Kernel Low-Rank Entropic Component Analysis for Hyperspectral Image Classification

Chengzu Bai, Ren Zhang, Zeshui Xu, *Fellow, IEEE*, Baogang Jin, Jian Chen, Shuo Zhang, and Longxia Qian

***Abstract*—Principal component analysis (PCA) and its variations are still the primary tool for feature extraction (FE) in the remote sensing community. This is unfortunate, as there has been a strong argument against using PCA for this purpose due to its inherent linear properties and uninformative principal components. Therefore, several critical issues still should be considered in the hyperspectral image classification when using PCA, among which: the large number of spectral channels and a small number of training samples; the nonlinearities of hyperspectral data; the small-sample issue. In order to alleviate these problems, this article employs a new information-theoretic FE method, the so-called kernel entropic component analysis (KECA), which can not only extract more nonlinear information but also can adapt to the limited-sample environment. A theorem of the pivoted Cholesky decomposition is also introduced to improve the efficiency of the KECA. The optimized version can more rapidly implement spectral-spatial features extraction, particularly for large-scale HSIs, while effectively maintaining the clustering performances of KECA. Experiments implemented on several real HSIs verify the effectiveness of the new method armed with a support vector machine classifier, in comparison with other PCA-based and state-of-the-art HSI classification algorithms. The code will also be made publicly available.**

***Index Terms*—Feature extraction (FE), image classification, maximum entropy methods.**

I. Introduction

DUE to the recently developed remote sensors, a number of spatial-spectral information contained in hyperspectral images (HSIs) offer a unique opportunity to conduct a detailed physical analysis of different land covers. In particular, HSI classification, i.e., assigning a class label to each pixel of the interest with a certain number of labeled training samples, can play a key role in many applications, such as environment monitoring [1], target detection [2], and land cover mapping [3], to name but a few. However, three critical problems still should be addressed in the HSI classification, which are as follows.

1) The curse of dimensionality caused by the comparison between the high dimension of spectral information and the low number of training samples [4].
2) The nonlinearities of hyperspectral data derived from the backscattering or spatial arrangement of distinct pixels [5], [6].
3) Poor performances under the small-sample environment.

In order to alleviate the first issue (i.e., the Hughes phenomenon), spectral-spatial classifiers and feature extraction (FE) are the two most adopted methods before the classification. Several well-known classifiers, e.g., support vector machines (SVMs) [7]–[10] and rotation forest classifiers [11]–[14], can achieve impressive results on the HSI classification. However, most of the spectral and/or spatial classifiers are unable to obtain an acceptable classification performance under a small number of labeled samples [15]. To further improve the classification results of HSIs, the feature-selection step is usually employed. It is a popular practice to apply PCA and its variants [16]–[24] to perform such data transformation. For instance, the extended morphological profiles (EMPs) are built on the first principal component of HSI to eliminate the effects of spectral variability [18]. Zabalza *et al*. [19] proposed the folded-PCA (FPCA) to handle the high computational cost and large memory requirement for the usage of PCA in HSI data reduction. Considering the superior properties of FPCA, several FPCA-based models are newly designed for HSI classification [20], [21]. In special, Uddin *et al*. [21] proposed a novel method so-called spectrally-segmented-FPCA (SSFPCA), in which they apply FPCA on the HSI datasets mixed with highly correlated and spectrally separated segments instead of directly applying the FPCA on the entire data, to obtain better classification performances. PCA is also employed to reduce the spectral dimension of HSIs preprocessed by the edge-preserving filters (EPFs) [22], [23] and the invariant attribute profiles (IAPs) [24], to name but a few (see [25] for more details of the PCA-based methods on extracting HSI features). While PCA has been proven to play a key role in the success of HSI classification, it is not necessarily an optimal FE method, since that PCA cannot describe the complex nonlinear structure in the HSIs very well due to its inherent linear characteristic. Additionally, the principal components do not always retain the discriminatory information stored in the original feature space. Therefore, how to appropriately tackle

Manuscript received June 16, 2020; revised August 11, 2020 and September 10, 2020; accepted September 11, 2020. Date of publication September 15, 2020; date of current version October 6, 2020. This work was supported by the National Natural Science Foundation of China under Grant 41706021. *(Corresponding author: Chengzu Bai.)*

Chengzu Bai, Baogang Jin, Jian Chen, and Shuo Zhang are with the Beijing Institute of Applied Meteorology, Beijing 100029, China, and also with the State Key Laboratory of Geo-Information Engineering, Xi'an 710054, China (e-mail: baiczu@163.com; jinbgang@163.com; chenj03@126.com; zhangshuo93@126.com).

Ren Zhang is with the College of Meteorology and Oceanography, National University of Defense Technology, Nanjing 211101, China (e-mail: zrpaper@163.com).

Zeshui Xu is with the Business School, Sichuan University, Chengdu 610064, China (e-mail: xuzeshui@263.net).

Longxia Qian is with the School of Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China (e-mail: qianlongxia@aliyun.com).

Digital Object Identifier 10.1109/JSTARS.2020.3024241

these issues associated with PCA is one of our main motivations in this article.

Our another center idea lies in taking the full advantage of nonlinear information contained in HSIs to extract more discriminative features from each pixel with limited training samples, i.e., the other two aforementioned issues. As well-known, kernel methods can provide a rigorous mathematical framework to implement nonlinear FE [4]. The kernel principal component analysis (KPCA) [26] is one of the most frequently adopted FE approaches in the remote sensing community [27]. Although KPCA has obtained impressive results in different fields, its data transformation is often performed by using the eigenvectors of the kernel matrix corresponding to the top eigenvalues, which cannot provide enough information and completely reveal the structure of raw data from the perspective of information theory [28].

Regarding the aforementioned issues, Jessen [29] proposed a new FE method named the kernel entropy component analysis (KECA), which is different from the conventional kernel methods of extracting nonlinear information. KECA attempts to search for optimal entropic components (i.e., eigenvectors) in the reproducing kernel Hilbert space (RKHS) that can compress the most Renyi entropy of the input data, instead of ranking eigenvalues and selecting the associated eigenvectors by PCA and KPCA. This is not only the distinct property of KECA but exactly the reason why KECA can mine more information than PCA and KPCA in FE under the small-sample environment [29], [30].

However, a direct use of the KECA in the HSI classification can be inefficient since adjacent noise-free hyperspectral bands characterized by high correlation and high spectral dimensionality will greatly increase the computational burden of the eigendecomposition. More specifically, let $N_1 = n_1 \times n_2$ denote the spatial size of an HSI $\mathbf{I}$ ($\mathbf{I} \in \Re^{n_1 \times n_2 \times n_3}$). For the classical KECA, $\mathbf{I}$ needs to be converted to an $N_1 \times n_3$ matrix to implement the singular value decomposition. There is the difficulty in obtaining the corresponding kernel matrix $\mathbf{K}$ ($\mathbf{K} \in \Re^{N_1 \times N_1}$) considering that the dimension N_1 is extremely large, usually over 100K, which can easily cause software tools crashed due to the memory management.

To solve the aforementioned problems existing in the HSI classification, we design a new three-stage classification scheme named as the kernel low-rank entropic component analysis (KLRECA). First, several sophisticated preprocessing techniques, such as EPFs or IAPs, are selected to remove noise and enhance the spectral-spatial information in the original HSI data according to different structures of the image. Second, the high spectral dimension of the processed data is reduced by the KECA, where the full kernel matrix is replaced by its Nyström-type low-rank approximation using the pivoted Cholesky decomposition [31], [32]. This proposal not only can highlight the separability of pixels but save much more computational time and memory space. It should be noted that the optimized KECA is utilized not only to explore spectral information but also to take the advantage of additional discriminant information from spatial dimension. Finally, the reduced data are classified by an SVM classifier. The rationale to select SVM is that it can be integrated into the kernel method category [6] so that more nonlinear information may be extracted from the entropic components. The novelty of this work consists of the following.

1) A new HSI classification framework consisting of the advanced preprocessing techniques and KECA via an appropriate strategy.
2) Introducing a novel FE technique, i.e., KECA improved by the pivoted Cholesky decomposition, to the remote sensing field.
3) Comparing the new classification scheme with not only the classical methods but also the state-of-the-art deep learning algorithms, such as the nonlinear spectral-spatial network (NSSNet) [33] as well as the rolling guidance filter and vertex component analysis network (RVCANet) [34].

The remainder of this article is organized as follows. Section II reviews KECA. Section III shows the proposed KLRECA classification scheme. Section IV examines the performance of the KLRECA on three real-world HSIs. We conclude this article in Section V.

II. Preliminaries

A. Kernel Entropy Component Analysis

The KECA algorithm is reviewed at first for readers' easy understanding.

The Renyi quadratic entropy is given by

$$H(p) = -\log \int p^2(\mathbf{x}) d\mathbf{x} \tag{1}$$

where $p(\mathbf{x})$ denotes the probability density function. (1) helps to measure the information preserved in the given input dataset $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ ($\mathbf{x}_i \in \Re^D$) [29]. Due to the monotonic property of logarithmic function, we can focus on the quantity of (1)

$$V(p) = \int p^2(\mathbf{x}) d\mathbf{x}. \tag{2}$$

Then, the kernel $k_\sigma(\mathbf{x}, \mathbf{x}_t)$ of the Parzen window density estimator, where σ represents the kernel size [29], is employed to estimate (2) (it should be noted here that a Gaussian kernel is used throughout the article)

$$\begin{aligned} V(p) &\approx \hat{V}(p) \\ &= \frac{1}{N} \sum_{\mathbf{x} \in \mathbf{X}} p(\mathbf{x}) \\ &= \frac{1}{N} \sum_{\mathbf{x}_i \in \mathbf{X}} \frac{1}{N} \sum_{\mathbf{x}_j \in \mathbf{X}} \kappa_\sigma(\mathbf{x}_i, \mathbf{x}_j) \\ &= \frac{1}{N^2} \mathbf{1}^T \mathbf{K} \mathbf{1} \end{aligned} \tag{3}$$

where the kernel matrix $\mathbf{K}$ is equal to $k_\sigma(\mathbf{x}, \mathbf{x}_t)$. It consists of $\mathbf{K}_{ij} = k_\sigma(\mathbf{x}, \mathbf{x}_t) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$, where ϕ can map the samples

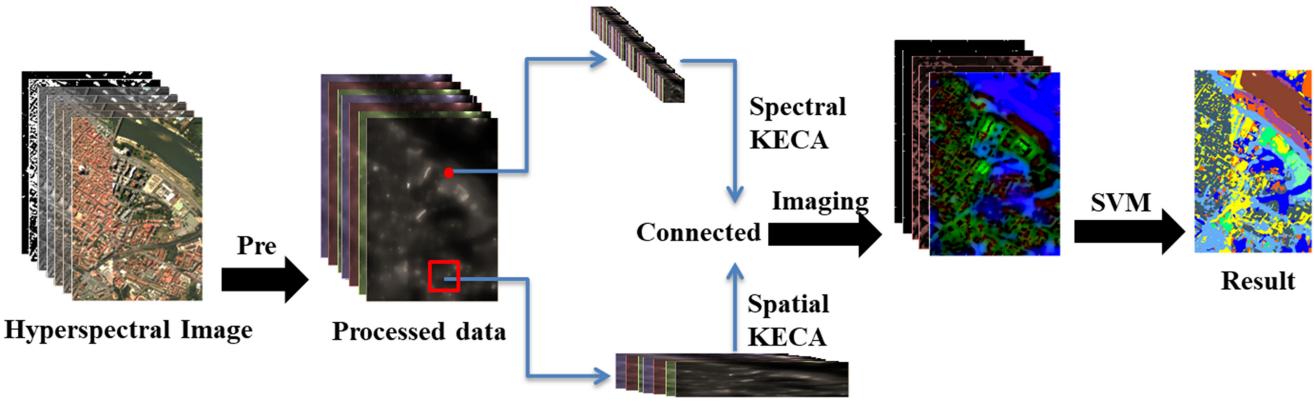

Fig. 1. Three-stage scheme of the KLRECA.

into the RKHS. $\mathbf{1}$ is the $(N \times 1)$ vector of ones. As is well-known, the kernel matrix can be eigendecomposed as follows:

$$\mathbf{K} = \mathbf{A}\mathbf{A}^T = \left(\mathbf{E}\mathbf{D}^{\frac{1}{2}}\right)\left(\mathbf{D}^{\frac{1}{2}}\mathbf{E}^T\right) \tag{4}$$

where the diagonal matrix $\mathbf{D}$ and the matrix $\mathbf{E}$ containing the eigenvalues $\lambda_1, \ldots, \lambda_N$ and the corresponding eigenvectors $\mathbf{e}_1, \ldots, \mathbf{e}_N$, respectively. Thus, (3) can be rewritten as

$$\hat{V}(p) = \frac{1}{N^2}\sum_{i=1}^{N}\left(\sqrt{\lambda_i}\mathbf{1}^T\mathbf{e}_i\right)^2 \tag{5}$$

where only the eigenvector $\mathbf{e}_i$ corresponding to $\lambda_i \neq 0$ and $\mathbf{1}^T\mathbf{e}_i \neq 0$ contributes to the entropy estimate. Therefore, KECA is an r-dimensional FE method by projecting ϕ onto a subspace spanned by r axis in RKHS contributing most to the Renyi entropy of the original data. This is the most striking difference between KECA and KPCA.

B. Several Basic Concepts Associated With a Positive Semidefinite (PSD) Matrix

Let $\mathbf{K}$ be a kernel matrix ($\mathbf{K} \in \Re^{N\times N}$) with $\mathrm{rank}(\mathbf{K}) = r$, which is a PSD and symmetric kernel function or Mercer kernel [29] in RKHS. Then, it has a unique Cholesky decomposition [31], [32]

$$\mathbf{K} = \mathbf{L}\mathbf{L}^T \tag{6}$$

where $\mathbf{L}$ is a lower triangular matrix. To reduce the complexity to both computation and storage, we can compute the low-rank approximations $\tilde{\mathbf{K}}$ of $\mathbf{K}$

$$\tilde{\mathbf{K}} = \mathbf{K}_{N\times r}\mathbf{K}_{r\times r}^{-1}\mathbf{K}_{N\times r}^T \tag{7}$$

where $\mathbf{K}_{N\times B}$ and $\mathbf{K}_{r\times r}$ are the submatrix of $\mathbf{K}$. If $\mathrm{rank}(\tilde{\mathbf{K}}) < r$, then $\mathbf{K}_{N\times r}\mathbf{K}_{r\times r}^{-1}\mathbf{K}_{N\times r}^T$ is called a Nyström approximation of $\mathbf{K}$ [32]; the approximation error $\mathbf{E} = \mathbf{K} - \tilde{\mathbf{K}}$ is named the Shur complement with respect to $\mathbf{K}$ [32].

As is well-known, a full eigendecomposition, which cannot be ignored in KECA, requires the computational complexity of $O(N^3)$. Additionally, considering the very high dimension of each band of HSI, it is imperative to develop an efficient algorithm to find the low-rank approximation of $\mathbf{K}$ so that we can explore more potential of KECA for the HSI classification.

III. Proposed Methodology for HSI Classification

Fig. 1 presents the whole schematic of our three-stage method. Specifically, first, implement the preprocessing techniques on the original hyperspectral imagery. Then, KECA is utilized to obtain the optimized kernel entropic components (KECs, i.e., the pivot elements in this article) to extract the joint spectral-spatial features of the processed data and highlight the discriminative information. Finally, we use SVM to perform the classification of the KECs.

A. Spectral-Spatial FE

Before implementing KECA on the HSI of our interest, it is useful to apply several image preprocessing methods to reduce the noise of the original imagery $\mathbf{I}$ ($\mathbf{I} \in \Re^{n_1\times n_2\times n_3}$) and perform the data transformation

$$\hat{\mathbf{I}} = \mathrm{Pre}(\mathbf{I}, \delta),\ \hat{\mathbf{I}} \in \Re^{n_1\times n_2\times m_3} \tag{8}$$

where $\mathrm{Pre}(\cdot)$ refers to the preprocessing filtering operation with the parameter setting δ. There are many commonly-used such techniques in the remote sensing community. However, only the adopted EPFs or IAPs neither deteriorate the performance nor require a high computational complexity. Then, motivated by Pan *et al*. [33], the spectral-spatial features of the HSI $\mathbf{I}$ can be extracted and combined as follows:

$$x_i \overset{\text{reshape}}{\rightarrow} \mathbf{X}_i^{\text{Spec}} \in \Re^{k_1\times k_2} \tag{9}$$

where $x_i \in \Re^{m_3}$ represents the spectrum vector of a pixel, $\mathbf{X}_i^{\text{Spec}}$ is the imaging form of x_i, and $k_1 \times k_2 = m_3$. The spatial features can be extracted from an image constructed by each pixel x_i and its neighbors

$$[x_{i-s}, \ldots, x_i, \ldots, x_{i+s}] \overset{\text{reshape}}{\rightarrow} \mathbf{X}_i^{\text{Spat}} \in \Re^{s^2\times m_3} \tag{10}$$

where s is the patch size for a pixel. In this article, s is determined from a set consisting of $\{1, 3, 5, 7\}$ through tenfold cross validation with due care. $s = 3$ is appropriate to describe the spatial structure. After extracting the spectral and spatial features,

we directly concatenate them as the final feature expression $\mathbf{X}_i = [\mathbf{X}_i^{\mathrm{Spec}}; \mathbf{X}_i^{\mathrm{Spat}}]$ for each pixel. The rationale to combine the spectral and spatial features together is to capture the multiscale structural information contained in the preprocessed data. Hence, the joint spatial-spectral features can preserve as much information derived from the original HSI as possible, which plays a role in improving the classification accuracy.

B. Feature Fusion With KECA

To prevent from being trapped in the Hughes phenomenon, an appropriate tool should be introduced to implement the feature fusion on the $\mathbf{X}_i$. As discussed in Section II, KECA not only can reduce the dimension of the data but also increase the spectral separability of objects with extracting more nonlinear information from HSIs. However, it is not efficient to directly employ the classical KECA into the HSI classification because of its very high computational complexities. Therefore, this section seeks to propose a much more efficient version of KECA to fuse the $\mathbf{X}_i$ on the support of the pivoted Cholesky decomposition.

For readers' easy understanding, we first give a mathematical analysis of the key to the new algorithm.

Let $\mathbf{K}$ be a kernel matrix ($\mathbf{K} \in \Re^{N_1 \times N_1}$) derived from $\mathbf{X}_i \in \Re^{N_1 \times N_2}$, a Nyström approximation of $\mathbf{K}$, the approximation error $\mathbf{E} = \mathbf{K} - \tilde{\mathbf{K}}$, where $N_1 = n_1 \times n_2$ and $N_2 = n_3 \times M$. If $\mathrm{trace}(\mathbf{E}) \le \varepsilon$ (ε is a minimum value), then $\|\mathbf{E}\|_2 \le \varepsilon$, i.e., $\tilde{\mathbf{K}}$ can be the simultaneous approximation of $\mathbf{K}$ [31], [32].

According to Proposal 1, we should propose an algorithm to decrease the maximum diagonal of $\mathbf{E}$ for minimizing $\mathrm{trace}(\mathbf{E})$, which will also minimize the upper bounds of the approximation error or $\|\mathbf{E}\|_2$. Therefore, we plan to adopt the pivoting Cholesky factorization [31], [32] to rapidly find the approximation of the kernel matrix $\mathbf{K}$. The reason why our new method can reduce the computational complexity and memory space is that only the diagonal elements are required in PCF. For simplicity, we refer to the newly proposed algorithm as kernel low-rank decomposition (KLRD), which is presented in Algorithm 1 in detail.

Theorem 1: The above-mentioned procedure can produce a low rank-r approximation $\tilde{\mathbf{K}}$ of $\mathbf{K}$.

Proof: According to (6) and (7), the rank-r approximation in the ith iteration should satisfy

$$\tilde{\mathbf{K}}^i = \mathbf{K}_{N\times i}\left(\mathbf{K}_{i\times i}\right)^{-1}\left(\mathbf{K}_{N\times i}\right)^T \tag{11}$$

where $\mathbf{K}_{i\times i}$ is also a PSD matrix [31], [32]. Thus, we can obtain the Cholesky factorization of $\mathbf{K}_{i\times i}$

$$\mathbf{K}_{i\times i} = \mathbf{R}_i^T\mathbf{R}_i. \tag{12}$$

Based on (11) and (12), we can compute $\mathbf{L}_i$ in Algorithm 1 as $\mathbf{L}_i = \mathbf{K}_{N\times i}\mathbf{R}_i^{-T}$. Additionally, motivated by the pivoted Cholesky decomposition [31], [32]

$$\mathbf{K}_{(i+1)\times(i+1)} = \begin{bmatrix} \mathbf{K}_{i\times i} & \mathbf{K}_{i\times 1}^t \\ (\mathbf{K}_{i\times 1}^t)^T & \mathbf{K}_{tt} \end{bmatrix} = \mathbf{R}_{i+1}^T\mathbf{R}_{i+1} \tag{13}$$

where $\mathbf{K}_{i\times 1}^t$ denotes the elements $\mathbf{K}_{jt}(j \in [1,\ i])$ of $\mathbf{L}_i$. Moreover, both $\mathbf{R}_{i+1}$ and $\mathbf{R}_i$ should be the lower triangular matrix.

Algorithm 1: KLRD.

Input: $\mathbf{K} \in \Re^{N_1\times N_1}$ with $\mathbf{K}_{ij} = k_\sigma(\mathbf{x}_i, \mathbf{x}_j)$, the width parameter σ, the error tolerance ε and the number r of the pivot elements being evaluated. $\mathbf{N}_1 = 1{:}N_1$. The iteration i starts with 1.

1: Initialization. Let $\mathbf{diag}_1 = \left[\mathbf{K}_{11},\ldots,\mathbf{K}_{N_1N_1}\right]^T$, $e_i = \left\|\mathbf{diag}_1\right\|_1$;

2: **While** $i = 1 \le r$ and $e_i \le \varepsilon$ **do**

3: set $t = \arg\max_{j\in\mathbf{N}_i}\mathbf{diag}_i^j$, $\mathbf{N}_{i+1} = \mathbf{N}_i \setminus t$, where $\mathbf{diag}_i^j$ represents the j the element of $\mathbf{diag}_i$;

4: $\mathbf{K}_{N_1\times 1}^t = \left[\mathbf{K}_{1t},\ldots,\mathbf{K}_{N_1t}\right]^T$, $\mathbf{V}_1 = \mathbf{K}_{N_1\times 1}^t / \sqrt{\mathbf{K}_{tt}}$;

5: $\mathbf{L}_{i+1} = \left[\mathbf{L}_i \quad \Gamma\right]$ with $\Gamma = \mu^{-1}\left(\mathbf{K}_{N_1\times 1}^t - \mathbf{L}_i\mathbf{u}\right)$, $\mathbf{u} = \left(\mathbf{L}_i^t\right)^T$ and

6: $\mu = \sqrt{\mathbf{K}_{tt} - \mathbf{u}^T\mathbf{u}}$, where $\mathbf{L}_i^t$ denotes the t th row of $\mathbf{L}_i$;

7: $\mathbf{diag}_{i+1}^j = \mathbf{diag}_i^j - \Gamma_j^2$, $j \in \mathbf{N}_{i+1}$, where Γ_j is the j th ele-

8: ment of Γ;

9: $e_{i+1} = sum\left(\mathbf{diag}_{i+1}\right)$, $i = i+1$;

10: **end while**

Output: the rank-r approximation $\tilde{\mathbf{K}} = \mathbf{L}^T\mathbf{L}$.

Then we can assume $\mathbf{L}_{i+1}$ satisfies

$$\mathbf{R}_{i+1} = \begin{bmatrix} \mathbf{R}_i & \mathbf{0} \\ \mathbf{u}^T & \mu \end{bmatrix}. \tag{14}$$

Substituting (12) and (14) into (13), we can obtain Steps 5 and 6 in Algorithm 1

$$\Gamma = \mu^{-1}\left(\mathbf{K}_{N_1\times 1}^t - \mathbf{L}_i\mathbf{u}\right), \mathbf{u} = \left(\mathbf{L}_i^t\right)^T, \mu = \sqrt{\mathbf{K}_{tt} - \mathbf{u}^T\mathbf{u}}. \tag{15}$$

Moreover, since $\mathbf{u}^T = \mathbf{L}_i^t$ is just the tth row of $\mathbf{L}_i$ and

$$\left(\mathbf{R}_{i+1}\right)^{-T} = \begin{bmatrix} \left(\mathbf{R}_i\right)^{-T} & -\mu^{-1}\left(\mathbf{R}_i\right)^{-T}\mathbf{u} \\ \mathbf{0} & \mu^{-1} \end{bmatrix} \tag{16}$$

by $\mathbf{L}_i = \mathbf{K}_{N\times i}\mathbf{R}_i^{-T}$ and $\mathbf{L}_{i+1} = \mathbf{K}_{N\times(i+1)}\mathbf{R}_{i+1}^{-T}$, we can derive that Step 5 of Algorithm 1

$$\mathbf{L}_{i+1} = [\mathbf{L}_i \quad \Gamma]. \tag{17}$$

Thus, Step 7 of the Algorithm

$$\mathbf{diag}_{i+1}^j = \mathbf{diag}_i^j - \Gamma_j^2 \tag{18}$$

can be also obtained. This step means that $\mathrm{trace}(\mathbf{E}_i)$ is monotonically decreasing until the Schur complement $\mathbf{E}_i$ vanishes or i meets r. ■

According to Algorithm 1, performing the steps of the pivoted Cholesky decomposition only requires the complexity of $O(r^2N)$ compared with the conventional one $O(N^3)$, which shows the proposed method offers significant savings of computation and storage especially when the size of samples is great.

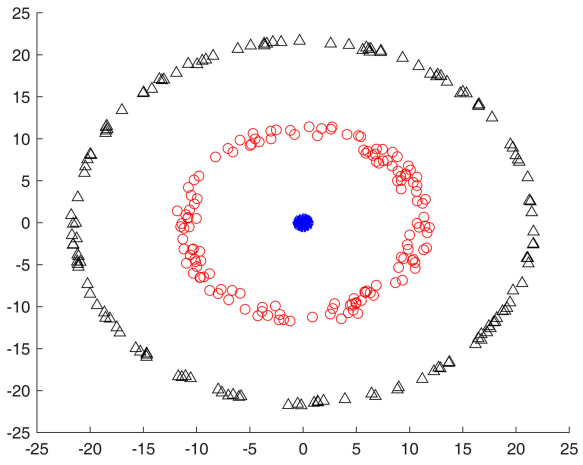

Fig. 2. Toy dataset.

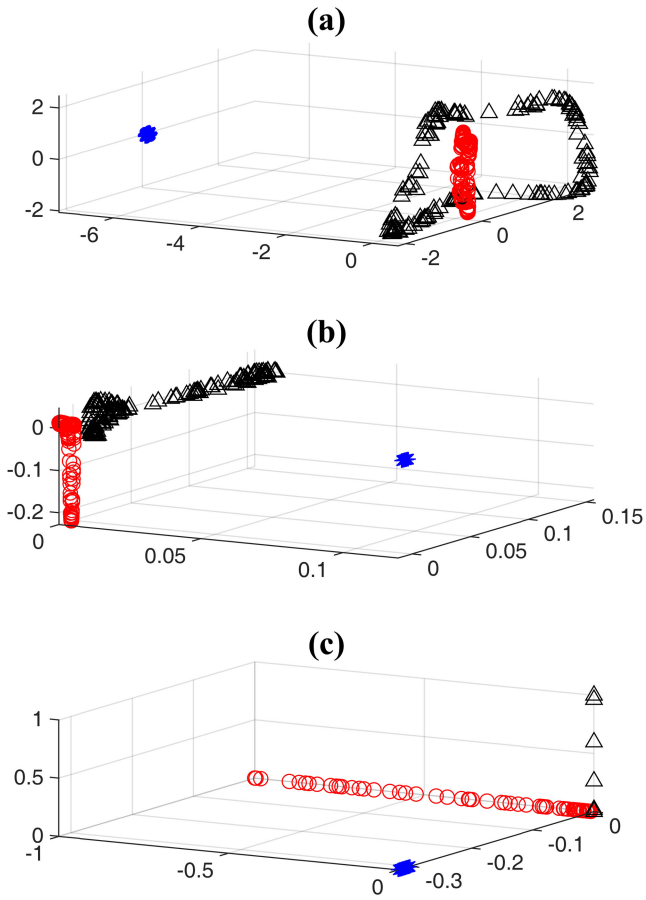

Fig. 3. Feature space analysis of (a) KPCA, (b) KECA, and (c) KLRECA, respectively.

C. Feature Space Analysis

In this section, we provide the feature space analysis to compare the visual class separation capability by the extracted features of the new method with the conventional KECA and KPCA. Fig. 2 shows a toy dataset, derived from [29], with three clusters, ordered from the innermost ring to outermost one with 70 140 and 140 samples, respectively. The dataset transformed by KPCA, KECA, and KLRECA is presented in Fig. 3(a)–(c), respectively. It can be obviously seen that Fig. 2(b) exhibits a clear angular structure. The innermost ring is mapped to a dense representation. The two other rings are separated by 90° angles. This distinct data structure makes KECA more effective for the classification. Fig. 3(c) shows KLRECA retains such superior property and yields the same angular structure. Moreover, the running time of KLRECA is 0.03 s compared with the one of KECA 0.40 s. On the other hand, the dataset produced by KPCA cannot be observed with the angular structure. The mapping of the outermost and middle rings is not very separated [see Fig. 3(a)]. Therefore, we can safely make a conclusion that KLRECA and the conventional KECA can produce strikingly transformed dataset with an angle-based structure compared with KPCA. Additionally, KLRECA is much more efficient than KECA when performing FE.

D. HSI Classification

Let $\mathbf{X}$ be an HSI being tested and the corresponding rank-r kernel matrix be $\tilde{\mathbf{K}} = \tilde{\mathbf{E}}\tilde{\mathbf{D}}\tilde{\mathbf{E}}^T$. Then, the projected image can be computed by $\tilde{\mathbf{X}} = \tilde{\mathbf{K}}\tilde{\mathbf{E}}$. Let $\tilde{\mathbf{x}}_u^i$ and $\tilde{\mathbf{x}}_l^i$ represent the r-dimensional representations of each testing pixel and each training one $\mathbf{x}_l^j$. Let $\mathbf{x}_u^*$ be an arbitrary one to be tested. If we have

$$\left\|\tilde{\mathbf{x}}_u^* - \tilde{\mathbf{x}}_l^j\right\| = \min_{h=1,\ldots,N} \left\|\tilde{\mathbf{x}}_u^* - \tilde{\mathbf{x}}_l^h\right\|_2 \tag{19}$$

then $\mathbf{x}_u^*$ and the jth pixel belong to the same class. In the subsequent experiments, we employ the SVM classifier so as to extract more nonlinear information from the testing hyperspectral data.

IV. Experiments

A. Data Description and Preprocessing Filtering Operation

1) Indian Pines Airborne Visible/Infrared Imaging Spectrometer (AVIRIS): The first HSI is recorded by the AVIRIS sensor over the Indian Pines in Northwestern Indiana, USA. This image comprises 200 spectral bands in the wavelength ranging from 0.4 to 2.5 μm. The spectral resolution and the size of this scene is 20 m and 145 $\times$ 145 pixels, respectively. The number of training samples is set to be 1% for each class. It should be noted that the size of the training data is relatively very small. A three-band color composite image and the ground truth of this AVIRIS hyperspectral data are shown in Fig. 4, from which it can be seen that the image is composed of 16 different classes. We selected the same filters with PCA-EPF as the preprocessing filtering operation $\mathrm{Pre}(\cdot)$ of KLRECA here.

2) Salinas AVIRIS: The second image is also collected by the AVIRIS sensor over the agricultural area of Salinas Valley, CA, USA. After discarding 20 water absorption bands, this image contains 204 bands of size 512 $\times$ 217. A small training dataset (only ten samples per class) is randomly selected from the reference data. Fig. 5 presents the three-band false color image and the corresponding ground truth image. It can be observed that the image includes the same number of classes with the first HSI. We also selected the same filters with PCA-EPF as the preprocessing filtering operation $\mathrm{Pre}(\cdot)$ of KLRECA here.

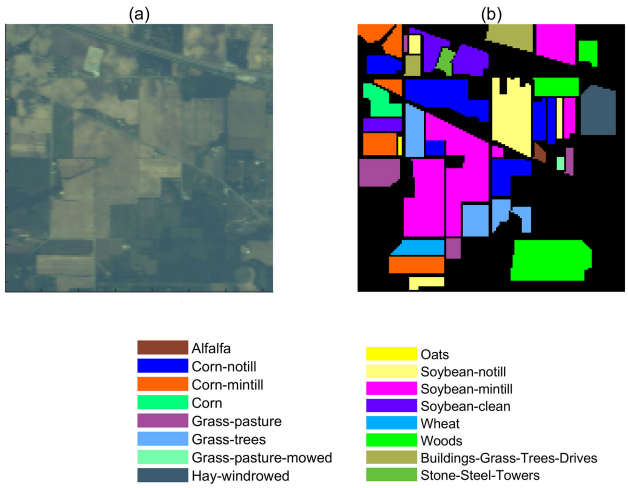

Fig. 4. Indian Pines dataset. (a) False-composite image with R-G-B = bands 35-16-9. (b) Ground truth map. Each color corresponds to a certain class.

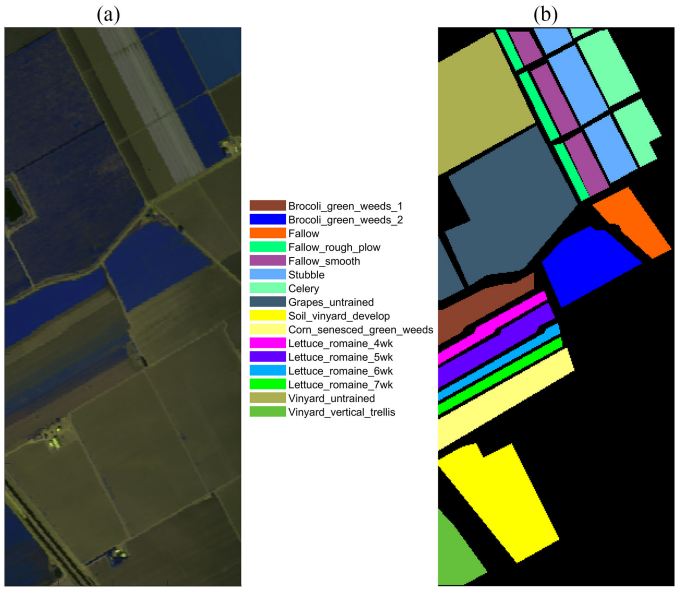

Fig. 5. Salinas dataset. (a) False-composite image with R-G-B = bands 35-22-78. (b) Ground truth map. Each color corresponds to a certain class.

3) Houston2013 ITRES CASI-1500: The third HSI was acquired by the ITRES CASI-1500 sensor during a flight campaign over the campus of the University of Houston and its surrounding areas. The number of spectral bands and the size of the HSI is 144 and 349 $\times$ 1046, respectively. The geometric resolution is 1.3 m. There are mostly different labeled pixels available in the ground truth, including 15 different classes. We artificially chose a small set of challenging labeled samples from the Houston2013 dataset as the training set (about 10% samples per class) and the remaining samples are then utilized for the test, as suggested in [24] and [37]. A false color composite image and the associated ground truth image are shown in Fig. 6(a) and (b), respectively. We selected the same filters with IAPs as the preprocessing filtering operation $\mathrm{Pre}(\cdot)$ of KLRECA here.

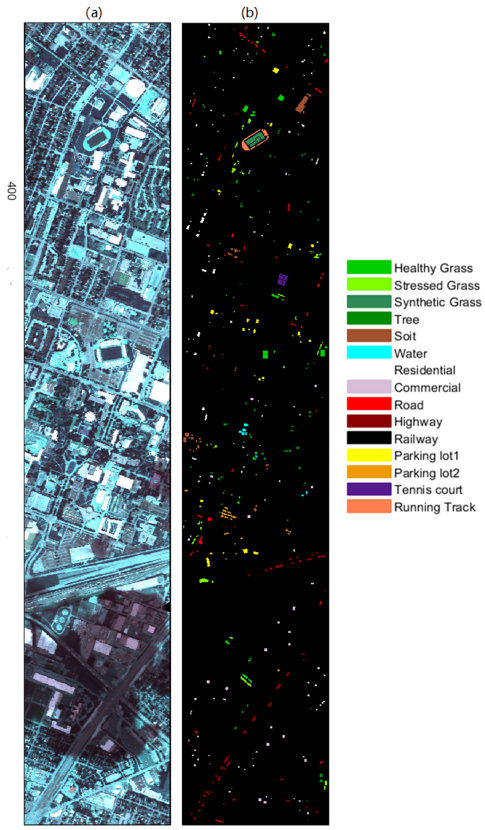

Fig. 6. Houston2013 dataset. (a) False-composite image with R-G-B = bands 15-63-52. (b) Ground truth map. Each color corresponds to a certain class.

B. Experimental Setup

1) Comparison in Related Works: We compare the performance of the KLRECA with the multiple HSI classification methods, including SVM [35], KPCA [26], EMPs [18], NSSNet [33], RVCANet [34], PCA-EPFs [22], SSFPCA [21], and IAPs [24]. KPCA is a classical kernel-based method. Since SVM plays a role as the classifier in our new scheme, thus the original SVM should be employed to evaluate the combination of KECA and SVM. EMPs, PCA-EPFs, SSFPCA, and IAPs are all PCA-associated methods used in extracting the spatial-spectral information from HSIs. NSSNet and RVCANet are recently developed deep learning algorithms for HSI classification and have shown promising performance since they aim at extracting the nonlinear features of remote sensing imagery as well.

2) Algorithm Configuration: The SVM algorithm is performed using the LIBSVM library [36] equipped with the radial basis function kernel. The best value C and gamma parameters γ are obtained through the tenfold cross validation suggested in [21] as follows:

1) *Indian Pines*: $C = 10, \gamma = 0.3$;
2) *Salinas*: $C = 10, \gamma = 0.125$;
3) *Houston2013*: $C = 10\,000, \gamma = 1$.

The MATLAB source code for performing PCA-EPFs and IAPs is available at Dr. Kang's and Dr. Hong's home page,[1,2],

[1][Online]. Available: http://xudongkang.weebly.com/
[2][Online]. Available: https://sites.google.com/view/danfeng-hong/data-code

respectively. The band number of the dimension-reduced data K and the number of the principal components L in PCA-EPFs are set as 15 and 30 suggested in [22]. For IAPs, the number of scaled convolution kernels n_s, the number of convolution kernels in spatially isotropic filtering r, and the number of Fourier orders m are selected as $n_s = 3, r = [2, 4, 6]$, and $m = [0, 1, 2, 3]$, respectively. To do a fair comparison between the new method and the other EPF- and IAP-based method, we analyze the performance of KLRECA using the same parameter setting with PCA-EPFs or IAPs in the phase of hyperspectral data preprocessing, respectively. For SSFPCA, the segmentation of bands of the three different HSIs is set according to the rules proposed in [21] as follows [visible wavelength (VIS); near-infrared wavelength (NIR); short-wavelength infrared (SWIR)]:

1) *Indian Pines*: VIS $= 1 - 36$, NIR $= 37 - 102$, and SWIR $= 103 - 200$;
2) *Salinas*: VIS $= 1 - 36$, NIR $= 37 - 102$, and SWIR $= 103 - 204$;
3) *Houston2013*: VIS $= 1 - 34$, NIR $= 35 - 74$, and SWIR $= 75 - 144$.

NSSNet and RVCANet are performed using the codes, which are available on Dr. Shi's homepage[3]. Both of them consist of two stages, kernels number $M = 8$ and kernel size $n = 7$. The mapping function in NSSNet and the rolling times in RVCANet are set as *Sigmoid* function and 30 as suggested in [33] and [34], respectively.

All of the experiments are conducted by MATLAB R2018a on a PC with Intel Core i7 processors, 8 GB memory, and Windows 10 operating system. It should be noted that the reason why the classical KECA is not considered here is that the PC is prone to be out of memory when we perform the conventional KECA.

3) Evaluation Metrics: In order to address the small sample issue, eight other compared methods are evaluated by the same *very limited* sample size with KLRECA. The reported values of average accuracy (AA), overall accuracy (OA), and Kappa coefficient (Kappa) are computed by the averaged results derived from ten repeated experiments with randomly selected training samples to calibrate all of the aforementioned models. The closer the values of these three performance evaluation criteria are to one, the more reliable is the model performance.

C. Parameter Analysis

1) Kernel Size: As is well-known, the length-scale parameter σ plays a role in the kernel-based methods, including KECA and KLRECA. Therefore, we first implement a sensitivity analysis of different kernel sizes σ. Regarding how to properly estimate σ employed by the Gaussian kernel, a number of rules have been proposed and achieved good results. The first good example is maximizing likelihood function [38], denoted by σ_1, which has been well verified in [26]. Another representative example is the mean distance among input datasets, denoted by σ_2. Shi and Malik [39] suggested utilizing a kernel size in an interval from 10% to 20% of the median distance among the training data. Therefore, 10%, 15%, and 20% from the interval are also selected as other potential choices, denoted by σ_{10}, σ_{15}, and σ_{20}, respectively. The final example is an empirical rule, i.e., Silverman's rule [40], which is the common rule of thumb for KDE, denoted by σ_{Silv}.

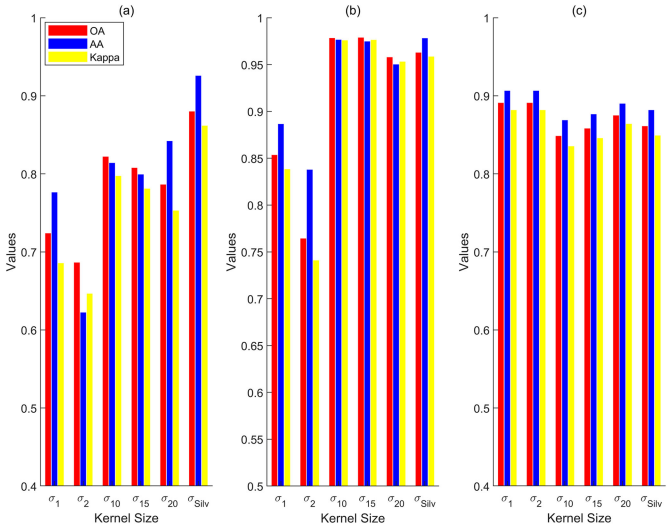

Fig. 7. Influence of kernel sizes on the performance of the KLRECA method. OAs, AAs, and Kappas recorded on (a) Indian Pines, (b) Salinas, and (c) Houston2013 dataset.

The classification performance of KLRECA on the three HSIs using such different kernel sizes are presented in Fig. 7. Under the limited training sample environment, it should be seen from Fig. 5 that KLRECA in coordination with σ_{Silv} can obtain the best OA, AA, and Kappa statistics for the first two HSIs while obtaining the best performances with σ_2 for the Houston2013 dataset. Therefore, σ_{Silv} is selected as the kernel size parameter in the subsequent experiments for the Indian Pines dataset and Salinas dataset, while σ_2 being adopted for the Houston2013 dataset.

2) Number of Pivot Elements: Next, an experiment is implemented to analyze the effects of the other important component, i.e., the number r of pivot elements or KECs, on the classification accuracies of the new method. It can be seen from Fig. 8 that the three performance evaluation criteria generally become more acceptable as r increases except for Houston2013 dataset. Specifically, for the HSIs of Indian Pines and Salinas, the proposed method obtains the best performance when r reaches the maximum. However, for the Houston2013 dataset, the best classification accuracy is achieved when $r = 15$, which means that more pivot elements in consideration cannot lead to more accurate classification results. From this experiment, we can see the number of pivoted elements r can also affect the final classification results.

For extended comparison, KPCA is also employed to implement the same experiment. Fig. 9 shows the classification performances provided by both KLRECA and KPCA. According to Fig. 9, KLRECA can always offer higher classification accuracy with respect to three different evaluation criteria. In addition, KPCA is not prone to obtain better performances with the growing number of input features as well as KLRECA. There

[3][Online]. Available: http://levir.buaa.edu.cn/Code.htm

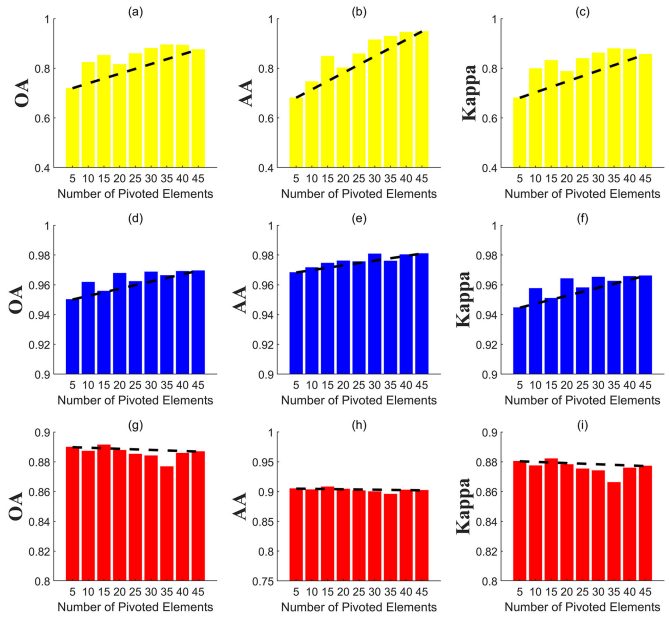

Fig. 8. Influence of the number of pivot elements to the performance of the KLRECA method. (a), (d), and (g) OAs, (b), (e), and (h) AAs, and (c), (f), and (i) Kappas recorded on the Indian Pines (yellow), Salinas (blue), and Houston2013 (red) dataset.

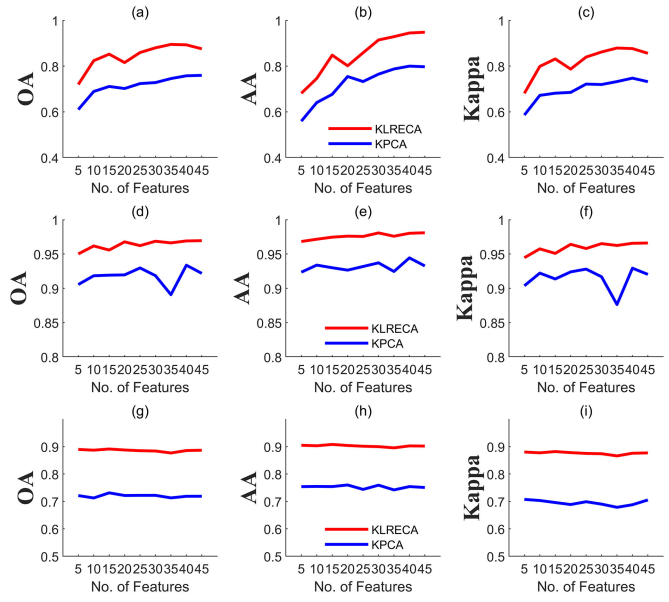

Fig. 9. Influence of the number of extracted features to the performance of KLRECA (red) and KPCA (blue). (a), (d), and (g) OAs, (b), (e), and (h) AAs, and (c), (f), and (i) Kappas recorded on the Indian Pines (the first row), Salinas (the second row), and Houston2013 (the third row) dataset.

seems to be no connection between the number of transformed features and good classification results since the FE step may bring redundant information and noise with much more input features or pivot elements.

D. Comparisons With Other HSI Classification Methods

In this section, we present comparisons of KLRECA against the aforementioned different HSI classification methods. The OA, AA, Kappa, and the classification accuracies for each class, derived from three different HSIs, are given in Tables I–III in the Appendix. Figs. 10–12 (in the Appendix) show the best visual classification result of each algorithm selected from the experiment. According to these results, the following can be seen.

1) Although there are limited pixels per class for training, KLRECA can still achieve more than or close to 90% OA, AA, and Kappa. This may be attributed to FE implemented by KECA optimally reveals the structure related to the most Renyi entropy of the original HSI data and encodes the discriminant information.
2) The results obtained by the proposed KLRECA method is better than the other methods in terms of OA, AA, and Kappa coefficient in most cases.
3) The boundaries and edges in the map of the KLRECA are in better accordance with the ground truth, while SVM, NSSNet, and RVCANet lead to oversmoothed classification maps.
4) Although IAPs obtained the nearly same best statistics for the Houston2013 dataset, KLRECA performs much better than IAPs on the other two real HSIs.
5) NSSNet and RVCANet tend to perform poorly when the size of the training samples is limited.

From the above-mentioned experimental results, we can safely draw a conclusion that KLRECA shows great potential for the HSI classification.

V. Conclusion

This article presents a new scheme for the spectral–spatial classification of hyperspectral imagery. The proposed algorithm utilizes a combination of several advanced preprocessing techniques and KECA. Specifically, our new algorithm consists of three stages, i.e., spectral-spatial FE, the KECA-based feature fusion, and the classification using the standard SVM. In particular, the rationale to use KECA is that it can extract more nonlinear and valuable information in HSIs. Moreover, we employ the theorem of the pivoted Cholesky decomposition to achieve a great improvement of computational efficiency and memory reduction for the conventional KECA. All of the experimental results show that, compared with the classical and state-of-the-art HSI classification methods, the newly proposed KLRECA scheme is of the ability to much efficiently obtain higher classification accuracies on the three real HSIs under a relatively small-sample environment.

In our future studies, we will further evaluate KLRECA using other real remote sensing data (e.g., multitemporal datasets). Additionally, another main focus is how to automatically determine the optimal number of the pivoted elements. Furthermore, it is interesting to explore the potential of the optimized KECA in deep learning algorithms for the HSI classification.

APPENDIX

TABLE I
CLASSIFICATION PERFORMANCE OF THE DIFFERENT METHODS FOR THE INDIAN PINES DATASET WITH 1% LABELED SAMPLES PER CLASS AS THE TRAINING SET

Class	Classification Results of Different methods								
	SVM	KPCA	EMP	NSSNet	RVCAnet	PCA-EPFs	SSFPCA	IAPs	KLRECA
Alfalfa	31.94(15.03)	86.51(13.04)	49.14(32.74)	0.00(0.00)	0.00(0.00)	**100.00(0.00)**	**100.00(0.00)**	90.49(4.67)	**100.00(0.00)**
Corn-notill	48.66(7.70)	65.44(14.21)	65.64(10.07)	65.52(11.09)	69.62(6.10)	74.84(10.10)	67.23(10.36)	64.80(8.04)	**79.85(9.16)**
Corn-mintill	44.76(14.04)	67.48(10.41)	54.99(9.02)	62.44(11.09)	70.35(3.68)	74.84(11.19)	**94.21(3.21)**	35.05(10.85)	84.47(9.92)
Corn	28.01(6.39)	49.92(14.16)	58.86(20.42)	36.60(14.68)	35.66(8.30)	69.62(12.55)	63.53(13.43)	67.62(12.43)	**89.71(10.15)**
Grass-pasture	65.59(16.68)	82.64(15.97)	80.80(14.08)	80.02(8.67)	73.60(14.43)	91.65(9.01)	**94.98(7.93)**	75.33(7.75)	94.88(8.09)
Grass-trees	79.09(4.16)	90.01(5.89)	86.91(8.23)	97.36(4.77)	95.99 (3.22)	95.22(3.76))	93.31(5.74)	**97.94(1.07)**	92.86(5.67)
Grass-pasture-mowed	30.63(14.85)	39.84(22.25)	49.08(30.24)	0.00(0.00)	0.00(0.00)	67.47(31.54)	**100.00(0.00)**	99.54(1.44)	73.81(25.56)
Hay-windrowed	93.98(5.09)	99.77(0.72)	**100.00(0.00)**	99.92(0.15)	**100.00(0.00)**	**100.00(0.00)**	**100.00(0.00)**	96.76(2.59)	**100.00(0.00)**
Oats	12.64(7.83)	74.63(32.90)	32.56(20.88)	0.00(0.00)	0.00(0.00)	65.44(29.97)	82.35(13.45)	90.71(10.13)	**95.16(7.86)**
Soybean-notill	49.77(9.00)	64.41(12.52)	59.73(16.89)	71.49(6.03)	70.79(5.24)	64.90(16.51)	**96.11(2.37)**	71.71(9.44)	88.70(7.22)
Soybean-mintill	67.12(5.38)	86.44(4.93)	81.02(10.07)	86.33(4.01)	82.28(4.78(	90.37(4.51)	**90.98(9.98)**	57.65(11.54)	86.01(6.71)
Soybean-clean	29.93(6.25)	65.75(9.92)	69.23(12.19)	59.35(12.89)	62.49(12.28)	70.73(11.74)	54.18(14.32)	61.22(16.18)	**80.91(10.27)**
Wheat	79.63(3.71)	94.55(5.37)	71.23(19.61)	94.83(4.61)	82.76(14.62)	99.90(0.21)	**100.00(0.00)**	98.44(1.78)	**100.00(0.00)**
Woods	88.51(3.46)	**98.92(0.71)**	96.81(2.65)	97.67(2.35)	97.59 (3.19)	98.79(1.99)	98.66(0.82)	88.70(4.54)	97.65(2.44)
Buildings-Grass-Trees-Drives	31.51(9.09)	85.97(9.65)	79.67(7.68)	62.07(20.04)	59.55 (9.50)	93.53(8.62)	**99.65(0.07)**	87.11(5.34)	98.94(2.40)
Stone-Steel-Towers	83.76(17.78)	95.80(5.84)	93.39(13.48)	87.93(22.97)	89.13(10.28)	95.43(6.68)	92.47(5.78)	99.19(0.95)	**99.74(0.55)**
OA (%)	55.97(2.58)	76.36(1.68)	71.82(4.28)	78.43(3.66)	79.03(0.86)	82.25(3.97)	85.13(4.13)	70.48(2.84)	**87.96(1.83)**
AA (%)	54.09(2.61)	78.01(3.26)	70.57(5.95)	62.59(3.83)	61.86(2.20)	84.55(3.37)	83.98(1.81)	80.14(1.43)	**91.42(2.72)**
Kappa×100	50.67(2.73)	73.41(1.90)	68.37(4.87)	75.21(3.11)	75.78(0.94)	79.94(4.40)	83.13(4.60)	66.79(3.02)	**86.22(2.09)**

Numbers in the parenthesis denote the standard variance of the accuracies obtained in repeated experiments.

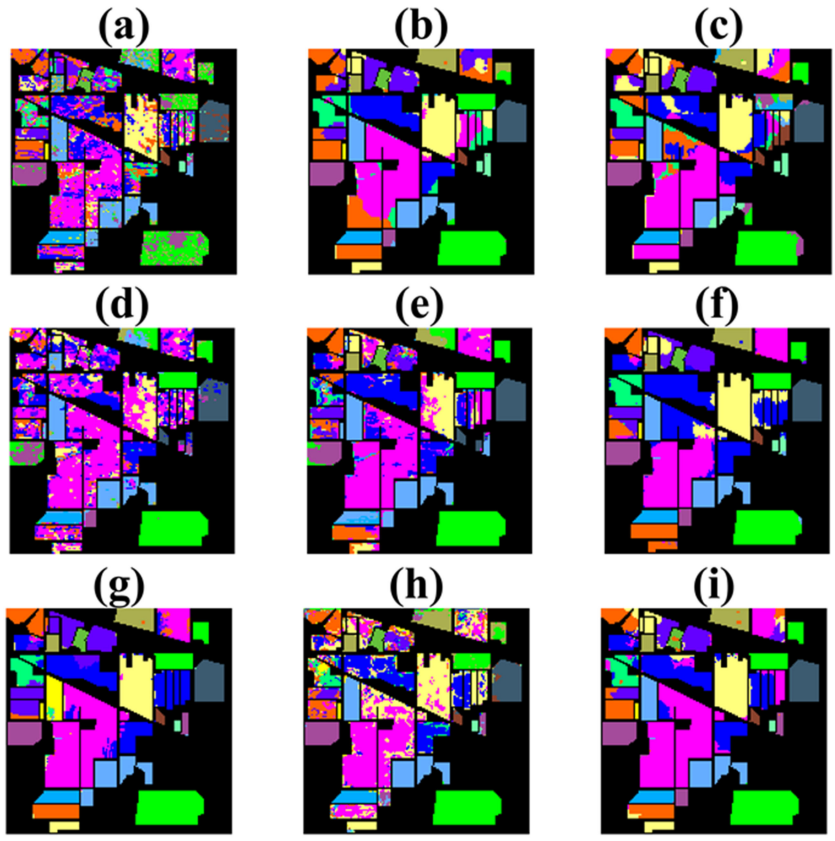

Fig. 10. Classification maps for Indian Pines dataset by (a) SVM, (b) KPCA, (c) EMPs, (d) NSSNet, (e) RVCANet, (f) PCA-EPFs, (g) SSFPCA, (h) IAPs, and (i) KLRECA.

TABLE II
CLASSIFICATION PERFORMANCE OF THE KPCA, SVM, IFRF, NSSNET, RVCANET, AND KLRECA FOR THE SALINAS DATASET WITH TEN LABELED SAMPLES PER CLASS AS THE TRAINING SET

Class	Classification Results of Different methods								
	SVM	KPCA	EMPs	NSSNet	RVCANet	PCA-EPFs	SSFPCA	IAPs	KLRECA
Brocoli_green_weeds_1	97.70(4.67)	80.13(7.34)	96.43(4.55)	**100.00(0.00)**	99.76(0.24)	99.89(0.36)	**100.00(0.00)**	99.13(0.79)	99.87(0.38)
Brocoli_green_weeds_2	98.02(1.31)	95.00(2.43)	**99.73(0.86)**	99.31(0.22)	99.49(0.17)	99.65(0.52)	99.45(0.42)	98.35(1.54)	98.84(1.92)
Fallow	82.24(4.56)	97.46(1.57)	**99.66(0.34)**	93.68（6.33）	94.93(4.47)	99.43(0.94)	99.85(0.09)	96.82(3.88)	97.48(1.36)
Fallow_rough_plow	96.75(0.96)	93.18(3.72)	86.18(7.12)	99.50（0.20）	**99.68(0.15)**	95.06(2.16)	74.84(7.13)	98.48(0.94)	96.11(2.28)
Fallow_smooth	95.83(3.64)	94.57(6.82)	99.39(1.79)	97.90（1.64）	98.98(0.33)	99.97(0.07)	98.98(0.31)	94.73(3.26)	**99.99(0.02)**
Stubble	99.98(0.04)	99.71(0.40)	**100.00(0.00)**	99.80（0,34）	99.80(0.30)	99.28(0.89)	**100.00(0.00)**	99.30(1.00)	97.27(3.13)
Celery	97.10(1.55)	98.49(1.17)	99.39(0.71)	99.69（0.19）	99.57(0.21)	99.42(1.43)	97.04(2.13)	99.80(0.15)	**99.95(0.02)**
Grapes_untrained	68.72(3.67)	94.78(3.67)	96.31(6.91)	60.10(10.22）	69.24(6.17)	95.60(3.65)	**100.00(0.00)**	80.28(7.14)	96.58(3.46)
Soil_vinyard_develop	98.80(0.66)	99.96(0.05)	99.95(0.14)	98.30（0.33）	98.04(0.39)	99.58（0.49）	**100.00(0.00)**	99.16(0.67)	**100.00(0.02)**
Corn_senesced_green_weeds	80.72(5.09)	99.35(0.59)	99.28(0.72)	85.34（6.42）	84.38(7.62)	99.66（0.35）	99.91(0.03)	90.50(3.06)	**99.94(0.09)**
Lettuce_romaine_4wk	79.80(5.75)	98.96(1.09)	98.77(1.62)	98.19（1.47）	98.28(1.12)	99.73（0.50）	**100.00(0.00)**	98.54(0.72)	99.88(0.21)
Lettuce_romaine_5wk	92.17(6.76)	98.23(1.44)	97.79(2.57)	**100.00(0.00)**	**100.00(0.00)**	99.53（0.83）	98.92(0.98)	98.25(1.07)	98.89(2.04)
Lettuce_romaine_6wk	81.14(9.61)	94.67(9.92)	91.86(13.20)	99.42(0.76)	99.71(0.31)	98.80（2.96）	**100.00(0.00)**	97.21(1.03)	98.03(3.55)
Lettuce_romaine_7wk	78.06(17.64)	92.76(9.30)	95.10(5.47)	94.94(1.03)	95.17(2.06)	94.04（3.51）	90.77(4.45)	**96.57(2.12)**	95.36(3.90)
Vinyard_untrained	49.14(4.53)	77.12(6.64)	87.44(7.47)	72.67(8.62)	68.48(7.76)	86.07（9.02）	83.07(5.54)	78.31(8.58)	**87.46(6.23)**
Vinyard_vertical_trellis	95.98(2.69)	99.30(2.22)	99.99(0.04)	94.52(4.37)	94.32(3.87)	99.99（0.04）	**100.00(0.00)**	98.36(0.92)	97.62(3.43)
OA (%)	81.78(1.37)	92.79(1.19)	96.07(1.79)	86.17(3.13)	87.52(0.97)	96.31（0.98）	95.89(2.69)	91.49(0.66)	**96.61(1.01)**
AA (%)	87.01(1.32)	94.60(1.23)	96.70(1.18)	93.34(1.61)	93.74(0.64)	97.70（0.38）	96.43(1.93)	95.26(0.44)	**97.86(0.48)**
Kappa×100	79.80(1.52)	92.01(1.31)	95.62(2.00)	84.66(3.46)	86..13(1.07)	95.89（1.08）	95.44(3.01)	90.54(0.72)	**96.23(1.11)**

Numbers in the parenthesis denote the standard variance of the accuracies obtained in repeated experiments.

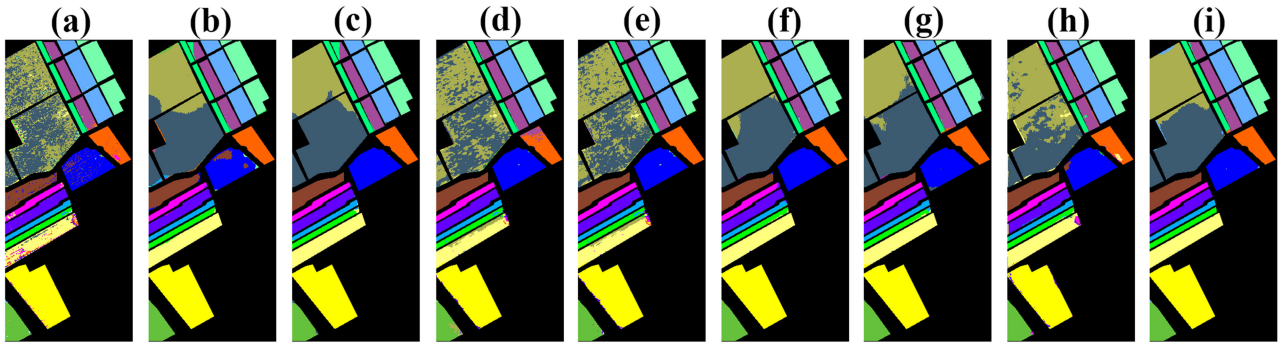

Fig. 11. Classification maps for Salinas dataset by (a) SVM, (b) KPCA, (c) EMPs, (d) NSSNet, (e) RVCANet, (f) PCA-EPFs, (g) SSFPCA, (h) IAPs, and (i) KLRECA.

TABLE III
CLASSIFICATION PERFORMANCE OF THE KPCA, SVM, IFRF, NSSNET, RVCANET, AND KLRECA FOR THE HOUSTON2013 DATASET

Class	Classification Results of Different methods								
	SVM	KPCA	EMPs	NSSNet	RVCANet	PCA-EPFs	SSFPCA	IAPs	KLRECA
Healthy Grass	**97.86**	79.58	93.53	81.29	81.39	88.21	81.67	82.81	83.10
Stressed Grass	**98.52**	76.50	65.44	85.24	84.96	64.52	75.09	85.06	85.06
Synthetic Grass	88.58	97.82	**100.00**	93.47	96.83	**100.00**	97.62	99.60	**100.00**
Tree	**97.22**	81.44	64.23	89.39	89.30	69.17	85.04	91.38	91.67
Soil	95.72	97.92	98.68	98.11	98.20	**100.00**	99.43	**100.00**	**100.00**
Water	10.77	88.81	57.32	98.60	98.60	45.36	**100.00**	98.60	**99.30**
Residential	83.32	82.37	66.18	87.69	87.87	65.29	**90.11**	85.35	85.35
Commercial	69.38	45.30	90.79	79.58	79.68	**93.47**	45.58	74.45	70.18
Road	68.38	67.71	74.21	87.82	**89.61**	63.71	75.73	83.76	82.25
Highway	82.16	44.21	93.60	60.33	59.94	82.20	43.05	94.02	**95.17**
Railway	89.97	47.82	45.16	68.88	69.45	54.04	68.31	90.32	**93.07**
Parking Lot1	81.78	80.79	88.32	76.56	78.00	84.29	88.76	90.39	**93.18**
Parking Lot2	36.47	66.67	71.02	**81.75**	**81.75**	75.91	56.84	76.14	81.05
Tennis Court	90.48	91.50	**100.00**	**100.00**	**100.00**	**100.00**	99.19	99.60	99.60
Running Track	99.57	83.30	**100.00**	**100.00**	**100.00**	**100.00**	97.04	99.79	**100.00**
OA (%)	77.48	72.62	77.19	83.34	83.78	79.74	78.31	88.79	**89.06**
AA (%)	79.34	75.45	80.57	85.91	86.37	79.08	80.23	90.09	**90.60**
Kappa×100	75.76	70.49	76.08	81.91	82.40	78.68	76.09	87.83	**88.12**

The standard variance of the accuracies obtained is 0.00 for all methods because of the fixed training and test set (see [24] and [37] for more details).

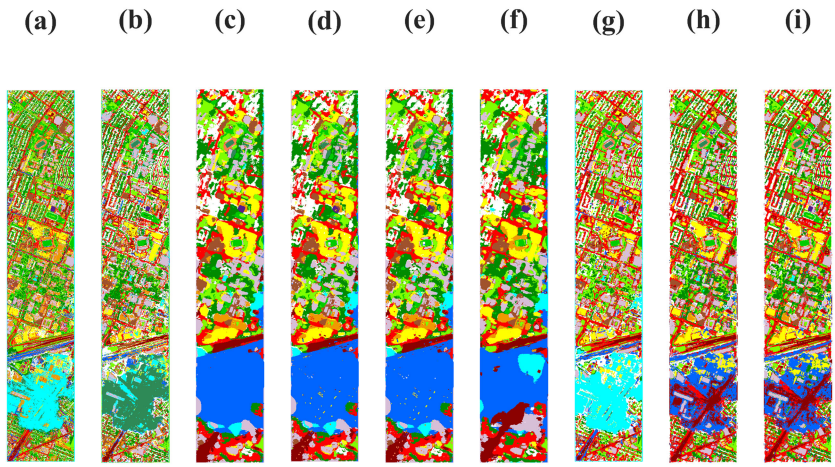

Fig. 12. Classification maps for the Houston2013 dataset by (a) SVM, (b) KPCA, (c) EMPs, (d) NSSNet, (e) RVCANet, (f) PCA-EPFs, (g) SSFPCA, (h) IAPs, and (i) KLRECA.

ACKNOWLEDGMENT

The authors would like to thank Dr. Kang, Dr. Pan, Dr. Zhou, Dr. Hong, and Dr. Uddin for generously sharing the codes of PCA-EPFs, RVCANet, the pivoted Cholesky decomposition, IAPs, and SSFPCA, respectively.

REFERENCES

[1] T. Sankey, J. Donager, J. McVay, and J. Sankey, "UAV lidar and hyperspectral fusion for forest monitoring in the southwestern USA," *Remote Sens. Environ.*, vol. 195, pp. 30–43, 2017.

[2] Z. Zou and Z. Shi, "Hierarchical suppression method for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 330–342, Jan. 2015.

[3] B. B. Damodaran and R. R. Nidamanuri, "Dynamic linear classifier system for hyperspectral image classification for land cover mapping," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2080–2093, Jun. 2014.

[4] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[5] P. Ghamisi, M. D. Mura, and J. A. Benediktsson, "A survey on spectral–spatial classification techniques based on attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2335–2353, May 2014.

[6] E. I. Verdiguier, "Kernel feature extraction methods for remote sensing data analysis," Univ. Valencia, Valencia, Spain, 2014.

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

[8] Y. Bazi and F. Melgani, "Toward an optimal SVM classification system for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3374–3385, Nov. 2006.

[9] D. K. Jain *et al.*, "An approach for hyperspectral image classification by optimizing SVM using self organizing map," *J. Comput. Sci.*, vol. 25, pp. 252–259, 2018.

[10] B. Tu, X. Zhang, X. Kang, G. Zhang, and S. Li, "Density peak-based noisy label detection for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1573–1584, Mar. 2019.

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

[12] J. Xia, J. Chanussot, P. Du, and X. He, "Spectral–spatial classification for hyperspectral data using rotation forests with local feature extraction and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2532–2546, May 2015.

[13] W. Feng, W. Huang, and W. Bao, "Imbalanced hyperspectral image classification with an adaptive ensemble method based on SMOTE and rotation forest with differentiated sampling rates," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1879–1883, Dec. 2019.

[14] J. Xia, P. Ghamisi, N. Yokoya, and A. G. Iwasaki, "Random forest ensembles and extended multiextinction profiles for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 202–216, Jan. 2018.

[15] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification via multiscale adaptive sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7738–7749, Dec. 2014.

[16] X. Jia and J. A. Richards, "Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 1, pp. 538–542, Jan. 1999.

[17] F. Tsai, E. K. Lin, and K. Yoshino, "Spectrally segmented principal component analysis of hyperspectral imagery for mapping invasive plant species," *Int. J. Remote Sens.*, vol. 28, no. 5, pp. 1023–1039, 2007.

[18] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.

[19] J. Zabalza *et al.*, "Novel folded-PCA for improved feature extraction and data reduction with hyperspectral imaging and SAR in remote sensing," *ISPRS J. Photogrammetry Remote Sens.*, vol. 93, pp. 112–122, 2014.

[20] S. Z. Mishu, B. Ahmed, M. A. Hossain, and M. P. Uddin, "Effective subspace detection based on the measurement of both the spectral and spatial information for hyperspectral image classification," *Int. J. Remote Sens.*, vol. 41, no. 19, pp. 7541–7564, 2020.

[21] M. P. Uddin, M. A. Mamun, and M. P. Hossain, "Effective feature extraction through segmentation-based folded-PCA for hyperspectral image classification," *Int. J. Remote Sens.*, vol. 40, no. 18, pp. 7190–7220, 2019.

[22] X. Kang, X. Xiang, S. Li, and J. A. Benediktsson, "PCA-based edge-preserving features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7140–7151, Dec. 2017.

[23] X. Kang, S. Li, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2013.

[24] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3791–3808, Jun. 2020.

[25] M. P. Uddin, M. A. Mamun, and M. P. Hossain, "PCA-based feature reduction for hyperspectral remote sensing image classification," *IETE Tech. Rev.*, to be published, doi: 10.1080/02564602.2020.1740615.

[26] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[27] J. Xia, N. Falco, J. A. Benediktsson, P. Du, and J. Chanussot, "Hyperspectral image classification with rotation random forest via KPCA," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 4, pp. 1601–1609, Apr. 2017.

[28] Z. Zhang and E. R. Hancock, "Kernel entropy-based unsupervised spectral feature selection," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 26, no. 5, 2012, Art. no. 1260002.

[29] R. Jenssen, "Kernel entropy component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 847–860, May 2010.
[30] C. Z. Bai, R. Zhang, Z. Xu, R. Cheng, B. Jin, and J. Chen, "L1-norm-based kernel entropy components," *Pattern Recognit.*, vol. 96, 2019, Art. no. 106990.
[31] H. Harbrecht, M. Peters, and R. Schneider, "On the low-rank approximation by the pivoted Cholesky decomposition," *Appl. Numer. Math.*, vol. 62, no. 4, pp. 428–440, 2012.
[32] S. Zhou, "Sparse LSSVM in primal using Cholesky factorization for large-scale problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 783–795, Apr. 2016.
[33] B. Pan, Z. Shi, N. Zhang, and S. Xie, "Hyperspectral image classification based on nonlinear spectral–spatial network," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1782–1786, Dec. 2016.
[34] B. Pan, Z. Shi, and X. Xu, "R-VCANet: A new deep-learning-based hyperspectral image classification method," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 1975–1986, May 2017.
[35] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.
[36] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, 2011, Art. no. 27.
[37] P. R. Marpu, M. Pedergnana, M. D. Mura, S. Peeters, and J. A. Benediktsson, and L. Bruzzone, "Classification of hyperspectral data using extended attribute profiles based on supervised and unsupervised feature extraction techniques," *Int. J. Image Data Fusion*, vol. 3, no. 3, pp. 269–298, 2012.
[38] R. P. W. Duin, "On the choice of smoothing parameters for parzen estimators of probability density functions," *IEEE Trans. Comput.*, vol. C-25, no. 11, pp. 1175–1179, Nov. 1976.
[39] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[40] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Boston, MA, USA: Chapman & Hall/CRC, 1998, pp. 296–297.

Chengzu Bai received the B.A.Sc. and M.A.Sc. degrees in ocean science from PLA National Defence University, Beijing, China, in 2012 and 2015, respectively and the Ph.D. degree in atmospheric science from the National University of Defense Technology, Changsha, China, in 2018.

His current research interests include data mining, machine learning, and remote sensing imagery processing.

Ren Zhang received the B.A.Sc. degree in meteorology from the Chengdu University of Information Technology, Chengdu, China, and the M.A.Sc. and Ph.D. degrees in meteorology from Nanjing University, Nanjing, China, in 1983, 1990, 2000, respectively.

He has contributed more than 150 journal articles to professional journals. He is currently a member of the advisory boards of *Computers and Industrial Engineering, International Journal of Information Technology & Decision Making, Journal of Atmospheric and Oceanic Technology*, etc.

His current research interests include data mining, machine learning, fuzzy information processing, and climate change.

Zeshui Xu (Fellow, IEEE) received the Ph.D. degree in management science and engineering from Southeast University, Nanjing, China, in 2003.

From October 2005 to December 2007, he was a Postdoctoral Researcher with the School of Economics and Management, Tsinghua University, Beijing, China. He is a Distinguished Young Scholar with the National Natural Science Foundation of China and the Chang Jiang Scholars of the Ministry of Education of China. He is currently a Professor with the Business School, Sichuan University, Chengdu, China. He has contributed more than 500 SCI/SSCI articles to professional journals, is among the world's top 1% most highly cited researchers with about 60 000 citations, and his h-index is 122. He has contributed more than 700 journal articles to professional journals. He is currently an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Information Sciences*, *Artificial Intelligence Review*, *Fuzzy Optimization and Decision Making*, etc., and a member of the Editorial (advisory) Boards of more than 30 professional journals. His current research interests include decision making, information fusion, clustering analysis, and fuzzy sets.

Dr. Xu has been elected as the Academician of IASCYS, IEEE Fellow, IFSA Fellow, IET Fellow, BCS Fellow, RSA Fellow, and ranked 431th among World's Top 10 0000 Scientists, in 2019.

Baogang Jin received the B.A.Sc., M.A.Sc., and Ph.D. degrees in atmosphere science from PLA National Defence University, Beijing, China, in 2004, 2007, 2011, respectively.

His current research interests include remote sensing imagery processing.

Jian Chen received the B.A.Sc., M.A.Sc., and Ph.D. degrees in atmosphere science from PLA National Defence University, Beijing, China, in 2009, 2011, 2014, respectively.

His current research interests include remote sensing imagery processing.

Shuo Zhang received the B.A.Sc. and M.A.Sc. degrees in meteorology from the Chengdu University of Information Technology, Chengdu, China, in 2016 and 2019, respectively.

Her current research interests include machine learning and remote sensing imagery processing.

Longxia Qian received the B.A.Sc. degree in mathematics from Anhui Normal University, Wuhu, China, in 2010, and the M.A.Sc. and Ph.D. degrees in atmospheric science from the National University of Defense Technology, Nanjing, China, in 2015 and 2018, respectively.

Her current research interests include data mining and remote sensing imagery processing.