Pseudolabel Guided Kernel Learning for Hyperspectral Image Classification

Shujun Yang, Junhui Hou, Member, IEEE, Yuheng Jia, Member, IEEE, Shaohui Mei, Member, IEEE, and Qian Du, Fellow, IEEE

Abstract—In this paper, we propose a new framework for hyperspectral image classification, namely pseudolabel guided kernel learning (PLKL). The proposed framework is capable of fully utilizing unlabeled samples, making it very effective to handle the task with extremely limited training samples. Specifically, with multiple initial kernels and labeled samples, we first employ support vector machine (SVM) classifiers to predict pseudolabels independently for each unlabeled sample, and consistency voting is applied to the resulting pseudolabels to select and add a few unlabeled samples to the training set. Then, we refine the kernels to improve their discriminability with the augmented training set and a typical kernel learning method. Such phases are repeated until stable. Furthermore, we enhance the PLKL in terms of both the computation and memory efficiencies by using a bagging-like strategy, improving its practicality for large scale datasets. In addition, the proposed framework is quite flexible and general. That is, other advanced kernel-based methods can be incorporated to continuously improve the performance. Experimental results show that the proposed frameworks achieve much higher classification accuracy compared with state-of-the-art methods. Especially, the classification accuracy improves more than 5% with very few training samples.

Index Terms—Classification, kernel learning, pseudolabel, semisupervised learning.

I. INTRODUCTION

HYPERSONTICAL imaging acquires digital images with hundreds of spectral bands, which often range from the visible to infrared spectrum [36]. The rich spectral information in hyperspectral images (HSIs) brings about breakthroughs in various application fields, such as military [27], [37], agriculture [10], [34], and mineralogy [18]. HSI classification is one of the most important tasks, where each high-dimensional pixel is assigned with label as a certain class. A variety of methods have been developed in HSI classification [6], [13], [21], [28], [29], [44]–[47]. Traditional HSI classification methods assume that the spectral information of samples within the same class should be similar to each other. Thus, learning an accurate similarity metric between samples is vital to the final classification performance.

Kernel methods have achieved great success in HSI classification [3], [30]. The key idea of kernel methods is to transform the data from the original sample space to a high-dimensional feature space where the linear separability is more easily achieved. Considering the insufficient discriminability of a single kernel for classification, representative multiple kernel learning (RMKL) [17], nonlinear multiple kernel learning (NMKL) [16], and the superpixel-based classification via multiple kernels (SC-MK) [12] were also proposed.

However, almost all of the aforementioned kernel methods only use data samples to construct kernels, which overlook the valuable label information in the training set. To make use of the label information for constructing more informative kernels, some kernel learning methods have been proposed, e.g., ideal regularized (IR) kernel [33], kernel metric learning using relative distance constraints [1], simultaneous kernel learning and label imputation with partially labeled data [20], constrained clustering by spectral kernel learning [23]. However, arising from the high cost, it is a nontrivial task to obtain a large amount of label information in many real applications, which limits their practical performance. Thus, how to effectively exploit the potential of unlabeled data samples is a promising direction for improving the classification performance of kernel-based methods for HSIs.

In this paper, we propose a new kernel-based framework for HSI classification, namely pseudolabel guided kernel learning (PLKL), by fully taking advantage of unlabeled samples. Specifically, our framework employs support vector machine (SVM) classifiers with multiple typical kernels to predict pseudolabels independently for each unlabeled sample, and then a few unlabeled samples are selected via a consistency voting to expand the original training set. With the augmented training set, efficient kernel refinement is performed to improve their discriminability. Such phases are repeated until stable. To further improve both the computation and memory efficiencies for dealing with large scale datasets, we enhance PLKL by using a bagging-like strategy. The proposed frameworks are very effective to han-
dle the case with extremely limited training samples. Moreover, our framework is flexible and general, which could be applied to any advanced kernel-based methods. Extensive experiments demonstrate that our proposed frameworks could significantly improve the performance of HSI classification when compared with several state-of-the-art methods. Especially, for the case with very few training samples, our frameworks achieve much more significant improvement on classification accuracy.

The remainder of this paper is organized as follows. In Section II, we introduce the related works about standard kernels for HSIs and kernel-based methods for HSI classification. We then present the proposed PLKL and efficient PLKL in Section III. Experimental results are given in Section IV. Finally, Section V concludes this paper.

II. RELATED WORK

Suppose we have a labeled training set with \( l \) samples \( \mathcal{L} = \{ \{ \mathbf{x}_1, y_1 \}, \ldots, \{ \mathbf{x}_l, y_l \} \} \) and an unlabeled testing set with \( n \) samples \( \mathcal{U} = \{ \mathbf{u}_1, \ldots, \mathbf{u}_n \} \), where \( \mathbf{x}_i \) and \( \mathbf{u}_j \) stand for the vectorial representations of the \( i \)th labeled and \( j \)th unlabeled samples, respectively, and \( y_i \in \{1, 2, \ldots, c\} \) is the label of \( \mathbf{x}_i \) with \( c \) being the number of classes. Let \( K \) be a kernel Gram matrix, a positive semidefinite matrix. The \((i, j)\)th entry of \( K \) is denoted as \( K_{i,j} \) or \( K(\mathbf{x}_i, \mathbf{x}_j) \), which measures the similarity between \( \mathbf{x}_i \) and \( \mathbf{x}_j \).

A. Standard Kernels for HSIs

1) Spectral Kernel: A spectral kernel measures the spectral similarity between sample points, and the spectral radial basis function (RBF) kernel between two samples \( \mathbf{x}_i \) and \( \mathbf{x}_j \) can be expressed as

\[
K_{i,j}^s = \exp \left( -\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_s^2} \right)
\]

where \( \sigma_s \) is the width of the spectral RBF kernel.

2) Spatial Kernel: In order to reduce sample noise, for a pixel \( \mathbf{x}_i \), we could extract a local spatial feature vector \( \mathbf{x}_i^s \) from its neighbourhoods \( \mathcal{N}_r(\mathbf{x}_i) = \{\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_r}\} \), e.g., the mean or standard deviation of its neighbourhoods. Then, the spatial RBF kernel between two samples \( \mathbf{x}_i \) and \( \mathbf{x}_j \) can be expressed as

\[
K_{i,j}^s = \exp \left( -\frac{||\mathbf{x}_i^s - \mathbf{x}_j^s||^2}{2\sigma_s^2} \right)
\]

where \( \sigma_s \) is the width of the spatial RBF kernel. And spatial mean map kernel can be defined as [14], [25]

\[
K_{i,j}^m = \frac{1}{r^2} \sum_{p=1}^{r} \sum_{q=1}^{r} K_{i,p,j,q}^s
\]

\[
= \frac{1}{r^2} \sum_{p=1}^{r} \sum_{q=1}^{r} \exp \left( -\frac{||\mathbf{x}_{i,p} - \mathbf{x}_{i,q}||^2}{2\sigma_m^2} \right).
\]

3) Spectral-Spatial Kernel: The typical composite spectral-spatial kernel [4] is a weighted summation kernel, which is defined as

\[
K_{i,j}^{sw} = (1-u)K_{i,j}^s + uK_{i,j}^m
\]

where \( u \in [0, 1] \) is a parameter to balance the spectral and spatial similarity. Similarly, the composite spectral-spatial mean map kernel can be expressed as

\[
K_{i,j}^{wm} = (1-u)K_{i,j}^s + uK_{i,j}^m.
\]

B. Kernel-Based Methods for HSI Classification

Many kernel-based methods have been proposed in HSI classification owing to its nonlinear nature [19]. One class of kernel-based methods focus on multiple kernel learning and achieve more robust discriminant capability than traditional single kernel learning. Several multiple kernel learning methods were proposed recently. RMKL aims to learn a kernel matrix which could capture the most variation of the original base kernels [17]. NMKL concentrates on the nonlinear combination of multiple kernels to incorporate both spectral and spatial features [16]. SC-MK [12] first utilizes oversgmentation algorithm to cluster an HSI into many superpixels, and then combines three kernels for HSI classification. Meanwhile, several methods focus on inducing the semisupervised learning into kernel learning for HSI classification, such as cluster kernel [42], mean map kernel [14], and graph Laplacian kernel [5], [7], [31], in which sample similarity and limited label information in training data are utilized to build a robust classifier. Through building a kernel-based similarity graph, which concerns sample similarity between both training and testing data, the kernel-based semisupervised learning methods could propagate the limited label information in training data to the testing data.

Besides, some other kernel methods for HSI classification focus on kernel-based representation, such as kernel basic thresholding classifier (KBTC) [39], kernel orthogonal matching pursuit (KOMP) [9], and simultaneous version of kernel orthogonal matching pursuit (KSOMP) [9], which assume that a testing pixel could be linearly represented by training samples. Specifically, KBTC [39] is a nonlinear kernel version of recently introduced basic threading classifier (BTC) [38], where BTC [38] is a sparsity-based linear classifier. BTC [38] utilizes the inner product for similarity measure, which aims to select a subset of atoms from the entire dictionary to approximate the signal. KOMP [9] and KSOMP [9] are different extended kerneled methods related with orthogonal matching pursuit (OMP) [40]. Specifically, KOMP is a kernel version of OMP [40], which aims to find a kernel sparse representation for each unlabeled pixel by assuming that each pixel could be represented by a kernel sparsity model of training samples. KSOMP [9] is a kernel version of the simultaneous orthogonal matching pursuit (SOMP) [41], where both KSOMP [9] and SOMP [41] assume that the neighboring pixels should share a common sparsity model. However, these two joint sparsity-based methods (KSOMP [9] and SOMP [41]) have expensive computational cost as they need to determine neighborhood window experimentally and simultaneously optimize the sparsity model for neighboring pixels.
in the labeled data to refine typical kernels. This strategy was introduced in HSI classification, which aims to improve the discriminability of original typical kernels by adding the label relationship constraints to the kernels [35]. However, arising from the high cost, it is a nontrivial task to obtain a large amount of label information in many real applications, which limits the performance of these methods for the tasks with very few labeled samples.

C. Semisupervised Methods for HSI Classification

Semisupervised learning could exploit the wealthy information in the unlabeled data for improving the performance [8]. Several semisupervised methods have been proposed in HSI classification, which can be roughly classified into three categories: 1) Generative models, such as multinomial logistic regression [22]; 2) low density separation related algorithms, which maximize the margin for labeled and unlabeled samples simultaneously, such as transductive SVM [2]; and 3) graph-based methods, which spread label information to its neighborhoods in the graph until the global stable state is achieved, such as L1-graph based classification [15] and spatio-spectral Laplacian SVM [43].

In addition, using unlabeled samples with high confidence pseudolabels as auxiliary training samples has been exploited. For example, computational baby learning [24] gradually learns a mature concept detector, in which high confidence unlabeled samples are selected as auxiliary training set by using an initial single concept detector. An recurrent neural network (RNN)-based deep learning network optimized by a transfer learning strategy is proposed in [26], which utilizes an initial deep learning network trained in the source domain to help select target samples to refine the network for target domain.

III. PROPOSED METHODS

As aforementioned, using label information to build more informative kernels can improve the HSI classification accuracy. However, the label information is usually limited in practice, arising from the high cost. To this end, by fully exploiting the potential of unlabeled samples, we propose a new kernel-based framework shown in Fig. 1, namely PLKL, which is mainly composed of three phases, i.e., initialization, training set augmentation, and kernel refinement. The proposed PLKL is very effective to handle the task with extremely few training samples. Moreover, we propose both computationally and memory efficient PLKL, i.e., E-PLKL.

A. Initialization

Let \( \mathcal{Z} \) be the training set that is dynamically changing in our framework, and we initialize it with the original training set, i.e., \( \mathcal{Z} = \mathcal{L} \). We also initialize a kernel set including different types of kernels with \( \mathcal{L} \), which is denoted as \( \mathcal{K} := \{K_1, K_2, \ldots, K_T\} \) with \( K_t \) being the \( t \)th kernel. Typically, in this paper, we employ the standard kernels given in Section II-A. Note that other advanced kernels could also be considered as the initial kernels to improve classification performance.

B. Training Set Expansion

In this phase, each kernel is equipped with SVM and thus we can derive a classifier set denoted as \( \mathcal{H} := \{h_1, h_2, \ldots, h_T\} \) with \( h_t \) being the \( t \)th classifier. Then, these classifiers are trained with \( \mathcal{Z} \) and then utilized to jointly predict pseudolabels for the unlabeled samples involved in \( \mathcal{U} \), and those samples with high confident pseudolabels will be selected to expand the training set.

Specifically, for each unlabeled sample \( \mathbf{u}_i \), the predicted pseudolabels by all \( T \) classifiers are denoted as \( \{h_1(\mathbf{u}_i), h_2(\mathbf{u}_i), \ldots, h_T(\mathbf{u}_i)\} \). We only select the unlabeled samples on whose pseudolabels all \( T \) classifiers achieve agreement, producing an auxiliary training set

\[
\mathcal{L}_\Delta = \{(\mathbf{u}_i, h_t(\mathbf{u}_i))| h_1(\mathbf{u}_i) = h_2(\mathbf{u}_i) = \ldots = h_T(\mathbf{u}_i)\}.
\] (6)

Thus, the expanded training set is derived as \( \mathcal{Z} = \{\mathcal{L} \cup \mathcal{L}_\Delta\} \).

C. Kernel Refinement

With the augmented training set \( \mathcal{Z} \), we could refine the corresponding kernels \( \{K_1, K_2, \ldots, K_T\} \) to make them more informative. Typically, we employ the IR kernel learning method [33] to realize our goal, considering its high efficiency owing to the closed-form solution. Notice that other advanced kernel learning methods can also be used in our framework. Specifically, the IR kernel learning method refines a kernel by embedding the known label information into the kernel, which is mathematically modeled as

\[
\min_{K \succeq 0} \text{tr}(K \log K - K \log K_0 - K + K_0) - \gamma(\text{tr}(K\bar{Z}))
\] (7)
Algorithm 1: PLKL: Pseudolabel Guided Kernel Learning for HSI Classification.

Input: $L = \{x_i, y_i\}_{i=1}^L, U = \{u_1, \ldots, u_n\}$.
1: Initialize the training set $\tilde{Z} = L$ and kernel set $K$.
2: while not converged do
3: Construct the classifier set $\hat{K}$ with $K$ and $\tilde{Z}$ and predict pseudolabels for unlabeled samples.
4: Generate the auxiliary training set $L_\Delta$ and expand the original training set $\tilde{Z} = \{L \cup L_\Delta\}$.
5: Refine the kernel set $K$ with $\hat{Z}$.
6: end while
Output: The final predicted label for each unlabeled sample $\{h_t(u_i)\}_{i=1}^U$.

where $\gamma$ is a positive parameter, $\text{tr}(\cdot)$ is the trace of a matrix, and $\text{tr}(K\tilde{Z})$ is a regularization term that incorporates available label relationships captured by matrix $\tilde{Z}$ for enhancement.

Specifically, the following three types of relationships should be considered during kernel refinement. The entries of the refined kernel matrix corresponding to two data samples $x_i$ and $x_j$ from the same class (i.e., $x_i, x_j \in L$ and $y_i = y_j$) should be larger than those corresponding to two samples from different classes (i.e., $y_i \neq y_j$). For each pair of a pseudolabeled sample $\hat{u}_i$ (i.e., $\hat{u}_i \in L_\Delta$) and a labeled sample $x_i$, if the pseudolabel of the unlabeled sample is the same as that of the labeled one (i.e., $h_t(\hat{u}_i) = y_j$), the corresponding entries of the refined kernel matrix should be larger; otherwise, it should be smaller. For each pair of the pseudolabeled samples $\hat{u}_i$ and $\hat{u}_j$ (i.e., $\hat{u}_i \in L_\Delta$ and $\hat{u}_j \in L_\Delta$), if their pseudolabels are the same (i.e., $h_t(\hat{u}_i) = h_t(\hat{u}_j)$), the corresponding entries of the refined kernel matrix should be larger; otherwise, it should be smaller. Accordingly, we define the label relationship $\Delta$ as

$$
\Delta(x_i, x_j) = \begin{cases} 
1, & y_i = y_j \\
0, & y_i \neq y_j 
\end{cases} \quad (8)
$$

$$
\Delta(\hat{u}_i, x_j) = \hat{\Delta}(\hat{u}_i, x_j) = \begin{cases} 
1, & h_t(\hat{u}_i) = y_j \\
0, & h_t(\hat{u}_i) \neq y_j 
\end{cases} \quad (9)
$$

$$
\Delta(\hat{u}_i, \hat{u}_j) = \begin{cases} 
1, & h_t(\hat{u}_i) = h_t(\hat{u}_j) \\
0, & h_t(\hat{u}_i) \neq h_t(\hat{u}_j). 
\end{cases} \quad (10)
$$

The problem in (7) has a closed-form solution, i.e., $K^* = \exp(\log K_0 + \gamma \Delta)$. According to [32], [33], and [35], the kernel value between an unlabeled sample $u_i \in U$ and a labeled or pseudolabeled sample $\tilde{x}_i \in \tilde{Z}$ can be computed as

$$
K(u_i, \tilde{x}_i) = \sum_{p,q=1}^{[\tilde{Z}]} S_{p,q} K_0(u_i, \tilde{x}_p) K_0(\tilde{x}_q, \tilde{x}_i) - K_0(u_i, \tilde{x}_i) \quad (11)
$$

where $S = K_0^{-1}(K_0 + K^*)K_0^{-1}$ with $K_0^{-1}$ being the inverse of $K_0$. The proposed PLKL algorithm is summarized in Algorithm 1.

D. Efficient PLKL (E-PLKL)

In PLKL, we expand the original training set $L$ with an auxiliary training set $L_\Delta$. However, when the size of $L_\Delta$ is very large, the proposed PLKL will be memory and time consuming due to the inverse operation of the Gram matrix in (11), i.e., $S = K_0^{-1}(K_0 + K^*)K_0^{-1}$. Here, we further propose a computationally and memory efficient PLKL (E-PLKL), demonstrating its practicality for handling large scale datasets. As shown in Fig. 2, to reduce the memory and computational costs without degenerating the classification accuracy, we adopt a bagging-like strategy to realize E-PLKL.

Specifically, we use the same methods as PLKL to initialize the framework and generate an auxiliary training set $L_\Delta$. In contrast to PLKL, which takes $L_\Delta$ as a whole and refines each kernel with all samples in $L_\Delta$, we partition $L_\Delta$ into $M$ nonoverlapping subsets which are denoted as $\{L_1, L_2, \ldots, L_M\}$ with $L_m$ being the $m$th subset. Combined with the original training set $L$, $L_m$ could produce an expanded subtraining set $\tilde{Z}_m = \{L \cup L_m\}$. And the union of the expanded subtraining sets is expressed as $\tilde{Z} = \{\tilde{Z}_m \mid m = 1, \ldots, M\}$.

Similar to PLKL, we then refine the kernels with the augmented subtraining set $\tilde{Z}_m$. Different from PLKL that only adopts a single kernel set, E-PLKL employs $M$ kernel sets denoted as $K := \{K_1, K_2, \ldots, K_M\}$, and each kernel set $K_m := \{K_m^1, K_m^2, \ldots, K_m^M\}$ is refined with the augmented subtraining set $\tilde{Z}_m$ via the IR kernel learning method shown in Section III-C. After the refinement process, each kernel of $K_m$ is equipped with an SVM to generate a base classifier, which is trained with $\tilde{Z}_m$ so that we can obtain a base classifier set.
Let $\mathcal{L}$ be the number of samples involved in $\mathcal{L}_\Delta$, and $l = l + |\mathcal{L}_\Delta|$ be the number of samples included in the expanded training set. The computational complexity of each step of PLKL and E-PLKL is summarized in Table I. To sum up, the total computational complexity for PLKL and E-PLKL are $\mathcal{O}(2\tilde{l}^3 + nT + n + n^2)$ and $\mathcal{O}(MT(l + \frac{|\mathcal{L}_\Delta|}{M})^3 + nTMc + n + M(l + \frac{|\mathcal{L}_\Delta|}{M})^3 + Mn(l + \frac{|\mathcal{L}_\Delta|}{M})^2)$, respectively.

2) Memory Consumption Analysis: The memory complexity for generating kernel-based classifiers is mainly dominated by kernel memory complexity. Table II summarizes memory consumption analysis for PLKL and E-PLKL, which shows that E-PLKL could handle a larger $\mathcal{L}_\Delta$.

### IV. EXPERIMENTS

In this section, we conducted experiments on two benchmark HSI datasets\(^1\) to evaluate our frameworks. We also compared them with other state-of-the-art HSI classification methods to demonstrate their advantages.

#### A. Hyperspectral Datasets

1) Indian Pines Dataset: This dataset was acquired by AVIRIS sensor over northern western Indian Pines test site. The image scene contains 145 × 145 pixels and 224 spectral bands in the 0.4–2.5 μm wavelength range and only 200 bands are used after water-absorption bands removal. Around two-thirds of the scene region are agriculture, while the remaining region are composed of forest and other natural perennial vegetation. There exists 16 classes in the Groundtruth map, which are not all mutually exclusive.

2) Pavia University Dataset: This dataset was collected by ROSIS sensor during a flight campaign over urban area of the University of Pavia, Pavia, Italy. The whole scene contains 610 × 340 pixels with the spatial resolution of 1.3 m. In total, the data contains 103 spectral bands with spectral range from 0.43 to 0.86 μm. Around 42 776 labeled pixels are labeled with nine classes in the Groundtruth map.

More details for these two benchmark datasets are given in Table III.

#### B. Effectiveness Verification of the Proposed Frameworks

For the parameter $\mu$ in $K_{ws}$ and $K_{wm}$, we set it according to the recommended values in [35]. For the parameter $\gamma$ and kernel widths in $K_s$, $K_{ws}$, $K_m$, and $K_{wm}$, we simply set it as the recommended values in [35]. More specifically, $K_{ws}$ has two kernel widths ($\sigma_w$, $\sigma_s$) and $K_s$ has one kernel width $\sigma_s$. For $K_{wm}$ and $K_m$, we define the kernel widths as $(\sigma_w, \sigma_m)$ and $\sigma_m$. The details for parameter settings are summarized in Tables IV and V.

To validate the effectiveness of our frameworks, we compared them with four standard kernels with SVM classifier, i.e., the spatial kernel ($K_s$), the spectral-spatial kernel ($K_{ws}$), the spatial mean map kernel ($K_m$), and the composite spectral-spatial mean map kernel ($K_{wm}$). All the spatial related kernels employ a

1) http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

---

### TABLE I

<table>
<thead>
<tr>
<th></th>
<th>Generate Classifiers</th>
<th>Pseudo-Labels Generation</th>
<th>Auxiliary Training Set Generation</th>
<th>Kernel Refinement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLKL</td>
<td>$\mathcal{O}(l +</td>
<td>\mathcal{L}_\Delta</td>
<td>)^3$</td>
<td>$\mathcal{O}(nT)$</td>
</tr>
<tr>
<td>E-PLKL</td>
<td>$\mathcal{O}(MT(l + \frac{</td>
<td>\mathcal{L}_\Delta</td>
<td>}{M})^3)$</td>
<td>$\mathcal{O}(nTMc)$</td>
</tr>
</tbody>
</table>

---

### TABLE II

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Memory Consumption Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLKL</td>
<td>$\mathcal{O}(2\tilde{l}^3 + nT + n + n^2)$</td>
</tr>
<tr>
<td>E-PLKL</td>
<td>$\mathcal{O}(MT(l + \frac{</td>
</tr>
</tbody>
</table>

---

### Algorithm 2: E-PLKL: Efficient Pseudolabel Guided Kernel Learning for HSI Classification.

**Input:** $\mathcal{L} = \{x_i, y_i\}_{i=1}^l$, $\mathcal{U} = \{u_1, \ldots, u_n\}$.

1. Initialize the training set $\mathcal{Z} = \mathcal{L}$, kernel set $\mathcal{K}$, and classifier set $\mathcal{H}$.
2. while not converged do
3. Generate the auxiliary training set $\mathcal{L}_\Delta$ and partition it into $M$ subsets $\{\mathcal{L}_\Delta^1, \mathcal{L}_\Delta^2, \ldots, \mathcal{L}_\Delta^M\}$, and produce the union of expanded subtraining sets $\mathcal{Z} = \{\mathcal{Z}_m\}_{m=1}^i$ with $\mathcal{Z}_m = \{\mathcal{L} \cup \mathcal{L}_\Delta^m\}$ being $m$th expanded subtraining set.
4. Refine each kernel set $\mathcal{K}^m$ with $\mathcal{Z}_m$.
5. Construct the base classifier set $\mathcal{H}^m$ with $\mathcal{K}^m$ and $\mathcal{Z}_m$ and predict pseudolabels for unlabeled samples.
6. end while

**Output:** The final predicted label for each unlabeled sample $\{h_t(u_i)\}_{i=1}^n$.

\[ h_t(u_i) = \arg \max_{y \in \{1, 2, \ldots, c\}} \sum_{m=1}^M I(h_t^m(u_i) = y) \] (12)

where $I$ is an indicator function, i.e.,

\[ I(\text{statement}) = \begin{cases} 1 & \text{if statement is true} \\ 0 & \text{if statement is false} \end{cases} \] (13)

In all, the proposed E-PLKL is summarized in Algorithm 2.
we used three popular evaluation criteria, i.e., overall accuracy (OA), average accuracy (AA), and Kappa coefficient ($\kappa$), to evaluate the performance of HSIs classification. We investigated the performance under different amounts of labeled training samples. More specifically, for both datasets, we randomly chose $P = 5, 10, 15, 20, 25, 30$ labeled samples from each class as training data, while the remaining ones are used as testing data. Note that for the class whose total number of samples is less than $P$, half of the samples are chosen. Here, we reported the average performance by repeating each method with randomly selected training samples for 10 times on Indian Pines and 20 times on Pavia University. The corresponding results are listed in Tables VI and VII, where we can observe that the proposed PLKL and E-PLKL significantly outperform other standard kernels, demonstrating the effectiveness of our frameworks in exploiting the characteristics of unlabeled samples.

Specifically, Table VI shows results on Indian Pines, where proposed $K^{w_m}$-PLKL always achieves the best performance on all cases. Compared with $K^{s}$-Ori, $K^{w_s}$-Ori, $K^{m}$-Ori, and $K^{w_m}$-Ori, our proposed $K^{w}$-PLKL, $K^{w_s}$-PLKL, $K^{m}$-PLKL, and $K^{w_m}$-PLKL improve the average OA with around 12.68%, 12.84%, 9.21%, and 9.33%, respectively. Similarly, our proposed $K^{s}$-E-PLKL, $K^{w_s}$-E-PLKL, $K^{m}$-E-PLKL, and $K^{w_m}$-E-PLKL improve the average OA with around 8.07%, 9.91%, 8.50%, and 9.14%, respectively, at both lower memory and computational costs. Moreover, it can be seen that the advantages of the proposed frameworks are more obvious as the labeled samples per class decreases. Specifically, when $P = 5, 10$, our PLKL and E-PLKL almost improve the accuracy more than 10% than the original kernel. For the larger Pavia University data, there are around 42 776 samples in total. Due to the memory limitation, we can implement the E-PLKL only. From Table VII, we can see that our proposed $K^{w}$-E-PLKL, $K^{w_s}$-E-PLKL, $K^{m}$-E-PLKL, and $K^{w_m}$-E-PLKL improve the average OA with around 12.68%, 12.84%, 9.21%, and 9.33%, respectively. Similarly, our proposed $K^{s}$-E-PLKL, $K^{w_s}$-E-PLKL, $K^{m}$-E-PLKL, and $K^{w_m}$-E-PLKL improve the average OA with around 8.07%, 9.91%, 8.50%, and 9.14%, respectively, at both lower memory and computational costs. Moreover, it can be seen that the advantages of the proposed frameworks are more obvious as the labeled samples per class decreases. Specifically, when $P = 5, 10$, our PLKL and E-PLKL almost improve the accuracy more than 10% than the original kernel.

For the larger Pavia University data, there are around 42 776 samples in total. Due to the memory limitation, we can implement the E-PLKL only. From Table VII, we can see that our proposed $K^{w}$-E-PLKL, $K^{w_s}$-E-PLKL, $K^{m}$-E-PLKL, and $K^{w_m}$-E-PLKL improve the average OA with around 12.68%, 12.84%, 9.21%, and 9.33%, respectively. Similarly, our proposed $K^{s}$-E-PLKL, $K^{w_s}$-E-PLKL, $K^{m}$-E-PLKL, and $K^{w_m}$-E-PLKL improve the average OA with around 8.07%, 9.91%, 8.50%, and 9.14%, respectively, at both lower memory and computational costs. Moreover, it can be seen that the advantages of the proposed frameworks are more obvious as the labeled samples per class decreases. Specifically, when $P = 5, 10$, our PLKL and E-PLKL almost improve the accuracy more than 10% than the original kernel.

The corresponding results are listed in Tables VI and VII, where we can observe that the proposed PLKL and E-PLKL significantly outperform other standard kernels, demonstrating the effectiveness of our frameworks in exploiting the characteristics of unlabeled samples.

Specifically, Table VI shows results on Indian Pines, where proposed $K^{w_m}$-PLKL always achieves the best performance on all cases. Compared with $K^{s}$-Ori, $K^{w_s}$-Ori, $K^{m}$-Ori, and $K^{w_m}$-Ori, our proposed $K^{w}$-PLKL, $K^{w_s}$-PLKL, $K^{m}$-PLKL, and $K^{w_m}$-PLKL improve the average OA with around 12.68%, 12.84%, 9.21%, and 9.33%, respectively. Similarly, our proposed $K^{s}$-E-PLKL, $K^{w_s}$-E-PLKL, $K^{m}$-E-PLKL, and $K^{w_m}$-E-PLKL improve the average OA with around 8.07%, 9.91%, 8.50%, and 9.14%, respectively, at both lower memory and computational costs. Moreover, it can be seen that the advantages of the proposed frameworks are more obvious as the labeled samples per class decreases. Specifically, when $P = 5, 10$, our PLKL and E-PLKL almost improve the accuracy more than 10% than the original kernel.
TABLE VI

<table>
<thead>
<tr>
<th>P</th>
<th>OA</th>
<th>ACCURACY (MEAN ± STD) COMPARISONS OF THE PROPOSED PLKL AND E-PLKL WITH STANDARD KERNELS ON INDIAN PINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>OA</td>
<td>58.75 ± 2.67</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>73.15 ± 1.71</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>54.16 ± 2.82</td>
</tr>
<tr>
<td>10</td>
<td>OA</td>
<td>73.10 ± 2.53</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>83.23 ± 1.30</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>69.68 ± 2.78</td>
</tr>
<tr>
<td>15</td>
<td>OA</td>
<td>77.02 ± 2.47</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>86.46 ± 0.60</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>74.26 ± 1.35</td>
</tr>
<tr>
<td>20</td>
<td>OA</td>
<td>82.02 ± 1.62</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>89.93 ± 1.31</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>79.76 ± 1.83</td>
</tr>
<tr>
<td>25</td>
<td>OA</td>
<td>85.41 ± 1.99</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>91.78 ± 0.35</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>83.46 ± 1.34</td>
</tr>
</tbody>
</table>

TABLE VII

<table>
<thead>
<tr>
<th>P</th>
<th>OA</th>
<th>ACCURACY (MEAN ± STD) COMPARISONS OF THE PROPOSED E-PLKL WITH STANDARD KERNELS ON PAVIA UNIVERSITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>OA</td>
<td>71.30 ± 2.57</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>75.00 ± 3.15</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>63.81 ± 2.16</td>
</tr>
<tr>
<td>10</td>
<td>OA</td>
<td>79.71 ± 2.47</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>82.51 ± 3.80</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>81.08 ± 1.31</td>
</tr>
<tr>
<td>15</td>
<td>OA</td>
<td>84.51 ± 3.44</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>85.59 ± 2.12</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>80.86 ± 1.03</td>
</tr>
<tr>
<td>20</td>
<td>OA</td>
<td>86.74 ± 2.26</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>87.82 ± 1.30</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>82.79 ± 2.82</td>
</tr>
<tr>
<td>25</td>
<td>OA</td>
<td>89.30 ± 1.73</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>89.60 ± 2.17</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>86.01 ± 1.77</td>
</tr>
<tr>
<td>30</td>
<td>OA</td>
<td>90.46 ± 1.75</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>90.42 ± 1.00</td>
</tr>
<tr>
<td></td>
<td>κ</td>
<td>87.48 ± 2.20</td>
</tr>
</tbody>
</table>

Fig. 3. (a) and (b) OA versus number of iterations when labeled samples per class P = 20 for Indian Pines data and Pavia University data, respectively.
TABLE VIII

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>61.36±2.50</td>
<td>67.33±1.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>64.28±2.52</td>
<td>70.21±3.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>68.68±4.51</td>
<td>80.42±4.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>66.70±4.05</td>
<td>82.20±2.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>69.59±4.36</td>
<td>84.44±2.12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>70.80±3.70</td>
<td>85.32±3.16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4. Classification maps obtained by different methods for Indian Pines when five samples per class was used for training data. (a)–(i) Corresponding classification map of Groundtruth, BagSVM (39.92%), KSOMP (71.64%), KBTC-WLS (65.93%), LCMR (72.60%), SC-MK (62.31%), $K_{wm}$-IR (79.34%), $K_{wm}$-PLKL (83.91%), and $K_{wm}$-E-PLKL (87.76%), respectively. (The OAs are reported in the parentheses).

C. Comparisons With State-of-the-Art Methods

We compared the proposed PLKL and E-PLKL with several state-of-the-art HSI classification methods, including the following:

1) semisupervised method, i.e., BagSVM [42];
2) two spectral-spatial sparse coding models, i.e., KBTC-WLS [39] and KSOMP [9];
3) two kernel learning methods, i.e., $K_{wm}$-IR [35] and SC-MK [12];
4) spatial-spectral feature extraction method, i.e., the local covariance matrix representation (LCMR) [11].

Tables VIII and IX show the results in terms of OA, where it can be seen that our proposed PLKL and E-PLKL outperform to a large extent. More importantly, the superiority of the proposed frameworks are more obvious when the labeled training samples per class are very few, e.g., $P = 5, 10, \text{ and } 15$.

The performance does not show significant improvements with the increase of iterations. The reason is that compared with Indian Pines, the samples of Pavia University have less noise and are easier to classify, and overfitting may be potentially caused as the number of iterations increases, resulting in even slightly decreased accuracy.

C. Comparisons With State-of-the-Art Methods

We compared the proposed PLKL and E-PLKL with several state-of-the-art HSI classification methods, including the following:

1) semisupervised method, i.e., BagSVM [42];
2) two spectral-spatial sparse coding models, i.e., KBTC-WLS [39] and KSOMP [9];
3) two kernel learning methods, i.e., $K_{wm}$-IR [35] and SC-MK [12];
4) spatial-spectral feature extraction method, i.e., the local covariance matrix representation (LCMR) [11].

Tables VIII and IX show the results in terms of OA, where it can be seen that our proposed PLKL and E-PLKL outperform to a large extent. More importantly, the superiority of the proposed frameworks are more obvious when the labeled training samples per class are very few, e.g., $P = 5, 10, \text{ and } 15$. Specifically, Table VIII shows the second highest OA by compared methods on case $P = 5$ is 78.60%, which is
much lower than ours, i.e., 83.93%. Moreover, the second best performance by the compared methods on case $P = 20$, (i.e., $OA = 92.59\%$) is even worse than ours with $P$ equal to 15 (i.e., $OA = 93.25\%$). Similar observations can be obtained from Table IX.

Figs. 4 and 5 show the classification maps by different methods with the same $L$ and $P$ equal to five on Indian Pines and Pavia University, respectively, which further demonstrate the superiority of the proposed frameworks. Finally, the average accuracy (AA) for each class is also shown in Tables X and XI, where we could see that our E-PLKL and PLKL produce the highest accuracy for most classes.

D. Visual Comparisons of Learned Kernels by Different Methods

A kernel Gram matrix measures the pairwise sample similarity in the RKHS. The kernel value of samples within the same class should be higher than that of samples across different classes. Here, we visualized the learned different types of Gram matrices in Fig. 6. Specifically, Fig. 6(a) shows the ideal similarity of randomly selected testing data from Indian Pines, where the samples were aligned according to the Groundtruth labels. Fig. 6(b)–(d) shows the corresponding original $K^{w_m}_E$, $K^{w_m}$ by E-PLKL, and $K^{w_m}$ by PLKL, respectively. From these figures, we could see that the Grams by our PLKL and E-PLKL are

![Image of classification maps](image_url)
TABLE X
AA(%) and Standard Deviation (Mean ± STD) of Ten Repeated Experiments on the Indian Pines Image Obtained by Different Methods With Five Training Samples Per Class

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>77.07±1.96</td>
<td>100.00±0.00</td>
<td>66.59±42.22</td>
<td>98.34±1.62</td>
<td>100.00±0.00</td>
<td>98.29±2.20</td>
<td>100.00±2.00</td>
<td>99.31±0.98</td>
</tr>
<tr>
<td>2</td>
<td>31.90±11.84</td>
<td>47.10±12.00</td>
<td>39.13±15.11</td>
<td>62.84±13.09</td>
<td>57.19±10.33</td>
<td>69.47±5.98</td>
<td>68.74±8.90</td>
<td>67.74±9.25</td>
</tr>
<tr>
<td>3</td>
<td>32.39±12.45</td>
<td>56.90±9.54</td>
<td>45.93±17.74</td>
<td>56.88±15.56</td>
<td>53.39±15.13</td>
<td>76.22±11.24</td>
<td>79.18±13.23</td>
<td>77.18±12.99</td>
</tr>
<tr>
<td>4</td>
<td>37.67±15.52</td>
<td>83.97±8.61</td>
<td>63.41±28.21</td>
<td>81.08±18.56</td>
<td>83.88±21.46</td>
<td>89.53±14.58</td>
<td>93.88±11.80</td>
<td>91.94±13.39</td>
</tr>
<tr>
<td>5</td>
<td>55.44±17.04</td>
<td>73.45±8.22</td>
<td>71.63±8.25</td>
<td>83.41±12.35</td>
<td>70.84±6.76</td>
<td>74.02±10.59</td>
<td>72.59±17.75</td>
<td>77.15±9.78</td>
</tr>
<tr>
<td>6</td>
<td>56.06±17.47</td>
<td>66.92±5.13</td>
<td>64.91±7.03</td>
<td>83.57±5.46</td>
<td>92.48±9.02</td>
<td>92.30±4.16</td>
<td>100.00±0.00</td>
<td>96.58±1.38</td>
</tr>
<tr>
<td>7</td>
<td>90.43±6.68</td>
<td>90.00±0.00</td>
<td>99.57±1.30</td>
<td>100.00±0.00</td>
<td>100.00±0.00</td>
<td>99.57±1.30</td>
<td>100.00±0.00</td>
<td>99.57±1.30</td>
</tr>
<tr>
<td>8</td>
<td>69.56±14.18</td>
<td>58.38±6.14</td>
<td>89.45±14.24</td>
<td>100.00±0.00</td>
<td>98.14±4.51</td>
<td>99.11±0.70</td>
<td>100.00±0.00</td>
<td>100.00±0.00</td>
</tr>
<tr>
<td>9</td>
<td>68.00±17.59</td>
<td>100.00±0.00</td>
<td>83.33±18.44</td>
<td>97.33±8.00</td>
<td>100.00±0.00</td>
<td>96.67±8.03</td>
<td>100.00±0.00</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>36.14±8.55</td>
<td>68.09±7.84</td>
<td>56.51±15.20</td>
<td>66.05±10.26</td>
<td>63.20±12.92</td>
<td>80.28±7.45</td>
<td>81.95±6.29</td>
<td>81.45±7.91</td>
</tr>
<tr>
<td>11</td>
<td>37.59±10.53</td>
<td>62.47±14.79</td>
<td>64.61±20.11</td>
<td>59.52±12.98</td>
<td>52.44±14.08</td>
<td>67.96±10.61</td>
<td>76.07±15.99</td>
<td>84.41±6.91</td>
</tr>
<tr>
<td>12</td>
<td>29.39±13.22</td>
<td>53.08±5.26</td>
<td>54.63±15.21</td>
<td>69.29±13.32</td>
<td>57.30±11.61</td>
<td>74.43±1.13</td>
<td>75.22±15.94</td>
<td>77.86±8.60</td>
</tr>
<tr>
<td>13</td>
<td>81.70±8.50</td>
<td>96.45±5.68</td>
<td>100.00±0.00</td>
<td>99.45±1.33</td>
<td>93.75±12.38</td>
<td>98.45±1.74</td>
<td>100.00±0.00</td>
<td>100.00±0.00</td>
</tr>
<tr>
<td>14</td>
<td>45.10±14.81</td>
<td>79.33±14.78</td>
<td>76.48±17.18</td>
<td>90.21±5.21</td>
<td>78.76±9.99</td>
<td>86.31±8.96</td>
<td>88.55±8.85</td>
<td>90.91±10.26</td>
</tr>
<tr>
<td>15</td>
<td>30.13±10.32</td>
<td>75.41±8.51</td>
<td>66.56±19.12</td>
<td>90.68±12.26</td>
<td>83.88±12.56</td>
<td>86.14±9.64</td>
<td>91.73±8.72</td>
<td>94.41±7.83</td>
</tr>
<tr>
<td>16</td>
<td>89.09±4.23</td>
<td>98.64±1.51</td>
<td>96.48±4.97</td>
<td>96.59±1.33</td>
<td>98.86±0.00</td>
<td>99.20±1.44</td>
<td>98.89±1.34</td>
<td>99.89±1.34</td>
</tr>
</tbody>
</table>

The best results are highlighted in bold.

The best results are highlighted in bold.

TABLE XI
AA(%) and Standard Deviation (Mean ± STD) of 20 Repeated Experiments on the Pavia University Image Obtained by Different Methods With Five Training Samples Per Class

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>54.35±2.88</td>
<td>79.37±1.80</td>
<td>73.06±4.21</td>
<td>83.71±2.19</td>
<td>80.26±2.45</td>
<td>87.15±1.31</td>
<td>89.03±1.70</td>
</tr>
<tr>
<td>OA</td>
<td>41.83±4.62</td>
<td>67.65±3.12</td>
<td>64.41±3.36</td>
<td>72.83±4.25</td>
<td>67.07±4.52</td>
<td>78.60±3.04</td>
<td>81.86±4.76</td>
</tr>
<tr>
<td>κ</td>
<td>35.41±4.67</td>
<td>67.31±3.32</td>
<td>59.69±3.58</td>
<td>69.52±4.51</td>
<td>63.04±4.92</td>
<td>75.94±3.32</td>
<td>79.52±5.19</td>
</tr>
</tbody>
</table>

The best results are highlighted in bold.

much closer to the Groundtruth. Consequently, they produce better class separability than the original kernel.

E. Computational Time Analysis

As analyzed in Section III-E, the E-PLKL could not only reduce the memory cost of PLKL, but also be beneficial to the computational complexity. Fig. 7 compares the running times of PLKL and E-PLKL over Indian Pines, where we randomly chose 15 samples per class as training data, and the remaining ones were used as testing data. From Fig. 7, it can be seen that the running time of E-PLKL is much lower than that of PLKL because PLKL uses a larger augmented training set during the learning process. And the running time of each
iteration gradually increases with the number of iterations increasing. The reason is that the classification performances of major kernels improve with the number of iterations increasing, leading to more samples to be contained in the augmented training set. Note that these experiments were implemented in MATLAB 2017 on a PC with E5-2640 CPU @ 2.40 GHz and 192 GB RAM.

V. CONCLUSION

In this paper, we have proposed a new framework for HSI classification, called PLKL, which is flexible and easy to extend to any existing kernel-based methods. Different from the traditional kernel methods that only consider the sample similarity or label similarity included in training data, the proposed framework is able to exploit the sample similarity and label similarity included in both training and testing data. It incorporates the pseudolabels of testing data into a standard kernel by means of the ensemble results of multiple classifiers. The proposed PLKL is suitable for various state-of-the-art kernels. To handle large scale datasets, we also propose an efficient PLKL which could keep performance comparable to PLKL, but at much lower memory and computational costs. Experimental results show that, by exploiting high confident information in unlabeled data, the proposed PLKL outperforms state-of-the-art spectral-spatial-classification methods. Moreover, our approach is more effective especially when the labeled samples are extremely limited.

REFERENCES


Shujun Yang received the B.S. degree from Jiangsu University, Zhenjiang, China, in 2011, and the M.S. degree from Nanjing University, Nanjing, China, in 2014, both in computer science. She is currently working toward the Ph.D. degree in computer science at the Department of Computer Science, City University of Hong Kong, China. Her research interests include hyperspectral remote sensing imaging processing and applications, pattern recognition, and machine learning.

Junhui Hou (S’13–M’16) received the B.Eng. degree in information engineering (Talented Students Program) from the South China University of Technology, Guangzhou, China, in 2009, the M.Eng. degree in signal and information processing from Northwestern Polytechnical University, Xi’an, China, in 2012, and the Ph.D. degree in electrical and electronic engineering from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2016. He has been an Assistant Professor with the Department of Computer Science, City University of Hong Kong, since 2017. His current research interests include multimedia signal processing, such as adaptive video representations and analysis (RGB/depth/light field/hyperspectral), static/dynamic 3-D geometry representations and processing (mesh/pixel/point cloud/Mocap), and discriminative modeling for clustering/classification. Dr. Hou was the recipient of the Prestigious Award from the Chinese Government for Outstanding Self-Financed Students Abroad, China Scholarship Council, in 2015, and the Early Career Award from the Hong Kong Research Grants Council in 2018. He currently serves as an Associate Editor for *The Visual Computer*, an Area Editor for *Signal Processing: Image Communication*, and the Guest Editor for the *Journal of Visual Communication and Image Representation*.

Yuheng Jia received the B.S. degree in automation and the M.S. degree in control theory and engineering from Zhejiang University, Zhejiang, China, in 2012 and 2015, respectively, and the Ph.D. degree in computer science from the City University of Hong Kong, China, in 2018. His research interests include machine learning, Bayesian method, and low rank modeling.

Shaohui Mei (S’10–M’12) received the B.S. degree in electronics and information engineering and the Ph.D. degree in signal and information processing from Northwestern Polytechnical University, Xi’an, China, in 2005 and 2011, respectively. He is currently an Associate Professor with the School of Electronics and Information, Northwestern Polytechnical University, China. He was a Visiting Student with the University of Sydney, from October 2007 to October 2008. His research interests include hyperspectral remote sensing image processing and applications, intelligent signal and information acquisition and processing, neural networks, pattern recognition, and machine learning.

Qian Du (S’98–M’09–S M’15–F’18) received the Ph.D. degree in electrical engineering from the University of Maryland at Baltimore County, Baltimore, MD, USA, in 2000. She is currently the Bobby Shackouls Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS, USA. Her research interests include hyperspectral remote sensing image analysis and applications, pattern classification, data compression, and neural networks. Dr. Du is a Fellow of the SPIE-International Society for Optics and Photonics. She was the recipient of the 2010 Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society. She was a Co-Chair of the Data Fusion Technical Committee of the IEEE Geoscience and Remote Sensing Society from 2009 to 2013, and the Chair of Remote Sensing and Mapping Technical Committee of the International Association for Pattern Recognition from 2010 to 2014. She served as an Associate Editor for the *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, the *Journal of Applied Remote Sensing*, and *IEEE Signal Processing Letters*. Since 2016, she has been the Editor-in-Chief for the *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. She is the General Chair of the 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing in Shanghai, China, in 2012.