

Superpixel Contracted Graph-Based Learning for Hyperspectral Image Classification

Philip Sellars[✉], Angelica I. Aviles-Rivero[✉], Member, IEEE, and Carola-Bibiane Schönlieb

Abstract—A central problem in hyperspectral image (HSI) classification is obtaining high classification accuracy when using a limited amount of labeled data. In this article we present a novel graph-based semi-supervised framework to tackle this problem. Our framework uses a superpixel approach, allowing it to define meaningful local regions in HSIs, which with high probability share the same classification label. We then extract spectral and spatial features from these regions and use them to produce a contracted weighted graph-representation, where each node represents a region rather than a pixel. The graph is then fed into a graph-based semi-supervised classifier which gives the final classification. We show that using superpixels in a graph representation is an effective tool for speeding up graphical classifiers applied to HSIs. We demonstrate through exhaustive quantitative and qualitative results that our proposed method produces accurate classifications when an incredibly small amount of labeled data is used. We show that our approach mitigates the major drawbacks of existing approaches, resulting in our approach outperforming several comparative state-of-the-art techniques.

Index Terms—Graph-based methods, hyperspectral image (HSI) classification, semi-supervised learning (SSL), superpixels.

I. INTRODUCTION

IN MODERN applications, hyperspectral images (HSIs) capture a detailed light distribution, over several hundreds of spectral bands. This detailed spectral and spatial information increases the discriminative ability of HSIs compared to conventional color images or multispectral images. As a result, hyperspectral imaging has been used in a wide range of applications including classification [1]–[3], object tracking [4]–[6], environmental monitoring [7], [8] and object detection [9]–[11].

In recent years, the classification of hyperspectral data has been an active topic of research. Classifying HSIs requires

Manuscript received May 29, 2019; revised July 19, 2019, September 20, 2019, and December 2, 2019; accepted December 16, 2019. The funding for this research was provided by the National Physics Laboratory and the Engineering and Physical Sciences Research Council. (*Corresponding author: Philip Sellars*)

Philip Sellars and Carola-Bibiane Schönlieb are with the Department of Theoretical Physics and Applied Mathematics, University of Cambridge, Cambridge CB3 0WA, U.K. (e-mail: ps644@cam.ac.uk; cbs31@cam.ac.uk).

Angelica I. Aviles-Rivero is with the Department of Pure Mathematics and Mathematical Statistics (DPMMS), University of Cambridge, Cambridge CB3 0WA, U.K. (e-mail: ai323@cam.ac.uk).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2019.2961599

assigning a class label to each pixel within the image. There are several large hurdles to overcome during the classification process: the high dimensionality of the spectral information, the large spatial variability of the data and the limited number of training samples available due to the cost of data labeling. There have been numerous different attempts to deal with these problems when classifying HSIs, in which the majority of solutions rely on supervised learning (SL). Here we briefly review the major SL approaches.

Kernel based classifiers such as support vector machines (SVMs) [1], [12] are commonly used in the field. Whilst initial kernel methods only used spectral features, many later kernel methods included spatial features. An example being the multiple kernel learning (MKL) approach of Fang *et al.* [13] which used MKL to combine spatial based feature vectors alongside spectral features.

To deal with the high dimensionality of the data, many different feature extraction (FE) methods have been investigated. These methods aim at finding a lower dimensional subspace in which the separability among samples is maximized. Kang *et al.* [14] used image fusion and recursive filtering (IFRF) to extract meaningful features, Li *et al.* [15] exploited local binary patterns (LBPs) to extract local features and textural information and Fang *et al.* [3] used local co-variance matrix representation (LCMR) to characterize the correlation between the spectral and spatial information in HSI data.

Motivated by the remarkable success of Deep Learning (DL), different works have used DL for HSI classification. Convolutions neural networks (CNNs) are commonly used to extract high level spectral and spatial features [16], [17]. Makantasis *et al.* [16] used a CNN to extract spatial and spectral features and passed these into a multilayer perceptron. In recent work, generative adversarial networks (GANs), which simultaneously train a generator and discriminator, have also been explored for HSI classification [18].

Although SL based classifiers have shown good results on HSI data, their performance is heavily reliant on having a large quality training set, that can be obtained either by on-the-spot investigations or visual determination, which is a costly investment. Moreover, the labeled data set might contain human bias, and therefore, propagating the labeling error and uncertainty. As an alternative to SL, we could use unsupervised learning (UL), in which the key idea is to rely on learning a set of classes from data that has not been labeled [19].

Although works such as [20] reported promising results on using UL for HSI classification, the major problem with UL is that the classification task becomes a massively ill-posed problem that needs specific assumptions to mitigate the lack of correspondence between the produced clusters and the known classes.

The aforementioned constraints associated with SL and UL make semi-supervised learning (SSL) [21] a clear alternative for obtaining an improved classification performance. The idea of SSL is to exploit both labeled and unlabelled data in the training process to produce a higher classification accuracy than solely using the labeled data. The advantages of SSL when using HSI data are twofold: we decrease the need for large amounts of labeled data and we gain further understanding of the relationships present in the data. In this article, we introduce for the first time a **superpixel contracted graph-based transductive learning** framework for semi-supervised HSI classification, that we named superpixel graph learning (SGL). It produces state-of-the-art results, especially when the amount of labeled data is extremely small. Our framework is composed of three main parts. Firstly, we modify a common superpixel algorithm, improving its performance on HSIs and allowing it to accurately partition our images into adaptive regions termed superpixels. Secondly, we perform FE on each superpixel to extract discriminative features. Finally, we use the superpixels and features to produce a weighted graphical representation of our image which is then classified using a graphical-learning method (LGC [22]). Our main contributions are as follows:

- 1) We propose a novel **computationally tractable** graphical framework for HSI classification, built upon a common superpixel method Manifold SLIC (MSLIC) [23]. We combine graphical models, SSL and over-segmentations. The key features of our approach are as follows.
 - a) *Hyperspectral Superpixels*: We create and implement a new modified state-of-the-art superpixel method specifically with HSI data in mind, which we call Hyper-Manifold SLIC (HMS). This modification uses a new clustering distance, which combines a Euclidean spectral distance with the Log-Euclidean distance (LED) of a covariance matrix representation. This allows us to define meaningful local regions to boost the overall classification performance. We demonstrate that our modified version outperforms other superpixel algorithms commonly used in hyperspectral approaches.
 - b) *Superpixel Graph Classification*: We show that combining superpixels with a graphical representation and a purely graphical classifier brings two major advantages. Firstly, it vastly decreases the size of the node set which allows for graph classification in computationally feasible times without the need for matrix approximation methods. Secondly, it allows for the intelligent regularization of the final classification map by using superpixels as adaptive local regions.

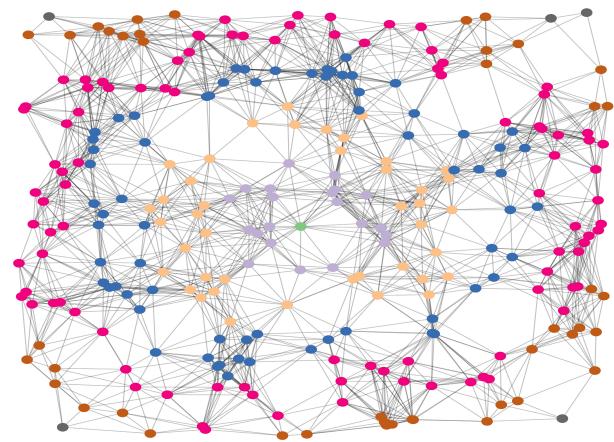


Fig. 1. Data visualization using a weighted undirected graph. The different node colors represent the minimum path length between each node and the central node of the graph, which is colored in green. Graphs are incredibly useful tools for capturing and visualizing the detailed information present in data. Furthermore, graphs are particularly useful for visualizing high dimensional data such as that present in HSIs.

- 2) We extensively validate our proposed approach by using three benchmarking data sets and six comparison methods. We provide a range of experimental results.
- 3) We demonstrate that our graphical superpixel approach (SGL) gives state of the art results for HSI classification across all data sets considered.

The remainder of this article is organized as follows. Section II explores the related work on SSL in the context of HSI classification. Section III is devoted to describing the proposed SGL method including superpixel generation, FE and graph-based semi supervised classification. Section IV contains the experimental results for testing upon three real HSIs and a comparison to other state-of-the-art classification methods. Finally, Section V presents the conclusions as well as discussion of further work.

II. RELATED WORK

The problem of semi-supervised classification of HSIs has been previously investigated by the remote sensing community. In this section, we review the existing techniques in turn. The literature regarding SSL algorithms can be roughly categorized into three different categories. These being **generative models**, **low-density separation** and graph-based methods.

Several previous methods have utilized graph-based learning, and our article is closely related to these. Graph-based methods rely upon constructing a graph representation, where the data points are represented by nodes and the similarity between these data points shown by edges and weights (see Fig. 1). The first graph-based learning method was proposed by Camps-Valls *et al.* [24]. This article used different spectral and spatial kernels alongside the Nyström extension, as a matrix approximation tool, to classify HSIs in computationally reasonable times. However, the produced accuracy was poor compared to other methods at the time. Gao *et al.* [25] used a bilayer graph-based learning algorithm to improve classification performance. The two layers were composed of

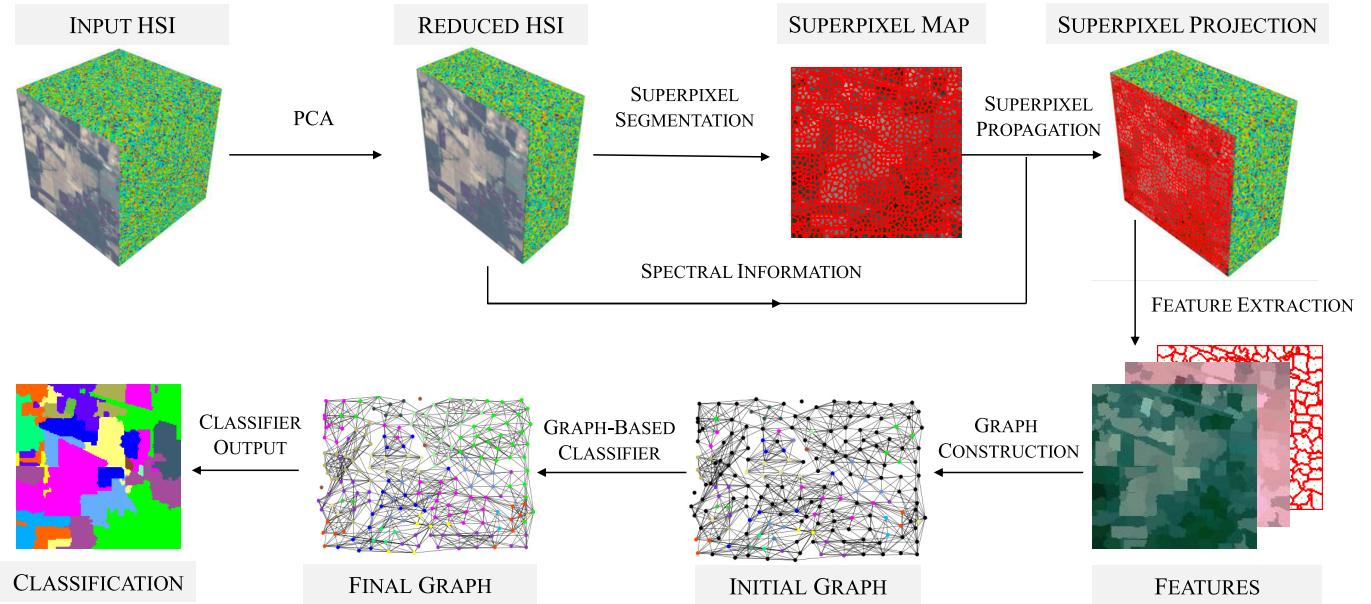


Fig. 2. Proposed framework for the method. An HSI is read in and dimensionally reduced before superpixel segmentation occurs. Features are then extracted from each superpixel and, when combined with the initial labeling, are used to create a superpixel based graph. A graph classifier is used to propagate information across the graph. The final labels are then combined with the superpixel map to give the classification of the HSI.

a pixel-based graph, similar to [24], and a hypergraph built from grouping relations estimated using UL. Cui *et al.* [26] used an extended random walker (ERW) on a superpixel-based graph to optimize a classification map produced from an SVM. Showing that the accuracy of the SVM could be greatly improved by using the information present in the graph.

Another group of semi-supervised methods seek to directly implement the low density separation assumption [21] by moving the decision boundary away from unlabeled points. The first article published in this area was by Bruzzone *et al.* [27] which used a novel **transductive** SVM (TSVM) for HSI classification. A TSVM differs from the standard SVM as it seeks to maximize the margin on a combination of labeled and unlabeled data. Building upon these ideas came semi-supervised self-learning algorithms such as the work by Dópido *et al.* [28], in which they sought to adapt active learning, in a which a user actively selects unlabeled samples, to a self learning framework in which the computer automatically selects the most informative unlabeled samples for classification purposes. Ratle *et al.* [29] took a different path and tackled low density separation using a semi-supervised neural network architecture. An embedding regularizer was added to the loss function to inject the unlabeled information and this approach produced higher classification accuracy than TSVMs.

The rise of DL methods, has led to an increase in popularity of generative methods for SSL. However, these methods are in still in their infancy. One of the most popular approaches by Zhan *et al.* [30] uses a GAN to simultaneous train a discriminator and generator. However, this article uses a 1D-GAN and can only exploit spectral feature and the produced accuracy suffers as a result. Zhu *et al.* [18] developed a 3D-GAN which used convolutional neural networks for the dis-

criminator and generator. This architecture allows the approach to exploit the spectral-spatial information present in the HSI. Therefore, the produced accuracy was much higher than [30].

Although works based on generative models and low-density separation have shown encouraging results, in this article, we concentrate on producing a graph-based method, the motivation for which is threefold. Firstly, data can be naturally represented on graphs. Secondly, a graph representation is motivated by its mathematical background and properties including sparseness. Thirdly, data can be represented in an uniform space even if the data is highly heterogeneous. We seek to produce a graph-based method that is based on superpixel representations similar to that of [26]. However, unlike [26] we seek to produce a fully graph-based learning method rather than a graph-based optimization of a non-graph based method.

III. PROPOSED METHOD

This section is devoted to explaining our proposed framework SGL which tackles the problem of HSI classification. We seek to find an accurate classification prediction for a large amount of unlabelled data given an extremely small amount of labeled data. SGL tackles the classification task under the transductive SSL paradigm.

Theorem 1 (Transductive Semi-Supervised Classification Task): Given a set of known observations $\{(x_i, y_i)\}_{i=1}^l$, $\{x_k\}_{k=l+1}^{l+u}$, and a label set $\mathcal{L} = \{1, \dots, c\}$ where $\{y_i\}_{i=1}^l \subseteq \mathcal{L}$, we seek to find a function $f : \mathcal{X} \mapsto \mathcal{Y}$, which utilizes the labeled and unlabelled data $\{x_k\}_{k=1}^{l+u}$, such that f gives a good prediction of the labels $\{y_k\}_{k=l+1}^{l+u}$.

SGL can be split into two major tasks: graph construction and label propagation. Graph construction is the method of creating a graphical representation from a given HSI. For

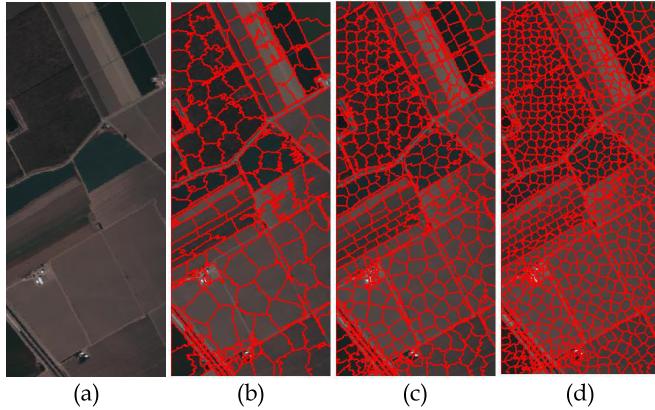


Fig. 3. Salinas HSI segmented using our proposed HMS algorithm. (a) RGB version of the image. (b)–(d) Image segmented using 280, 569, and 1034 superpixels, respectively. Note that due to content sensitive nature of the HMS extension, there are a larger number of smaller superpixels in content dense regions.

our framework, (see Fig. 2), this involves dimensionality reduction, superpixel segmentation and FE. Label propagation is the process of propagating the label information from the initially labeled samples to all nodes in the graph, for which we use a popular graphical classifier LGC [22] but any graph classifier could be used in theory.

We begin by briefly explaining the dimensionality reduction used in this article. Given the closeness of neighboring hyperspectral bands there is a large correlation between them. Therefore, we applied PCA [31] to our HSIs to extract non-correlated features and improve the computational efficiency. Denoting an HSI as $\mathbf{I} = \{I_b\}$, $b = 1, \dots, \mathcal{B}$ with dimensions $\mathcal{W} \times \mathcal{H} \times \mathcal{B}$ representing the width, height, and number of bands, respectively, we produce a reduced image $\hat{\mathbf{I}} = \{\hat{I}_a\}$, $a = 1, \dots, \mathcal{A}$ where $\mathcal{A} \ll \mathcal{B}$. To select the dimensionality \mathcal{A} , we required that the principle components must explain at least 99.8% of the variance present in the original data.

A. Superpixel Segmentation

Superpixels are perceptually meaningful connected regions which group pixels similar in color or other features and were initially introduced by Ren and Malik [32]. In subsequent years, many different algorithmic approaches have been proposed (see [23], [33], [34]). For a detailed survey on superpixel algorithms (see [35]). Fig 3 shows the application of a superpixel algorithm to an HSI. Superpixel maps such as the one shown in Fig 3 have many desirable properties: they are computationally and representationally efficient, the individual superpixels are perceptually meaningful and as superpixels are the result of an over-segmentation they are very good at conserving image structures.

Why use superpixels as a tool for HSIs? In order to extract spatial features for use in spectral-spatial models, it is important to be able to define good local regions. Whilst setting a fixed size window (see [36]) has shown good results, a fixed size does not allow for the full exploitation of spatial context. Using superpixels as adaptive regi-

ons [13], [37]–[40] has been shown to produce discriminative information for HSI classification. Cui *et al.* [26] demonstrated this by using a superpixel based random walker to optimize an SVM probability map to great effect. Furthermore, Cui *et al.* [26] additionally demonstrated that a superpixel spectrum is more stable and less affected by noise than an individual pixel spectrum. Therefore by using superpixels we become more resistant to noise present in the data.

The most common algorithm used in clustering based superpixel methods is Lloyd's algorithm [41], a modified version of the popular k -means clustering algorithm. In the context of Lloyd's algorithm, let us first formalize the definition of a superpixel segmentation.

Theorem 2 (Segmentation in k -Means Clustering): An image of integer width w and integer height h is a function $I : \mathcal{X} \rightarrow \Omega$, where Ω is the image domain and $\mathcal{X} = [w] \times [h] \subset \mathbb{Z}^2$, then a segmentation into superpixels is a partition $\{S_i\}_{i=1}^n \subset \mathcal{X}$ such that for each $1 \leq i \leq n$ we have $S_i = \{x : d((x, I(x)), F(S_i)) = \min_{1 \leq j \leq n} d((x, I(x)), F(S_j))\}$, and each S_i is path-connected to \mathbb{Z}^2 . Where d is a metric on the space $\mathcal{X} \times \Omega$ and F is the feature function on the set of all partitions.

In this article, we build on this definition to propose our algorithmic approach. Denoting an individual pixel as $p \in \hat{\mathbf{I}}$, we seek to partition $\hat{\mathbf{I}}$ into superpixels. This corresponds to splitting $\hat{\mathbf{I}}$ into a family of disjoint sets, $\hat{\mathbf{I}} = \cup_{i=1}^K S_i$, $S_i \cap S_j = \emptyset$, where S_i corresponds to an individual superpixel and K is the number of superpixels. Each superpixel S_i is made up of a set of n_i connected pixels, $S_i = \{p_{i,1}, \dots, p_{i,n_i}\}$.

When constructing our hyperspectral superpixels, we need to ensure that our algorithm extracts effective information from hyperspectral data. Whilst the vast majority of superpixel approaches for HSIs, including [13], feed the first three principal components of HSIs into RGB designed superpixel algorithms, we sought to ensure that our approach can use any number of spectral bands to extract higher dimensional features.

As the base for our algorithm, we began with MSLIC [23]. MSLIC has two features that make it highly useful for our purpose. Firstly it produces content sensitive superpixels by mapping the image I to a 2-D manifold \mathcal{M} and measuring the area of Voronoi cells on \mathcal{M} . Secondly, the number of superpixels will change from the initial selection to fit the content structure in the image, thereby lowering the chances of a poor initial choice of K greatly reducing the final accuracy.

We then designed and implemented several key modifications to MSLIC to allow it to produce accurate results for hyperspectral data. We name this extension Hyper Manifold SLIC (HMS). HMS involves three major changes over MSLIC.

1) *High Dimensional Adaption:* We alter the MSLIC algorithm to take image data with any number of bands \mathcal{B} . This involves changing several steps such as mapping the image \mathbf{I} to a 2-D manifold $\mathcal{M} \in \mathbb{R}^{\mathcal{B}+2}$ rather than the standard $\mathcal{M} \in \mathbb{R}^5$.

2) *Hyperspectral Clustering Distance:* Based on our previous work [42], we design a more effective clustering distance as a combination of the Euclidean spectral distance [34] and LED [43] of a covariance matrix representation [44]. This combination effectively combines the spatial and spectral data

present in the image. For each pixel $p \in \hat{\mathbf{I}}$ we construct a covariance matrix \mathbf{C}_p using the same methodology as Fang *et al.* [3] and use the LED metric to calculate the distances between these matrices. The distance between two pixels p_x, p_y is given by

$$\begin{aligned} d(p_x, p_y) = & \|\text{logm}(\mathbf{C}_{p_x}) - \text{logm}(\mathbf{C}_{p_y})\|_F \\ & + \|\hat{\mathbf{I}}(p_x) - \hat{\mathbf{I}}(p_y)\| + \frac{m}{S} \|p_x - p_y\|. \end{aligned} \quad (1)$$

From (1), the parameter m controls the compactness of superpixels whilst S scales the spatial distance and, for a image with N pixels, $S = (N/K)^{1/2}$ as in the MSLIC algorithm.

3) Spectral Merging: In the original MSLIC algorithm when the area of a seed s_i is below a threshold it is randomly merged with a neighboring seed s_j . However, in our implementation we instead choose the neighboring seed which satisfies

$$j = \underset{s_j \in \mathcal{N}}{\operatorname{argmin}} \|\mathcal{P}_i^m - \mathcal{P}_j^m\| \quad (2)$$

where \mathcal{P}_i^m is the average spectral information of the seed s_i and \mathcal{N} is the set of neighboring seeds. We choose to merge superpixels, which are most similar in their spectral properties, as this yields a better form of adaptation to the hyperspectral data.

These proposed changes allow HMS to produce accurate superpixels for HSIs.

B. FE

Now we seek to extract meaningful features from the extracted superpixels ready for graph construction. In this article, we use the same features as we did in our previous work on superpixels [42]. From each superpixel \mathcal{S}_i we extract three different features. To extract localized spatial information we apply a mean filter to each superpixel to produce a mean feature vector $\vec{\mathcal{S}}_i^m$ which is defined as

$$\vec{\mathcal{S}}_i^m = \frac{\sum_{j=1}^{n_i} \hat{\mathbf{I}}(p_{i,j})}{n_i}. \quad (3)$$

Using a weighted combination of the mean feature vectors of a superpixel's adjacent neighbors, we can obtain a measure of the spatial information between superpixels. Note that adjacency is defined based on four-connectivity on the image grid. For each superpixel \mathcal{S}_i , we define the set $\mathcal{Z}_i = \{z_1, z_2, \dots, z_J\}$ which contains the J indexes of its adjacent superpixels. From this, we construct the weighted feature vector $\vec{\mathcal{S}}_i^w$ which reads

$$\vec{\mathcal{S}}_i^w = \sum_{j=1}^J w_{i,z_j} \vec{\mathcal{S}}_{z_j}^m \quad (4)$$

where the weight between adjacent superpixels w_{i,z_j} is defined as

$$w_{i,z_j} = \frac{\exp(-\|\vec{\mathcal{S}}_{z_j}^m - \vec{\mathcal{S}}_i^m\|_2^2/h)}{\sum_{j=1}^J \exp(-\|\vec{\mathcal{S}}_{z_j}^m - \vec{\mathcal{S}}_i^m\|_2^2/h)} \quad (5)$$

with h as a predefined scalar parameter. Finally, we propose to extract the centroidal location of each superpixel $\vec{\mathcal{S}}_i^p$ which we calculate as

$$\vec{\mathcal{S}}_i^p = \frac{\sum_{j=1}^{n_i} p_{i,j}}{n_i}. \quad (6)$$

C. Graph Based Classification

After defining how to get our superpixel set and extracted features, we now turn to explain how we create our weighted graph-representation. However, we first give some background into challenges associated with the computational implementation of graph-based methods and how superpixels can be used to overcome some of these.

As noted by Camps-Valls *et al.* [24], many graphical algorithms rely on calculating and manipulating large kernel matrices formed by the labeled and unlabelled data. As an example, for an image with n pixels the associated graph Laplacian is a matrix of size $n \times n$. If we seek to inverse the graph Laplacian via singular value decomposition then the computational complexity would be $O(n^3)$, ruining the scaling that we seek.

Approximation methods do exist to speed up such matrix inversions. One commonly used technique is the Nyström extension [45] and it is regularly used to speed up matrix calculations [24], [46]. However, the Nyström extension has several drawbacks. It is unsuitable for sparse applications as the Nyström extension acts as an approximation for complete matrices.

In this article, we implement a novel solution to increase the speed and reduce the complexity of graphical classifiers applied to HSIs. Instead of having a graphical representation where each node represents a pixel, we instead use our segmented superpixels as the node set. This greatly reduces the size of our node set as $K \ll n$ and allows us to perform matrix inversion and other calculations without approximations such as the Nyström extension. Furthermore, a superpixel representation should help to boost the classification accuracy as we are defining strong local regions in our data. Therefore, from these previously discussed features and our superpixel node set, a weighted, undirected graph $G = (V, E, W)$ can be created. The weight between two connected superpixels \mathcal{S}_i and \mathcal{S}_j is constructed based on two Gaussian kernels and is given as

$$w_{ij} = s_{ij} l_{ij} \quad (7)$$

where

$$s_{ij} = \exp\left(\frac{(\beta - 1)\|\vec{\mathcal{S}}_i^w - \vec{\mathcal{S}}_j^w\|_2^2 - \beta\|\vec{\mathcal{S}}_i^m - \vec{\mathcal{S}}_j^m\|_2^2}{\sigma_s^2}\right) \quad (8)$$

$$l_{ij} = \exp\left(\frac{-\|\vec{\mathcal{S}}_i^p - \vec{\mathcal{S}}_j^p\|_2^2}{\sigma_l^2}\right) \quad (9)$$

where β balances the influence between the mean and weighted features and σ_s, σ_l determine the width of the Gaussian kernels. Note that weights are limited in value between $[0, 1]$ with 1 implying most similar. The edge set

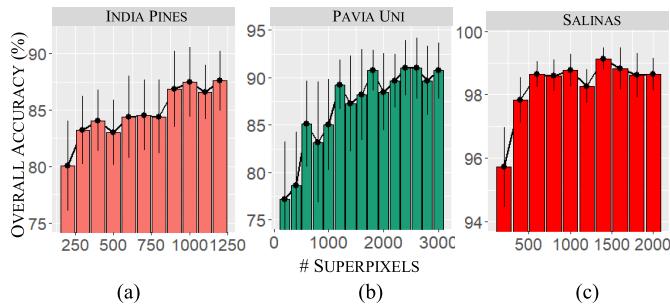


Fig. 4. Sensitivity analysis of the parameter K , the number of superpixels, for (a) Indian Pines, (b) Pavia University, and (c) Salinas. Each data point is the accuracy average of ten repetitions whilst the error bars reflect one standard deviation. For all three data sets the accuracy increases with increasing values of K . However, once the number of superpixels is high enough to accurately over-segment the image, there are diminishing returns for increasing the number of superpixels as the accuracy flattens out.

is constructed using k -nearest neighbors. Therefore, the edge weights are defined as

$$W_{ij} = \begin{cases} w_{ij}, & \text{if } i \text{ is one of the } k \text{ nearest neighbor of } j \\ & \text{or vice versa} \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

For our propagation stage, following standard protocol [24]–[26], a set of labeled spectral pixels are randomly selected from the original HSI. The initial label of each superpixel is taken as the average initial label of its corresponding set of pixels. If no pixel within a superpixel is initially labeled then the superpixel is initially unlabelled. After the superpixel labels are created there is no further use of the pixel labels themselves. The labeling information for the superpixels are specified using a matrix $Y \in \mathbb{R}^{K \times c}$, where c is the number of classes present and K is the number of superpixels. Y_{vl} specifies the value of the seed label l for node v . The weight matrix and the initial labeling are then passed into local and global consistency (LCG) algorithm [22]. LCG is a graph based SSL approach that formalizes the smoothness and clustering assumptions of SSL by designing a classification function which is smooth upon the graphical structure generated by all the data. The final labeling is specified using a matrix $F \in \mathbb{R}^{K \times c}$. The cost function associated with the matrix F is given by

$$\mathcal{Q}(F) = \frac{1}{2} \sum_{i,j=1}^n W_{ij} \left\| \frac{F_i}{\sqrt{D_{ii}}} - \frac{F_j}{\sqrt{D_{jj}}} \right\|^2 + \frac{\mu}{2} \sum_{i=1}^n \|F_i - Y_i\|^2 \quad (11)$$

where $\mu > 0$ is a regularization parameter. \mathcal{F} denotes the set of $n \times c$ matrices with non-negative entries. The labeling matrix is given by $F^* = \operatorname{argmin}_{F \in \mathcal{F}} \mathcal{Q}(F)$.

The first term in the cost function is the smoothness constraint, which encourages connected nodes to have similar labeling, whilst the second term fits the finally labeling to the initially labeled data. Balance between these constraints is set by the parameter μ . The above cost function has a closed

TABLE I

PARAMETER VALUES USED FOR ALL EXPERIMENTS IN THIS ARTICLE.
NOTE THAT $\{x, y\}$ SIGNIFIES A RANDOM UNIFORM DISTRIBUTION
BETWEEN x AND y

FIXED PARAMETERS		
Parameter	Description	Value
m	Controls the compactness of superpixels	10.0
h	Weighted filtering kernel	15.0
σ_s	Kernel parameter for constructing s_{ij}	0.20
k	k -NN construction	8
μ	Weighting in the LGC classifier	{0.1, 0.15}

DATA-BASED PARAMETERS			
Parameter	Indian Pines	Salinas	Pavia University
β	0.9	0.9	0.1
σ_l	{0.4, 0.5}	{3.2, 4.0}	{17, 20}
K	1200	1400	2400

form solution which reads: $F^* = \beta(I - \alpha D^{-(1/2)} W D^{-(1/2)})Y$, where $\beta = (\mu/1 + \mu)$ and $\alpha = 1 - \beta$. The final labeling of the nodes is then computed as: $y_i = \operatorname{argmax}_{j \leq c} F_{ij}$. The superpixel labels and the superpixel segmentation are used to construct the final pixel classification map.

IV. EXPERIMENTAL RESULTS

In this section, we detail the experiments conducted to validate the proposed approach.

A. Data Description

We use three benchmark HSI data sets to evaluate our approach, which have the following characteristics.

- 1) *Indian Pines Data set*: The data set was collected by an Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over an agricultural site in Indiana and has 16 classes. The data set consists of 145×145 pixels, 200 spectral channels, a spectral range of 0.4 to $2.5 \mu\text{m}$ and a spatial resolution of 20 m.
- 2) *Salinas*: This image was also collected by the AVIRIS sensor over Salinas Valley, California, and contains 16 classes. The image size is 512×217 pixels and identical to Indian Pines has 200 spectral channels over 0.4 to $2.5 \mu\text{m}$. The data has a spatial resolution of 3.7 m/pixel.
- 3) *University of Pavia*: This data set was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS). The image (610×340 pixels) covers the Engineering School at the University of Pavia and has nine classes. The image contains 115 spectral channels from 0.43 to $0.86 \mu\text{m}$ and has a spatial resolution of 1.3 m.

B. Evaluation Protocol

For all experiments carried out in this article, each one is repeated ten times and the average and standard deviation are provided for each measurement. To evaluate the performance of each HSI classifier, we use three commonly implemented evaluation criteria overall accuracy (OA), average accuracy (AA) and the Kappa Coefficient.

TABLE II
OA(%) OF TEN REPEATED EXPERIMENTS WITH DIFFERING NUMBERS OF TRAINING SAMPLES PER CLASS

SALINAS							
SAMPLES PER CLASS	OURS	LCMR [3]	SC-MK [13]	EPF [47]	LBP [15]	IFRF [14]	SVM [1]
3	98.0 ± 0.8%	83.7 ± 4.0%	82.8 ± 2.9%	78.2 ± 4.8%	78.5 ± 3.4%	87.7 ± 5.3%	76.6 ± 2.3%
5	99.1 ± 0.5%	89.9 ± 2.1%	85.0 ± 3.0%	80.9 ± 4.4%	84.8 ± 2.8%	93.1 ± 1.8%	79.6 ± 1.9%
7	99.1 ± 0.3%	92.3 ± 1.6%	88.2 ± 2.2%	84.9 ± 3.2%	88.4 ± 2.0%	93.8 ± 1.5%	81.3 ± 1.9%
10	99.0 ± 0.4%	93.8 ± 1.1%	90.1 ± 2.2%	86.4 ± 4.3%	91.4 ± 1.2%	95.4 ± 1.3%	82.4 ± 1.2%
15	99.1 ± 0.3%	94.7 ± 1.0%	93.1 ± 1.1%	89.2 ± 2.4%	93.0 ± 1.2%	97.1 ± 1.2%	84.3 ± 1.4%
20	99.3 ± 0.2%	96.1 ± 0.8%	93.3 ± 1.1%	89.8 ± 3.3%	95.1 ± 1.1%	97.3 ± 1.0%	84.5 ± 1.5%

INDIANA PINES							
SAMPLES PER CLASS	OURS	LCMR [3]	SC-MK [13]	EPF [47]	LBP [15]	IFRF [14]	SVM [1]
3	78.7 ± 5.3%	66.1 ± 3.9%	59.8 ± 4.1%	44.6 ± 5.0%	58.9 ± 3.7%	57.4 ± 3.9%	37.7 ± 5.0%
5	82.6 ± 3.9%	74.1 ± 3.3%	67.8 ± 3.8%	49.7 ± 9.4%	67.3 ± 3.9%	67.2 ± 6.3%	42.4 ± 5.3%
7	87.8 ± 2.1%	78.5 ± 3.0%	73.6 ± 5.1%	57.6 ± 5.4%	75.6 ± 2.9%	75.7 ± 3.8%	48.1 ± 2.2%
10	90.7 ± 2.2%	82.7 ± 3.1%	80.7 ± 2.5%	67.3 ± 3.2%	78.9 ± 2.7%	80.3 ± 1.8%	53.0 ± 3.3%
15	92.9 ± 0.9%	86.9 ± 2.0%	86.2 ± 2.2%	74.5 ± 3.6%	85.9 ± 1.8%	87.9 ± 1.2%	59.5 ± 1.6%
20	94.4 ± 1.4%	90.0 ± 2.0%	89.7 ± 1.6%	80.8 ± 2.3%	88.6 ± 1.4%	89.9 ± 1.9%	63.3 ± 1.4%

UNIVERSITY OF PAVIA							
SAMPLES PER CLASS	OURS	LCMR [3]	SC-MK [13]	EPF [47]	LBP [15]	IFRF [14]	SVM [1]
3	84.4 ± 4.9%	70.3 ± 7.3%	63.6 ± 6.4%	56.1 ± 7.1%	55.1 ± 6.4%	57.6 ± 5.8%	57.1 ± 8.3%
5	88.1 ± 4.6%	78.8 ± 5.1%	71.4 ± 4.5%	64.0 ± 7.4%	65.4 ± 4.2%	67.4 ± 4.3%	62.4 ± 4.4%
7	92.1 ± 1.9%	83.0 ± 4.9%	77.9 ± 3.9%	67.0 ± 7.6%	71.1 ± 3.6%	71.7 ± 4.9%	62.2 ± 7.0%
10	93.7 ± 1.4%	87.4 ± 3.5%	81.6 ± 4.7%	72.7 ± 9.1%	75.4 ± 3.1%	77.3 ± 5.8%	67.4 ± 4.7%
15	94.5 ± 1.8%	90.1 ± 2.6%	87.3 ± 2.4%	79.2 ± 6.6%	79.2 ± 2.0%	83.1 ± 3.5%	73.0 ± 3.8%
20	95.4 ± 0.9%	92.3 ± 2.1%	88.3 ± 2.1%	85.7 ± 3.4%	83.4 ± 1.9%	88.5 ± 2.1%	74.1 ± 4.0%

To validate the performance of our proposed classification framework SGL, several state-of-the-art HSI classification methods have been implemented to act as comparisons. These are LCMR [3], superpixel-based classification via multiple kernels (SC-MK) [13], the edge preserving filter (EPF) based method [47], LBPs [15], an SVM method [1] and IFRF [14].

C. Parameter Selection

In our proposed framework, there are eight hyperparameters that come from the four tasks of our framework.

- 1) Superpixel construction: K and m .
- 2) FE: h .
- 3) Graph construction: σ_s , σ_l , k , and β .
- 4) LGC classification: μ .

For the superpixels construction step, we enforce that the ratio of the number of pixels to the number of superpixels (N/K) must be at least 15. The parameters m , h , σ_s , k , and μ have the same value for all data sets used. These values were found using empirical testing in a coarse to fine search method. The other three parameters, σ_l , β , and K , change value depending on the HSI used. The parameter values used in the experiments are given in Table I. We note that that the parameters m , β , k , and β had a relatively small effect on the final classification accuracy.

Given that we are using a superpixel based classifier, it is critically important to understand how the superpixel number K effects the accuracy. This is especially true when it is unclear what value of K to pick for a given image. To investigate the effect of changing the parameter K , we classified all three HSIs using a varying number of superpixels and seven randomly selected samples from each class and plotted

the classification accuracy against the superpixel number. The results for this analysis are given in Fig. 4. In general the classification accuracy increases with the number of superpixels, due to the underlying over-segmentation being more accurate. However, once the image is accurately over-segmented, there are diminishing returns for further increasing the superpixel number. Combined with the fact that increasing the number of superpixels increases the size of the graph and thus the running time, we used the smallest number superpixels that reliably gave a good classification accuracy for each HSI.

For the compared methods the parameters were set using the default values provided in the demo code or referenced in this articles themselves. The SVM method was implemented using the LIBSVM [49] library and uses a Gaussian kernel and fivefold cross validation.

D. Experimental Results

Our experiments are organized into two parts. Firstly, we compare the classification accuracy of our proposed framework with the comparison classifiers detailed above. Due to the semi-supervised nature of our method, we will be testing the classification performance using very limited amounts of training data. Secondly, we will seek to use visual classification maps to understand and explain the performance of our classifier to relation to the other classifiers.

E1: In our first experiment, we evaluate the OA of our method against the state-of-the-art when using a reduced amount of labeled data for training ($\{3, 5, 7, 10, 15, 20\}$ randomly selected samples per class). The accuracy of the different classifiers for the three benchmark data sets are given in Table II and the graphical representation of the results is shown in Fig. 5.

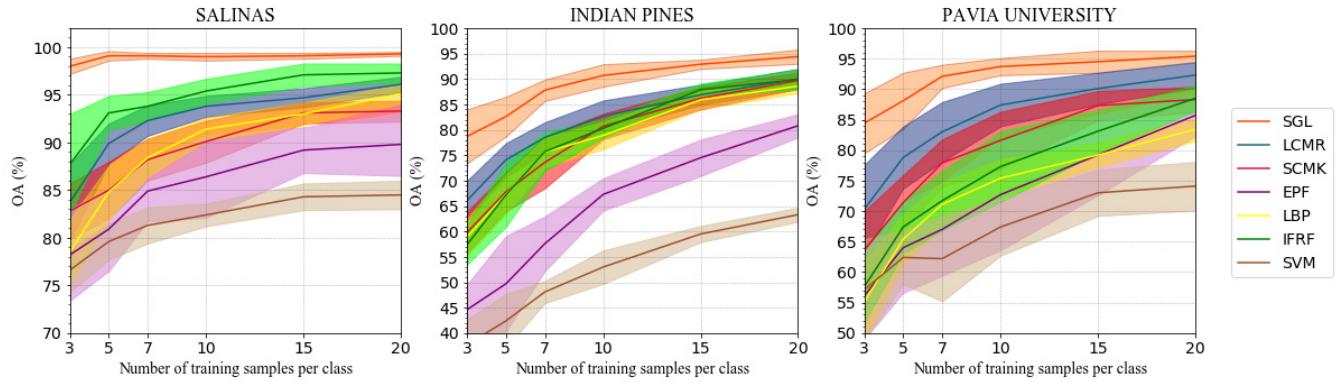


Fig. 5. Comparison of the classification accuracy of different methods with varying number of training samples. The methods used are LCMR [3], SC-MK [13], EPF [47], LBP [15], IFRF [14], SVM [1] and the proposed SGL method. The solid lines represent the average of the different methods whilst the shaded area covers one standard deviation from the mean.

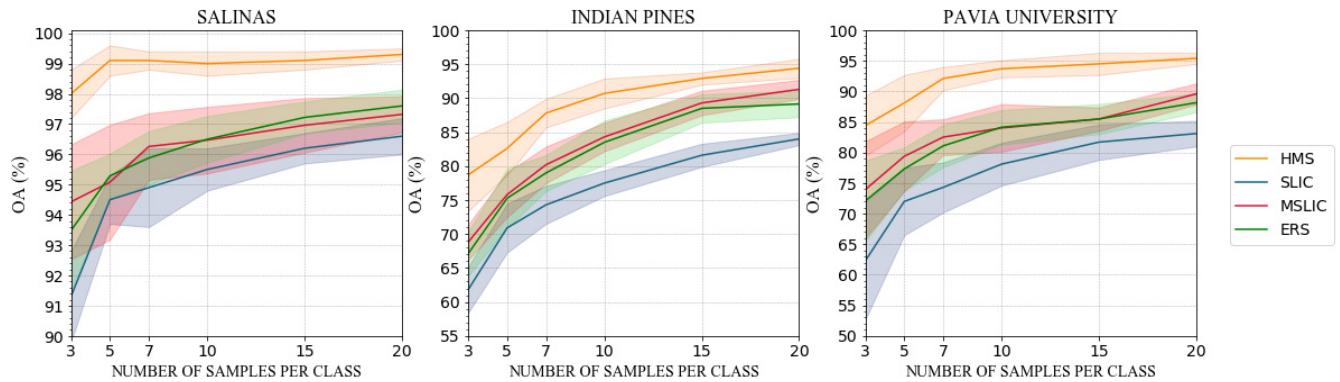


Fig. 6. Comparison of the classification accuracy obtained when using different superpixel algorithms for over-segmentation. HMS is the new modification proposed in this article whilst SLIC [33], MSLIC [23] and ERS [48] are three commonly used superpixel methods which have been used for HSIs.

We see that the accuracy produced by the SGL framework is, by a significant margin, the best of any classifier considered in this article. The SGL framework produces the best accuracy for all three benchmark images for each differing amount of labeled data. In particular, the average difference in OA between SGL and its nearest competitor LCMR [3], across the three data sets, was 9% when using 5 samples per class and was 13.7% when using 3 samples per class. Highlighting the semi-supervised nature of the SGL framework that allows it to exploit information present in the unlabelled data to overcome the limited amount of labeled samples.

E2: Secondly, we sought to demonstrate that our superpixel modification HMS produces more accurate over-segmentations than other superpixel algorithms which are commonly used for hyperspectral data. To show this, we evaluated the OA when using different superpixel algorithms. Acting as comparisons to our method are entropy rate superpixels ERS [48], SLIC [33], and MSLIC [23]. For these comparative methods the parameters were kept the same as in the cited articles, the number of superpixels were set using Table I and three spectral bands, extracted by PCA, were used as input. Note that unlike these comparison methods, HMS is being fed several ten's of spectral bands, extracted using the previously

described dimensional reduction method, as it can cope with input images of any dimension. The accuracy comparison is given in Fig. 6 and it clearly shows that the results produced by HMS are substantially more accurate. Thus highlighting the extra information that can be obtained by using more spectral bands and high dimensional features such as covariance matrices.

In Figs. 7, 8 and 9 we provide additional visual results for the HMS algorithm. We provide over-segmentations with differing numbers of superpixels highlighting the content sensitivity of the algorithm. In particular, note that Indian Pines and Salinas are easily over-segmented using a small number of superpixels whilst the more complex structure of Pavia University requires more superpixels to achieve an accurate over-segmentation.

E3: To gain an understanding about how each classifier was performing and the explanation for the large increase in classification accuracy obtained by SGL, we produced visual classification maps. For each HSI we use ten labeled samples per class and ran the methods again to calculate the OA, AA, the Kappa coefficient, a full class by class breakdown and full classification maps. The numerical results for this experiment are reported in Table III whilst Figs. 10, 11, and 12 give color

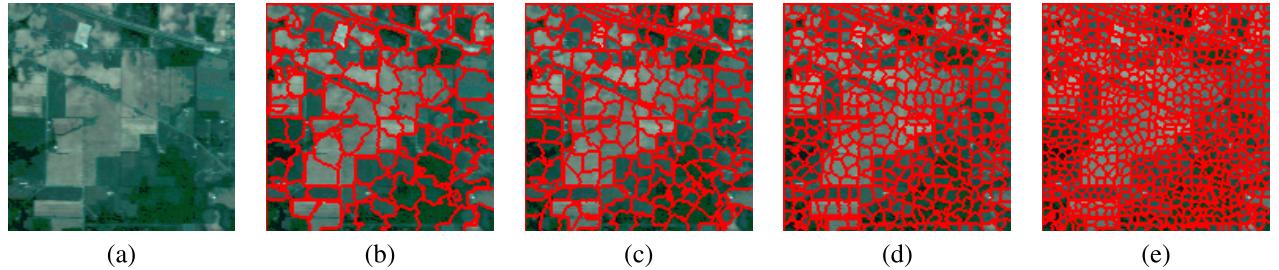


Fig. 7. Superpixel over-segmentations on the Indian Pines scene generated by the HMS extension. (a) Composite RGB image. (b)–(e) Superpixel segmentations with 129, 287, 434, and 791 superpixels, respectively.

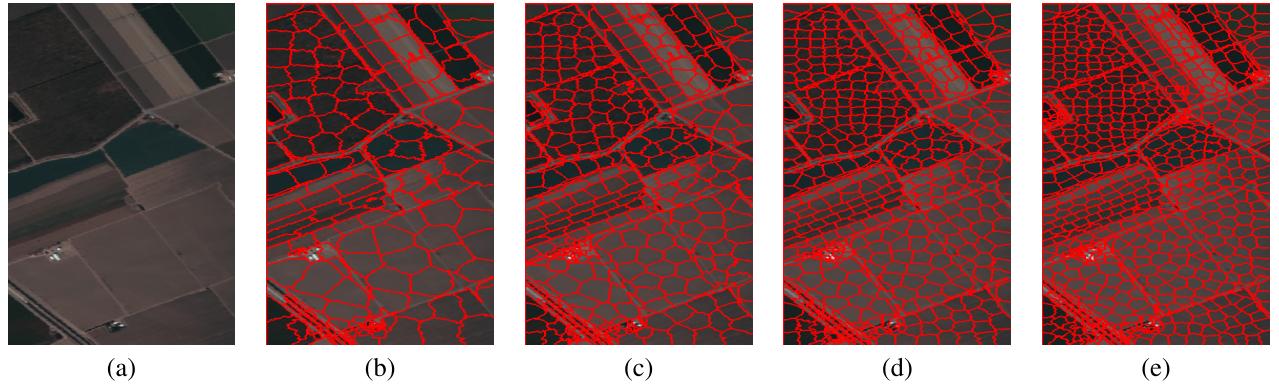


Fig. 8. Superpixel over-segmentations on the Salinas scene generated by the HMS extension. (a) Composite RGB image. (b)–(e) Superpixel segmentations with 244, 465, 639, and 919 superpixels, respectively.

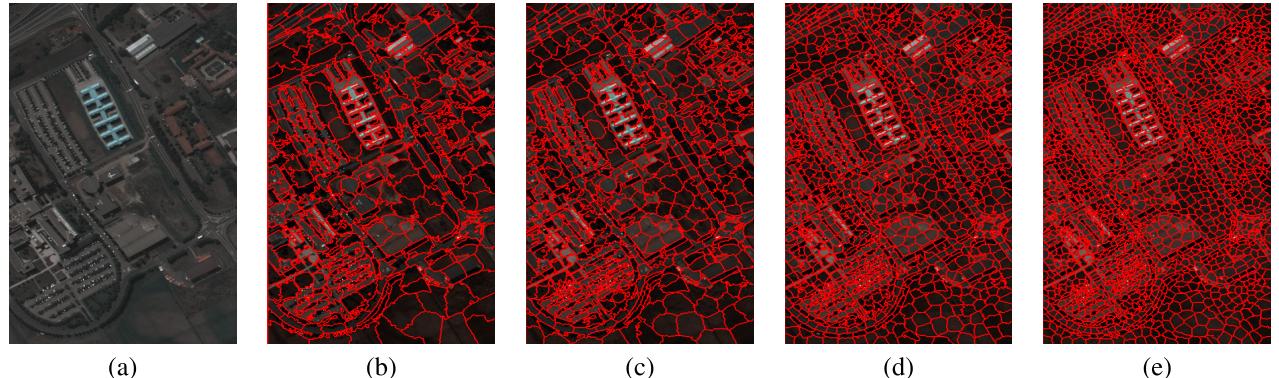


Fig. 9. Superpixel over-segmentations on the University of Pavia scene generated by the HMS extension. (a) Composite RGB image. (b)–(e) Superpixel segmentations with 948, 1286, 1662, and 1963 superpixels, respectively.

composite images, ground truth images and final classification maps for the seven considered methods for the three data sets used.

Examining the OA, AA, and Kappa coefficient of the differing methods, we observe that SGL is again the best performing method with an average improvement of OA +12.3%, AA +9.5%, and Kappa +11.6% in the Indian pines scene, OA +15.4%, AA +13.8%, and Kappa +17.3% in the Pavia University scene and OA +7.8%, AA +4.3%, and Kappa +8.4% in the Salinas scene compared to the other classifiers (excluding the SVM).

To provide an explanation for the fantastic performance of SGL compared to the other methods let us examine the classification maps. The poorest performing classifier was the

SVM. The SVM method only uses spectral information and as a result produces very noisy classification maps. The EPF method seeks to optimize the SVM classification map with an EPF to smooth out some of this noise and from these results we can see it successfully does so. However, the poor performance of the underlying SVM classification prevents the EPF method from achieving good classification. The LBP and IFRF methods produce over-smooth classification results when only a limited amount of data is available. This causes poor performance in the more complicated Indian Pines and Pavia University images. The LCMR and SCMK methods are the closest competitors to the SGL method with LCMR slightly outperforming the SCMK method due to a slightly higher amount of smoothing. Both of these methods manage

TABLE III

OA(%) AA(%), KAPPA AND A CLASS BY CLASS BREAKDOWN OBTAINED BY DIFFERENT CLASSIFIERS WITH TEN TRAINING SAMPLES PER CLASS. THE BEST RESULTS ARE HIGHLIGHTED IN GREEN

INDIANA PINES							
CLASS	SGL	LCMR [3]	SC-MK [13]	EPF [47]	LBP [15]	IFRF [14]	SVM [1]
1	98.04 \pm 0.65%	99.44 \pm 1.17%	98.33 \pm 1.43%	53.78 \pm 28.30%	100.0 \pm 0%	53.51 \pm 25.90%	20.53 \pm 5.10%
2	76.06 \pm 9.19%	75.99 \pm 10.18%	78.82 \pm 6.80%	55.88 \pm 11.32%	70.94 \pm 7.05%	70.35 \pm 10.65%	43.14 \pm 8.51%
3	84.27 \pm 6.04%	70.78 \pm 8.08%	77.65 \pm 9.18%	60.19 \pm 17.99%	70.28 \pm 12.13%	67.99 \pm 7.80%	39.15 \pm 8.39%
4	96.67 \pm 2.73%	93.88 \pm 10.78%	86.39 \pm 12.62%	34.57 \pm 12.83%	97.93 \pm 3.23%	79.37 \pm 12.24%	21.25 \pm 3.97%
5	92.30 \pm 7.48%	90.32 \pm 10.14%	82.33 \pm 10.56%	93.39 \pm 5.30%	82.92 \pm 7.94%	79.40 \pm 13.64%	59.34 \pm 11.59%
6	98.71 \pm 0.60%	91.40 \pm 4.18%	89.54 \pm 7.43%	86.32 \pm 10.34%	90.36 \pm 5.49%	93.63 \pm 4.96%	83.29 \pm 3.83%
7	100.0 \pm 0.00%	100.0 \pm 0.00%	100.0 \pm 0.00%	70.92 \pm 39.13%	100.0 \pm 0.00%	39.65 \pm 23.77%	24.85 \pm 11.06%
8	100.0 \pm 0.00%	99.68 \pm 0.23%	97.09 \pm 9.19%	98.41 \pm 3.83%	100.0 \pm 0.00%	99.97 \pm 0.07%	93.12 \pm 4.02%
9	100.0 \pm 0.00%	100.0 \pm 0.00%	100.0 \pm 0.00%	59.44 \pm 26.70%	100.0 \pm 0.00%	28.81 \pm 21.34%	12.59 \pm 8.36%
10	88.94 \pm 6.52%	76.46 \pm 7.31%	71.32 \pm 10.49%	61.19 \pm 11.43%	79.90 \pm 5.05%	75.95 \pm 9.75%	36.93 \pm 10.25%
11	91.04 \pm 7.49%	71.40 \pm 6.33%	69.22 \pm 12.69%	81.05 \pm 8.72%	73.78 \pm 6.78%	93.81 \pm 4.23%	61.50 \pm 3.96%
12	90.05 \pm 4.14%	90.50 \pm 3.66%	78.47 \pm 17.31%	44.31 \pm 14.00%	70.58 \pm 6.99%	74.08 \pm 10.25%	28.20 \pm 5.78%
13	99.56 \pm 0.15%	99.33 \pm 0.25%	99.90 \pm 0.22%	98.34 \pm 3.38%	98.31 \pm 2.97%	75.32 \pm 15.02%	80.12 \pm 6.41%
14	100.0 \pm 0.00%	98.18 \pm 3.54%	88.14 \pm 2.69%	95.15 \pm 4.04%	91.32 \pm 5.03%	98.31 \pm 1.34%	88.22 \pm 4.03%
15	97.69 \pm 6.92%	91.62 \pm 10.39%	90.96 \pm 11.42%	64.91 \pm 23.49%	90.59 \pm 10.07%	77.14 \pm 11.42%	39.10 \pm 8.79%
16	100.0 \pm 0.00%	98.67 \pm 3.79%	97.59 \pm 1.50%	84.46 \pm 7.54%	98.43 \pm 1.14%	92.62 \pm 14.18%	87.71 \pm 20.71%
OA	90.89 \pm 2.98%	82.74 \pm 2.32%	79.91 \pm 2.60%	68.95 \pm 2.01%	80.52 \pm 2.03%	80.86 \pm 3.76%	51.20 \pm 3.92%
AA	92.16 \pm 6.77%	90.48 \pm 1.56%	87.86 \pm 1.53%	71.39 \pm 3.49%	88.46 \pm 1.29%	74.99 \pm 3.16%	51.19 \pm 3.22%
Kappa	87.5 \pm 3.33%	80.51 \pm 2.59%	77.31 \pm 2.93%	65.02 \pm 2.25%	78.09 \pm 2.23%	78.45 \pm 4.16%	45.41 \pm 4.11%

UNIVERSTY OF PAVIA							
CLASS	SGL	LCMR [3]	SC-MK [13]	EPF [47]	LBP [15]	IFRF [14]	SVM [1]
1	86.64 \pm 4.39%	79.29 \pm 7.09%	72.48 \pm 13.89%	94.80 \pm 4.52%	59.64 \pm 5.07%	68.30 \pm 7.67%	94.09 \pm 5.44%
2	95.87 \pm 3.17%	87.67 \pm 8.05%	80.05 \pm 8.03%	89.55 \pm 6.61%	69.72 \pm 8.12%	94.90 \pm 2.19%	85.59 \pm 2.55%
3	85.37 \pm 10.53%	90.90 \pm 4.45%	76.84 \pm 9.10%	62.03 \pm 23.65%	79.52 \pm 7.30%	53.78 \pm 10.49%	42.74 \pm 13.46%
4	87.44 \pm 3.77%	95.10 \pm 3.40%	94.77 \pm 2.74%	57.08 \pm 11.72%	66.44 \pm 7.33%	66.44 \pm 22.53%	59.85 \pm 10.48%
5	95.84 \pm 2.91%	97.03 \pm 6.17%	99.66 \pm 0.08%	91.20 \pm 5.64%	89.91 \pm 12.78%	99.63 \pm 1.10%	93.69 \pm 5.78%
6	99.92 \pm 0.19%	95.37 \pm 2.36%	76.24 \pm 6.62%	49.32 \pm 13.91%	89.33 \pm 4.03%	82.47 \pm 9.46%	39.38 \pm 9.21%
7	96.59 \pm 1.10%	92.58 \pm 8.13%	76.06 \pm 14.92%	66.86 \pm 13.46%	89.15 \pm 8.91%	63.32 \pm 12.76%	42.26 \pm 10.38%
8	94.03 \pm 5.63%	84.67 \pm 5.57%	79.79 \pm 3.85%	75.57 \pm 10.98%	80.78 \pm 16.55%	55.33 \pm 7.28%	73.22 \pm 5.74%
9	97.55 \pm 0.43%	93.80 \pm 3.55%	100.0 \pm 0.00%	98.48 \pm 1.70%	59.40 \pm 7.01%	49.33 \pm 9.07%	99.87 \pm 0.10%
OA	93.70 \pm 1.35%	88.29 \pm 4.06%	80.23 \pm 4.06%	73.92 \pm 7.06%	72.66 \pm 4.29%	76.36 \pm 3.81%	67.40 \pm 4.66%
AA	93.25 \pm 5.03%	90.72 \pm 1.67%	83.99 \pm 2.15%	76.10 \pm 5.06%	75.99 \pm 2.71%	70.39 \pm 3.24%	70.08 \pm 2.48%
Kappa	91.71 \pm 1.73%	84.91 \pm 4.89%	74.63 \pm 4.60%	67.40 \pm 8.23%	72.66 \pm 4.25%	69.70 \pm 4.55%	59.38 \pm 4.88%

SALINAS							
CLASS	SGL	LCMR [3]	SC-MK [13]	EPF [47]	LBP [15]	IFRF [14]	SVM [1]
1	100.0 \pm 0.00%	99.95 \pm 0.06%	99.93 \pm 0.13%	100.0 \pm 0.00%	97.97 \pm 2.64%	95.77 \pm 6.65%	97.54 \pm 2.53%
2	100.0 \pm 0.00%	93.21 \pm 5.19%	98.66 \pm 1.82%	99.87 \pm 0.29%	96.57 \pm 2.58%	100.0 \pm 0.00%	99.10 \pm 0.49%
3	100.0 \pm 0.00%	99.56 \pm 0.42%	96.94 \pm 4.04%	93.84 \pm 2.01%	98.59 \pm 2.03%	99.32 \pm 0.78%	86.62 \pm 3.31%
4	99.71 \pm 0.01%	100.0 \pm 0.00%	98.79 \pm 0.77%	97.70 \pm 0.79%	97.84 \pm 2.98%	87.42 \pm 8.54%	96.92 \pm 0.73%
5	98.09 \pm 0.00%	96.88 \pm 1.09%	95.63 \pm 1.92%	99.48 \pm 0.98%	92.37 \pm 4.34%	99.92 \pm 0.08%	97.57 \pm 2.25%
6	99.93 \pm 0.02%	98.53 \pm 0.67%	99.53 \pm 0.81%	99.98 \pm 0.02%	92.14 \pm 4.40%	100.0 \pm 0.00%	99.97 \pm 0.05%
7	99.48 \pm 0.98%	97.57 \pm 1.96%	94.22 \pm 5.79%	97.92 \pm 2.40%	92.68 \pm 6.81%	98.88 \pm 1.14%	97.68 \pm 1.84%
8	99.38 \pm 0.62%	87.84 \pm 4.77%	74.47 \pm 11.72%	84.19 \pm 7.87%	85.27 \pm 6.12%	96.83 \pm 4.43%	70.82 \pm 3.92%
9	100.0 \pm 0.00%	96.90 \pm 2.63%	99.40 \pm 0.81%	99.47 \pm 0.19%	93.05 \pm 2.72%	98.82 \pm 0.18%	98.84 \pm 0.90%
10	96.72 \pm 2.92%	93.71 \pm 7.73%	88.34 \pm 7.20%	86.13 \pm 5.46%	93.65 \pm 3.28%	99.21 \pm 8.00%	79.64 \pm 4.15%
11	95.882 \pm 2.19%	99.94 \pm 0.05%	97.03 \pm 3.65%	91.81 \pm 8.68%	97.83 \pm 3.26%	98.96 \pm 0.45%	83.29 \pm 6.77%
12	99.90 \pm 0.00%	99.60 \pm 1.14%	97.50 \pm 6.22%	99.42 \pm 0.56%	89.96 \pm 4.07%	98.19 \pm 1.15%	94.42 \pm 1.60%
13	98.80 \pm 0.00%	98.65 \pm 0.73%	95.36 \pm 4.40%	96.32 \pm 2.87%	91.59 \pm 6.07%	92.20 \pm 8.00%	88.15 \pm 8.68%
14	95.38 \pm 1.48%	95.23 \pm 2.86%	90.29 \pm 6.73	95.27 \pm 11.66%	88.18 \pm 6.84%	87.05 \pm 14.28%	84.51 \pm 17.07%
15	99.28 \pm 0.07%	88.12 \pm 7.50%	84.36 \pm 6.15%	56.02 \pm 5.52%	82.42 \pm 12.45%	86.51 \pm 8.12%	49.19 \pm 2.71%
16	100.0 \pm 0.00%	94.46 \pm 6.47%	96.17 \pm 3.18%	98.67 \pm 4.09%	97.96 \pm 3.91%	99.77 \pm 0.56%	92.51 \pm 8.59%
OA	99.24 \pm 0.16%	93.90 \pm 1.29%	90.38 \pm 2.42%	86.53 \pm 1.99%	90.68 \pm 1.35%	95.87 \pm 1.62%	82.42 \pm 1.15%
AA	98.9 \pm 1.51%	96.26 \pm 1.02%	94.16 \pm 1.11%	93.51 \pm 0.91%	93.00 \pm 1.03%	96.24 \pm 1.43%	88.55 \pm 0.99%
Kappa	99.15 \pm 0.17%	93.22 \pm 1.44%	89.33 \pm 2.663%	85.10 \pm 2.15%	90.68 \pm 1.36%	95.41 \pm 1.80%	80.53 \pm 1.25%

to preserve edges and boundaries whilst producing smooth classification maps. This is due to the inclusion of spatial information via local neighboring pixel construction and superpixel based kernels respectively.

What sets SGL apart from the other methods considered is that the classification map has been intelligently smoothed with near complete preservation of edges and boundaries. Primarily, this is due to the use of superpixels as the node

set in our graph. The superpixels produce by HMS have accurately preserved the edges and boundaries in the image. Therefore, when we assign labels to each superpixel, rather than each pixel, we smooth our classification map across the homogeneous superpixels whilst retaining boundaries.

E4: For our final experiment, we compare the computing time of the methods when applied to the three HSIs reported in Table IV. We ran all test under the same conditions

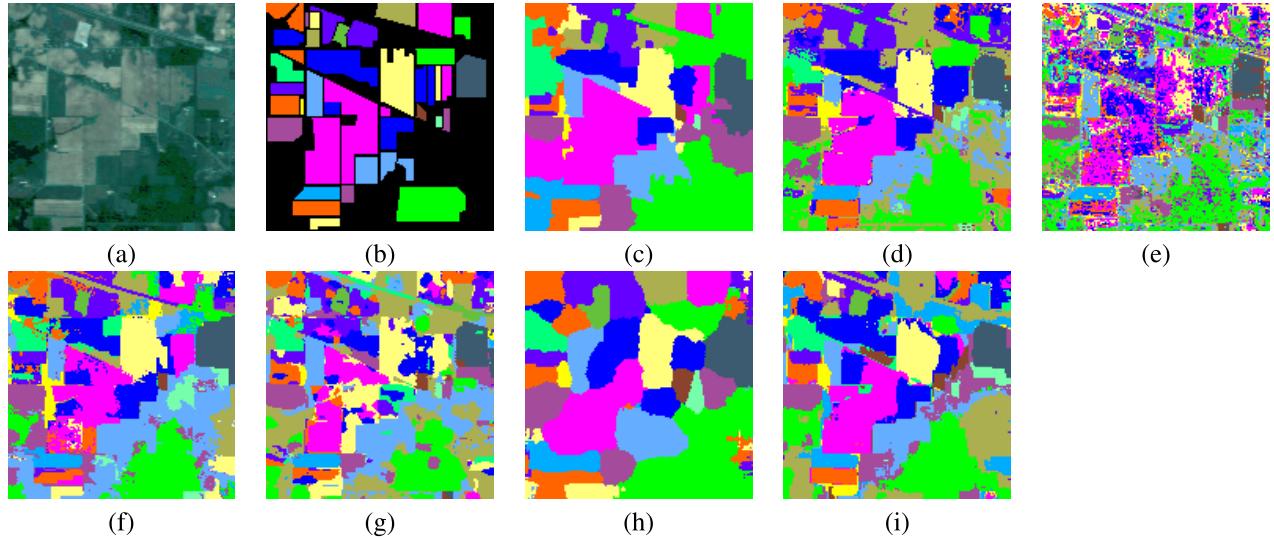


Fig. 10. Indian Pines data set. (a) Color composite image. (b) Ground truth. (c)–(h) Classifications maps produced using ten labeled samples for each class. The methods used were: (c) proposed SGL, (d) LCMR [3], (e) SVM [1], (f) SC-MK [13], (g) EPF [47], (h) LBP [15], and (i) IFRF [14].

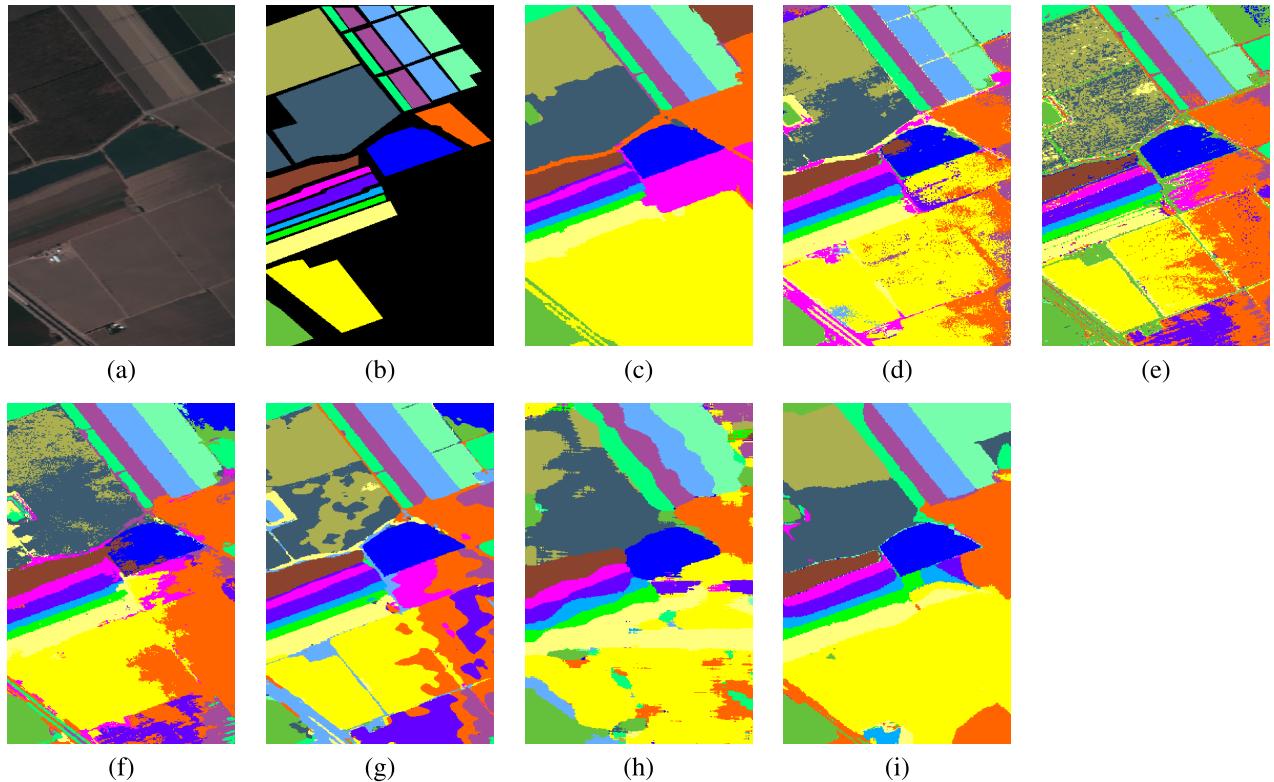


Fig. 11. Salinas data set. (a) Color composite image. (b) Ground truth. (c)–(h) Classifications maps produced using ten labeled samples for each class. The methods used were: (c) proposed SGL, (d) LCMR [3], (e) SVM [1], (f) SC-MK [13], (g) EPF [47], (h) LBP [15], and (i) IFRF [14].

using an Intel Core i5-4670 processor and 16 GB of RAM. As we can see, our full method (OURS) is one of the slower methods considered in this analysis. This is due to the time required to extract and manipulate the co-variance matrices. If we exclude the time needed to produce the covariance matrices the average computational time is instead 6.6, 22.2, and 24.6 s for the three data sets. Showing that by using a

superpixel representation we have greatly reduced the time required for the graphical learning component of our framework. As the covariance matrices are only used in the over-segmentation step, the implementation of a new faster over-segmentation approach, which preserves performance would allow our framework to achieve very fast performance in line with the compared methods.

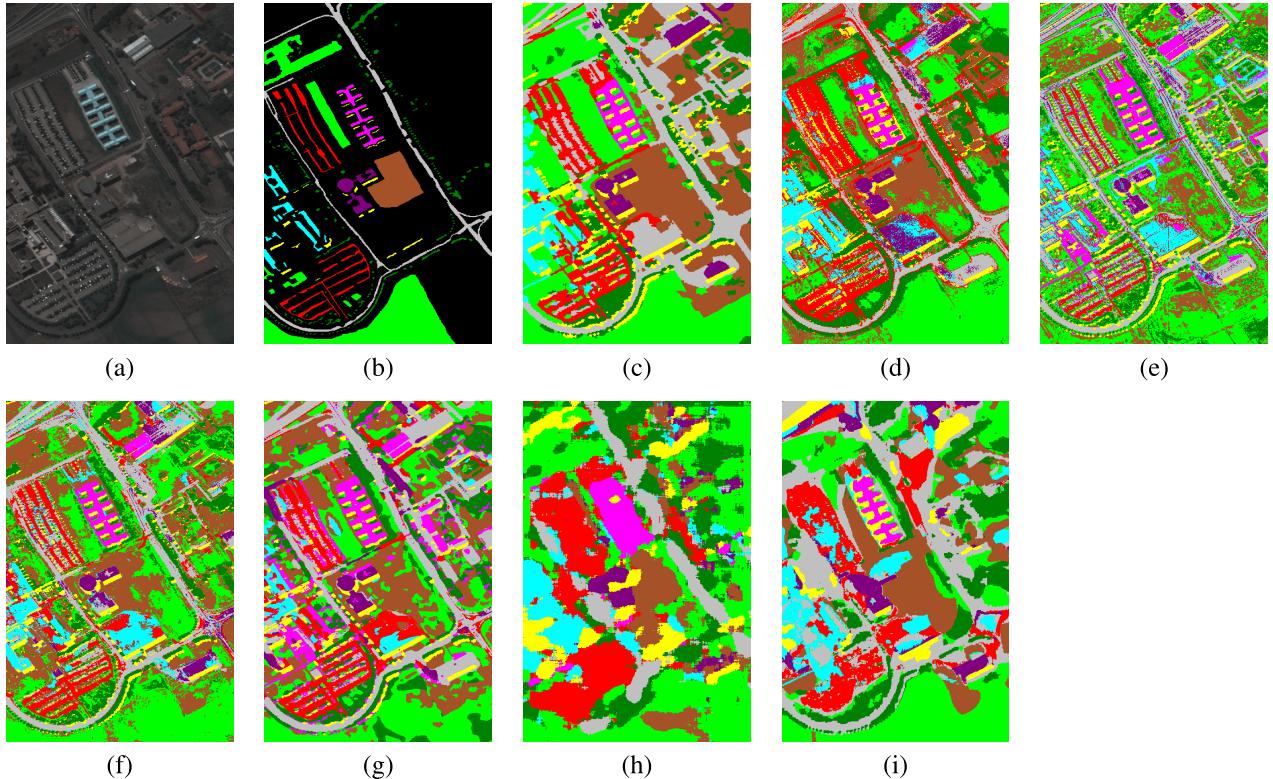


Fig. 12. Pavia University data set. (a) Color composite image. (b) Ground truth. (c)–(h) Classifications maps produced using ten labeled samples for each class. The methods used were: (c) proposed SGL, (d) LCMR [3], (e) SVM [1], (f) SC-MK [13], (g) EPF [47], (h) LBP [15], and (i) IFRF [14].

TABLE IV
COMPARISON OF COMPUTATIONAL TIME(SECONDS) WITH TEN LABELED SAMPLES FOR EACH CLASS

TECHNIQUE	OURS	LCMR [3]	SC-MK [13]	EPF [47]	LBP [15]	IFRF [14]	SVM [1]
Indian Pines	16.9	9.6	2.8	5.2	41.1	2.1	5.8
Salinas	72.5	51.7	10.2	10.5	174.5	3.9	9.5
Pavia University	137.6	91.8	10.7	6.7	268.1	4.6	4.5

V. CONCLUSION

In this article, we have developed a novel transductive semi-semi-supervised graph-based approach, SGL, for the classification of HSIs. Our experiments with real benchmark HSIs demonstrate that our proposed method greatly outperforms other state-of-the-art classifiers in terms of qualitative and quantitative results, especially when using an incredibly small amount of data.

The semi-supervised nature of our solution exploits information present in the unlabelled data and can overcome the issue of having a highly limited training set, a common problem in the field of remote sensing. Furthermore, for the first time we propose using superpixels as the nodes of a pure graphical classifier which has two large benefits. Firstly, the size of the superpixel graph is much smaller than a pixel based graph allowing for computational reasonable run times without the need for matrix approximations. Secondly, applying labels to superpixels intelligent smooths our classification maps with near perfect preservation of edges and boundaries.

Additionally, we modified a common superpixel algorithm MSLIC to create an alternative version HMS which accurately over-segments HSIs. These over-segments are shown to

be more effective than the over-segments produced by other commonly used superpixel methods such as ERS and SLIC.

In our future work, we intend on applying DL to automate the extraction of deep features. Furthermore, we seek to apply recent work on heterogeneous graphs to investigate a combined superpixel/pixel representation.

ACKNOWLEDGMENT

The authors would like to thank Prof. D. Landgrebe from Purdue University and the NASA Jet Propulsion Laboratory for providing the hyperspectral data sets. They would also like to thank Prof. D. Coomes from the Department of Plant Sciences, University of Cambridge, for his advice and support. They would also like to thank the support of NVIDIA Corporation with the donation of a Quadro P6000 GPU used for this research.

REFERENCES

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

- [2] L. Y. Fang, S. T. Li, X. D. Kang, and J. A. Benediktsson, "Spectral-spatial classification of hyperspectral images with a superpixel-based discriminative sparse model," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4186–4201, Aug. 2015.
- [3] 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, Mar. 2018.
- [4] T. Wang, Z. Zhu, and E. Blasch, "Bio-inspired adaptive hyperspectral imaging for real-time target tracking," *IEEE Sensors J.*, vol. 10, no. 3, pp. 647–654, Mar. 2010.
- [5] B. Uzkent, M. J. Hoffman, and A. Vodacek, "Real-time vehicle tracking in aerial video using hyperspectral features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2016, pp. 36–44.
- [6] B. Uzkent, A. Rangnekar, and M. J. Hoffman, "Aerial vehicle tracking by adaptive fusion of hyperspectral likelihood maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 233–242.
- [7] R. J. Ellis and P. W. Scott, "Evaluation of hyperspectral remote sensing as a means of environmental monitoring in the St. Austell China clay (kaolin) region, Cornwall, UK," *Remote Sens. Environ.*, vol. 93, nos. 1–2, pp. 118–130, Oct. 2004.
- [8] S. Manfreda *et al.*, "On the use of unmanned aerial systems for environmental monitoring," *Remote Sens.*, vol. 10, no. 4, p. 641, 2018.
- [9] Z. Pan, G. Healey, M. Prasad, and B. Tromberg, "Face recognition in hyperspectral images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1552–1560, Dec. 2003.
- [10] Y. Liu, G. Gao, and Y. Gu, "Tensor matched subspace detector for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 4, pp. 1967–1974, Apr. 2017.
- [11] Y. Zhang, B. Du, L. Zhang, and T. Liu, "Joint sparse representation and multitask learning for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 894–906, Feb. 2017.
- [12] G. Mercier and M. Lennon, "Support vector machines for hyperspectral image classification with spectral-based kernels," in *Proc. Int. IEEE Geosci. Remote Sens. Symp.*, vol. 1, May 2003, pp. 288–290.
- [13] L. Fang, S. Li, W. Duan, J. Ren, and J. A. Benediktsson, "Classification of hyperspectral images by exploiting spectral-spatial information of superpixel via multiple kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6663–6674, Dec. 2015.
- [14] X. Kang, S. Li, and J. A. Benediktsson, "Feature extraction of hyperspectral images with image fusion and recursive filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3742–3752, Jun. 2014.
- [15] 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.
- [16] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Milan, Italy, Jul. 2015, pp. 4959–4962.
- [17] W. Zhao and S. Du, "Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.
- [18] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.
- [19] B. H. Barlow, "Unsupervised learning," *Neural Comput.*, vol. 1, no. 3, pp. 295–311, Mar. 1989.
- [20] W. Zhu *et al.*, "Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2786–2798, May 2017.
- [21] O. Chapelle, A. Zien, and B. Schölkopf, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [22] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. NIPS*, 2004, pp. 595–602.
- [23] Y.-J. Liu, C.-C. Yu, M.-J. Yu, and Y. He, "Manifold SLIC: A fast method to compute content-sensitive superpixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 651–659.
- [24] G. Camps-Valls, T. Marsheva, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3044–3054, Oct. 2007.
- [25] Y. Gao, R. Ji, P. Cui, Q. Dai, and G. Hua, "Hyperspectral image classification through bilayer graph-based learning," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2769–2778, Jul. 2014.
- [26] B. Cui, X. Xie, X. Ma, G. Ren, and Y. Ma, "Superpixel-based extended random walker for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3233–3243, Jun. 2018.
- [27] L. Bruzzone, M. Chi, and M. Marconcini, "Transductive SVMs for semisupervised classification of hyperspectral data," in *Proc. Int. Geosci. Remote Sens. Symp.*, 2005.
- [28] I. Dópido, J. Li, P. Marpu, A. Plaza, J. Dias, and J. A. Benediktsson, "Semisupervised self-learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, pp. 4032–4044, Jan. 2013.
- [29] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised self-learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2271–2282, Oct. 2013.
- [30] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 212–216, Feb. 2018.
- [31] I. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 2005.
- [32] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. Int. Conf. Comput. Vis.*, 2003, pp. 10–17.
- [33] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
- [34] G. Maierhofer, D. Heydecker, A. I. Aviles-Rivero, S. M. Alsah, and C.-B. Schönlieb, "Peekaboo—where are the objects? Structure adjusting superpixels," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018.
- [35] D. Stutz, A. Hermans, and B. Leibe, "Superpixels: An evaluation of the state-of-the-art," *Comput. Vis. Image Understand.*, vol. 166, pp. 1–27, Jan. 2018.
- [36] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.
- [37] S. Jia, B. Deng, J. Zhu, X. Jia, and Q. Li, "Superpixel-based multitask learning framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2575–2588, May 2017.
- [38] S. Jia, B. Deng, J. Zhu, X. Jia, and Q. Li, "Local binary pattern-based hyperspectral image classification with superpixel guidance," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 749–759, Feb. 2018.
- [39] S. Jia, X. Deng, J. Zhu, M. Xu, J. Zhou, and X. Jia, "Collaborative representation-based multiscale superpixel fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7770–7784, Oct. 2019.
- [40] S. Jia, Z. Lin, B. Deng, J. Zhu, and Q. Li, "Cascade superpixel regularized Gabor feature fusion for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.
- [41] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
- [42] P. Sellars, A. Aviles-Rivero, N. Papadakis, D. Coomes, A. Faul, and C.-B. Schönlieb, "Semi-supervised learning with graphs: Covariance based superpixels for hyperspectral image classification," Jan. 2019, *arXiv:1901.04240*. [Online]. Available: <https://arxiv.org/abs/1901.04240>
- [43] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 29, no. 1, pp. 328–347, Jan. 2007.
- [44] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 589–600.
- [45] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Neural Inf. Process. Syst.*, 2001.
- [46] A. L. Bertozzi and A. Flenner, "Diffuse interface models on graphs for classification of high dimensional data," *Multiscale Model. Simul.*, vol. 10, no. 3, pp. 1090–1118, Jan. 2012.
- [47] X. Kang, S. Li, and J. A. Benediktsson, "Spectral-spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2014.
- [48] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 2097–2104.
- [49] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27–1–27–27, Apr. 2011.



Philip Sellars received the B.A. and M.Sc. degrees from the University of Cambridge, Cambridge, U.K., in 2017, where he is currently pursuing the Ph.D. in mathematics under the supervision of Prof. C.-B. Schönlieb.

His research interests are focused on semi-supervised learning and its applications in computer vision and remote sensing.



Carola-Bibiane Schönlieb received the degree in mathematics from the Institute for Mathematics, University of Salzburg, Salzburg, Austria, in 2004, and the Ph.D. degree from the University of Cambridge, Cambridge, U.K., in 2009.

From 2004 to 2005, she held a teaching position in Salzburg. After one year of post-doctoral activity at the University of Göttingen, Germany, she became a Lecturer with the Department of Applied Mathematics and Theoretical Physics (DAMTP) in 2010, where she was promoted to a Reader in 2015 and

a Professor in 2018. She is currently a Professor of applied mathematics with DAMTP, University of Cambridge, where she is also the Head of the Cambridge Image Analysis Group, the Director of the Cantab Capital Institute for Mathematics of Information, the Director of the EPSRC Centre for Mathematical and Statistical Analysis of Multimodal Clinical Imaging, and has been a fellow of the Jesus College Cambridge since 2011. Her research interests focus on variational methods, partial differential equations and machine learning for image analysis, image processing, and inverse imaging problems.



Angelica I. Aviles-Rivero (Member, IEEE) received the Ph.D. degree from the Polytechnic University of Catalonia, Barcelona, Spain, in 2017.

She is currently a Research Associate with the Department of Pure Mathematics and Mathematical Statistics (DPMMS), University of Cambridge, Cambridge, U.K. Her research lies at the intersection of computational mathematics, computer vision, and machine learning for applications to large-scale real-world problems. Her central research is to develop new data-driven algorithmic techniques that allow computers to gain high-level understanding from vast amounts of data, with the aim of aiding the decisions of users from multiple disciplines. This line of research has allowed her to gain expertise with a wide range of various data types, including medical imaging, computational photography, computer graphics, and remote sensing to name a few.

Dr. Aviles-Rivero is a member of Society for Industrial and Applied Mathematics (SIAM), Societ Mathematique de France (SMF), International Society for Magnetic Resonance in Medicine (ISMRM), and Applied and Computational Mathematics (ACM).