

NSCKL: Normalized Spectral Clustering With Kernel-Based Learning for Semisupervised Hyperspectral Image Classification

Yuanchao Su¹, Member, IEEE, Lianru Gao², Senior Member, IEEE, Mengying Jiang³, Antonio Plaza⁴, Fellow, IEEE, Xu Sun¹, Member, IEEE, and Bing Zhang¹, Fellow, IEEE

Abstract—Spatial–spectral classification (SSC) has become a trend for hyperspectral image (HSI) classification. However, most SSC methods mainly consider local information, so that some correlations may not be effectively discovered when they appear in regions that are not contiguous. Although many SSC methods can acquire spatial–contextual characteristics via spatial filtering, they lack the ability to consider correlations in non-Euclidean spaces. To address the aforementioned issues, we develop a new semisupervised HSI classification approach based on normalized spectral clustering with kernel-based learning (NSCKL), which can aggregate local-to-global correlations to achieve a distinguishable embedding to improve HSI classification performance. In this work, we propose a normalized spectral clustering (NSC) scheme that can learn new features under a manifold assumption. Specifically, we first design a kernel-based iterative filter (KIF) to establish vertices of the undirected graph, aiming to assign initial connections to the nodes associated with pixels. The NSC first gathers local correlations in the Euclidean space and then captures global correlations in the manifold. Even though homogeneous pixels are distributed in noncontiguous regions, our NSC can still aggregate correlations to generate new (clustered) features. Finally, the clustered features and a kernel-based extreme learning machine (KELM)

Manuscript received 25 June 2022; revised 12 September 2022; accepted 27 October 2022. Date of publication 17 November 2022; date of current version 16 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 42001319 and Grant 62161160336; in part by the European Union’s Horizon Research; in part by the Scientific Research Program of the Education Department of Shaanxi Province under Grant 21JK0762; and in part by the Scientific Research Foundation of Xi’an University of Science and Technology under Grant 2050120010. This article was recommended by Associate Editor M. Zhang. (*Corresponding author: Lianru Gao.*)

Yuanchao Su is with the Department of Remote Sensing, College of Geomatics, Xi’an University of Science and Technology, Xi’an 710054, China (e-mail: suych3@xust.edu.cn).

Lianru Gao and Xu Sun are with the Key Laboratory of Computational Optical Imaging Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (e-mail: gaolr@aircas.ac.cn; sunxu@aircas.ac.cn).

Mengying Jiang is with the School of Electronics and Information Engineering, Xi’an Jiaotong University, Xi’an 710049, China (e-mail: myjiang@stu.xjtu.edu.cn).

Antonio Plaza is with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, 10071 Cáceres, Spain (e-mail: aplaza@unex.es).

Bing Zhang is with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China, and also with the College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zb@radi.ac.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCYB.2022.3219855>.

Digital Object Identifier 10.1109/TCYB.2022.3219855

are employed to achieve the semisupervised classification. The effectiveness of our NSCKL is evaluated by using several HSIs. When compared with other state-of-the-art (SOTA) classification approaches, our newly proposed NSCKL demonstrates very competitive performance. The codes will be available at <https://github.com/yuanchaosu/TCYB-nsckl>.

Index Terms—Graph structure learning, hyperspectral image (HSI) classification, semisupervised classification, spatial–spectral classification (SSC), spectral clustering (SC).

I. INTRODUCTION

HYPERSPECTRAL imagery can be utilized for accurately defining land cover types as the sensors can obtain spatial details and spectral information from land surfaces [1], [2], [3]. Spectral signatures can combine with spatial details to assign a unique class label to each pixel. Therefore, the application of HSI enables land cover/use classifications more accurately [4]. To further enhance classification performance, many researchers combine spatial and spectral features for HSI classification, such as the local covariance matrix representation (LCMR) [5], region-kernel-based support vector machine (RKSVM) [6], or spectral–spatial networks (SSNs) [7].

Over the past few years, spatial–spectral classification (SSC) methods have been effective in many scenarios because they combine spatial information and spectral signatures to improve classification performance [8]. Nevertheless, some spatial–contextual characteristics may not be captured effectively if they hide in noncontiguous regions, and most SSC methods only focus on the spatial homogeneity of contiguous pixels. In HSIs, a pixel and its neighbors usually belong to the same material and possess similar spectral signatures when located in a homogeneous region [9]. Since the neighboring pixels only involve local spatial information, conventional SSC methods may neglect some essential information about the spatial correlation among objects if the homogeneous pixels are distributed in noncontiguous regions [8]. Recently, some SSC methods employed filters to gather local spatial characteristics from neighboring pixels. However, these methods face a challenge that conventional neighborhood hypotheses cannot characterize correlations from noncontiguous pixels.

Additionally, classification techniques need to use a large number of training samples if HSIs are processed by supervised schemes [10]. Considering that semisupervised schemes

can reduce the dependence on training samples, many researchers combined labeled training samples with available unlabeled pixels to develop semisupervised SSC methods, such as the geometric low-rank Laplacian regularized with composite kernel support vector machine (GLLR-CKSVM) [11]. In recent years, the development of deep learning (DL) has fostered many newcome semisupervised methods, such as spectral-spatial graph convolutional networks (S^2 GCNs) [12] or semisupervised deep feature learning with feature consistency (SDFL-FC) [13]. These methods have been shown to be very effective for SSC. However, they might suffer from the local oversmoothing problem when characterizing neighboring correlations. Moreover, DL-based methods are often computationally costly since they adopt deep structures.

Recently, spectral clustering (SC) has attracted wide attention in the cybernetics community and has been successfully applied to various fields, for example, information processing, image segmentation, social network partitioning, gene expression analysis, and many more [14]. As a new clustering technology, SC derives from graph theory and is not restricted to the condition that data follows a Gaussian distribution, compared with most conventional clustering algorithms [15]. The fundamental idea of SC is to construct an undirected graph using the observation data and a similarity metric function, aiming to embed the input data into a low-dimensional space [16]. In processing HSIs, SC can partition the image into several clusters by comparing the spectral signature associated with each pixel [17]. The pixels contained in a cluster are assumed to be very similar, while the pixels assigned to different clusters are considered to be distinct [18]. Although SC has been recently applied to processing HSIs, most of the related works focus on unsupervised problems rather than supervised or semisupervised classification. Furthermore, it is also challenging to construct and decompose the weighted adjacency matrix involved in SC due to the large number of pixels and bands in HSIs.

The extreme learning machine (ELM) is a single-hidden-layer neural network that only involves feedforward connections without the need to fine-tune the node parameters of hidden layers [19], [20], [21], [22]. The kernel-based ELM (KELM) is an extension of ELM that further enhances the stability of learning by using a kernel trick [23]. The KELM can also be regarded as a single-hidden-layer network, whereby kernel functions are treated as activation functions in the hidden layer, and the number of training samples equals the number of nodes [24]. In the implementation of KELM, the training error and the norm of the output weights are minimized simultaneously [25]. Because of its good generalization and robustness, KELM (a kernel-based learning (KL) tool) has been employed to develop HSI classification approaches, such as the SSN, and KELM with composite kernels [25]. Although these methods reveal that KELM exhibits good potential for HSIs classification, the correlations among non-contiguous objects cannot be properly captured because they mainly concern the local information from contiguous pixels.

Resulting from the above, we can conclude that the fast acquisition of global information is an important goal of SSC

applications and the main motivation of this work. To achieve this, we employ the normalized spectral clustering with KL (NSCKL) to develop a semisupervised HSI classification approach. The implementation of our NSCKL approach includes two main steps: 1) feature representation and 2) pixel classification. We first develop a new NSC scheme that can represent clustered features by aggregating correlations from both Euclidean and manifold spaces. In the NSC scheme, we design a new kernel-based iterative filter (KIF) to establish the vertices of an undirected graph. For the KIF, each node is taken as a center to aggregate the local characteristics from neighboring nodes, and the nodes with identical properties constitute the vertices, and all nodes and vertices are updated via a KL process iteratively. With the updates of nodes, the consistency between neighboring pixels increases, and the similarity relationships between neighboring pixels gradually stabilize. Utilizing the KL scheme, our filter can adaptively calculate the aggregation weights between the center pixel and its neighbors. Moreover, a second-order difference is embedded into the filter to avoid oversmoothing of local features. The vertices obtained from the KIF provide the local connections of nodes, and thus we adopt a fully connected scheme to acquire the weighted adjacency matrix of the graph (considering global characteristics). Regardless of the distribution of pixels within the same material, the clustered features consider the correlation between these pixels after conducting the NSC. Finally, we use the clustered features to train a KELM, aiming to classify pixels in the HSI. Our NSCKL can better collect the global correlations to improve classification performance by obtaining the features from homogenous regions. The main contributions of our NSCKL can be summarized as follows.

- 1) The idea of SC is innovatively applied to semisupervised HSI classification, significantly improving classification performance by means of a graph structure learning-based feature representation.
- 2) The proposed NSCKL can efficiently process HSIs because the new NSC is a fast method that can reduce the dimensionality of the data.
- 3) The newly developed KIF refines the spatial information features via iterative learning, avoiding the oversmoothing of local features.

The remainder of this article is organized as follows. In Section II, the related works are briefly reviewed. Section III describes the proposed NSCKL in detail. Afterward, in Section IV, we evaluate the proposed NSCKL by drawing comparisons with other state-of-the-art (SOTA) methods. Finally, Section V concludes this work with some remarks and hints at plausible future research lines. Table I lists the key notations involved in this article unless otherwise stated.

II. RELATED WORKS

The proposed approach adopts the SC idea to explore the correlations from noncontiguous pixels. In our work, KL is used for feature representation and pixel classification. In the following, we introduce some relevant works related to our newly proposed methodology.

TABLE I
EXPLANATION OF KEY NOTATIONS REFERRED TO THIS ARTICLE

Notation	Explanation
$\mathbf{X} \in \mathbb{R}^{n \times h}$	HSI, including n pixels with h bands
$\mathbf{x}_i \in \mathbb{R}^{h \times 1}$	Pixel: a vector with h dimensions, $i = 1, \dots, n$
$\hat{\mathbf{x}}_i \in \mathbb{R}^{h \times 1}$	Node: a vector with h dimensions
$\tilde{\mathbf{X}} \in \mathbb{R}^{n \times h}$	Node matrix: all nodes
$\tilde{\mathbf{X}} \in \mathbb{R}^{n \times h}$	Transformed matrix of $\tilde{\mathbf{X}}$
$\tilde{\mathbf{X}} \in \mathbb{R}^{n \times n}$	Sparse matrix, including n sparse vectors
$\tilde{\mathbf{x}}_i \in \mathbb{R}^{n \times 1}$	Sparse vector: local similarity between \mathbf{x}_i and all pixels
$\tilde{\mathbf{X}} \in \mathbb{R}^{k \times h}$	Anchor matrix: all anchors
c	Number of cuttings
k	Number of anchors
γ	Width factor of the Gaussian kernel
$.$	Element-wise division
\mathcal{G}	Undirected graph
\mathcal{G}_{Cut}	Graph cut
$\mathcal{S}_{\mathcal{G}}$	Subgraph
$\mathbf{A} \in \mathbb{R}^{n \times n}$	Clustering indicator matrix
$\mathbf{B} \in \mathbb{R}^{n \times c}$	Clustered feature matrix: all clustered features
$\mathbf{b}_i \in \mathbb{R}^{c \times 1}$	Clustered feature: a vector with c dimensions
$\mathbf{D} \in \mathbb{R}^{n \times n}$	Degree matrix
$\mathbf{E} \in \mathbb{R}^{n \times n}$	Normalized similarity matrix
$\mathbf{F} \in \mathbb{R}^{n \times k}$	Node-to-anchor matrix
\mathbf{F}	Transform matrix of \mathbf{F}
\mathbf{I}_c and \mathbf{I}_ϕ	Identity matrix
$\mathbf{K} \in \mathbb{R}^{\phi \times \phi}$	Kernel matrix, using ϕ training samples
$\mathbf{L} \in \mathbb{R}^{n \times n}$	Graph Laplacian matrix
$\mathbf{U} \in \mathbb{R}^{n \times n}$	Left singular vectors
$\mathbf{\Sigma} \in \mathbb{R}^{n \times h}$	Singular value matrix
$\mathbf{V} \in \mathbb{R}^{h \times h}$	Right singular vectors
$\mathbf{W} \in \mathbb{R}^{n \times n}$	Adjacency matrix of nodes: all edges of \mathcal{G}
w_{ij}	Edge between two nodes, $j = 1, 2, \dots, n$
$\text{Tr}(\cdot)$	Trace of a matrix
$\mathcal{N}(\cdot)$	Neighborhood function
$\Delta(\cdot)$	Differential function
$\mathcal{F}(\cdot)$	Mapping function
$K(\cdot)$	Kernel function

A. Spectral Clustering

The fundamentals of SC depend on the manifold assumption, which assumes that two samples have the same characteristics if they are located in the high density-region of a low-dimensional manifold [26]. The implementation of SC refers to two main steps: 1) structuring an undirected graph and 2) cutting the undirected graph [14]. Regarding the first step, the samples can be regarded as nodes, and all nodes can be interconnected as an undirected graph by means of edges. The edge value between two nodes that are farther away is lower, while the edge value between two nodes that are close is higher [14]. After cutting the undirected graph, the sum of edge values between different subgraphs will be as low as possible, while the sum of edge values within the subgraphs is intended to be as high as possible, achieving the final clustering [27].

When applying SC, we usually structure an undirected graph to denote weights among nodes, and the edges of the undirected graph can be expressed as a symmetric affinity matrix. Let $\mathbf{X} \equiv [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times h}$ be the input data, and $\mathbf{x}_i \in \mathbb{R}^{h \times 1}$ is a vector with h dimensions, $i = 1, \dots, n$. Given an undirected graph $\mathcal{G} = (\mathbf{X}, \mathbf{W})$, \mathbf{X} represents the nodes of the undirected graph, and $\mathbf{W} \in \mathbb{R}^{n \times n}$ is the adjacency matrix

that denotes all edges of the graph and is a symmetric affinity matrix [28]. Each element w_{ij} in \mathbf{W} is an edge of two nodes and represents the pairwise similarity. Note that \mathbf{W} in a fully connected graph can directly be used as the weighted adjacency matrix [29]. According to [14], the objective function of cutting the graph is denoted as

$$\mathcal{G}_{\text{Cut}} = \min \frac{1}{2} \sum_{k=1}^c \mathbf{W}(\mathcal{S}_{\mathcal{G}}, \bar{\mathcal{S}}_{\mathcal{G}k}) \quad (1)$$

where c is the number of cuttings, $\mathcal{S}_{\mathcal{G}}$ is a subgraph via cutting, and $\bar{\mathcal{S}}_{\mathcal{G}k}$ is a complement corresponding to $\mathcal{S}_{\mathcal{G}}$, and the term $1/2$ is aimed at avoiding the consideration of each edge twice [14]. The normalized cut (Ncut) [30] is the most common way for cutting graphs. The Ncut involves the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} \in \mathbb{R}^{n \times n}$ is the degree matrix of \mathbf{W} and is a diagonal matrix. Let d_i be the diagonal element of \mathbf{D} , and d_i is calculated by using $w_{i,j}$, and the acquisition of \mathbf{D} is denoted as

$$\mathbf{D} \Leftarrow d_i = \sum_{j=1}^n w_{i,j}. \quad (2)$$

Following [30], the objective function in (1) can be rewritten as

$$\mathcal{G}_{\text{Cut}} = \min_{\mathbf{A}^\top \mathbf{D} \mathbf{A} = \mathbf{I}_c} \text{Tr}(\mathbf{A}^\top \mathbf{L} \mathbf{A}) \quad (3)$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_c] \in \mathbb{R}^{n \times c}$ is the clustering indicator matrix, $\mathbf{I}_c \in \mathbb{R}^{c \times c}$ is an identity matrix, and \mathbf{a}_g is the indicator vector of $\mathcal{S}_{\mathcal{G}}$, $g = 1, \dots, c$. To relax the discreteness condition, according to [14] and given that $\mathbf{B} = \mathbf{D}^{1/2} \mathbf{A} \in \mathbb{R}^{n \times c}$, and the objective function can be defined as

$$\mathcal{G}_{\text{Cut}} = \min_{\mathbf{B}^\top \mathbf{B} = \mathbf{I}_c} \text{Tr}(\mathbf{B}^\top \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \mathbf{B}) \quad (4)$$

where $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]^\top \in \mathbb{R}^{n \times c}$, \mathbf{b}_i is regarded as a clustered feature associated with \mathbf{x}_i . SC includes a family of methods that adopt different ways to calculate \mathbf{W} and \mathbf{B} [27]. Finally, the input data \mathbf{X} can be partitioned into c clusters.

B. Kernel-Based Learning

KL provides a ubiquitous way for conducting various learning tasks, such as dimensionality reduction, filtering, clustering, and classification. A kernel is an arbitrary function that maps objects in a complex space to a high-dimensional space that enables comparisons of these objects easily [31]. In this higher-dimensional space, complex data may become better separated or structured, and the mapping with kernels has no constraints [32].

Kernel tricks can bring linearity to any nonlinear algorithm by expressing a product between two samples [33]. Let \mathcal{X} be the original space, and $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ are two samples. The original data set uses n samples, $j = 1, 2, \dots, n$. The kernel function denotes an inner product in a feature space, and it can be written as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \quad (5)$$

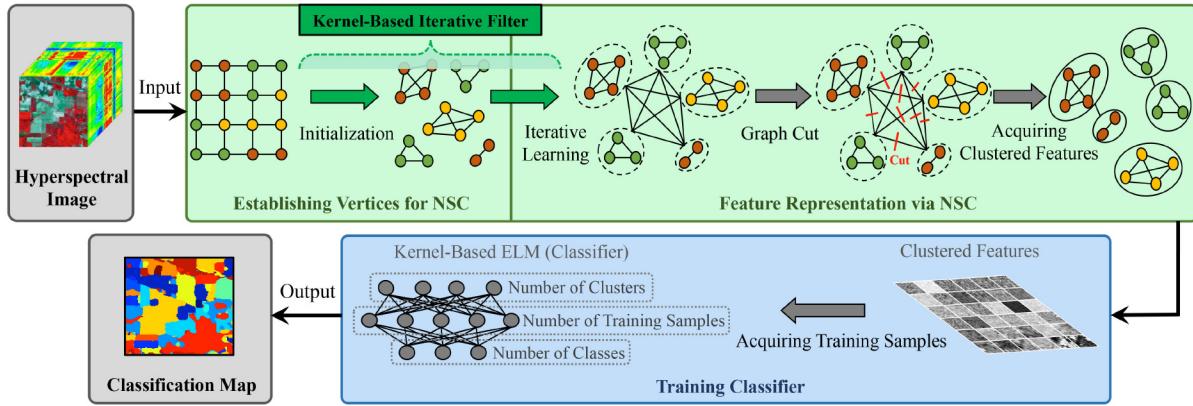


Fig. 1. Flowchart of the proposed NSCKL, involving two parts: feature representation (Jade-green part) and pixel classification (Wathet-blue part). The newly developed KIF first establishes the vertices of the undirected graph, then the NSC achieves feature representation by cutting the graph. The clustered features are used to train KELM for classifying pixels. Note that a red/green/orange dot represents a node associated with a pixel, and a dotted circle represents the vertex that carries the spatial information from the neighboring pixels.

where $\phi(\cdot)$ denotes a mapping relationship, $\phi(\cdot) : \mathcal{X} \rightarrow \mathfrak{R}$, and \mathfrak{R} represents the feature space. If we use a kernel function, the related algorithm would generate a higher-dimensional space without explicitly mapping the data into this space [34]. A kernel function is able to transform the original data into the required form, while KL methods involve various kernel functions [35]. For a KL method, the appropriate choice of kernels is an indispensable part of the overall work [36].

Nowadays, many kernel functions are used for learning particular tasks, mainly including linear kernel (LK) and Gaussian kernel (GK). LK is suitable for objects represented by a large number of features with a fixed length, and its expression is an inner product between two sample vectors; GK is a radial approximation function and can ensure high accuracy in measuring similarity when data become dense [37]. In practice, the motivation behind the choice of a kernel depends on information acquired by users' needs [33].

III. METHODOLOGY

Despite being noncontiguous with each other, some pixels may actually record the same materials in the HSI. In this work, we propose a new NSC algorithm that can aggregate related characteristics from contiguous and noncontiguous pixels to acquire clustered features. The NSC is combined with the KELM to achieve semisupervised HSI classification. Since the similarity between pixels is determined by a local-to-global strategy, the acquired clustered features can help the KELM to improve classification performance significantly. The framework of our classification approach contains two main parts, as illustrated in Fig. 1. The first part develops the KIF to establish the vertices of the undirected graph involved in the NSC, and then we implement the graph cut to acquire clustered features. In the second part, the acquired clustered features provide training samples to train the KELM to classify pixels in the HSI.

A. Establishing Vertices for NSC

HSIs can be regarded as a cross-grid structure data in which a node corresponds to a pixel. In the undirected graph, a vertex

aggregates the local characteristics of neighboring pixels, and these vertices can provide the initial connections between pixels. In order to establish vertices, we develop the KIF to aggregate the neighboring spatial characteristics for each node by iterative learning. Let a vector $\hat{\mathbf{x}}_i \in \mathbb{R}^{h \times 1}$ denote a node, and the initial $\hat{\mathbf{x}}_i$ equals the corresponding pixel \mathbf{x}_i . Let $\hat{\mathbf{X}}$ be the initial-node matrix, $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_i\}_{i=1}^n \in \mathbb{R}^{n \times h}$. Before implementing the KIF, all the elements of pixels are normalized to the range of $[0, 1]$ to ensure reasonable spectral reflectance. The normalization is given by the following equation:

$$x_m = \frac{x_m - x_{\min}}{x_{\max} - x_{\min}} \quad (6)$$

where x_m is the m th element in a pixel, and x_{\min} and x_{\max} are the minimum and maximum elements, respectively.

Let (α_i, β_i) represent the coordinate of the node $\hat{\mathbf{x}}_i$ in a image, and let η be the width of the neighborhood window. Note that η is a hyperparameter set as an odd number. Following previous work in [38], we regard $\hat{\mathbf{x}}_i$ as the center of the neighborhood window, and then its neighborhood $\mathcal{N}(\hat{\mathbf{x}}_i)$ can be defined as

$$\mathcal{N}(\hat{\mathbf{x}}_i) = \left\{ \hat{\mathbf{x}}_{ir} \mid \hat{\mathbf{x}}_{ir} := (\alpha, \beta), r = 1, 2, \dots, \eta^2 - 1 \right\} \quad (7)$$

where α and β denote the coordinate range of neighboring nodes, $\alpha \in [\alpha_i - \theta, \alpha_i + \theta]$, $\beta \in [\beta_i - \theta, \beta_i + \theta]$, and $\theta = (\eta - 1)/2$. A larger window may collect too many heterogeneous pixels, while a small window would increase the computational burden significantly. Thus, we set $\eta = 5$ to balance efficiency and accuracy in this work. According to (7), all nodes included in the window are represented as $\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{i1}, \dots, \hat{\mathbf{x}}_{i\eta^2-1}$.

With respect to the center $\hat{\mathbf{x}}_i$, we mainly focus on its similarity with the nodes in the window because nearby pixels are more likely to belong to the same material than distant ones. The KIF adopts a GK to obtain the similarity between each node and the other pixels in the neighborhood window. As a result, we obtain η^2 similarity values because the window has η^2 pixels. Meanwhile, the number of nodes outside the window is $n - \eta^2$, and their similarity values are 0. Typically, the number of bands is usually much less than the number of

pixels, that is, $h \ll n$. The i th node will be expanded as a sparse vector $\hat{\mathbf{x}}_i \in \mathbb{R}^{n \times 1}$ by using the indexes of pixels, and it can be written as

$$\hat{\mathbf{x}}_i \Leftarrow \text{Index}(K(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{ij})) \quad (8)$$

where $K(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{ij}) = \exp(-\gamma \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{ij}\|_2^2)$ is the GK, and γ is a hyperparameter that determines the width of the GK. The sparse vector $\hat{\mathbf{x}}_i \in \mathbb{R}^{n \times 1}$ represents the similarity between the pixel \mathbf{x}_i and all pixels, and these vectors can compose a sparse matrix $\ddot{\mathbf{X}} = \{\ddot{\mathbf{x}}_i\}_{i=1}^n$, $\ddot{\mathbf{X}} \in \mathbb{R}^{n \times n}$. The matrix $\ddot{\mathbf{X}} \in \mathbb{R}^{n \times n}$ is a symmetric matrix, and its diagonal elements are 1. The node matrix $\hat{\mathbf{X}}$ can be updated by the sparse matrix $\ddot{\mathbf{X}}$

$$\hat{\mathbf{X}} = (\ddot{\mathbf{X}} \hat{\mathbf{X}}) ./ (\ddot{\mathbf{X}} \mathbf{1}_{(n \times h)}) \quad (9)$$

where $./$ is the element-wise division, and $\mathbf{1}_{(n \times h)}$ is a matrix with all elements being 1. Each central node $\hat{\mathbf{x}}_i$ in the window can aggregate the spatial information from neighboring pixels by means of filtering.

According to the iterative learning procedure in [39], finite differences can be applied to avoid oversmoothing phenomena. However, it should be noted that first-order differences may face the problem that a change in $\ddot{\mathbf{X}}$ could be small if all its nonzero elements are very close to 1. To avoid such phenomenon, we employ second-order differences to accomplish this task

$$\begin{aligned} \Delta(\Delta(\ddot{\mathbf{X}}^{(t)})) &= \Delta(\ddot{\mathbf{X}}^{(t)} - \ddot{\mathbf{X}}^{(t-1)}) \\ &= \Delta(\ddot{\mathbf{X}}^{(t)}) - \Delta(\ddot{\mathbf{X}}^{(t-1)}) \\ &= \ddot{\mathbf{X}}^{(t)} - \ddot{\mathbf{X}}^{(t-1)} - (\ddot{\mathbf{X}}^{(t-1)} - \ddot{\mathbf{X}}^{(t-2)}) \\ &= \ddot{\mathbf{X}}^{(t)} - 2\ddot{\mathbf{X}}^{(t-1)} + \ddot{\mathbf{X}}^{(t-2)} \end{aligned} \quad (10)$$

where $\Delta(\cdot)$ represents a differential function, and t denotes the iteration times. The stopping condition of the iterative learning process is defined as follows:

$$\text{Stop : } \left\| \left(2\ddot{\mathbf{X}}^{(t-1)} - \ddot{\mathbf{X}}^{(t)} - \ddot{\mathbf{X}}^{(t-2)} \right) \right\|_2^2 / n \leq \Gamma \quad (11)$$

where Γ is a threshold for the stopping iteration, $0 < \Gamma \ll 1$. In each iteration, the filter runs n times to optimize n nodes. With the iterative process, the changes of nonzero elements in the sparse matrix $\ddot{\mathbf{X}}$ become stable dynamically. The nodes with an identical class will have higher consistency regarding the GK-based similarity. These nodes with local spatial consistency define the vertices of the undirected graph. Fig. 2 illustrates the process adopted for obtaining the vertices by using the KIF. Considering the GK and the iterative process, the computational complexity of this step is $\mathcal{O}(n\eta^2ht)$.

B. Graph Cut for NSC

The newly proposed NSC is dependent on the manifold assumption, where the samples belonging to identical classes are closer to each other. Many relevant characteristics may be aggregated via cutting the graph. According to (4), the SC would be completed after acquiring \mathbf{B} . First, we adopt $\mathbf{L} = \mathbf{D} - \mathbf{W}$ to transform $\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ in (4), and then we can obtain the following conversion:

$$\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}. \quad (12)$$

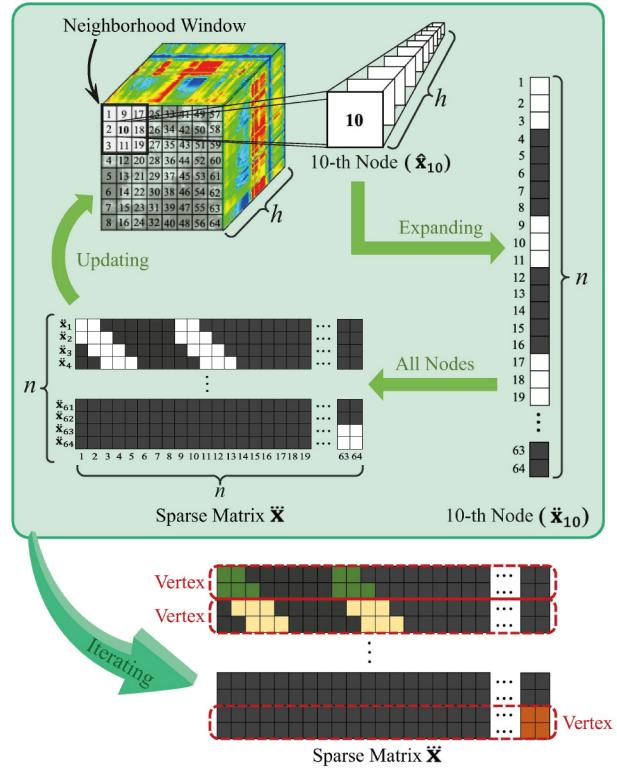


Fig. 2. Illustration of how the vertices of the undirected graph are established. As an example, the 10th node (associated with the 10th pixel) is the center of the window, and pixels 1–3, 9–11, and 17–19 are included in the window. The node $\hat{\mathbf{x}}_{10} \in \mathbb{R}^{h \times 1}$ is expanded to a sparse vector $\hat{\mathbf{x}}_{10} \in \mathbb{R}^{n \times 1}$ by using the indexes of pixels. In the sparse vector $\hat{\mathbf{x}}_{10}$, all black squares represent zeros, while all white squares are nonzero values. The sparse vectors form a sparse matrix $\ddot{\mathbf{X}}$. These nodes (with local spatial consistency) define the vertices after iterative learning.

Introducing (12) into (4), the objective function in (4) can be rewritten as

$$\begin{aligned} \mathcal{G}_{\text{Cut}} &= \min_{\mathbf{B}^T \mathbf{B} = \mathbf{I}_c} \text{Tr}(\mathbf{B}^T (\mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}) \mathbf{B}) \\ &= \max_{\mathbf{B}^T \mathbf{B} = \mathbf{I}_c} \text{Tr}(\mathbf{B}^T \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} \mathbf{B}). \end{aligned} \quad (13)$$

The expression above (13) can be used to calculate an optimal solution with respect to \mathbf{B} . Let $\mathbf{E} = \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$ be a normalized similarity matrix, $\mathbf{E} \in \mathbb{R}^{n \times n}$. The objective function in (13) can be converted to

$$\mathcal{G}_{\text{Cut}} = \max_{\mathbf{B}^T \mathbf{B} = \mathbf{I}_c} \text{Tr}(\mathbf{B}^T \mathbf{E} \mathbf{B}) \quad (14)$$

where the matrix \mathbf{B} can be composed by c eigenvectors corresponding to the largest eigenvalues in \mathbf{E} . To acquire a separable adjacency matrix, we adopt the LK to obtain \mathbf{W} , and the procedure is defined as

$$\mathbf{W} = \hat{\mathbf{X}} \hat{\mathbf{X}}^T \Leftarrow w_{ij} = \langle \hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j \rangle. \quad (15)$$

The degree matrix \mathbf{D} can be acquired by using (2) and (15). Afterward, \mathbf{E} can be redefined as follows:

$$\mathbf{E} = \mathbf{D}^{-1/2} \hat{\mathbf{X}} \hat{\mathbf{X}}^T \mathbf{D}^{-1/2} = \tilde{\mathbf{X}} \tilde{\mathbf{X}}^T \quad (16)$$

where $\tilde{\mathbf{X}} = \mathbf{D}^{-1/2} \hat{\mathbf{X}}$, $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times h}$ is the transform matrix of $\hat{\mathbf{X}}$. Following [40], the eigendecomposition on \mathbf{E} equals the

singular value factorization (SVD) on $\tilde{\mathbf{X}}$. With respect to $\tilde{\mathbf{X}}$, the SVD can be denoted as

$$\tilde{\mathbf{X}} = \mathbf{U}\Sigma\mathbf{V}^\top \quad (17)$$

where $\Sigma \in \mathbb{R}^{n \times h}$ is the singular value matrix and is diagonal with non-negative real numbers, $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{h \times h}$ are two orthogonal matrices, $\mathbf{U}\mathbf{U}^\top = \mathbf{1}_{(n \times n)}$, and $\mathbf{V}\mathbf{V}^\top = \mathbf{1}_{(h \times h)}$. The matrix Σ can be denoted as

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_h \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (18)$$

where σ_z is an eigenvalue in Σ , $z = 1, 2, \dots, h$. Introducing (17) into (16), we have

$$\mathbf{E} = (\mathbf{U}\Sigma\mathbf{V}^\top)(\mathbf{U}\Sigma\mathbf{V}^\top)^\top = \mathbf{U}\Sigma\Sigma^\top\mathbf{U}^\top = \mathbf{U}\Omega\mathbf{U}^\top \quad (19)$$

where Ω is a diagonal matrix, $\Omega \in \mathbb{R}^{n \times n}$. According to (14) and (19), we can observe that the undirected graph can be cut by keeping the c largest eigenvalues in Ω and, moreover, the column vectors of \mathbf{U} are the eigenvectors of \mathbf{E} . Since Σ is a rectangular diagonal matrix with non-negative real numbers on the diagonal, the matrix Ω can be written as

$$\Omega = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_h^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}. \quad (20)$$

As a consequence, our approach can use the c largest eigenvalues in Ω to obtain the corresponding eigenvectors $[\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_c]$ from \mathbf{U} , and then these eigenvectors are combined into the assignment matrix \mathbf{B} . Following previous studies in [41], the diagonal elements of Ω are sorted in descending order and, thus, \mathbf{B} can be formed as follows:

$$\mathbf{B} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_c] \xleftarrow{\text{Corresponding}} \left\{ \sigma_g^2 \right\}_{g=1}^c \quad (21)$$

where σ_g^2 is a diagonal element in Ω , $g = 1, 2, \dots, c$. For an SC algorithm, \mathbf{B} can be regarded as the low-dimensional embedding of nodes [27]. Theoretically, the graph cut requires the computational complexity to be $\mathcal{O}(nhc)$. Conventional SC algorithms usually adopt the GK to calculate each element w_{ij} and use eigendecomposition to obtain \mathbf{B} , where the computational complexity is $\mathcal{O}(n^2c)$ [17]. Considering that $h \ll n$, our NSC exhibits reduced computational complexity when compared to conventional SC algorithms.

C. Feature Representation via NSC

To further improve the efficiency, we randomly select k nodes to construct anchors, $k \ll n$ and $k < h$. Meanwhile,

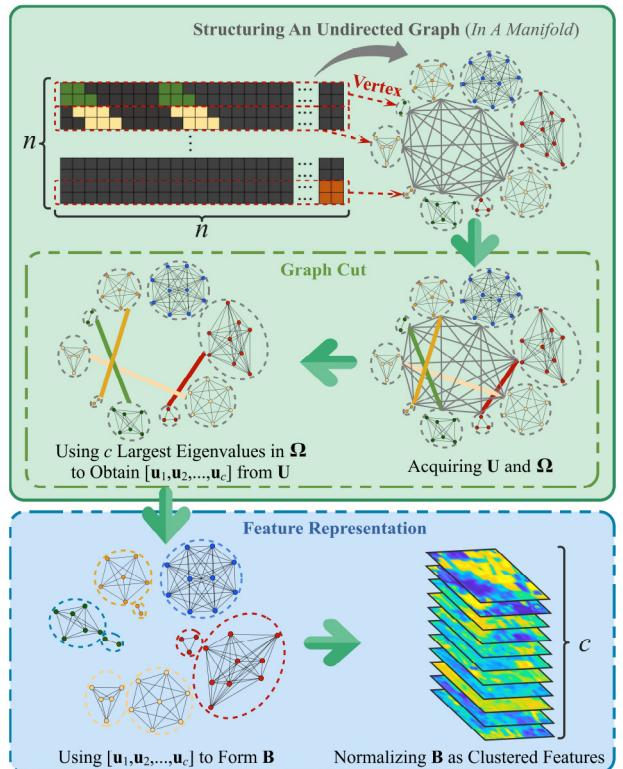


Fig. 3. Illustrations of graph cut and feature representation. The green part is related to the procedure of cutting graph, and the blue part displays the feature representation. By means of structuring the graph and cutting the graph, the node matrix is processed in a manifold. Finally, we obtain the matrix \mathbf{B} as the clustered features.

we still adopt the LK to obtain the node-to-anchor matrix $\mathbf{F} \in \mathbb{R}^{n \times k}$, and the procedure is denoted as

$$\mathbf{F} = \hat{\mathbf{X}}\bar{\mathbf{X}}^\top \Leftarrow f_{pq} = \langle \hat{\mathbf{x}}_p, \bar{\mathbf{x}}_q \rangle \quad (22)$$

where $\bar{\mathbf{X}} \in \mathbb{R}^{k \times h}$ represents the anchor matrix, $\bar{\mathbf{X}} \subset \hat{\mathbf{X}}$, $\bar{\mathbf{x}}_q$ denotes an anchor and is a row vector in $\bar{\mathbf{X}}$, $p = 1, \dots, n$, $q = 1, \dots, k$. We use \mathbf{F} to replace $\hat{\mathbf{X}}$, and thus \mathbf{W} in (15) is replaced as $\mathbf{W} = \mathbf{F}\mathbf{F}^\top$. Meanwhile, we adopt $\tilde{\mathbf{F}} = \mathbf{D}^{-1/2}\mathbf{F}$ to replace $\tilde{\mathbf{X}}$ in (16), where $\tilde{\mathbf{F}}$ is the transformed matrix of \mathbf{F} . After replacing (15) and (16), we implement the SVD on the new $\tilde{\mathbf{X}}$ to obtain \mathbf{B} , using (17) and (21). Owing to the utilization of anchors, the implementation complexity of the SVD is reduced from $\mathcal{O}(nhc)$ to $\mathcal{O}(nkc)$.

Since the SVD might introduce some outliers, all elements of \mathbf{B} are also normalized to the range $[0, 1]$ for enhancing the robustness of our approach. Let us assume that \mathbf{b}_i is a row vector from \mathbf{B} , $\mathbf{B} \equiv [\mathbf{b}_1, \dots, \mathbf{b}_n]^\top$. Similarly, with the definition in (6), the normalization with respect to \mathbf{b} can be written as

$$b_m = \frac{b_m - b_{\min}}{b_{\max} - b_{\min}} \quad (23)$$

where b_m is the m th element in \mathbf{b}_i , and b_{\min} and b_{\max} are the minimum and maximum elements in \mathbf{b}_i , respectively. For the proposed NSC, \mathbf{b}_i is a clustered feature after implementing the normalization. Finally, \mathbf{B} can be regarded as a new feature representation of nodes, called clustered features. Fig. 3 illustrates the whole procedure of feature representation via our

NSC. Moreover, the graph cut in Section III-B is also shown in Fig. 3.

D. NSCKL for Semisupervised HSI Classification

In the framework of the proposed NSCKL, our NSC generates clustered features as the inputs of the KELM. When we establish vertices for the undirected graph, the KIF aggregates the spatial neighbor characteristics from neighboring pixels to update nodes. Among these neighboring pixels, some are labeled and some are unlabeled. Since we use all nodes to construct the adjacency matrix \mathbf{W} and the normalized similarity matrix \mathbf{E} , the clustered features contain the information available from labeled and unlabeled samples. Afterward, all training samples are selected from the cluster features randomly. Therefore, the proposed NSCKL is a semisupervised classification method for processing HSIs.

We assume that the dataset contains ϕ training samples and ψ test samples, and the number of classes is ε . A sample can be represented as $\{\mathbf{b}, \mathbf{y}\}$, where $\mathbf{b} \in \mathbb{R}^{c \times 1}$ represents the sample vector, and $\mathbf{y} \in \mathbb{R}^{\varepsilon \times 1}$ denotes the label vector of the sample. Let λ be the dimensions of \mathbf{y} , and y_λ is an element in \mathbf{y} and can be indicated as

$$y_\lambda = \begin{cases} 1, & \mathbf{b} \in \lambda\text{-th Target Class} \\ 0, & \text{Otherwise} \end{cases} \quad (24)$$

where $\lambda = 1, 2, \dots, \varepsilon$, $\mathbf{y} = [y_1, y_2, \dots, y_\varepsilon]$. All label vectors of training samples form the label matrix $\mathbf{Y} \in \mathbb{R}^{\phi \times \varepsilon}$. Following the kernel expressions in [7] and [23], the output function of the KELM classifier can be indicated as

$$\mathcal{F}(\mathbf{b}_l) = \begin{pmatrix} K(\mathbf{b}_l, \mathbf{b}_1) \\ K(\mathbf{b}_l, \mathbf{b}_2) \\ \vdots \\ K(\mathbf{b}_l, \mathbf{b}_\phi) \end{pmatrix}^\top \left(\frac{\mathbf{I}_\phi}{\Psi} + \mathbf{K} \right)^{-1} \mathbf{Y} \quad (25)$$

where \mathbf{b}_l denotes a test sample, and Ψ is a regularization parameter that can be estimated by using cross-validation [42]. In (25), $\mathbf{I}_\phi \in \mathbb{R}^{\phi \times \phi}$ is also an identity matrix, and $\mathbf{K} \in \mathbb{R}^{\phi \times \phi}$ is the kernel matrix (a symmetric matrix calculated by using the GK). Since a given sample \mathbf{b}_l is associated with pixel \mathbf{x}_l , $\mathcal{F}(\mathbf{b}_l)$ in (25) can be utilized to label pixels for obtaining the final classification result. If the test set contains all clustered features, all pixels in an HSI can be labeled via the KELM model in (25). According to [24], the complexity of the model is $\mathcal{O}(c\phi + \phi\varepsilon)$. The proposed NSCKL needs to use training and test samples, and thus the classification includes a training procedure and a testing procedure. A pseudocode of our NSCKL is given in Algorithm 1.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The effectiveness of the proposed NSCKL is evaluated by using four real HSIs, including the Indian Pines, Salinas Valley, Houston University, and WHU-Hi-HongHu data.¹ The

¹The Indian Pines data and the Salinas Valley data were acquired from <http://www.ehu.eus/ccwintco/index.php/Hyperspectral-Remote-Sensing>. The Houston University 2013 data were downloaded from <http://www.grss-ieee.org/community/technical-committees/data-fusion/2013-ieee-grss-data-fusion-contest/>. The WHU-Hi-HongHu data were provided by Prof. Yanfei Zhong, Wuhan University, in 2020.

Algorithm 1: Pseudocode of the Proposed NSCKL

```

Input: HSI  $\mathbf{X}$ , Label matrix  $\mathbf{Y}$ , Number of clusters  $c$ ,  

        Indexes of training samples  $Index_{train}$ ,  

        Indexes of test samples  $Index_{test}$ .
Output: Labels of test samples.

Step 1. {* — Feature Representation — *}
1. Normalize all samples by using (6).
/* Establishing Vertices for NSC */
repeat
    2. Obtain a sparse vector  $\tilde{\mathbf{x}}$  by using (7) and (8).
    3. Use  $n$  sparse vectors to compose the matrix  $\tilde{\mathbf{X}}$ .
    4. Update the node matrix  $\hat{\mathbf{X}}$  by using (9).
until Reach the stopping condition of (11).
/* Graph Cut for NSC */
5. Obtain the node-to-anchor matrix  $\mathbf{F}$  by using (22).
6. Use  $\mathbf{F}$  to replace  $\hat{\mathbf{X}}$ , then acquire the adjacency matrix  $\mathbf{W}$  by using (15).
7. Use  $\mathbf{W}$  to acquire the degree matrix  $\mathbf{D}$  via (2).
8. Use  $\mathbf{D}$  and  $\mathbf{F}$  to obtain  $\mathbf{F}$  for replacing  $\tilde{\mathbf{X}}$  in (16).
9. Obtain a matrix  $\mathbf{B}$  by using (17) and (21).
10. Normalize  $\mathbf{B}$  as clustered features via (23).

Step 2. {* — Pixel Classification — *}
/* Training Procedure */
11. Acquire training samples from  $\mathbf{B}$  by  $Index_{train}$ .
12. Use training samples to train the KELM.
13. Estimate  $\Psi$  in (25) by using cross-validation.
/* Testing Procedure */
14. Acquire test samples from  $\mathbf{B}$  by  $Index_{test}$ .
15. Predict labels of test samples by using the trained KELM model in (25).

```

first three HSIs were obtained by satellite-carried sensors, while the WHU-Hi-HongHu HSI was acquired by unmanned aerial vehicle (UAV). The experiments with the WHU-Hi-HongHu data can further demonstrate the practicability of our NSCKL. Additionally, the parameters of our NSCKL are evaluated by considering different scenarios.

The competitiveness of our NSCKL is evaluated by comparing it with other SOTA HSI classification methods. These SOTA methods include the aforementioned KELM [23], RKSVM [6], SSN [7], LCMR [5], GLLR-CKSVM [11], S²GCN [12], and SDFL-FC [13]. Furthermore, to embody the contribution of our approach on HSI classification, we perform a complementary test which compares our NSCKL with an unsupervised SC-based method. The unsupervised SC-based method combines fast SC with anchor graph (FSCAG) for implementing an unsupervised HSI classification framework [17]. All the methods mentioned above (with the exception of KELM, that only considers spectral information) consider both the spatial and the spectral information contained in the HSIs. The RKSVM, SSN, and LCMR are three classic SSC approaches, while the GLLR-CKSVM, S²GCN, and SDFL-FC are three new SSC methods. The S²GCN and SDFL-FC use deep neural networks for implementing semisupervised HSI classification. The SDFL-FC uses a convolutional neural network (CNN) to implement HSI classification, while the S²GCN employs GCNs. The RKSVM, LCMR, and GLLR-CKSVM adopt SVM as a classifier, while the SSN uses the KELM. In this article, following [12], we use 200 epochs to train the network models of the S²GCN and the SDFL-FC, and the other methods use the default settings provided by the published codes.

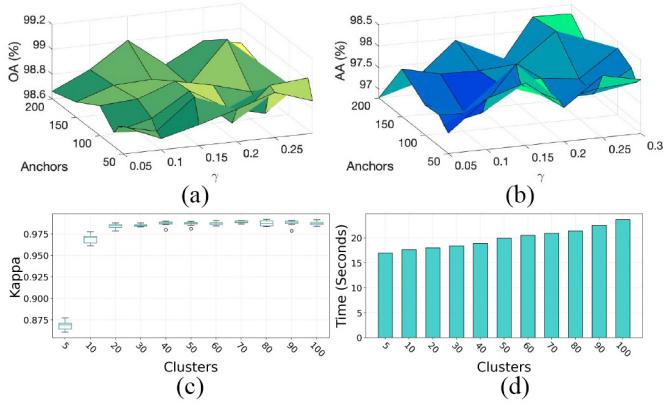


Fig. 4. Top: OAs and AAs from NSCKL under different parameter settings (k and γ), where the number of clusters is set to $c = 50$. (a) OAs. (b) AAs. Bottom: Changes of Kappa coefficient and computation time, along with clusters, where k and γ are set as $k = 100$ and $\gamma = 0.2$, respectively. (c) Kappa coefficient. (d) Computation time.

To evaluate the classification performance of our NSCKL, we use four metrics: 1) class accuracy (CA); 2) overall accuracy (OA); 3) average accuracy (AA); and 4) Kappa coefficient. Moreover, the computation time is considered in the comparison. In our experiments, all classification methods are run ten times and we report mean values and standard deviations.

A. Experiments With Indian Pines Data

1) *Analysis of Parameters*: The Indian Pines dataset was obtained by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), and it covers an agricultural area in northwest Indiana, USA. The image contains 145×145 pixels with 200 bands, ranging from 0.2 to $2.4 \mu\text{m}$. Moreover, the Indian Pines data contains ground-truth information that comprises 16 classes for evaluating classification performances.

In the experiments with the Indian Pines data, we randomly select 10% pixels in each class (utilizing the ground truth) to acquire the training samples of each class. To evaluate the effect of parameters on performance, our NSCKL was implemented using different parameter settings. The proposed NSCKL involves three user-defined parameters, including γ , k , and c . Considering that $k \leq h$, the value of k is restrained to $[50, 100, 150, 200]$. The parameter γ is set to $[0.05, 0.1, 0.15, 0.2, 0.25, 0.3]$. Fig. 4(a) and (b) shows the accuracy comparisons for various values of k and γ . We can observe that the results are insensitive to variations of k and γ . In subsequent experiments, k and γ are set to $k = 100$ and $\gamma = 0.2$, respectively. The number of clusters is an important parameter for our NSCKL. Fig. 4(c) displays the changes of the Kappa coefficient along with the number of clusters. We can observe that the changes are very small if $c > 30$. Fig. 4(d) shows the changes in computation time along with the number of clusters. Fig. 4(c) and (d) reveals that the accuracy does not significantly rise if $c > 30$, while the computation time continues to increase. To balance accuracy and efficiency, we set $c = 50$ in the following experiments. Our NSCKL can obtain clustered features \mathbf{B} . For illustrative purposes, \mathbf{B} is

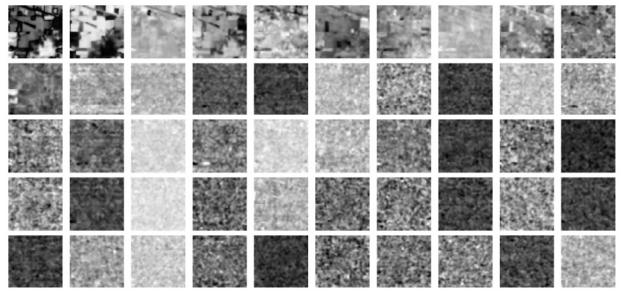


Fig. 5. Visualization of clustered features from the Indian Pines data, $k = 100$, $\gamma = 0.2$, and $c = 50$.

TABLE II
COMPARISON BETWEEN UNSUPERVISED AND SEMISUPERVISED SC-BASED METHODS. THE NUMBER OF ANCHORS IS SET TO DIFFERENT VALUES. ACCURACIES (%) AND COMPUTATION TIME (SECONDS) ARE REPORTED

Anchors	Unsupervised (FSCAG)				
	10	20	50	100	200
OA	35.96 ± 2.52	37.26 ± 3.15	37.82 ± 4.09	41.25 ± 1.88	41.63 ± 2.08
AA	22.14 ± 5.23	32.53 ± 2.02	42.73 ± 2.59	50.49 ± 3.13	50.71 ± 1.92
Kappa	26.40 ± 2.02	29.21 ± 1.37	30.22 ± 2.27	35.09 ± 1.48	35.18 ± 1.97
Time	17.56 ± 3.35	31.27 ± 4.45	36.07 ± 7.34	56.50 ± 6.98	62.93 ± 7.08
Anchors	Semisupervised (NSCKL)				
	10	20	50	100	200
OA	95.79 ± 0.37	97.42 ± 0.43	98.89 ± 0.30	99.02 ± 0.36	99.02 ± 0.15
AA	96.68 ± 0.78	96.93 ± 1.03	97.94 ± 0.76	98.59 ± 0.97	98.61 ± 1.11
Kappa	96.34 ± 0.42	97.06 ± 0.49	98.73 ± 0.41	98.81 ± 0.34	98.89 ± 0.17
Time	17.64 ± 2.41	17.98 ± 3.32	19.66 ± 2.54	23.52 ± 5.07	30.71 ± 4.23

converted to a 3-D cube with size $145 \times 145 \times c$. Fig. 5 visualizes the clustered features \mathbf{B} obtained by our NSCKL. In Fig. 5, it can be observed that many relevant characteristics have been aggregated into the set of clustered features, especially the plots in the first two rows.

2) *Analysis of Semisupervised Training*: The FSCAG is a classical unsupervised SC-based approach and it also adopts anchors to improve the efficiency of HSI classification. To confirm the significance of the semisupervised training process, we compare our NSCKL with the FSCAG. Following [17], we set the parameters involved in FSCAG. Table II provides a performance comparison, including classification accuracies and computation time. The comparisons between Kappa coefficients are shown at the top of Fig. 6, which indicates our semisupervised training significantly improves classification accuracy. Moreover, Fig. 6 also shows the classification maps obtained by FSCAG and our NSCKL. Fig. 6(a)–(d) provides an RGB composition, the ground truth, the unsupervised classification map, and the semisupervised classification map. In Fig. 6, we can observe that our semisupervised approach significantly improves the classification performance compared with the unsupervised approach.

3) *Performance Comparison*: In this section, we compare the classification results from NSCKL to those obtained by other SOTA methods. The ground truth of Indian Pines provides a way for evaluating these classification methods quantitatively. Table III lists the classification performances obtained from the aforementioned approaches. We can observe that our NSCKL significantly improves classification accuracy compared with the single KELM. The improvement indicates that the clustered features obtained by our newly proposed

TABLE III
CLASSIFICATION ACCURACIES (%) OF DIFFERENT APPROACHES, OBTAINED BY USING THE INDIAN PINES DATA.
CAS ARE LISTED FOR COMPARISON. BEST RESULTS ARE IN BOLD

	Class Name	Training Samples	Test Samples	Total	KELM	RKSVM	SSN	LCMR	GLLR-CKSVM	S^2 GCN	SDFL-FC	NSCKL
CA	Alfalfa	6	48	54	69.79±2.41	81.45±9.13	92.50±7.29	97.26±1.17	92.97±4.21	99.40 ±5.04	93.03±6.72	97.25±4.25
	Corn-notill	144	1290	1434	82.90±1.97	96.62±0.64	97.64±1.09	96.49±1.06	97.42±2.24	91.17±2.17	97.63±1.98	98.73 ±0.72
	Corn-min till	84	750	834	68.37±4.67	95.77±3.02	98.65±1.29	96.06±2.44	97.23±3.40	84.87±4.65	93.26±3.47	98.52±1.37
	Corn	24	210	234	62.38±4.14	91.95±4.26	98.12±2.43	97.09±1.56	96.58±1.98	90.09±3.02	89.78±2.38	99.14 ±1.60
	Grass-pasture	50	447	497	91.86±2.39	93.89±3.96	97.24±2.06	97.64±1.30	97.39±2.72	97.63±4.05	95.34±3.84	97.89 ±2.16
	Grass-trees	75	627	747	97.20±2.39	98.51±1.04	99.44±0.27	99.61±0.21	98.52±0.53	98.93±2.25	98.69±2.97	99.67 ±0.43
	Grass-pasture-mowed	3	23	26	79.13±6.74	86.52±10.33	97.39±4.67	98.95±0.47	92.77±1.92	99.36±2.01	85.24±9.46	99.57 ±1.37
	Hay-win drowed	49	440	489	98.89±0.66	97.59±1.71	100.00 ±0.00	100.00 ±0.00	99.15±1.10	100.00 ±0.00	99.63±0.12	100.00 ±0.00
	Oats	2	18	20	74.71±18.19	67.22±24.63	88.33±13.72	85.29±9.30	86.62±10.39	91.21±15.32	85.40±14.29	92.35 ±9.90
	Soybean-notill	97	871	968	80.16±2.49	94.63±1.98	97.23±1.21	94.00±0.98	95.20±2.05	93.14±4.13	94.19±3.36	98.00 ±1.57
	Soybean-min till	247	2221	2468	86.89±1.17	97.04±1.31	98.74±0.39	98.45±0.65	98.41±1.54	84.78±1.87	98.05±2.13	99.54 ±0.41
	Soybean-clean	62	552	614	82.19±2.37	94.38±2.59	98.35 ±1.09	96.04±1.64	95.54±0.59	90.23±5.50	97.19±6.24	98.12±1.18
	Wheat	22	190	212	98.84±0.69	99.47±0.49	99.05±0.81	99.67±0.28	99.03±0.99	98.84 ±3.01	99.68 ±2.83	99.01±0.94
	Woods	130	1164	1294	95.98±1.10	97.82±1.10	99.71±0.57	99.74±0.04	98.86±0.52	99.02±1.98	99.74±0.98	99.82 ±0.32
	Buildings-Grass-Tress-Drives	38	342	380	62.46±4.85	93.59±3.44	99.02±1.07	99.97±0.09	98.78±1.37	89.16±5.74	89.65±5.08	99.88 ±0.28
	Stone-steel-towers	10	85	95	78.24±8.06	93.52±12.46	95.41±3.65	92.77±5.32	94.73±3.03	96.43±1.79	96.70±2.24	98.53 ±1.44
OA					85.21±0.57	96.19±0.58	98.57±0.32	97.71±0.40	97.24±0.49	91.49±0.73	96.85±0.90	99.06 ±0.35
AA					81.87±1.75	92.50±1.50	97.30±1.33	96.81±0.69	96.20±1.69	94.02±3.52	94.57±3.17	98.50 ±0.88
Kappa					83.09±0.66	95.66±0.66	98.37±0.36	97.39±0.46	96.93±0.53	90.95±0.95	96.38±0.93	98.89 ±0.39
Time (seconds)					3.63±1.94	82.17±9.61	113.08±7.52	2.03±0.04	35.62±5.61	319.86±8.39	537.83±7.96	19.57±5.27

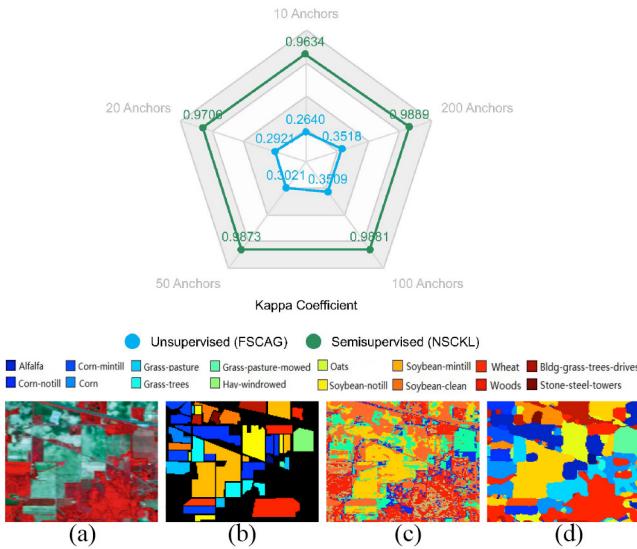


Fig. 6. Top: Kappa coefficients from the unsupervised and semisupervised SC approaches. Bottom: SC-based classification on the Indian Pines data, where pixels are labeled by implementing different approaches. (a) RGB composition. (b) Ground truth. (c) Unsupervised classification map (FSCAG). (d) Semisupervised classification map (NSCKL).

NSC algorithm can significantly enhance the classification performance. In Table III, our NSCKL obtains the best CAS in 12 out of 16 classes. As a result, the proposed NSCKL achieves the best results compared with other SOTA methods. Moreover, we can see that the computation time of our NSCKL is significantly lower than that of the DL-based methods.

Fig. 7 reports the classification maps obtained by different approaches. Additionally, we can observe that the classification map of our NSCKL is smoother than others, and the classification map of the original KELM contains salt and pepper noise. Comparing Fig. 7 with Fig. 6(b), it can be seen that the classification map obtained by our NSCKL is the most similar to the ground truth. The results in Table III and Figs. 6 and 7 demonstrate that the proposed NSCKL is highly competitive when processing HSIs.

B. Experiment With Salinas Valley Data

The Salinas Valley dataset was acquired by the AVIRIS over the Salinas Valley, CA, USA. The image contains 512×217

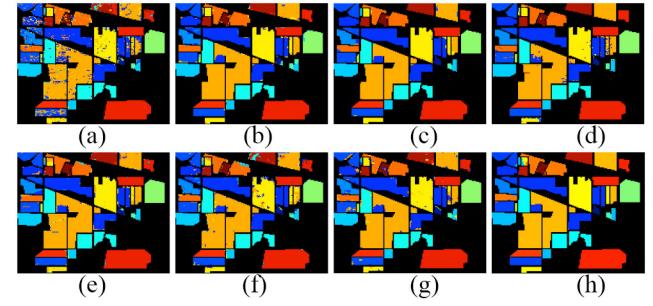


Fig. 7. Comparison of classification results on the Indian Pines data. These results are obtained from different approaches. (a) KELM. (b) RKSVM. (c) SSN. (d) LCMR. (e) GLLR-CKSVM. (f) S^2 GCN. (g) SDFL-FC. (h) NSCKL.

pixels with 204 bands. The spatial resolution is 3.7-m spatial resolution and the image contains 16 land-over classes. Moreover, 54 049 labeled samples are available as the ground truth of the dataset.

To evaluate the performance of the NSCKL more comprehensively, we design a new comparison in which the number of training samples is changed. Table IV gives the classification accuracies obtained by the aforementioned classification approaches using the Salinas Valley data. In the comparison, we randomly select 1%, 3%, 5%, and 10% pixels from each ground-truth class to acquire the training samples. In Table IV, we can observe that our NSCKL achieves the best classification accuracies, regardless of the number of training samples used. Moreover, it can be seen that the OAs, AAs, and Kappa coefficients from our NSCKL are higher than those obtained by other methods in each considered scenario.

The classification maps obtained by different algorithms are illustrated in Fig. 8. As it can be seen in Fig. 8, the classification performance of our NSCKL is better than that of the other classification methods. Specifically, the classification map obtained by the proposed NSCKL is smoother than the classification maps obtained by other methods; meanwhile, mislabels rarely appear in each class when using the proposed NSCKL. Furthermore, in Fig. 8, we can observe that the classification map from our NSCKL is very similar to the ground-truth map. Notwithstanding the fact that we employ a framework without deep structure, the classification

TABLE IV
CLASSIFICATION ACCURACIES (%) OF DIFFERENT APPROACHES FOR THE SALINAS IMAGE. THE TRAINING SAMPLES ARE OBTAINED BY RANDOMLY SELECTING A PERCENTAGE OF SAMPLES IN THE GROUND TRUTH. BEST RESULTS ARE IN BOLD

Ratio (Training Samples)	1 %	3 %	5 %	10 %	
OA	KELM	90.55±0.93	92.56±0.32	93.42±0.23	94.32±0.17
	RKSVM	94.86±1.04	98.35±0.29	99.01±0.19	99.55±0.04
	SSN	95.25±0.44	96.66±0.34	97.43±0.23	98.59±0.14
	LCMR	98.43±0.37	99.33±0.09	99.53±0.05	99.64±0.04
	GLLR-CKSVM	95.09±1.95	97.41±1.52	99.02±0.49	99.35±0.15
	S ² GCN	93.62±3.27	95.25±2.00	97.65±1.25	98.82±0.86
	SDFL-FC	96.18±3.06	98.17±3.01	98.85±0.87	99.46±0.18
AA	NSCKL	99.29±0.24	99.79±0.09	99.85±0.06	99.95±0.02
	KELM	94.15±0.73	95.66±0.29	96.41±0.32	97.11±0.15
	RKSVM	95.67±0.82	98.57±0.25	99.04±0.11	99.57±0.10
	SSN	96.28±0.52	97.39±0.22	98.02±0.16	98.90±0.07
	LCMR	98.60±0.56	99.34±0.08	99.51±0.06	99.63±0.05
	GLLR-CKSVM	94.68±2.12	98.10±0.84	98.62±0.25	99.00±0.22
	S ² GCN	94.35±3.40	96.83±1.61	98.49±0.90	99.03±0.35
Kappa	SDFL-FC	95.15±1.98	97.27±0.95	98.37±0.93	99.14±0.17
	NSCKL	98.99±0.32	99.61±0.21	99.73±0.11	99.93±0.08
	KELM	89.46±10.31	91.71±0.35	92.67±0.25	93.67±0.19
	RKSVM	94.28±1.16	98.16±0.32	98.88±0.27	99.50±0.05
	SSN	94.71±0.49	96.29±0.38	97.14±0.23	98.33±0.16
	LCMR	98.22±0.42	99.26±0.15	99.47±0.06	99.67±0.02
	GLLR-CKSVM	94.31±1.78	97.93±0.92	98.14±0.30	99.16±0.09
	S ² GCN	92.46±2.52	96.04±1.16	97.82±0.79	98.75±0.43
	SDFL-FC	95.87±1.92	97.98±1.35	98.51±0.69	99.28±0.28
	NSCKL	99.21±0.27	99.76±0.11	99.83±0.06	99.94±0.04

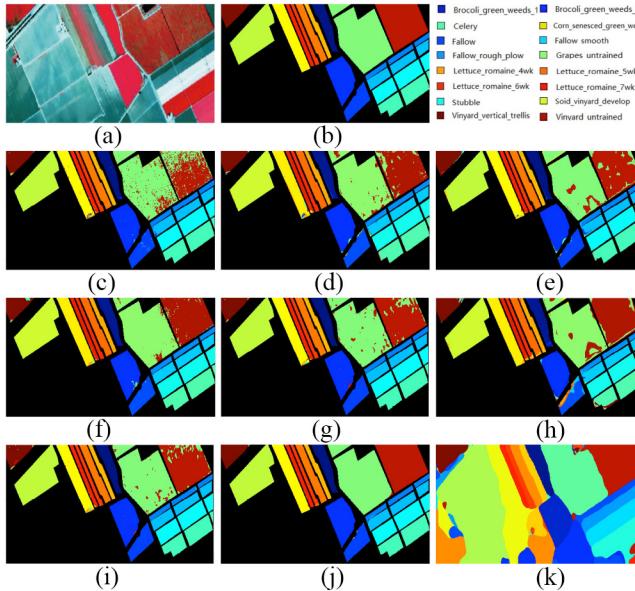


Fig. 8. Classification results obtained by different approaches for the Salinas Valley data. (a) RGB composition. (b) Ground truth. (c) KELM. (d) RKSVM. (e) SSN. (f) LCMR. (g) GLLR-CKSVM. (h) S²GCN. (i) SDFL-FC. (j) NSCKL. (k) NSCKL labeling all pixels.

performance of our NSCKL is quite remarkable. The reason is that the newly developed NSC can explore more hidden correlations from both contiguous and noncontiguous pixels. Table IV and Fig. 8 further verify the effectiveness of the proposed NSCKL.

C. Experiment With WHU-Hi-HongHu Data

The WHU-Hi-HongHu dataset was obtained by the Headwall Nano-Hyperspec sensor mounted on a UAV platform, and data were acquired over Honghu City, Hubei Province, China. The region comprises 22 land-cover types when the UAV passed over it in 2017, and Wuhan University provided the ground truth for testing classification approaches.

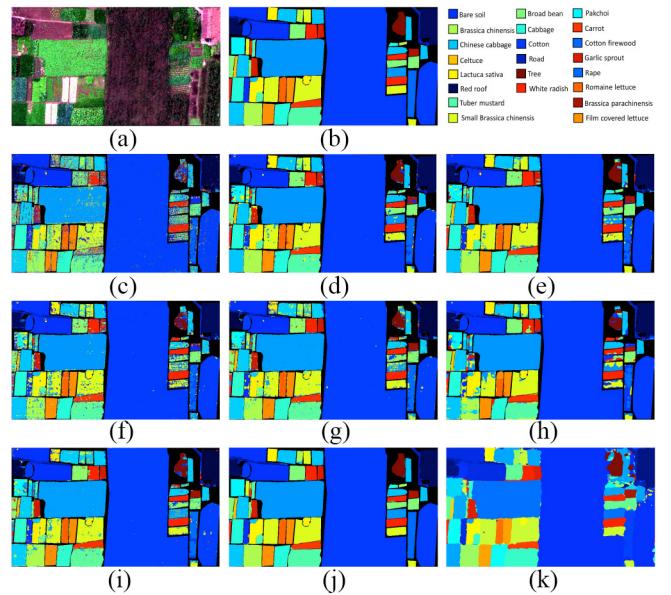


Fig. 9. Classification results obtained by different approaches for the WHU-Hi-HongHu data. (a) RGB composition. (b) Ground truth. (c) KELM. (d) RKSVM. (e) SSN. (f) LCMR. (g) GLLR-CKSVM. (h) S²GCN. (i) SDFL-FC. (j) NSCKL. (k) NSCKL labeling all pixels.

The HSI contains 940×475 pixels with 270 bands, and the spectral range is from 0.4 to $1.0 \mu\text{m}$ [43].

In this test, we use the ground truth to randomly select 1% pixels in each class to acquire training samples. Table V lists the classification accuracies from different methods. In Table V, we can observe that the NSCKL obtains better classification results than other SOTA methods. For comparative purposes, Fig. 9 illustrates the classification maps obtained by using different methods. In Fig. 9, it can be seen that the NSCKL obtains better classification performances than other methods. Furthermore, in Fig. 9, we observe that the classification map obtained by our NSCKL is smoother than those provided by other methods and preserves the boundary details of classes well. The results in Table V and Fig. 9 further demonstrate the effectiveness of our NSCKL.

D. Experiment With Houston University Data

The Houston University dataset was collected by the ITRES CASI-1500 sensor from the University of Houston, Houston, TX, USA. As a contest dataset, it was released by the National Center for Airborne Laser Mapping (NCALM) in 2013. The image contains 349×1905 pixels with 144 bands, and the spectral range is from 0.36 to $1.05 \mu\text{m}$. The scenario comprises 15 land-cover classes.

In this experiment, the training and test samples are fixed and provided by the NCALM. The ground truth of the Houston University data is therefore composed of training and test samples. Since the training samples are not randomly selected, this experiment does not involve multiple runs. Table VI lists the classification accuracies from different methods and reveals that our NSCKL can still obtain satisfactory results even in the presence of poor-quality training samples. After

TABLE V
CLASSIFICATION ACCURACIES (%) OBTAINED DIFFERENT APPROACHES FOR THE WHU-HI-HONGHU DATA.
CAS ARE LISTED FOR COMPARISON. BEST RESULTS ARE IN BOLD

	Class Name	Training Samples	Test Samples	Total	KELM	RKSVM	SSN	LCMR	GLLR-CKSVM	S^2 GCN	SDFL-FC	NSCKL
CA	Red-roof	141	13900	14041	91.20 \pm 1.56	93.14 \pm 1.78	97.94 \pm 0.79	96.42 \pm 0.71	97.42 \pm 0.68	98.21 \pm 0.85	97.36 \pm 0.91	98.64 \pm 0.47
	Road	36	3476	3512	72.98 \pm 5.40	78.40 \pm 4.87	83.98 \pm 7.40	85.45 \pm 3.22	80.83 \pm 1.89	81.38 \pm 8.42	81.42 \pm 7.36	86.38 \pm 3.40
	Bare-soil	219	21602	21821	89.91 \pm 0.88	92.07 \pm 0.89	96.55 \pm 0.91	97.76 \pm 0.25	97.92 \pm 0.45	90.66 \pm 3.39	93.50 \pm 3.53	98.06 \pm 0.65
	Cotton	1633	161625	163285	99.49 \pm 0.16	98.73 \pm 0.27	98.66 \pm 0.14	95.23 \pm 0.58	98.99 \pm 0.09	98.82 \pm 2.47	97.44 \pm 2.16	99.95 \pm 0.03
	Cotton-firewood	63	6155	6218	57.20 \pm 6.99	80.43 \pm 8.23	87.96 \pm 2.43	94.45 \pm 0.61	96.71 \pm 1.15	98.41 \pm 1.15	99.00 \pm 0.93	98.95 \pm 0.99
	Rape	446	44111	44557	88.57 \pm 1.00	88.96 \pm 1.05	95.58 \pm 0.35	99.01 \pm 0.09	88.62 \pm 3.31	89.04 \pm 8.20	98.51 \pm 1.14	99.86 \pm 0.09
	Chinese-cabbage	242	23861	24103	84.35 \pm 2.45	88.48 \pm 1.67	93.26 \pm 0.47	95.14 \pm 0.49	98.10 \pm 0.49	97.51 \pm 2.30	92.73 \pm 3.72	98.58 \pm 0.37
	Pakchoi	41	4013	4054	54.40 \pm 1.89	98.25 \pm 4.20	83.03 \pm 3.16	98.38 \pm 0.52	88.97 \pm 4.29	85.92 \pm 5.57	92.95 \pm 3.03	99.23 \pm 0.74
	Cabbage	109	10710	10819	86.38 \pm 1.80	95.96 \pm 1.19	96.79 \pm 6.51	96.89 \pm 0.70	95.98 \pm 0.80	96.41 \pm 3.02	96.28 \pm 1.63	99.14 \pm 0.62
	Tuber-mustard	124	12270	12394	54.35 \pm 3.92	74.45 \pm 2.24	88.92 \pm 2.61	93.39 \pm 0.24	96.27 \pm 0.64	86.62 \pm 5.45	89.90 \pm 6.18	97.76 \pm 0.54
	Brassica-parachinensis	111	10904	11015	53.46 \pm 2.21	82.16 \pm 3.16	86.82 \pm 2.02	97.71 \pm 0.54	95.97 \pm 0.48	85.39 \pm 6.73	96.22 \pm 2.20	98.37 \pm 0.77
	Brassica-chinensis	90	8864	8954	65.33 \pm 2.20	90.29 \pm 1.95	85.25 \pm 2.90	95.28 \pm 0.69	97.40 \pm 0.61	85.91 \pm 7.04	97.25 \pm 1.84	97.36 \pm 1.07
	Small-brassica-chinensis	226	22281	22507	72.76 \pm 2.19	86.80 \pm 2.36	89.07 \pm 2.39	89.82 \pm 4.46	92.43 \pm 3.46	87.25 \pm 4.32	93.87 \pm 1.13	97.98 \pm 0.78
	Lactuca-sativa	74	7282	7356	72.34 \pm 2.61	87.37 \pm 2.30	89.04 \pm 3.01	95.90 \pm 0.53	87.88 \pm 6.44	85.84 \pm 5.52	94.45 \pm 1.42	95.60 \pm 0.85
	Celtuce	11	991	1002	52.45 \pm 7.56	70.46 \pm 7.42	83.31 \pm 6.15	82.98 \pm 7.42	79.91 \pm 6.37	88.79 \pm 8.42	85.70 \pm 5.65	93.15 \pm 4.54
	Film-covered-lettuce	73	7189	7262	81.63 \pm 3.72	94.39 \pm 0.93	94.93 \pm 1.41	99.10 \pm 0.29	99.05 \pm 0.55	99.14 \pm 0.31	99.20 \pm 0.30	99.31 \pm 0.51
	Romaine-lettuce	31	2979	3010	64.53 \pm 4.60	86.50 \pm 3.38	89.29 \pm 2.41	95.79 \pm 3.78	90.36 \pm 5.27	87.23 \pm 6.15	91.82 \pm 3.72	97.91 \pm 1.87
	Carrot	33	3184	3217	59.14 \pm 6.85	66.54 \pm 5.19	94.75 \pm 1.75	89.99 \pm 6.32	85.90 \pm 7.16	95.36 \pm 7.20	95.51 \pm 3.64	96.14 \pm 2.15
	White-radish	88	8624	8712	61.94 \pm 2.93	86.62 \pm 3.39	90.86 \pm 1.55	92.91 \pm 3.11	93.34 \pm 2.05	90.46 \pm 4.23	95.04 \pm 2.21	94.90 \pm 2.08
	Garlic-sprout	35	3451	3486	70.20 \pm 5.58	82.55 \pm 4.48	92.23 \pm 2.24	95.52 \pm 0.93	96.52 \pm 0.93	83.29 \pm 5.52	92.36 \pm 4.03	97.43 \pm 1.40
	Broad-bean	14	1314	1328	61.78 \pm 1.03	78.06 \pm 1.28	92.61 \pm 1.13	99.20 \pm 1.08	99.11 \pm 0.31	82.85 \pm 4.97	95.89 \pm 3.17	99.57 \pm 0.27
	Tree	41	3999	4040	53.39 \pm 4.94	88.14 \pm 3.32	88.14 \pm 3.77	94.41 \pm 2.25	91.94 \pm 2.28	92.20 \pm 3.53	92.42 \pm 4.07	98.49 \pm 1.75
	OA				81.60 \pm 0.10	89.73 \pm 0.13	93.76 \pm 0.09	96.01 \pm 0.09	95.37 \pm 0.12	89.62 \pm 0.15	95.89 \pm 0.14	98.93 \pm 0.08
	AA				70.35 \pm 0.92	85.85 \pm 0.81	90.86 \pm 0.24	94.57 \pm 0.55	93.16 \pm 0.75	90.30 \pm 0.97	94.04 \pm 0.92	97.40 \pm 0.41
	Kappa				77.85 \pm 0.24	87.42 \pm 0.27	92.46 \pm 0.31	95.67 \pm 0.20	94.35 \pm 0.43	88.45 \pm 0.51	95.49 \pm 0.53	98.64 \pm 0.11

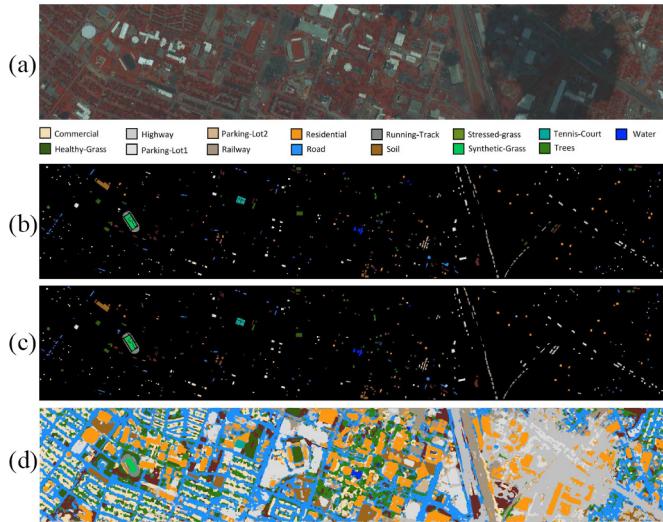


Fig. 10. Classification performance on the Houston University data. (a) RGB composition. (b) Ground truth. (c) NSCKL. (d) NSCKL labeling all pixels.

comparing Fig. 10(b) and (c), we can observe that the results from our NSCKL are very close to the ground truth. Additionally, a comparison between Fig. 10(a) and (d) reveals that our NSCKL accurately preserves object details and boundary information.

V. CONCLUSION AND FUTURE LINES

In HSI classification, it is quite challenging to capture correlations from noncontiguous regions. SC, a branch of graph theory, provides some strategies to deal with this problem. Although SC has been used for processing HSIs, existing approaches are still restricted to unsupervised classification, which cannot achieve the high accuracies needed in land cover/use applications. This article proposes a new SSC method, called NSCKL, which can solve the aforementioned problems to significantly improve the classification performance. Furthermore, an important contribution of our

TABLE VI
CLASSIFICATION ACCURACIES (%) OBTAINED BY DIFFERENT APPROACHES FOR THE HOUSTON UNIVERSITY DATA. CAS ARE LISTED FOR COMPARISON. BEST RESULTS ARE IN BOLD

	Class Name	Training Samples	Test Samples	GLLR-CKSVM	S^2 GCN	SDFL-FC	NSCKL
CA	Healthy-Grass	198	1053	80.20	77.93	82.04	83.10
	Stressed-grass	190	1064	82.81	80.29	84.17	84.31
	Synthetic-Grass	192	505	89.86	93.15	90.30	
	Trees	188	1056	86.42	93.05	85.99	90.25
	Soil	186	1056	99.26	98.41	100.00	100.00
	Water	182	143	81.54	85.77	79.38	79.72
	Residential	196	1072	65.94	63.68	75.32	74.16
	Commercial	191	1053	78.84	57.90	84.27	88.60
	Road	193	1059	81.35	78.64	80.98	84.61
	Highway	191	1036	73.42	69.95	72.01	76.74
	Railway	181	1054	83.95	81.50	85.39	87.19
	Parking-Lot1	192	1041	93.67	90.86	95.10	94.62
	Parking-Lot2	184	285	86.49	81.92	79.90	86.67
	Tennis Court	181	247	98.05	100.00	98.51	100.00
	Running Track	187	473	97.16	96.08	97.29	98.73
	OA			84.65	81.63	85.80	87.19
	AA			85.51	83.06	86.23	87.93
	Kappa			83.17	79.63	84.55	86.09

work is the use of SC to implement semisupervised classification. We first design a KIF to establish the vertices of the undirected graph, which assigns the initial correlation by using local spatial information. Afterward, our newly developed NSC aggregates correlations from contiguous and noncontiguous pixels. Finally, the KELM is employed by the semisupervised framework to achieve pixel-level classification. This work provides a new theoretical foundation for the development of SSC. However, in the current implementation, the computation of clustered features is nondifferentiable, resulting in the limitation that the NSC cannot be directly utilized to construct deep networks. Our future work will investigate differentiable schemes to achieve the graph cut in order to further aggregate hidden correlations hierarchically.

REFERENCES

- [1] D. Hong, N. Yokoya, J. Chanussot, J. Xu, and X. X. Zhu, “Joint and progressive subspace analysis (JPSA) with spatial-spectral manifold alignment for semisupervised hyperspectral dimensionality reduction,” *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3602–3615, Jul. 2021.

- [2] Y. Dong, W. Shi, B. Du, X. Hu, and L. Zhang, "Asymmetric weighted logistic metric learning for hyperspectral target detection," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 11093–11106, Oct. 2022, doi: [10.1109/TCYB.2021.3070909](https://doi.org/10.1109/TCYB.2021.3070909).
- [3] Y. Su, X. Xu, J. Li, H. Qi, P. Gamba, and A. Plaza, "Deep autoencoders with multitask learning for bilinear hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8615–8629, Oct. 2021.
- [4] Q. Zhu et al., "A spectral-spatial-dependent global learning framework for insufficient and imbalanced hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 11709–11723, Nov. 2022, doi: [10.1109/TCYB.2021.3070577](https://doi.org/10.1109/TCYB.2021.3070577).
- [5] L. Fang, N. He, S. Li, A. J. Plaza, and J. Plaza, "A new spatial-spectral feature extraction method for hyperspectral images using local covariance matrix representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3534–3546, Jun. 2018.
- [6] J. Peng, Y. Zhou, and C. L. P. Chen, "Region-kernel-based support vector machines for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 9, pp. 4810–4824, Sep. 2015.
- [7] Y. Zhou and Y. Wei, "Learning hierarchical spectral-spatial features for hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1667–1678, Jul. 2016.
- [8] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.
- [9] Y. Xu, Z. Wu, and Z. Wei, "Markov random field with homogeneous areas priors for hyperspectral image classification," in *Proc. IEEE Geosci. Remote Sens. Symp.*, Nov. 2014, pp. 3426–3429.
- [10] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.
- [11] Z. Feng, S. Yang, M. Wang, and L. Jiao, "Learning dual geometric low-rank structure for semisupervised hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 346–358, Jan. 2021.
- [12] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral-spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, Feb. 2019.
- [13] Y. Cao, Y. Wang, J. Peng, C. Qiu, L. Ding, and X. X. Zhu, "SDFL-FC: Semisupervised deep feature learning with feature consistency for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 12, pp. 10488–10502, Dec. 2021.
- [14] U. von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, Jul. 2007.
- [15] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, Aug. 2015.
- [16] K. Taşdemir, "Vector quantization based approximate spectral clustering of large datasets," *Pattern Recognit.*, vol. 45, no. 8, pp. 3034–3044, Aug. 2012.
- [17] R. Wang, F. Nie, and W. Yu, "Fast spectral clustering with anchor graph for large hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2003–2007, Nov. 2017.
- [18] Y. Zhao, Y. Yuan, and Q. Wang, "Fast spectral clustering for unsupervised hyperspectral image classification," *Remote Sens.*, vol. 11, no. 4, p. 399, Feb. 2019.
- [19] G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
- [20] H. Zhou, G.-B. Huang, Z. Lin, H. Wang, and Y. C. Soh, "Stacked extreme learning machines," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2013–2025, Sep. 2015.
- [21] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.
- [22] Y. Su, M. Jiang, L. Gao, X. Sun, X. You, and P. Li, "Graph-cut-based collaborative node embeddings for hyperspectral images classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, Jun. 2022, doi: [10.1109/LGRS.2022.3184817](https://doi.org/10.1109/LGRS.2022.3184817).
- [23] M. Pal, A. E. Maxwell, and T. A. Warne, "Kernel-based extreme learning machine for remote-sensing image classification," *Remote Sens. Lett.*, vol. 4, no. 9, pp. 853–862, Jun. 2013.
- [24] C.-M. Wong, C.-M. Vong, P.-K. Wong, and J. Cao, "Kernel-based multilayer extreme learning machines for representation learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 757–762, Mar. 2018.
- [25] Y. Zhou, J. Peng, and C. L. P. Chen, "Extreme learning machine with composite kernels for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2351–2360, Jun. 2015.
- [26] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, Nov. 2011.
- [27] G. Chen, "Scalable spectral clustering with cosine similarity," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, 2018, pp. 314–319.
- [28] H. Gao, Y. Liu, and S. Ji, "Topology-aware graph pooling networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4512–4518, Dec. 2021, doi: [10.1109/TPAMI.2021.3062794](https://doi.org/10.1109/TPAMI.2021.3062794).
- [29] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2001, pp. 849–856.
- [30] 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.
- [31] B. Schlkopf and A. J. Smola, *Learning With Kernels*. Cambridge, MA, USA: MIT Press, 2001.
- [32] Y. Liu, S. Liao, H. Zhang, W. Ren, and W. Wang, "Kernel stability for model selection in kernel-based algorithms," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 5647–5658, Dec. 2021, doi: [10.1109/TCYB.2019.2923824](https://doi.org/10.1109/TCYB.2019.2923824).
- [33] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 480–492, Mar. 2012.
- [34] M. G. Genton, "Classes of kernels for machine learning: A statistics perspective," *J. Mach. Learn. Res.*, vol. 2, pp. 299–312, Jan. 2001.
- [35] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.
- [36] S. Christoff, S. Rene, A. Denis, and B. Tristan, "Kernel-based machine learning for efficient simulations of molecular liquids," *J. Chem. Theory Comput.*, vol. 16, no. 5, pp. 3194–3204, 2020.
- [37] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on Riemannian manifolds with Gaussian rbf kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2464–2477, Dec. 2015.
- [38] M. Jiang, Y. Fang, Y. Su, G. Cai, and G. Han, "Random subspace ensemble with enhanced feature for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1373–1377, Aug. 2020.
- [39] Y. Chen, L. Wu, and M. Zaki, "Iterative deep graph learning for graph neural networks: Better and robust node embeddings," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 19314–19326.
- [40] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Singular Value Decomposition*. Cambridge, U.K.: Cambridge Univ. Press, 1992.
- [41] G. H. Golub and C. F. Van Loan, *The Singular Value Decomposition*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.
- [42] Y. Liu, S. Liao, S. Jiang, L. Ding, H. Lin, and W. Wang, "Fast cross-validation for kernel-based algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1083–1096, May 2020.
- [43] Y. Zhong, X. Hu, C. Luo, X. Wang, J. Zhao, and L. Zhang, "WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H^2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF," *Remote Sens. Environ.*, vol. 250, pp. 1–18, Dec. 2020.



Yuanchao Su (Member, IEEE) received the B.S. and M.Sc. degrees from the Xi'an University of Science and Technology (XUST), Xi'an, China, in 2012 and 2015, respectively, and the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2019.

He is currently a Lecturer and a Master's Supervisor with the Department of Remote Sensing, College of Geomatics, XUST. From 2013 to 2015, he was an Exchange Postgraduate with the Optical Laboratory, Institute of Remote Sensing and Digital

Earth, Chinese Academy of Sciences (CAS), Beijing, China. From 2018 to 2019, he was a Visiting Scholar with the Advanced Imaging and Collaborative Information Processing Group, Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA. In 2019, he joined XUST and built the Hyperspectral Information and Intelligent Computation Group. In recent years, he has presided over several scientific research projects at national and ministerial levels. Since 2021, he has been a Visiting Researcher with the Key Laboratory of Computational Optical Imaging Technology, Aerospace Information Research Institute, CAS. His current research interests include hyperspectral unmixing, hyperspectral image classification, neural network, and DL.

Dr. Su is a member of the IEEE Geoscience and Remote Sensing Society and serves as a Reviewer for many international journals, including the IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, and IEEE GEOSCIENCE AND REMOTE SENSING LETTERS.



Mengying Jiang received the B.S. degree in electronic information engineering from Wuhan Technology and Business University, Wuhan, China, in 2013, and the M.Sc. degree from the School of Electronic Engineering, Guangdong University of Technology, Guangzhou, China, in 2019. She is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China.

Her current research interests include graph neural network, transformer, bioinformatics, and hyperspectral image classification.



Lianru Gao (Senior Member, IEEE) received the B.S. degree in civil engineering from Tsinghua University, Beijing, China, in 2002, and the Ph.D. degree in cartography and geographic information system from the Institute of Remote Sensing Applications, Chinese Academy of Sciences (CAS), Beijing, China, in 2007.

He is currently a Professor with the Key Laboratory of Computational Optical Imaging Technology, Aerospace Information Research Institute, CAS. He also has been a Visiting Scholar

with the University of Extremadura, Caceres, Spain, in 2014, and with Mississippi State University, Starkville, MS, USA, in 2016. In the last ten years, he was the Principal Investigator of ten scientific research projects at national and ministerial levels, including projects by the National Natural Science Foundation of China (2016–2019, 2018–2020, and 2022–2025) and by the National Key Research and Development Program of China (2021–2025). He has authored more than 200 peer-reviewed articles, and more than 120 journal articles are included in Science Citation Index. He has coauthored three academic books and holds 29 National Invention Patents in China. His primary research focuses on hyperspectral image processing and information extraction.

Dr. Gao was a recipient of the Outstanding Science and Technology Achievement Prize of the CAS in 2016. He was supported by the China National Science Fund for Excellent Young Scholars in 2017 and won the Second Prize of the State Scientific and Technological Progress Award in 2018. He received the recognition of the best reviewers for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING in 2015 and the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING in 2017.



Antonio Plaza (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in computer engineering from the University of Extremadura, Badajoz, Spain, in 1999 and 2002, respectively.

He is currently a Full Professor and the Head of the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura. He has authored or coauthored more than 600 publications, including 364 JCR journal papers (265 in IEEE journals), 24 book chapters, and 330 peer-reviewed conference proceeding papers, in the remote sensing field. His current research interests comprise hyperspectral data processing and parallel computing of remote sensing data.

Dr. Plaza has guest-edited ten special issues on hyperspectral remote sensing for different journals. He was the recipient of the Recognition of Best Reviewers for the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS in 2009, the Recognition of Best Reviewers of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING in 2010 for which he was an Associate Editor from 2007 to 2012, the Best Column Award of the *IEEE Signal Processing Magazine* in 2015, the 2013 Best Paper Award of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, the most highly cited paper (2005–2010) in the *Journal of Parallel and Distributed Computing*, and the Best Paper Awards at the IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, the IEEE International Conference on Space Technology, and the IEEE Symposium on Signal Processing and Information Technology. He is an Associate Editor of IEEE ACCESS (receiving the recognition of Outstanding Associate Editor for the journal in 2017) and was a member of the Editorial Board of the IEEE GEOSCIENCE AND REMOTE SENSING NEWSLETTER from 2011 to 2012, *IEEE Geoscience And Remote Sensing Magazine* in 2013, and the Steering Committee of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. He was the Director of Education Activities for the IEEE Geoscience and Remote Sensing Society in 2011 and 2012, and the President of the Spanish Chapter of IEEE Geoscience and Remote Sensing Society in 2012 and 2016. He is a Chair of the Publications Awards Committee of IEEE GRSS and a Vice Chair of the Fellow Evaluations Committee of IEEE GRSS. He has reviewed more than 500 manuscripts for more than 50 different journals. He was the Editor-in-Chief of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2013 to 2017 and IEEE JOURNAL ON MINIATURIZATION FOR AIR AND SPACE SYSTEMS from 2019 to 2020. He has been included in the 2018–2021 Highly Cited Researchers List (Clarivate Analytics). He is a Fellow of IEEE “for contributions to hyperspectral data processing and parallel computing of Earth observation data” and a member of Academia Europaea, The Academy of Europe.



Xu Sun (Member, IEEE) received the B.S. degree in mathematics and application mathematics from Tsinghua University, Beijing, China, in 2006, and the Ph.D. degree in cartography and geographical information system from the Graduate University of Chinese Academy of Sciences (CAS), Beijing, in 2011.

He is currently an Associate Researcher with the Aerospace Information Research Institute, CAS. His current research interests include hyperspectral image processing, artificial intelligence algorithm, hyperspectral unmixing, and high-resolution remote sensing image information mining.



Bing Zhang (Fellow, IEEE) received the B.S. degree in geography from Peking University, Beijing, China, in 1991, and the M.S. and Ph.D. degrees in remote sensing from the Institute of Remote Sensing Applications, Chinese Academy of Sciences (CAS), Beijing, in 1994 and 2003, respectively.

He is currently a Full Professor and the Deputy Director of the Aerospace Information Research Institute, CAS, where he has been leading lots of key scientific projects in the area of Hyperspectral Remote Sensing for more than 25 years. He has

developed five software systems in image processing and applications. His creative achievements were rewarded with ten important prizes from the Chinese Government, and special government allowances of the Chinese State Council. He has authored more than 300 publications. His research interests include the development of mathematical and physical models and image processing software for analyzing hyperspectral remote sensing data in many different areas.

Dr. Zhang was awarded the National Science Foundation for Distinguished Young Scholars of China in 2013. He was awarded the 2016 Outstanding Science and Technology Achievement Prize of the Chinese Academy of Sciences, and the highest level of awards for the CAS scholars. He is serving as an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. He has been serving as a Technical Committee Member of the IEEE Workshop on Hyperspectral Image and Signal Processing since 2011, the President of the Hyperspectral Remote Sensing Committee of China National Committee of International Society for Digital Earth since 2012, and the Standing Director of Chinese Society of Space Research since 2016. He got the IEEE GRSS Regional Leader Award in 2020. He was the Student Paper Competition Committee Member in International Geoscience and Remote Sensing Symposium from 2015 to 2021.