moreover, we make use of the four

hyperspectral image datasets listed below, whose size range from 21,025 to 783,640. We provide descriptions of these datasets, following the guidance of [55,56]:

- Indian Pines: In 1992, this information about Indian Pines was collected by an AVIRIS sensor. The photographic scene has 145 × 145 pixels as well as 220 spectral bands. Water absorption and noise bands were eliminated from the initial 220 bands (i.e., 104–108, 150–163, and 220, respectively). As a result, there are 200 bands of spectral characteristics left. The dataset has 21,025 samples and 16 clusters. Figure 2a,e display the false color picture that contains bands of 46, 18, and 29, corresponding with the map of the ground truth results.
- Salinas: In 1998, the information about Salinas was also obtained by the AVIRIS sensor. The visual scene has 512 × 217 pixels as well as 224 spectral bands. Specifically, 20 bands (i.e., 108–112, 154–167, and 224) of water absorption were deleted to leave the experiment with 204 bands of spectral characteristics. There were 16 clusters in the research, and 111,104 samples were collected. Figure 2b,f display the false color picture that contains bands of 42, 36, and 12, corresponding with the map of the ground truth results.
- Pavia University: In 2000, the information about Pavia University was obtained by the ROSIS sensor. The picture scene has around 610 × 340 pixels and 115 spectral bands. The dataset has been cleansed of some noisy spectral bands, after which 103 bands were left for the experiment. The dataset contains 207,340 samples and *nine* clusters. Figure 2c,g display the false color picture that contains bands of 77, 36, and 22, corresponding with the map of the ground truth results.
- Pavia Centre: In 2000, the information about the Pavia Centre was also obtained by the ROSIS sensor. The photographic scene has 1096 × 1096 pixels as well as 102 spectral bands. The dataset has been cleansed of some noisy spectral bands, as well as the blank samples in the picture, after which 1096 × 715 pixels were left for the experiment. There are 783,640 samples in total divided into *nine* clusters for usage. Figure 2d,h display the false color picture that contains bands of 99, 36, and 24, corresponding with the map of the ground truth results.

In these four aforementioned datasets, there three kinds of sizes: small-size dataset, medium-size dataset, and large-size dataset. In particular, we consider Indian Pines to be a small-size dataset in our studies, Salinas to be a medium-size dataset, and Pavia University and Pavia Centre to be large-size datasets. We also established the ground truth in each dataset to be the number of clusters. All the parameters for the *k*-means [32], FCM [35,36], and FCM-S1 [38] were manually adjusted to their ideal values. The best outcome for each approach is reported after 20 repetitions of the three ways. For a comprehensive assessment of each method, we provide visual clustering maps alongside quantitative metrics, including the user's accuracy, average accuracy (AA), overall accuracy (OA), and kappa coefficient for each conducted experiment. In our EKM-SPL, λ is the parameters controlling the self-learning progress in EKM-SPL, and we set the range at $[2^{-7}, 2^{7}]$ and use grid search to select the best parameter configuration.



(**a**) (**e**)

Figure 2. *Cont.*

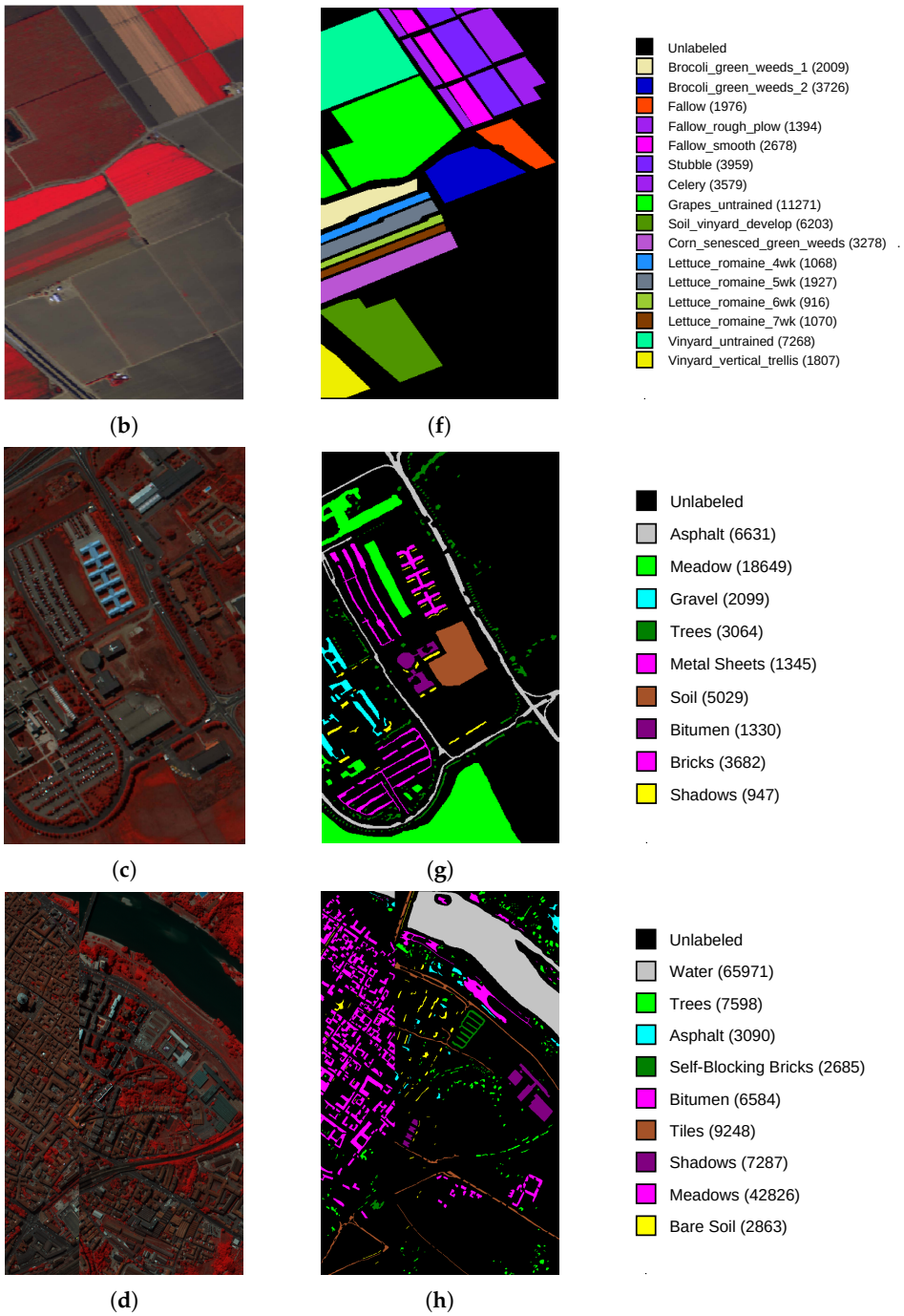


Figure 2. Ground truth maps of four datasets with the false color pictures. (**a**) Indian Pines with bands 46, 18, and 29. (**b**) Salinas image of bands 42, 36, and 12. (**c**) Pavia University image of bands 77, 36, and 22. (**d**) Pavia Centre image with bands 99, 36, and 24. (**e**–**h**) The corresponding ground truth maps. (The number of each cluster is displayed in brackets).

4.1. Small-Size Dataset

To test the efficiency of our proposed EKM-SPL method, we conduct tests on the Indian Pines dataset. From Figure 3a to Figure 3f, the clustering maps created using each clustering technique are displayed. Table 1 provides the corresponding quantitative assessments of the clustering outcomes. The table highlights the second-best outcome by underlining it and displays the best values for each row in bold.

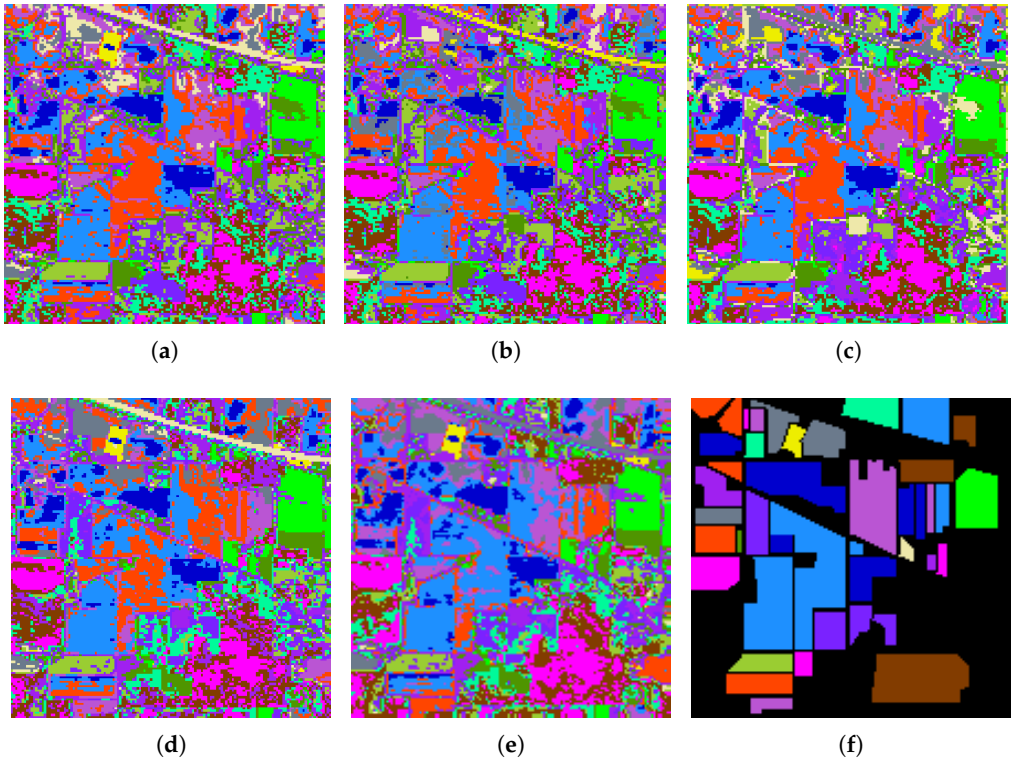


Figure 3. Clustering maps on the Indian Pines dataset. (**a**) *k*-means. (**b**) FCM. (**c**) FCM-S1. (**d**) EKM-SPL-hard (**e**) EKM-SPL-soft. (**f**) Ground truth.

Table 1. Quantitative evaluations on the Indian Pines HSI dataset.

Method	Class	*k*-Means	FCM	FCM-S1	EKM-SPL-Hard	EKM-SPL-Soft
User's accuracy (%)	Alfalfa	0.00	0.00	0.00	0.00	0.00
	Corn-notill	28.15	19.89	28.64	28.57	**36.13**
	Corn-min	**41.08**	38.55	17.95	0.24	14.22
	Corn	**21.94**	14.77	20.68	13.50	5.49
	Grass/Pasture	49.48	49.69	52.59	51.97	**57.76**
	Grass/Trees	33.56	40.96	38.77	**48.49**	**48.49**
	Grass/Pasture-mowed	**82.14**	**82.14**	78.57	0.00	35.71
	Hay-windrowed	75.52	85.98	70.50	**98.12**	87.87
	Oats	55.00	0.00	65.00	**75.00**	0.05
	Soybean-notill	27.98	27.06	37.45	**52.78**	37.86
	Soybean-min	33.85	33.85	36.05	44.89	**58.62**
	Soybean-clean	12.98	25.30	18.04	**26.31**	20.57
	Wheat	96.10	**97.07**	**97.07**	80.00	80.98
	Woods	41.03	41.98	43.32	41.42	**46.72**
	Bldg-Grass-Trees-Drives	16.84	**18.13**	**18.13**	16.06	16.85
	Stone-steel towers	**87.10**	2.15	0.00	81.72	83.87
	AA (%)	**43.92**	36.10	38.92	41.19	39.76
	OA (%)	36.26	35.69	35.99	40.28	**44.30**
	Kappa	29.41	29.34	28.73	32.93	**36.69**

It is evident from Figure 3 and Table 1 that the clustering result of FCM is extremely subpar and has the lowest AA and OA. When compared to FCM, *k*-means and FCM-S1 are more precise, and the AA coefficient increases from 36.10% to 43.92% and 38.92%, respectively. While *k*-means, FCM, and FCM-S1 enhance the clustering performance to produce significantly smoother clustering maps, we see that EKM-SPL-hard and EKM-SPL-soft achieve worse clustering performances. With a kappa value of 36.69% and the best OA of 44.30%, our proposed EKM-SPL approach has the best accuracy. The HESSC (hierarchical sparse subspace clustering) algorithm studied in reference [57] has a kappa of 28.66% on the Indian Pines dataset, while the FECBG (fuzzy embedded clustering based on bipartite graph) algorithm proposed in reference [58] has a kappa of only 33.33% on this dataset.

Additionally, the EKM-SPL-hard technique achieves the maximum precision of 48.49%, 98.12%, 75.00%, 52.78% and 26.31% for the clusters of Grass/Trees, Hay-windrowed, Oats, Soybean-notill and Soybean-clean.

4.2. Medium-Size Dataset

The Salinas dataset is used for the experiments. From Figure 4a to Figure 4f, the clustering maps generated using each clustering technique are displayed, and Table 2 contains the quantitative assessments of the clustering outcomes that correspond to each. The ideal value for each row in this table is displayed in bold, and the second-best outcome is underlined. With the lowest AA of 61.73%, OA of 61.12%, and kappa coefficient of 58.71%, respectively, FCM obtains poor clustering results with a lot of misclassifications, as shown by the clustering maps in Figure 4 and the quantitative evaluations that correspond to those maps in Table 2. By drastically reducing the number of misclassifications, *k*-means and EKM-SPL-soft perform better, improving their kappa scores by about 4.92% and 7.99%, respectively. Additionally, it can be shown that in this picture, *k*-means, EKM-SPL-soft, and FCM-S1 produce much better clustering results than FCM with advantages. With the best AA of 67.47%, OA of 68.20%, and kappa coefficient of 68.20%, our proposed EKM-SPL-soft approach successfully distinguishes practically all clusters to produce the best visual result and the greatest accuracy. The HESSC algorithm [57] has a kappa of 47.87% on the Salinas dataset, while the FECBG algorithm [58] has a kappa of only 64.27%. It is important to note that whereas other methods have almost little recognition, the Lettuce_4wk class has been effectively discriminated to some extent. Additionally, the Lettuce_6wk cluster stands out nicely and has accuracy close to 100%, which is superior to all the other approaches.

Table 2. Quantitative evaluations on the Salinas dataset.

Method	Class	*k*-Means	FCM	FCM-S1	EKM-SPL-Hard	EKM-SPL-Soft
User's accuracy (%)	Brocoli_weeds_1	0.00	44.21	0.00	**98.11**	92.11
	Brocoli_weeds_2	**93.24**	62.21	85.06	91.98	91.14
	Fallow	**00.07**	0.00	0.00	0.00	0.00
	Fallow_rough	**99.64**	99.35	97.27	99.07	99.14
	Fallow_smooth	97.31	97.68	90.10	98.13	**98.21**
	Stubble	88.31	94.14	95.20	**96.39**	96.29
	Celery	97.76	98.30	94.78	**99.33**	99.16
	Grapes_untrained	62.67	55.74	51.34	**65.54**	59.07
	Soil	**93.66**	47.86	78.43	67.55	81.49
	Corn	1.07	46.92	30.11	48.75	**49.73**
	Lettuce_4wk	0.00	0.09	0.00	0.00	**65.92**
	Lettuce_5wk	47.64	0.00	**97.04**	86.20	90.56
	Lettuce_6wk	97.16	98.69	96.07	98.91	**99.02**
	Lettuce_7wk	86.54	**88.97**	70.93	87.57	87.66
	Vinyard_untrained	58.60	**67.47**	53.45	30.23	62.08
	Vinyard_trellis	0.00	5.01	0.00	67.28	**72.83**
	AA (%)	65.23	61.73	65.44	62.82	**67.47**
	OA (%)	65.54	61.12	63.33	62.23	**68.20**
	Kappa	62.51	58.71	57.59	58.73	**64.58**

4.3. Large-Size Datasets

We conduct experiments on the Pavia University and Pavia Centre datasets to illustrate the scalability of the EKM-SPL method. Figures 5 and 6 illustrate the clustering maps for the Pavia University and Pavia Centre datasets, respectively. The accompanying quantitative analyses of the clustering findings are shown in Tables 3 and 4. The total number of samples increases as the scale of the Pavia University and Pavia Centre databases grows to 207,340 and 783,640, respectively.

According to Figure 5 and Table 3, *k*-means, FCM, and FCM-S1 gain poor clustering results and contain considerable quantities of misclassifications, with low OAs of 52.44%, 52.85% and 50.36%, respectively. The proposed EKM-SPL-soft method excels both visually and quantitatively, attaining the superior clustering performance with the highest overall accuracy of 57.98% and kappa coefficient of 48.15%. The HESSC algorithm [57] has a kappa of 36.26% on the Pavia University dataset, while the FECBG algorithm [58] has a kappa of only 43.02%. It is stated that the classifications of meadows, trees, and self-blocking bricks are successfully discriminated, and the recognition level is substantially greater than that of other approaches. Furthermore, the Painted metal sheets class is recognized well in EKM-SPL-hard, and the accuracy is near 92.19%, but the recognition level of other techniques is quite low.

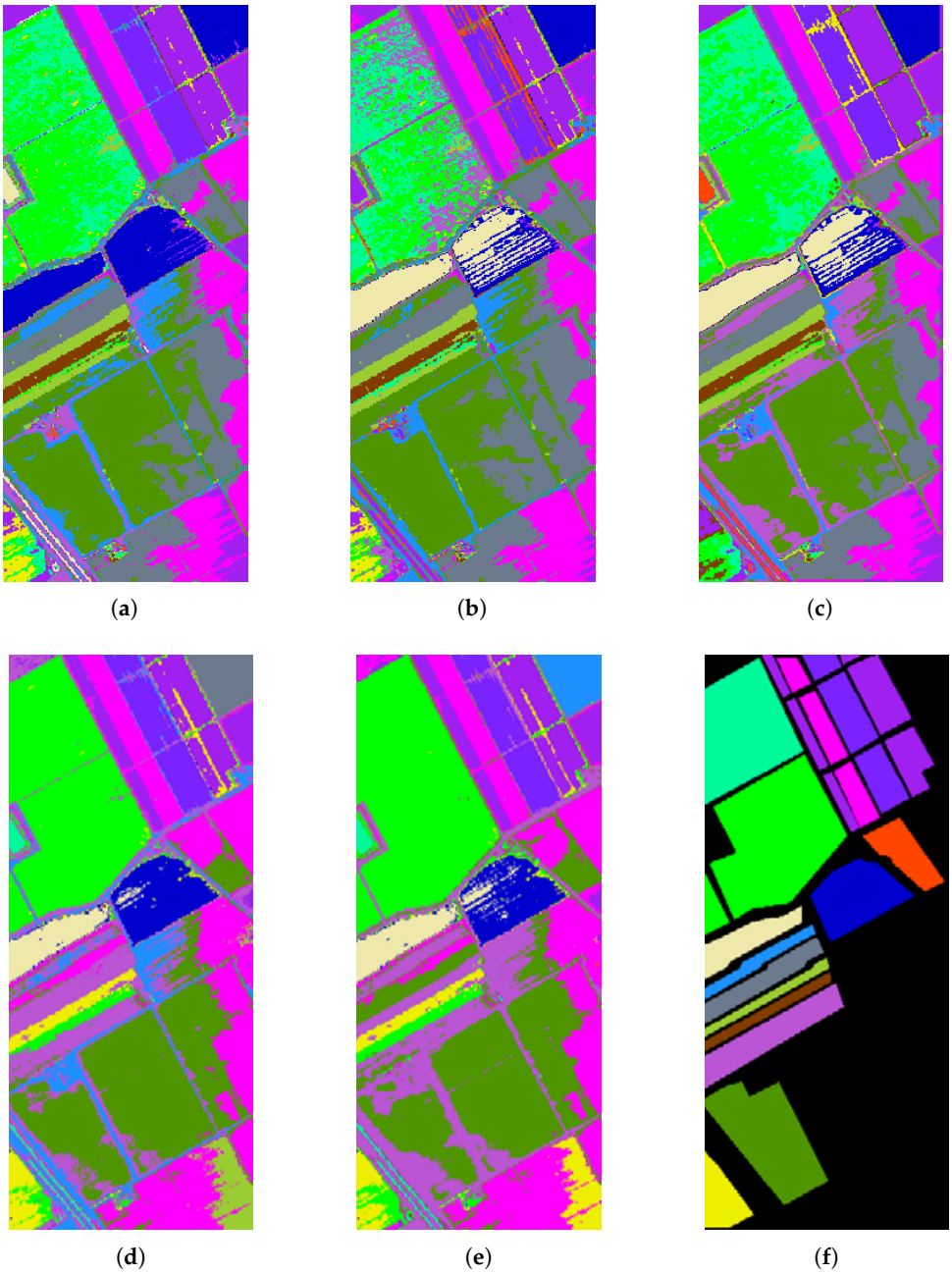


Figure 4. Clustering maps on the Salinas dataset. (**a**) *k*-means. (**b**) FCM. (**c**) FCM-S1. (**d**) EKM-SPL-hard (**e**) EKM-SPL-soft. (**f**) Ground truth.

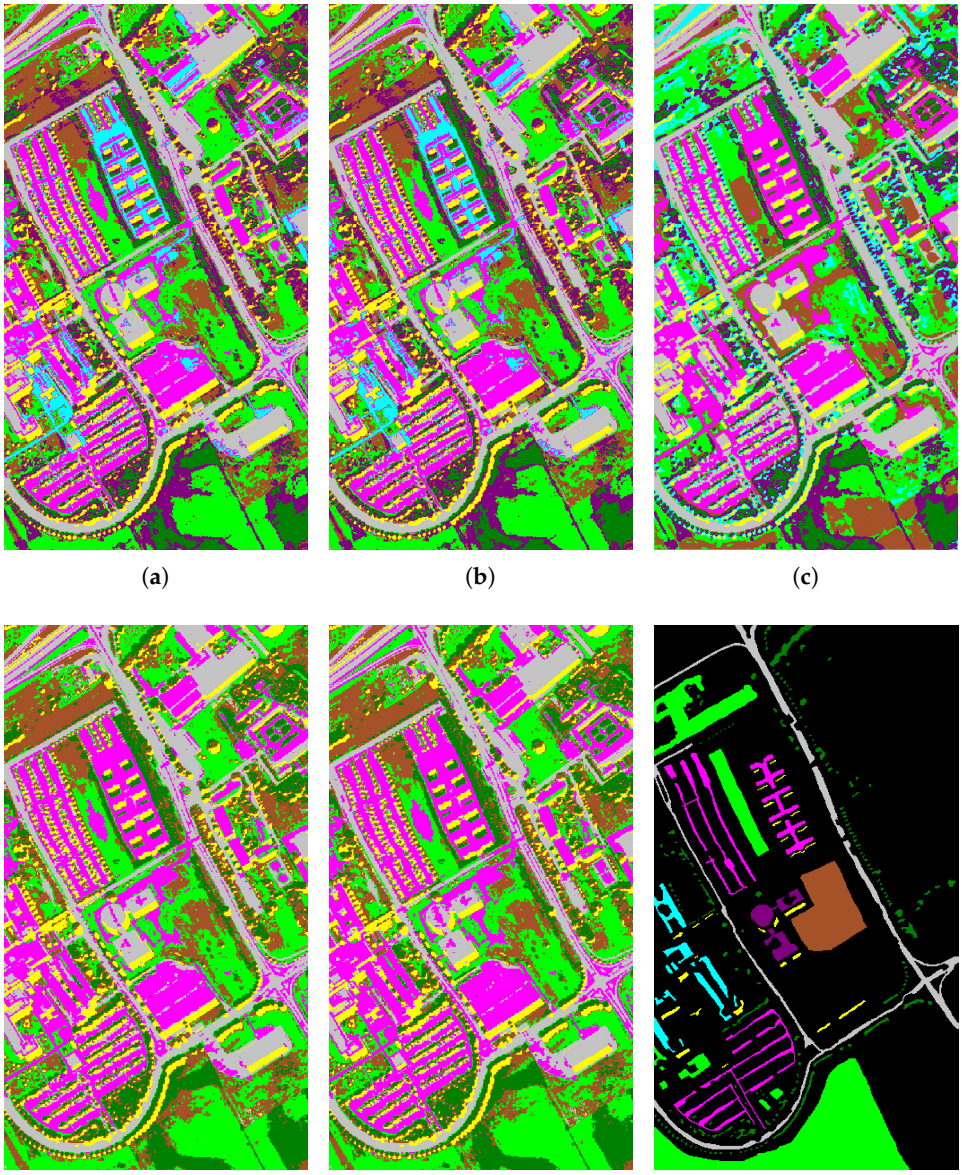


Figure 5. Clustering maps on the Pavia University dataset. (**a**) *k*-means. (**b**) FCM. (**c**) FCM-S1. (**d**) EKM-SPL-hard. (**e**) EKM-SPL-soft. (**f**) Ground truth.

Table 3. Quantitative evaluations on the Pavia University dataset.

Method	Class	*k*-Means	FCM	FCM-S1	EKM-SPL-Hard	EKM-SPL-Soft
User's accuracy (%)	Asphalt	90.24	90.01	**96.12**	90.60	85.48
	Meadows	42.35	44.11	35.19	46.67	**48.55**
	Gravel	**0.03**	**0.01**	0.00	0.00	0.00
	Trees	63.22	61.98	59.27	86.62	**87.60**
	Painted metal sheets	43.82	50.74	77.70	**92.19**	85.13
	Bare Soil	30.11	32.66	29.59	**35.00**	33.82
	Bitumen	0.00	0.00	0.00	0.00	0.00
	Self-Blocking Bricks	93.20	94.55	79.14	94.02	**97.85**
	Shadows	**100.00**	**100.00**	99.89	**100.00**	**100.00**
AA (%)		51.99	52.39	53.31	**60.57**	59.83
OA (%)		52.44	52.85	50.36	57.92	**57.98**
Kappa		43.27	42.87	40.29	48.12	**48.15**

According to Figure 6 and Table 4, *k*-means, FCM, and FCM-S1 provide lower clustering results and have a high number of misclassifications with low AA values of 44.92%,

45.09% and 45.18%, respectively. With the best AA of 52.10%, OA of 74.52% and kappa coefficient of 64.98%, the proposed EKM-SPL-soft obtains the best clustering performance both visually and statistically. It is worth noting that the Tiles class in EKM-SPL-hard is effectively differentiated with a substantially higher recognition level than the others.

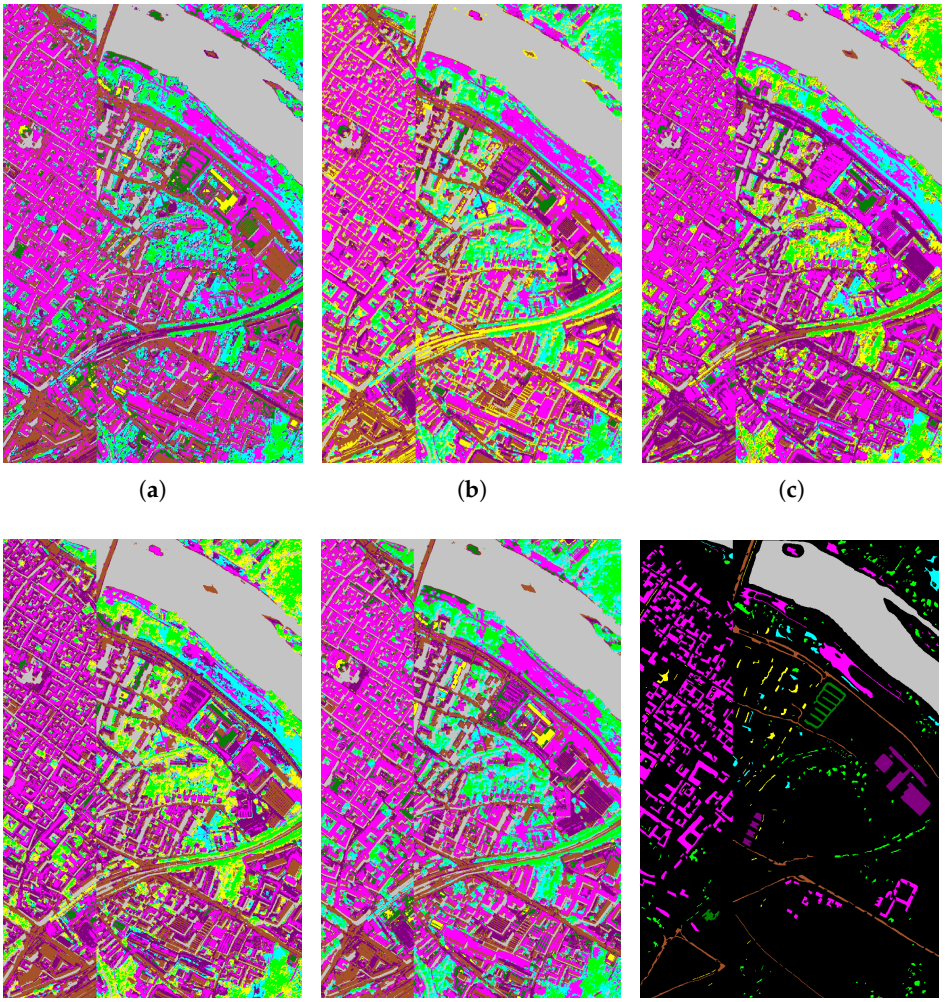


Figure 6. Clustering maps on the Pavia Centre dataset. (**a**) *k*-means. (**b**) FCM. (**c**) FCM-S1. (**d**) EKM-SPL-hard. (**e**) EKM-SPL-soft (**f**) Ground truth.

Table 4. Quantitative evaluations on the Pavia Centre dataset.

Method	Class	*k*-Means	FCM	FCM-S1	EKM-SPL-Hard	EKM-SPL-Soft
User's accuracy (%)	Water	98.32	97.32	97.12	**99.37**	99.36
	Trees	62.60	63.27	71.22	**75.74**	72.90
	Asphalt	12.90	**12.99**	10.01	9.97	8.51
	Self-Blocking Bricks	**58.11**	11.02	11.43	13.05	50.58
	Bitumen	23.35	49.01	**50.30**	36.60	25.26
	Tiles	82.23	85.01	20.99	**93.99**	91.72
	Shadows	0.91	23.21	**80.71**	24.23	74.50
	Meadows	46.95	51.39	52.10	**55.54**	52.04
	Bare Soil	0.00	**0.05**	0.00	0.03	0.00
AA (%)		44.92	45.09	45.18	45.37	**52.10**
OA (%)		68.33	70.28	71.56	73.31	**74.52**
Kappa		55.78	62.33	63.18	63.19	**64.98**

5. Conclusions

In this work, we proposed a new Euler kernel mapping for hyperspectral images clustering via self-paced learning. To better represent the complex internal structure of HSIs, it quietly leverages the Euler kernel mapping to shift the HSI data to a complex feature space, where nonlinearly inseparable clusters may become linearly separable. Furthermore, the Euler kernel mapping can remove the influence of noisy features, making it ideal for HSI clustering. Furthermore, self-paced learning (SPL) is used to train the clustering model in a logical order by moving sample pixels from simple to complex, which can help to jump out of the undesirable local optimums. Meanwhile, rather than categorizing samples as "easy" or "complex", a probabilistic smoothing weighting method is used to quantify difficulty across them, making the optimization process more logical. Finally, thorough testings using 4 classic and common HSI datasets demonstrate the efficacy and superiority of our proposed EKM-SPL method.

Author Contributions: Conceptualization, F.Z. and H.Y.; methodology, F.Z.; software, H.Y.; validation, J.Z.; formal analysis, H.H.; investigation, J.Z.; resources, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant number 42401499.

Data Availability Statement: Publicly available datasets were analyzed in this study, which can be found here: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes (accessed on 1 March 2024).

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

References

1. Li, X.; Zhang, L.; You, J. Domain Transfer Learning for Hyperspectral Image Super-Resolution. *Remote Sens.* **2019**, *11*, 694. [CrossRef]
2. Cheng, F.; Chen, N.; Wang, C.; Wang, Q.; Du, B. A Global-to-local Evolutionary Algorithm for Hyperspectral Endmember Extraction. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5504117. [CrossRef]
3. Zhai, H.; Lv, C.; Liu, W.; Yang, C.; Fan, D.; Wang, Z.; Guan, Q. Understanding Spatio-Temporal Patterns of Land Use/Land Cover Change under Urbanization in Wuhan, China, 2000–2019. *Remote Sens.* **2021**, *13*, 3331. [CrossRef]
4. Zhai, H.; Zhang, H.; Li, P.; Zhang, L. Hyperspectral Image Clustering: Current achievements and future lines. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 35–67. [CrossRef]
5. Lu, H.; Su, H.; Hu, J.; Du, Q. Dynamic Ensemble Learning with Multi-view Kernel Collaborative Subspace Clustering for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 2681–2695. [CrossRef]
6. Bhatti, U.A.; Huang, M.; Neira-Molina, H.; Marjan, S.; Baryalai, M.; Tang, H.; Wu, G.; Bazai, S.U. MFFCG–Multi Feature Fusion for Hyperspectral Image Classification Using Graph Attention Network. *Expert Syst. Appl.* **2023**, *229*, 120496. [CrossRef]
7. Zhai, H.; Zhang, H.; Xu, X.; Zhang, L.; Li, P. Kernel Sparse Subspace Clustering with a Spatial Max Pooling Operation for Hyperspectral Remote Sensing Data Interpretation. *Remote Sens.* **2017**, *9*, 335. [CrossRef]
8. Hu, H.; He, F.; Zhang, F.; Ding, Y.; Wu, X.; Zhao, J.; Yao, M. Unifying Label Propagation and Graph Sparsification for Hyperspectral Image Classification. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
9. Kruse, F.A.; Boardman, J.W.; Huntington, J.F. Comparison of airborne hyperspectral data and EO-1 Hyperion for mineral mapping. *IEEE Trans. Geosci. Remote. Sens.* **2003**, *41*, 1388–1400. [CrossRef]
10. Kong, B.; Yu, H.; Du, R.; Wang, Q. Quantitative estimation of biomass of alpine grasslands using hyperspectral remote sensing. *Rangel. Ecol. Manag.* **2019**, *72*, 336–346. [CrossRef]
11. Shimoni, M.; Haelterman, R.; Perneel, C. Hypersectral imaging for military and security applications: Combining myriad processing and sensing techniques. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 101–117. [CrossRef]
12. Ikotun, A.M.; Ezugwu, A.E.; Abualigah, L.; Abuhaija, B.; Heming, J. K-means Clustering Clgorithms: A comprehensive Review, Variants Analysis, and Advances in the Era of Big Data. *Inf. Sci.* **2023**, *622*, 178–210. [CrossRef]
13. Lu, J.; Lu, Y.; Wang, R.; Nie, F.; Li, X. Multiple Kernel K-Means Clustering with Simultaneous Spectral Rotation. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 4143–4147.
14. Lu, Y.; Zheng, X.; Lu, J.; Wang, R.; Nie, F.; Li, X. Self-Paced and Discrete Multiple Kernel *k*-Means. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 4284–4288.
15. Wang, R.; Chen, H.; Lu, Y.; Zhang, Q.; Nie, F.; Li, X. Discrete and Balanced Spectral Clustering with Scalability. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 14321–14336. [CrossRef] [PubMed]

16. Ding, Y.; Zhang, Z.; Zhao, X.; Cai, Y.; Li, S.; Deng, B.; Cai, W. Self-supervised locality preserving low-pass graph convolutional embedding for large-scale hyperspectral image clustering. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5536016. [CrossRef]
17. Zhai, H.; Zhang, H.; Zhang, L.; Li, P. Nonlocal Means Regularized Sketched Reweighted Sparse and Low-Rank Subspace Clustering for Large Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 4164–4178. [CrossRef]
18. Zhai, H.; Zhang, H.; Zhang, L.; Li, P. Total Variation Regularized Collaborative Representation Clustering with a Locally Adaptive Dictionary for Hyperspectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 166–180. [CrossRef]
19. Zhai, H.; Zhang, H.; Zhang, L.; Li, P. Sparsity-Based Clustering for Large Hyperspectral Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 10410–10424. [CrossRef]
20. Ding, Y.; Zhang, Z.; Zhao, X.; Cai, W.; Yang, N.; Hu, H.; Huang, X.; Cao, Y.; Cai, W. Unsupervised self-correlated learning smoothy enhanced locality preserving graph convolution embedding clustering for hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5536716. [CrossRef]
21. Mao, J.; Jain, A.K. A self-organizing network for hyperellipsoidal clustering (HEC). *IEEE Trans. Neural Netw.* **1996**, *7*, 16–29.
22. Ma, Y.; Lao, S.; Takikawa, E.; Kawade, M. Discriminant analysis in correlation similarity measure space. In Proceedings of the 24th Annual International Conference on Machine Learning Held in Conjunction with the 2007 International Conference on Inductive Logic Programming, Corvalis OR, USA, 20–24 June 2007; Volume 227, pp. 577–584.
23. Chen, J.; Jia, X.; Yang, W.; Matsushita, B. Generalization of Subpixel Analysis for Hyperspectral Data with Flexibility in Spectral Similarity Measures. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 2165–2171. [CrossRef]
24. Zhai, H.; Zhang, H.; Zhang, L.; Li, P. Laplacian-Regularized Low-Rank Subspace Clustering for Hyperspectral Image Band Selection. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1723–1740. [CrossRef]
25. Chen, J.; Liu, S.; Zhang, Z.; Wang, H. Diffusion Subspace Clustering for Hyperspectral Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 6517–6530. [CrossRef]
26. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [CrossRef] [PubMed]
27. Shi, J.; Malik, J. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
28. Zhu, W.; Chayes, V.; Tiard, A.; Sánchez, S.; Dahlberg, D.; Bertozzi, A.L.; Osher, S.J.; Zosso, D.; Kuang, D. Unsupervised Classification in Hyperspectral Imagery with Nonlocal Total Variation and Primal-Dual Hybrid Gradient Algorithm. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2786–2798. [CrossRef]
29. Huang, N.; Xiao, L.; Xu, Y. Bipartite Graph Partition Based Coclustering with Joint Sparsity for Hyperspectral Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2019**, *12*, 4698–4711. [CrossRef]
30. Nie, F.; Zhu, W.; Li, X. Unsupervised Large Graph Embedding Based on Balanced and Hierarchical K-Means. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 2008–2019. [CrossRef]
31. Zhong, Y.; Zhang, L.; Huang, B.; Li, P. An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 420–431. [CrossRef]
32. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A *k*-means clustering algorithm. *J. R. Stat. Soc.* **1979**, *28*, 100–108. [CrossRef]
33. El Rahman, S.A. Hyperspectral imaging classification using ISODATA algorithm: Big data challenge. In Proceedings of the 2015 Fifth International Conference on e-Learning (econf), Manama, Bahrain, 18–20 October 2015; pp. 247–250.
34. Zhang, B.; Li, S.; Wu, C.; Gao, L.; Zhang, W.; Peng, M. A neighbourhood-constrained k-means approach to classify very high spatial resolution hyperspectral imagery. *Remote Sens. Lett.* **2013**, *4*, 161–170. [CrossRef]
35. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Springer: New York, NY, USA, 1981.
36. Ahn, C.W.; Baumgardner, M.; Biehl, L. Delineation of soil variability using geostatistics and fuzzy clustering analyses of hyperspectral data. *Soil Sci. Soc. Am. J.* **1999**, *63*, 142–150. [CrossRef]
37. Wang, X.; Bu, J. A fast and robust image segmentation using FCM with spatial information. *Digit. Signal Process.* **2010**, *20*, 1173–1182. [CrossRef]
38. Chen, S.; Zhang, D. Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. *IEEE Trans. Syst. Man Cybern. Part B* **2004**, *34*, 1907–1916. [CrossRef] [PubMed]
39. Zhong, Y.; Ma, A.; Zhang, L. An Adaptive Memetic Fuzzy Clustering Algorithm with Spatial Information for Remote Sensing Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2014**, *7*, 1235–1248. [CrossRef]
40. Lu, Y.; Xin, H.; Wang, R.; Nie, F.; Li, X. Scalable Multiple Kernel *k*-means Clustering. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 4279–4283.
41. Tzimiropoulos, G.; Zafeiriou, S.; Pantic, M. Subspace Learning from Image Gradient Orientations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2454–2466. [CrossRef]
42. Wang, R.; Lu, J.; Lu, Y.; Nie, F.; Li, X. Discrete and Parameter-Free Multiple Kernel k-Means. *IEEE Trans. Image Process.* **2022**, *31*, 2796–2808. [CrossRef]
43. Schölkopf, B.; Smola, A.; Müller, K. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Comput.* **1998**, *10*, 1299–1319. [CrossRef]
44. Rengasamy, S.; Murugesan, P. K-means-Laplacian Clustering Revisited. *Eng. Appl. Artif. Intell.* **2022**, *107*, 104535. [CrossRef]
45. Wang, R.; Lu, J.; Lu, Y.; Nie, F.; Li, X. Discrete Multiple Kernel k-means. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21), Virtual, 19–27 August 2021; pp. 3111–3117.
46. Liu, X. Simplemkkm: Simple Multiple Kernel K-means. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 5174–5186. [CrossRef]

47. Lu, Y.; Lu, J.; Wang, R.; Nie, F. Discrete Multi-Kernel K-Means with Diverse and Optimal Kernel Learning. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 4153–4157.
48. Schölkopf, B. The Kernel Trick for Distances. *Adv. Neural Inf. Process. Syst.* **2000**, *13*, 301–307.
49. Wu, J.; Zheng, W.; Lai, J.; Suen, C.Y. Euler Clustering on Large-Scale Dataset. *IEEE Trans. Big Data* **2018**, *4*, 502–515. [CrossRef]
50. Liwicki, S.; Tzimiropoulos, G.; Zafeiriou, S.; Pantic, M. Euler Principal Component Analysis. *Proc. Int. J. Comput. Vis.* **2013**, *101*, 498–518. [CrossRef]
51. Klink, P.; Abdulsamad, H.; Belousov, B.; D'Eramo, C.; Peters, J.; Pajarinen, J. A Probabilistic Interpretation of Self-paced Learning with Applications to Reinforcement Learning. *J. Mach. Learn. Res.* **2021**, *22*, 1–52.
52. Kumar, M.; Packer, B.; Koller, D. Self-Paced Learning for Latent Variable Models. *Adv. Neural Inf. Process. Syst.* **2010**, *23*, 1189–1197.
53. Tang, K.; Ramanathan, V.; Li, F.-F.; Koller, D. Shifting Weights: Adapting Object Detectors from Image to Video. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 647–655.
54. Wright, S. Coordinate descent algorithms. *Math. Program.* **2015**, *151*, 3–34. [CrossRef]
55. Wang, R.; Nie, F.; Hong, R.; Chang, X.; Yang, X.; Yu, W. Fast and Orthogonal Locality Preserving Projections for Dimensionality Reduction. *IEEE Trans. Image Process.* **2017**, *26*, 5019–5030. [CrossRef]
56. Deng, C.; Xue, Y.; Liu, X.; Li, C.; Tao, D. Active Transfer Learning Network: A Unified Deep Joint Spectral-Spatial Feature Learning Model for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1741–1754. [CrossRef]
57. Shahi, K.R.; Khodadadzadeh, M.; Tusa, L.; Ghamisi, P.; Gloaguen, R. Hierarchical Sparse Subspace Clustering (HESSC): An Automatic Approach for Hyperspectral Image Analysis. *Remote Sens.* **2020**, *12*, 2421. [CrossRef]
58. Yang, X.; Xu, Y.; Li, S.; Liu, Y.; Liu, Y. Fuzzy Embedded Clustering Based on Bipartite Graph for Large-Scale Hyperspectral Image. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 5505605. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.