Noname manuscript No.
(will be inserted by the editor)

A two-stage method for spectral-spatial classification of hyperspectral images

In memory of Mila Nikolova

Raymond H. Chan · Kelvin K. Kan
Mila Nikolova · Robert J. Plemmons

Received: date / Accepted: date

Abstract We propose a novel two-stage method for the classification of hyperspectral images. Pixel-wise classifiers, such as the classical support vector machine (SVM), consider spectral information only. As spatial information is not utilized, the classification results are not optimal and the classified image may appear noisy. Many existing methods, such as morphological profiles, superpixel segmentation, and composite kernels, exploit the spatial information. In this paper, we propose a two-stage approach inspired by image denoising and segmentation to incorporate the spatial information. In the first stage, SVMs are used to estimate the class probability for each pixel. In the second stage, a convex variant of the Mumford-Shah model is applied to each probability map to denoise and segment the image into different classes. Our proposed method effectively utilizes both spectral and spatial information of the data sets and is fast as only convex minimization is needed in addition to the SVMs. Experimental results on three widely utilized real hyperspectral data sets indicate that our method is very competitive in accuracy, timing, and the number of parameters when compared with current state-of-the-art methods, especially when the inter-class spectra are similar or the percentage of training pixels is reasonably high.

Keywords Hyperspectral Image Classification · Image Segmentation · Image Denoising · Mumford-Shah Model · Support Vector Machine · Alternating Direction Method of Multipliers

Raymond H. Chan
Department of Mathematics, City University of Hong Kong, Tat Chee Avenue, KLN, Hong Kong
E-mail: rchan.sci@cityu.edu.hk

Kelvin K. Kan
Department of Mathematics, Emory University, Atlanta, GA 30322, USA
E-mail: kelvin.kan@emory.edu

Mila Nikolova
CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France
E-mail: nikolova@cmla.ens-cachan.fr

Robert J. Plemmons
Department of Computer Science and Department of Mathematics, Wake Forest University, Winston-Salem, NC 27106, USA
E-mail: plemmons@wfu.edu

1 Introduction

Remotely-sensed hyperspectral images (HSI) are images taken from drones, airplanes or satellites that record a wide range of electromagnetic spectrum, typically more than 100 spectral bands from visible to near-infrared

wavelengths. Since different materials reflect different spectral signatures, one can identify the materials at each pixel of the image by examining its spectral signatures. HSI is used in many applications, including agriculture [1, 2], disaster relief [3, 4], food safety [5, 6], military [7, 8] and mineralogy [9].

One of the most important problems in hyperspectral data exploitation is HSI classification. It has been an active research topic in past decades [10, 11]. The pixels in the hyperspectral image are often labeled manually by experts based on careful review of the spectral signatures and investigation of the scene. Given these ground-truth labels of some pixels (also called "training pixels"), the objective of HSI classification is to assign labels to part or all of the remaining pixels (the "testing pixels") based on their spectral signatures and their locations.

Numerous methods have been developed for HSI classification. Among these, machine learning is a well-studied approach. It includes multinomial logistic regression [12–14], artificial neural networks [15–19], and support vector machines (SVMs) [20–22]. Since our method is partly based on SVMs, we will discuss it in more detail here. Early SVM classification methods [23, 24] perform pixel-wise classification that utilizes spectral information but not spatial dependencies. Numerous spectral-spatial SVM classification methods have been introduced since then. They show better performance when compared to the pixel-wise SVM classifiers. Here we discuss some of them.

SVMs with composite kernels [25] use composite kernels that are weighted summations of spectral kernels and spatial kernels. The spatial information is extracted by taking the average of the spectra in a fixed window around each pixel. To further utilize the spatial information, the method in [26] first applies superpixel segmentation to break the hyperspectral image into small regions with flexible shapes and sizes. Then it extracts the spatial information based on the segmentation and finally performs the classification using SVMs with multiple kernels. In [27], a pixel-wise SVM classification is first used to produce classification maps, then a partitional clustering is applied to obtain a segmentation of the hyperspectral image. Then a majority vote scheme is used in each cluster and finally a filter is applied to denoise the result. The method in [28] first produces pixel-wise classification maps using SVMs and then applies edge-preserving filtering to the classification maps. In addition to these methods, techniques based on Markov random fields [29], segmentation [26, 27, 30, 31] and morphological profiles [31, 32] have also been incorporated into SVMs to exploit the spatial information.

Besides machine learning approaches, another powerful approach is sparse representation [33]. It is based on the observation that spectral signatures within the same class usually lie in a low-dimensional subspace; therefore test data can be represented by a few atoms in a training dictionary. A joint sparse representation method is introduced in [34] to make use of the spatial homogeneity of neighboring pixels. In particular, each testing pixel and its neighboring pixels inside a fixed window are jointly sparsely represented. In [35], a kernel-based sparse algorithm is proposed which incorporates the kernel functions into the joint sparse representation method. It uses a fixed size local region to extract the spatial information. Approaches with more flexible local regions were proposed in [36] and [37]. They incorporate a multiscale scheme and superpixel segmentation into the joint sparse representation method respectively. Multiple-feature-based adaptive sparse representation was proposed in [38]. It first extracts various spectral and spatial features and then the adaptive sparse representations of the features are computed. The method in [39] first estimates the pixel-wise class probabilities using SVMs, then applies sparse representation to obtain superpixel-wise class probabilities in which spatial information is utilized and the final result is obtained by combining both probabilities.

A pixel-wise classifier such as SVM considers only spectral information. It generates

results with decent accuracy but would appear "noisy" as spatial information is not used, see [23] and also Fig. 1. Segmentation techniques have been used to incorporate the spatial information, see [26, 27, 30, 31]. Indeed, image segmentation is a well-studied subject in image processing and numerous effective segmentation methods for noisy images have been introduced [40–45]. Among them, a variational method called the Mumford-Shah model [40, 41] is one of the most important and successful image segmentation techniques. In this paper, we propose a simple but effective two-stage classification method inspired by our previous methods for image segmentation [43–45] which are based on the Mumford-Shah model. In the first stage, we apply a pixel-wise SVM method that exploits the spectral information to estimate a pixel-wise probability map for each class. In the second stage, we apply a convex variant of the Mumford-Shah model to denoise the maps and exploit the spatial information so as to segment the image into different classes accurately. Traditional methods like that in [27] apply a pixel-wise classification to obtain an initialization. Then they use a segmentation algorithm followed by a denoising algorithm to do the classification. In comparison, in our proposed method, since our convex Mumford-Shah model performs denoising and segmentation simultaneously, we just need one step here. Besides, because of the superior segmentation accuracy of our convex Mumford-Shah model, our method has much better classification results.

Our method utilizes spectral information in the first stage and spatial information in the second stage. Experiments show that our method generates very accurate results when compared to the state-of-the-art methods on real HSI data sets, especially when the interclass spectra are similar. This is because our method can effectively exploit the spatial information even when the other methods cannot distinguish between the spectra. Moreover, our method has a much smaller number of parameters and shorter computation time than the state-of-the-art methods.

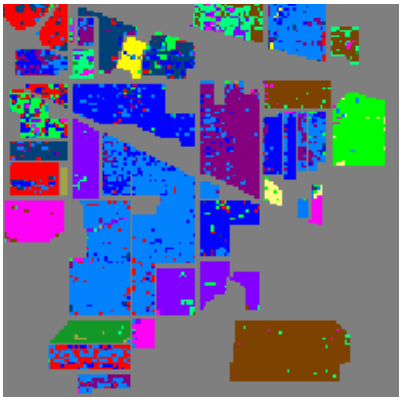


Fig. 1: An example of classification result using pixel-wise SVM classifiers

This paper is organized as follows. In Sect. 2, support vector machines and variational methods for denoising and segmentation are reviewed. In Sect. 3, our proposed two-stage classification method is presented. In Sect. 4, experimental results are presented to illustrate the effectiveness of our method. Sect. 5 concludes the paper.

2 Support Vector Machines and Variational Methods

2.1 Review of ν-Support Vector Classifiers

Support vector machines (SVMs) have been used successfully in pattern recognition [46], object detection [47, 48], and financial time series forecasting [49, 50] etc. This approach also has superior performance in hyperspectral classification, especially when the dimensionality of the data is high and the number of training data is limited [23, 24]. In this subsection, we review the ν-support vector classifier (ν-SVC) [22] which will be used in the first stage of our method.

Consider for simplicity a supervised binary classification problem. We are given m training data $\{\mathbf{x}_i\}_{i=1}^m$ in $\mathbb{R}^{d_1}$, and each data is associated with a binary label $y_i \in \{-1, +1\}$ for

$i = 1, 2, ..., m$. In the training phase of SVM, one aims to find a hyperplane to separate the two classes of labels and maximize the distance between the hyperplane and the closest training data, which is called the support vector. In the kernel SVM, the data is mapped to a higher dimensional feature space by a feature map $\phi : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ in order to improve the separability between the two classes.

The ν-SVC is an advanced support vector classifier which enables the user to specify the maximum training error before the training phase. Its formulation is given as follows:

$$\begin{cases} \min\limits_{\mathbf{w},b,\boldsymbol{\xi},\rho} \frac{1}{2}\|\mathbf{w}\|_2^2 - \nu\rho + \frac{1}{m}\sum\limits_{i=1}^{m}\xi_i \\ \text{subject to:} \\ y_i(\mathbf{w}\cdot\phi(\mathbf{x}_i)+b) \geq \rho - \xi_i,\ i = 1, 2, \ldots, m, \\ \xi_i \geq 0,\ i = 1, 2, \ldots, m, \\ \rho \geq 0, \end{cases} \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^{d_2}$ and $b \in \mathbb{R}$ are the normal vector and the bias of the hyperplane respectively, ξ_i's are the slack variables which allow training errors, and $\rho/\|\mathbf{w}\|_2$ is the distance between the hyperplane and the support vector. The parameter $\nu \in (0, 1]$ is shown to be an upper bound on the fraction of training errors [22].

The optimization problem (1) can be solved through its Lagrangian dual:

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} \ -\frac{1}{2}\sum\limits_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) \\ \text{subject to: } 0 \leq \alpha_i \leq \frac{1}{m},\ i = 1, 2, \ldots, m, \\ \sum\limits_{i=1}^{m}\alpha_i y_i = 0, \\ \sum\limits_{i=1}^{m}\alpha_i \geq \nu. \end{cases} \tag{2}$$

Its optimal Lagrange multipliers can be calculated using quadratic programming methods [51]. After obtaining them, the parameters of the optimal hyperplane can be represented by the Lagrange multipliers and the training data. The decision function for a test pixel $\mathbf{x}$ is given by:

$$\begin{aligned} &g(\mathbf{x}) = \operatorname{sgn}(f(\mathbf{x})), \\ &\text{where } f(\mathbf{x}) = \sum_{i=1}^{m}\alpha_i y_i K(\mathbf{x}_i,\mathbf{x}) + b. \end{aligned} \tag{3}$$

Mercer's Theorem [51, p. 423-424] states that a symmetric function K can be represented as an inner product of some feature maps ϕ, i.e. $K(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x})\cdot\phi(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y}$, if and only if K is positive semi-definite. In that case, the feature map ϕ need not be known in order to perform the training and classification, but only the kernel function K is required. Examples of K satisfying the condition in Mercer's Theorem include: $K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$ and $K(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i\cdot\mathbf{x}_j + 1)^p$.

2.2 Review of Variational Methods for Denoising and Segmentation

Let $\Omega = \{1, ..., N_1\} \times \{1, ..., N_2\}$ be the index set of pixel locations of an image, $\mathbf{v}$ be the noisy image and $\mathbf{u}$ be the restored image. One famous variational method to denoise images with Gaussian noise is the total variation (TV) method [52]. It involves an optimization model with a TV regularization term which corresponds to the function $\|\nabla\cdot\|_1$. However, it is known that it reproduces images with staircase effect, i.e. with piecewise constant regions. One approach to improve it is to add a higher-order term, see, *e.g.*, [53–57]. In [56], the authors considered minimizing

$$H(\mathbf{u}) = \frac{1}{2}\|\mathbf{v}-\mathbf{u}\|_2^2 + \alpha_1\|\nabla\mathbf{u}\|_1 + \frac{\alpha_2}{2}\|\nabla\mathbf{u}\|_2^2. \tag{4}$$

Here the first term is the ℓ_2 data-fitting term that caters for Gaussian noise. The second term is the TV term while the third term is the extra higher order term added to introduce smoothness to the restored image $\mathbf{u}$. By setting the parameters $\{\alpha_i\}_{i=1}^2$ appropriately, one can control the trade off between a piece-wise constant and a piece-wise smooth $\mathbf{u}$.

In [43–45], the authors derived the same minimizational function (4) as a convex approximation of the Mumford-Shad model for segmentation. In [43–45], (4) is first applied to obtain a smooth denoised image and then thresholding is applied to the restored image to obtain the segmentation. The method is successful for segmenting greyscale and color images corrupted by different noises (Gaussian, Poisson, Gamma), information loss and/or blur. We note that the denoising and segmentation are intimately related. Indeed, Cai and Steidl showed that the famous Chan-Vese segmentation model [58] can be obtained by thresholding the TV denoising model with some properly chosen regularization parameter, see [59] for more details.

The 2-stage approach for denoising has also been applied to impulse noise removal, see [60]. In the first stage a standard impulse noise detector, the Adaptive Median Filter [61], is used to detect the locations of possible noisy pixels. Then in the second stage, it restores the noisy pixels while keeping the non-noisy pixels unchanged by minimizing:

$$F(\mathbf{u}) = \|\mathbf{v}-\mathbf{u}\|_1 + \frac{\beta}{2}\|\nabla \mathbf{u}\|_\alpha^\alpha, \quad \text{s.t. } \mathbf{u}|_\Upsilon = \mathbf{v}|_\Upsilon, \tag{5}$$

where Υ is the set of non-noisy pixels detected by the Adaptive Median Filter, $\mathbf{u}|_\Upsilon = (u_i)_{i\in\Upsilon}$, and $1 < \alpha \leq 2$. We remarked that in [62], Nikolova showed that the 1-norm data-fitting term (used in (5) above) is the correct norm for impulse noise. This 2-stage method is the first method that can successfully restore images corrupted with extremely high level of impulse noise (e.g. 90%).

Our proposed method is inspired by the image denoising/segmentation methods in [43–45, 56, 60], which apply (4) successfully to denoise/segment images with various noises. In the first stage of our proposed method, we use the spectral classifer ν-SVC to generate a pixel-wise probability map for each class. Then in the second stage, we use a combination of (4) and (5) to denoise and segment the result from the first stage.

3 Our Two-stage Classification Method

SVMs yield decent classification accuracy [23] but their results can be noisy (see Fig. 1) since only spectral information is used. We therefore propose to use an image denoising/segmentation scheme to incorporate the spatial information into the classification. Our method first estimates the pixel-wise probability map for each class using SVMs. Then the spatial positions of the training pixels are used in a variational denoising/segmentation method to effectively segment the image into different classes.

3.1 First Stage: Pixel-wise Probability Map Estimation

3.1.1 SVM Classifier

HSI classification is a multi-class classification but the SVM is a binary classifier. To extend SVM to multi-class, we use the One-Against-One (OAO) strategy [63] where $[c(c-1)/2]$ SVMs are built to classify every possible pair of classes. Here c is the number of classes. In this paper, we choose the SVM method ν-SVC [22] with OAO strategy for the HSI multiclass classification in our first stage. Moreover, the radial basis function kernel (RBF kernel) [21] is used as the kernel function in our SVM method. The RBF kernel is defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\Big(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\Big). \tag{6}$$

We remark that one can use other SVMs, other multiclass strategies such as the One-Against-All strategy in [63], or other kernel functions such as the polynomial kernel [21] instead.

3.1.2 Probability Estimation of SVM Outputs

Given a testing pixel $\mathbf{x}$ and a SVM classifier with decision function $f(\mathbf{x})$ in (3), we can label

$\mathbf{x}$ with a class according to the sign of $f(\mathbf{x})$, see [21]. Under the OAO strategy, there are $[c(c-1)]/2$ such pairwise functions $f_{h,l}$, $1 \le h, l \le c$, $h \ne l$. We use them to estimate the probability p_h that $\mathbf{x}$ is in the h-th class. The idea is given in [64, 65]. We first estimate the pairwise class probability $\text{Prob}(y = h \mid y = h \text{ or } y = l)$ by computing

$$r_{h,l} = \frac{1}{1 + e^{\eta f_{h,l}(\mathbf{x}) + \tau}}, \tag{7}$$

where η and τ are computed by minimizing a negative log likelihood problem over all the training pixels [64].

Then the probability vector $\mathbf{p} = [p_1, p_2, ..., p_c]^\intercal$ of the testing pixel $\mathbf{x}$ is estimated by solving:

$$\begin{aligned} &\min_{\mathbf{p}} \frac{1}{2} \sum_{h=1}^{c} \sum_{l \ne h} (r_{l,h} p_h - r_{h,l} p_l)^2, \\ &\text{s.t. } p_h \ge 0, \forall h, \sum_{h=1}^{c} p_h = 1. \end{aligned} \tag{8}$$

Its optimal solution can be obtained by solving the following simple $(c+1)$-$(c+1)$ linear system, see [65]:

$$\begin{bmatrix} Q & \mathbf{e} \\ \mathbf{e}^\intercal & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \tag{9}$$

where

$$Q_{hl} = \begin{cases} \sum\limits_{s \ne h} r_{s,h}^2 & \text{if } h = l, \\ -r_{l,h} r_{h,l} & \text{if } h \ne l, \end{cases}$$

b is the Lagrange multiplier of the equality constraint in (8), $\mathbf{e}$ is the c-vector of all ones, and $\mathbf{0}$ is the c-vector of all zeros. In our tests, the probability vectors $\mathbf{p}(\mathbf{x})$ for all testing pixels $\mathbf{x}$ are computed by this method using the toolbox of LIBSVM library [66].

We finish Stage 1 by forming the 3D tensor $\mathcal{V}$ where $\mathcal{V}_{i,j,k}$ gives the probability that pixel (i, j) is in class k. More specifically, if pixel (i, j) is a testing pixel, then $\mathcal{V}_{i,j,:} = \mathbf{p}(\mathbf{x}_{i,j})$; if pixel (i, j) is a training pixel belonging to the c-th class, then $\mathcal{V}_{i,j,c} = 1$ and $\mathcal{V}_{i,j,k} = 0$ for all other k's.

3.2 Second Stage: Denoising/Segmentation of the Pixel-wise Probability Map

Given the probability tensor $\mathcal{V}$ obtained in Stage 1, one can obtain an HSI classification by taking the maximum probability for each pixel [28]. However, the result will appear noisy as no spatial information is taken into account. The goal of our second stage is to incorporate the spatial information into $\mathcal{V}$ by a denoising/segmentation method that keeps the value of the training pixels unchanged during the optimization, as their ground-truth labels are given a priori.

Let $\mathbf{v}_k := \mathcal{V}_{:,:,k}$, $k = 1, ..., c$, be the probability map of the k-th class obtained from Stage 1. We improve them by minimizing:

$$\begin{aligned} &\min_{\mathbf{u}} \frac{1}{2} \|\mathbf{u} - \mathbf{v}_k\|_2^2 + \beta_1 \|\nabla \mathbf{u}\|_1 + \frac{\beta_2}{2} \|\nabla \mathbf{u}\|_2^2, \\ &\text{s.t. } \mathbf{u}|_\Upsilon = \mathbf{v}_k|_\Upsilon, \end{aligned} \tag{10}$$

where β_1, β_2 are regularization parameters and Υ is the set of training pixels. We choose this minimization functional because it gives superb performance in denoising [56] and segmentation [43–45]. The higher-order $\|\nabla \mathbf{u}\|_2^2$ term encourages smoothness of the solution and can improve the classification accuracy, see Sect. 4.4. In our tests, we use anisotropic TV [67] and periodic boundary condition for the discrete gradient operator, see [68, p. 258].

Alternating direction method of multipliers (ADMM) [69] is used to solve (10). First, we rewrite (10) as follows:

$$\begin{aligned} &\min_{\mathbf{u}} \frac{1}{2} \|\mathbf{u} - \mathbf{v}_k\|_2^2 + \beta_1 \|\mathbf{s}\|_1 + \frac{\beta_2}{2} \|D\mathbf{u}\|_2^2 + \iota_{\mathbf{w}} \\ &\text{s.t. } \mathbf{s} = D\mathbf{u} \text{ and } \mathbf{w} = \mathbf{u}. \end{aligned} \tag{11}$$

Here D denotes the discrete operator of ∇, $D = \begin{pmatrix} D_x \\ D_y \end{pmatrix} \in \mathbb{R}^{2n \times n}$, where D_x and D_y are the first-order difference matrices in the horizontal and vertical directions respectively and n is the total number of pixels in the hyperspectral image, $\iota_{\mathbf{w}}$ is the indicator function, where $\iota_{\mathbf{w}} =$

0 if $\mathbf{w}|_{\Upsilon} = \mathbf{v}_k|_{\Upsilon}$ and $\iota_{\mathbf{w}} = \infty$ otherwise. Its augmented Lagrangian is given by:

$$\begin{aligned} &L(\mathbf{u}, \mathbf{s}, \mathbf{w}, \boldsymbol{\lambda}) \\ &= \frac{1}{2}\|\mathbf{u} - \mathbf{v}_k\|_2^2 + \beta_1\|\mathbf{s}\|_1 + \frac{\beta_2}{2}\|D\mathbf{u}\|_2^2 \\ &\quad + \iota_{\mathbf{w}} + \frac{\mu}{2}\|E\mathbf{u} - \mathbf{g} - \boldsymbol{\lambda}\|_2^2, \end{aligned} \tag{12}$$

where $\mu > 0$ is a positive constant, $E = \begin{pmatrix} D \\ I \end{pmatrix}$, $\mathbf{g} = \begin{pmatrix} \mathbf{s} \\ \mathbf{w} \end{pmatrix}$ and $\boldsymbol{\lambda} = \begin{pmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{pmatrix}$ the Lagrange multipliers.

The formulation (12) allows us to solve $\mathbf{u}$ and $\mathbf{g}$ alternately as follows:

$$\begin{aligned} \mathbf{u}^{(t+1)} = \underset{\mathbf{u}}{\operatorname{argmin}} \ &\left\{ \frac{1}{2}\|\mathbf{u} - \mathbf{v}_k\|_2^2 + \frac{\beta_2}{2}\|D\mathbf{u}\|_2^2 \right. \\ &\left. + \frac{\mu}{2}\|E\mathbf{u} - \mathbf{g}^{(t)} - \boldsymbol{\lambda}^{(t)}\|_2^2 \right\} \end{aligned} \tag{13a}$$

$$\begin{aligned} \mathbf{g}^{(t+1)} = \underset{\mathbf{g}}{\operatorname{argmin}} \ &\left\{ \beta_1\|\mathbf{s}\|_1 + \iota_{\mathbf{w}} \right. \\ &\left. + \frac{\mu}{2}\|E\mathbf{u}^{(t+1)} - \mathbf{g} - \boldsymbol{\lambda}^{(t)}\|_2^2 \right\} \end{aligned} \tag{13b}$$

$$\boldsymbol{\lambda}^{(t+1)} = \boldsymbol{\lambda}^{(t)} - E\mathbf{u}^{(t+1)} + \mathbf{g}^{(t+1)} \tag{13c}$$

The $\mathbf{u}$-subproblem (13a) is a least squares problem. Its solution is

$$\begin{aligned} \mathbf{u}^{(t+1)} = &(I + \beta_2 D^\intercal D + \mu E^\intercal E)^{-1} \\ &(\mathbf{v}_k + \mu E^\intercal(\mathbf{g}^{(t)} + \boldsymbol{\lambda}^{(t)})). \end{aligned} \tag{14}$$

Since periodic boundary conditions are used, the solution can be computed efficiently using the two-dimensional fast Fourier transform (FFT) [70] in $O(n \log n)$ complexity.

For the $\mathbf{g}$-subproblem (13b), the optimal $\mathbf{s}$ and $\mathbf{w}$ can be computed separately as follows:

$$\begin{aligned} \mathbf{s}^{(t+1)} = \underset{\mathbf{s}}{\operatorname{argmin}} \ &\left\{ \beta_1\|\mathbf{s}\|_1 \right. \\ &\left. + \frac{\mu}{2}\|D\mathbf{u}^{(t+1)} - \mathbf{s} - \boldsymbol{\lambda}_1^{(t)}\|_2^2 \right\} \end{aligned} \tag{15}$$

and

$$\mathbf{w}^{(t+1)} = \underset{\mathbf{w}}{\operatorname{argmin}} \ \left\{ \iota_{\mathbf{w}} + \frac{\mu}{2}\|\mathbf{u}^{(t+1)} - \mathbf{w} - \boldsymbol{\lambda}_2^{(t)}\|_2^2 \right\} \tag{16}$$

The solution of (15) can be obtained by soft thresholding [71]:

$$\begin{aligned} [\mathbf{s}^{(t+1)}]_i = \operatorname{sgn}([\mathbf{r}]_i) \cdot \max\{|[\mathbf{r}]_i| - \frac{\beta_1}{\mu}, 0\}, \\ i = 1, ..., 2n, \end{aligned} \tag{17}$$

where $\mathbf{r} = D\mathbf{u}^{(t+1)} - \boldsymbol{\lambda}_1^{(t)}$. The solution of (16) is simply

$$[\mathbf{w}^{(t+1)}]_i = \begin{cases} [\mathbf{v}_k]_i & \text{if } i \in \Upsilon, \\ [\mathbf{u}^{(t+1)} - \boldsymbol{\lambda}_2^{(t)}]_i & \text{otherwise.} \end{cases} \tag{18}$$

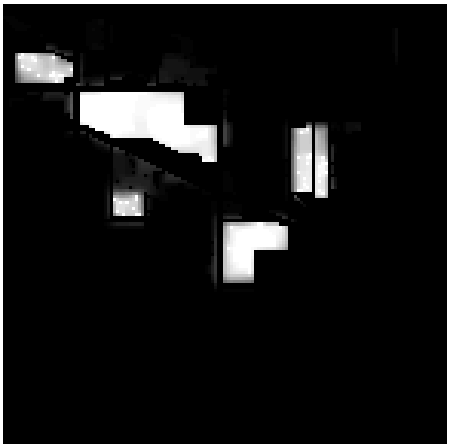
The computation of (13c), (17) and (18) have a computational complexity of $O(n)$. Hence the computational complexity of our ADMM is $O(n \log n)$ for each iteration, where n is the total number of pixels.

The convergence of our ADMM to the global minimum is guaranteed by [69]. Once it finishes, we obtain the enhanced probability map $\mathbf{u}$ for class k. We denote it as $\mathcal{U}_{:,:,k}$. After the map for each class is obtained, we get a 3D tensor $\mathcal{U}$. The final classification of the (i, j)-th pixel is given by finding the maximum value in $\mathcal{U}_{i,j,:}$, i.e. $\underset{k}{\operatorname{argmax}}\ \mathcal{U}_{i,j,k}$. Our proposed method is summarized in Algorithm 1.

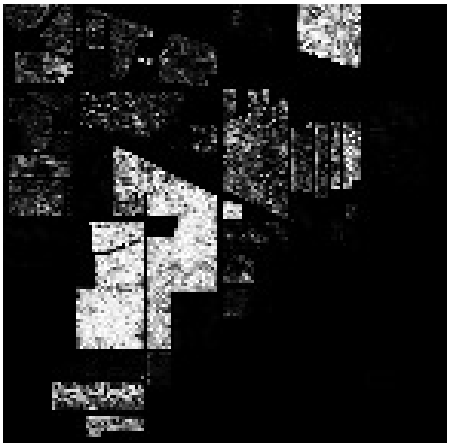
We remark that in Stage 1, the operation is along the spectral dimension, i.e. the third index of the tensor, while in Stage 2, the operation is along the spatial dimension, i.e. the first two indices of the tensor. The techniques of Stage 2 are essentially similar to our segmentation methods in [43–45], where a smooth denoised image is first computed and then thresholding (here maximizing) is applied to it to segment (here classify) it.

Fig. 2 shows the probability maps before the second stage and the enhanced probability maps after the second stage. The figures are in gray scale, i.e., completely white represents probability one and completely black represents probability zero. Note that the second stage does not guarantee the enhanced probability maps to have a sum to one property. In Figures 2b and 2d, the enhanced probability maps are normalized to sum to one.

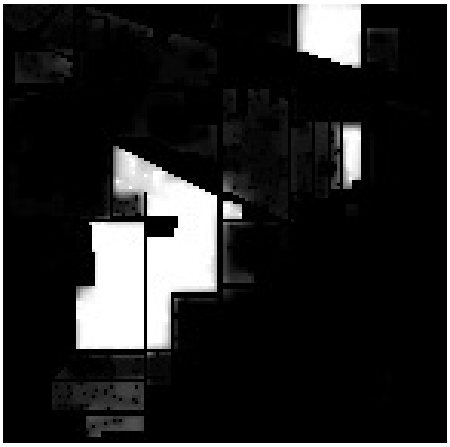
(a) Probability map of class 2 before the second stage



(b) Normalized probability map of class 2 after the second stage



(c) Probability map of class 11 before the second stage



(d) Normalized probability map of class 11 after the second stage

Fig. 2: Examples of probability maps on Indian Pines before and after the second stage. Here, completely white represents probability one and completely black represents probability zero.

4 Experimental Results

4.1 Experimental Setup

4.1.1 Data Sets

Three commonly-tested hyperspectral data sets are used in our experiments. These data sets have pixels labeled so that we can compare the methods quantitatively. The first one is the "Indian Pines" data set acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pines test site in North-western Indiana. It has a spatial resolution of 20 m per pixel and a spectral coverage ranging from 0.2 to 2.4 μm in 220 spectral bands. However, due to water absorption, 20 of the spectral bands (the 104-108th, 150-163th and 220th bands) are discarded in experiments in previous papers. Therefore our data set is of size $145 \times 145 \times 200$, and there are 16 classes in the given ground-truth labels.

Algorithm 1 Our two-stage method

Stage 1: Estimation of pixel-wise probability using SVMs
for all pairs of classes **do**
  Solve for the SVM:

$$\begin{cases} \max\limits_{\alpha} \ -\frac{1}{2}\sum\limits_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) \\ \text{subject to: } 0\le\alpha_i\le\frac{1}{m},\ i=1,2,\ldots,m, \\ \qquad\qquad \sum\limits_{i=1}^{m}\alpha_i y_i = 0, \\ \qquad\qquad \sum\limits_{i=1}^{m}\alpha_i \ge \nu \end{cases}$$

end for
for all pixels $\mathbf{x}_{i,j}$ **do**
  for all pairs of classes $1\le h,l\le c$ **do**
    Compute: $r_{h,l}=\frac{1}{1+e^{\eta f_{h,l}(\mathbf{x}_{i,j})+\tau}}$
  end for
  Solve for the probability $\mathbf{p}_{i,j}$:
  $\begin{bmatrix} Q & \mathbf{e} \\ \mathbf{e}^\intercal & 0 \end{bmatrix}\begin{bmatrix} \mathbf{p}_{i,j} \\ b \end{bmatrix}=\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$
end for
Output of stage 1: probability maps $\mathcal{V}$, where $\mathcal{V}_{i,j,:}=\mathbf{p}_{i,j}$
Stage 2: Denoising/Segmentation using ADMM
for $k=1,2,...,c$ **do**
  Initialize
  Set $\mathbf{v}_k=\mathcal{V}_{:,:,k}$ and $t=0$. Choose $\mu>0$, $\mathbf{u}_k^{(0)}$, $\mathbf{s}^{(0)}$, $\boldsymbol{\lambda}^{(0)}$ and $\mathbf{w}^{(0)}$, where $\mathbf{w}^{(0)}|_\Upsilon=\mathbf{v}_k|_\Upsilon$
  while stopping criterion is not satisfied **do**
    $\mathbf{u}_k^{(t+1)}\leftarrow(I+\beta_2 D^\intercal D+\mu E^\intercal E)^{-1}(\mathbf{v}_k+\mu E^\intercal(\mathbf{g}^{(t)}+\boldsymbol{\lambda}^{(t)}))$
    $\mathbf{s}^{(t+1)}\leftarrow \mathrm{sgn}(\mathbf{r})\cdot\max\{|\mathbf{r}|-\frac{\beta_1}{\mu},0\}$, where $\mathbf{r}=D\mathbf{u}_k^{(t+1)}-\boldsymbol{\lambda}_1^{(t)}$
    $\mathbf{w}^{(t+1)}|_{\Omega\setminus\Upsilon}\leftarrow(\mathbf{u}_k^{(t+1)}-\boldsymbol{\lambda}_2^{(t)})|_{\Omega\setminus\Upsilon}$
    $\boldsymbol{\lambda}^{(t+1)}\leftarrow\boldsymbol{\lambda}^{(t)}-E\mathbf{u}_k^{(t+1)}+\mathbf{g}^{(t+1)}$
  end while
end for
Classification result of the (i,j)-th pixel: $\underset{k}{\mathrm{argmax}}\ \mathcal{U}_{i,j,k}$, where $\mathcal{U}_{:,:,k}=\mathbf{u}_k$

The second and third images are the "University of Pavia" and "Pavia Center" data sets acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor over Pavia in northern Italy. The sensor has 1.3 m spatial resolution and spectral coverage ranging from 0.43 to 0.86 μm. The data set sizes are 610×340×103 and 1096 × 715 × 102 respectively, where the third dimension is the spectral dimension. Both sets have 9 classes in the ground-truth labels.

4.1.2 Methods Compared and Parameters Used

We have compared our method with five well-known classification methods: ν-support vector classifiers (ν-SVC) [22, 23] (i.e. the first stage of our method), SVMs with composite kernels (SVM-CK) [25], edge-preserving filtering (EPF) [28], superpixel-based classification via multiple kernels (SC-MK) [26] and multiple-feature-based adaptive sparse representation (MFASR) [38]. All the tests are run on a laptop computer with an Intel Core i5-7200U CPU, 8 GB RAM and the software platform is MATLAB R2016a.

In the experiments, the parameters are chosen as follows. For the ν-SVC method, the parameters are obtained by performing a five-fold cross-validation [72]. For the SVM-CK method, the parameters are tuned such that it gives the highest classification accuracy. All parameters of the EPF method, the SC-MK method, and the MFASR method are chosen as stated in [26, 28, 38] respectively, except the window size in the EPF method, the number of superpixels and the parameters of the superpixel segmentation algorithm in the SC-MK method, and the sparsity level of the MFASR are tuned such that the highest classification accuracies are obtained. For our method, the parameters of the ν-SVC (1) in the first stage are obtained by performing a five-fold cross-validation and the parameters of the optimization problem (10) in the second stage are tuned such that it gives the highest classification accuracy. The optimal parameters in the second stage β_1 and β_2 are 0.4 and 3; 0.1 and 3; 0.2 and 4 for Indian Pines, University of Pavia and Pavia Center

respectively. By [69], Algorithm 1 converges for any $\mu > 0$, so μ is fixed as 5 for all the tests on the three data sets.

4.1.3 Performance Metrics

To quantitatively evaluate the performance of the methods, we use the following three widely-used metrics: (i) overall accuracy (OA): the percentage of correctly classified pixels, (ii) average accuracy (AA): the average percentage of correctly classified pixels over each class, and (iii) kappa coefficient (kappa): the percentage of correctly classified pixels corrected by the number of agreements that would be expected purely by chance [73].

For each method, we perform the classification ten times where each time we randomly choose a different set of training pixels. In the tables below, we give the averages of these metrics over the ten runs. The accuracies are given in percentage, and the highest accuracy of each category is listed in boldface. In order to graphically show the classification results in an objective way, we also count the number of mis-classifications for each testing pixel over the ten runs. The numbers of mis-classifications are shown in the corresponding heatmap figures, with the heatmap colorbar indicating the number of mis-classifications.

4.2 Classification Results

4.2.1 Indian Pines

The Indian Pines data set consists mainly of big homogeneous regions and has very similar inter-class spectra (see Fig. 3 for the spectra of the training pixels of Indian Pines data where there are three similar classes of corns, three similar classes of grasses and three similar classes of soybeans). It is therefore very difficult to classify it if only spectral information is used. In the experiments, we choose exactly the same number of training pixels as in [26, 37] and they amount to about 10% of the pixels from each class. Some classes have small numbers of pixels and hence 10 pixels are taken as training pixels for each of these classes. The rest of the labeled pixels are used as testing pixels.

The number of training and testing pixels as well as the classification accuracies obtained by different methods are reported in Table 1. We see that our method generates the best results for all three metrics (OA, AA and kappa) and outperforms the comparing methods by a significant margin. They are at least 0.95% higher than the others. Also, the second stage of our method improves the overall accuracy of ν-SVC (used in the first stage of our method) by almost 20%.

Fig. 4 shows the heatmaps of mis-classifications. The results of the ν-SVC, SVM-CK and EPF methods produce large area of mis-classifications. The SC-MK also produces mis-classification at the top-right region and the middle-right region which are soybeans-clean and soybeans-no till respectively. This shows that SC-MK cannot distinguishing these two similar classes well. The heatmap of MFASR method contains scattered regions of mis-classification. In contrast, our method generates smaller regions of mis-classifications and less errors as it effectively utilizes the spatial information to give an accurate result.

4.2.2 University of Pavia

The University of Pavia data set consists of regions with various shapes, including thin and thick structures and large homogeneous regions. Hence it can be used to test the ability of the classification methods on handling different shapes. In the experiments, we choose the same number of training pixels (200 for each class) as in [26]. This accounts for approximately 4% of the labeled pixels. The remaining ones are used as testing pixels.

Table 2 reports the classification accuracies obtained by different methods. We see that the performance of SC-MK, MFASR, and our method are very close: approximately 99% in all three metrics (OA, AA and kappa)

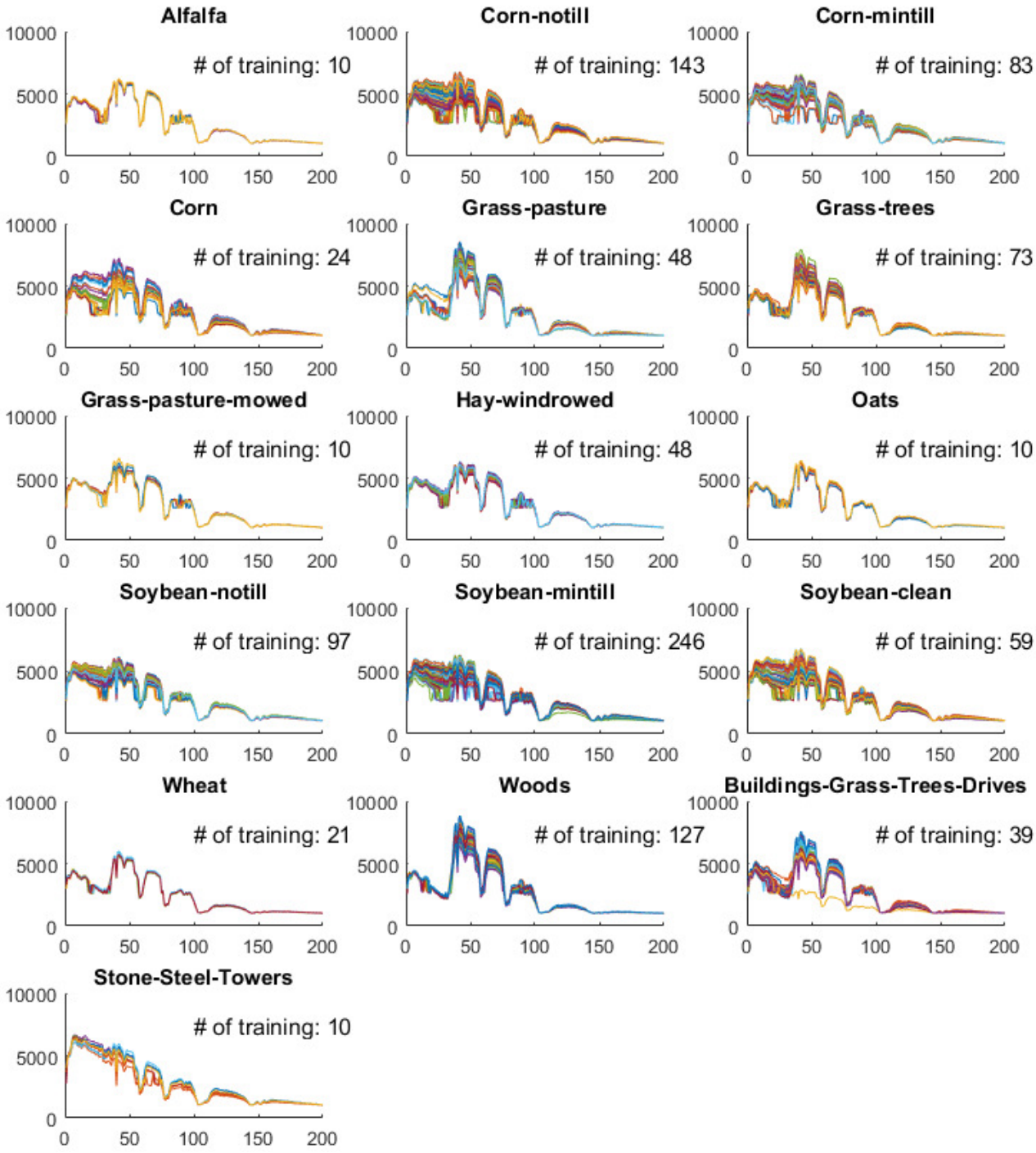
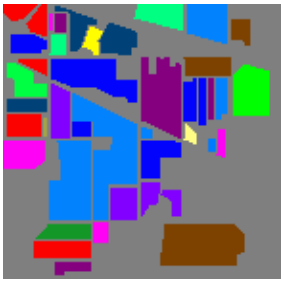
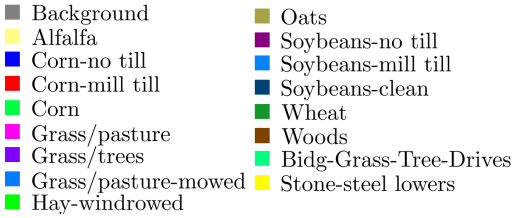


Fig. 3: Spectra of training pixels of Indian Pines data

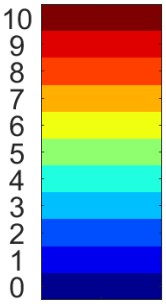


(a) Ground Truth

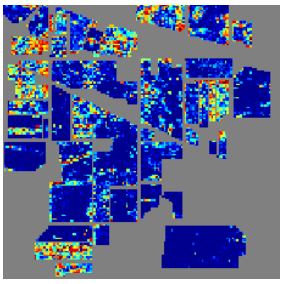


(b) Label color

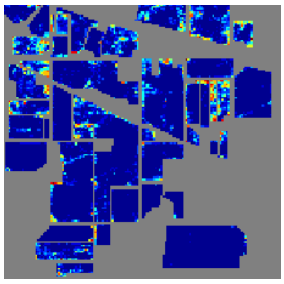
(c) False color image



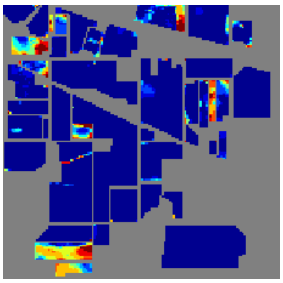
(d) Heatmap colorbar



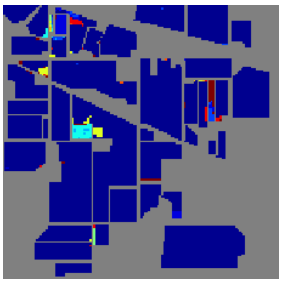
(e) ν-SVC [22, 23]



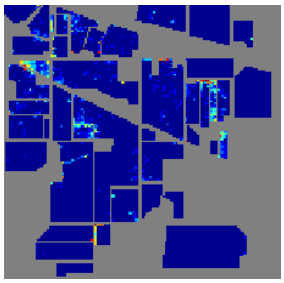
(f) SVM-CK [25]



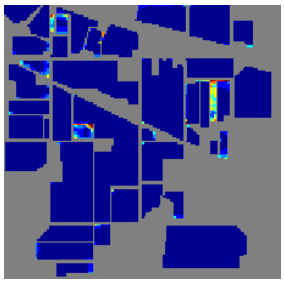
(g) EPF [28]



(h) SC-MK [26]



(i) MFASR [38]



(j) Our method

Fig. 4: Indian Pines data set. (a) ground-truth labels, (b) label color of the ground-truth labels, (c) false color image, (d) heatmap colorbar, (e)–(j) classification results by different methods.

Table 1: Number of training/testing pixels and classification accuracies for Indian Pines data set.

Class	train/test	ν-SVC	SVM-CK	EPF	SC-MK	MFASR	Our method
Alfalfa	10/36	70.28%	81.94%	97.29%	**100%**	98.06%	99.17%
Corn-no till	143/1285	77.90%	89.98%	96.03%	95.44%	96.66%	**97.89%**
Corn-mill till	83/747	67.80%	89.68%	97.75%	97.16%	97.94%	**98.73%**
Corn	24/213	52.96%	86.24%	93.03%	**99.25%**	91.69%	99.01%
Grass/pasture	48/435	89.13%	93.31%	**99.17%**	96.67%	94.62%	96.92%
Grass/trees	73/657	96.15%	98.98%	96.02%	99.70%	99.56%	**99.74%**
Grass/pasture-mowed	10/18	93.33%	96.11%	99.47%	**100%**	**100%**	**100%**
Hay-windrowed	48/430	93.93%	98.42%	**100%**	**100%**	99.98%	**100%**
Oats	10/10	90.00%	**100%**	96.25%	**100%**	**100%**	**100%**
Soybeans-no till	97/875	72.26%	88.81%	92.21%	94.62%	**96.03%**	96.01%
Soybeans-mill till	246/2209	79.71%	91.57%	86.65%	**98.80%**	98.58%	99.54%
Soybeans-clean	59/534	67.66%	85.90%	96.26%	96.29%	97.06%	**99.64%**
Wheat	21/184	96.09%	98.64%	**100%**	99.67%	99.57%	**100%**
Woods	127/1138	91.89%	96.85%	95.24%	**99.99%**	99.89%	99.91%
Bridg-Grass-Tree-Drives	39/347	56.97%	88.01%	93.70%	98.39%	98.01%	**99.14%**
Stone-steel lowers	10/83	85.66%	98.43%	96.11%	97.71%	**98.92%**	96.39%
OA		79.78%	92.11%	93.34%	97.83%	97.88%	**98.83%**
AA		80.11%	92.68%	95.95%	98.35%	97.91%	**98.88%**
kappa		0.769	0.910	0.924	0.975	0.976	**0.987**

and they outperform the ν-SVC, SVM-CK and EPF methods. However, we note that MFASR requires twice the number of parameters as ours and 12 times longer to run, see Tables 7–8. Fig. 5 shows the heatmaps of mis-classifications. The ν-SVC, SVM-CK and EPF methods produce large regions of mis-classifications. The SC-MK method produces many mis-classifications at the middle and bottom regions where the meadows are. The MFASR method and our method generate smaller regions of mis-classification.

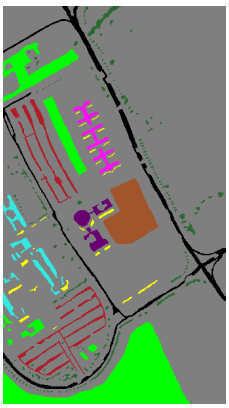
4.2.3 Pavia Center

The Pavia Center data set also consists of regions with various shapes. In the experiments, we use the same number of training pixels as in [31] (150 training pixels per class). This accounts for approximately 1% of the labeled pixels. The rest of the labeled pixels are used as testing pixels. Table 3 reports the number of training/testing pixels and the classification accuracies of different methods. We see that the EPF method gives the highest OA and kappa while our method gives the second highest and their values differ by about 0.1%. However, our method gives the highest AA (99.12%) which outperforms the EPF method by almost 1%. The SC-MK and MFASR methods give slightly worse accuracies than our method. Fig. 6 shows the heatmaps of mis-classifications.

4.3 Advantages of Our 2-stage Method

4.3.1 Percentage of Training Pixels

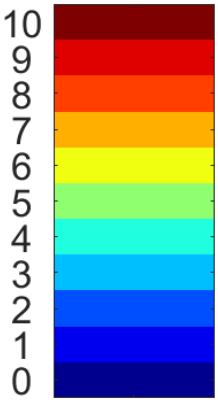
Since our method improves on the classification accuracy by using spatial information, it is expected to be a better method if the training percentage (percentage of training pixels) is higher. To verify that, Tables 4-6 show the overall accuracies obtained by our method on the three data sets with different levels of training percentage. We see that our method out-



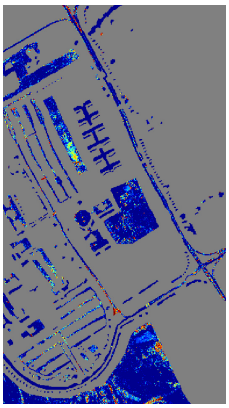
(a) Ground Truth

(b) Label color

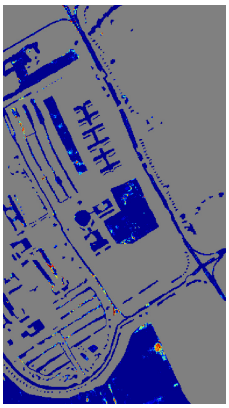
(c) False color image



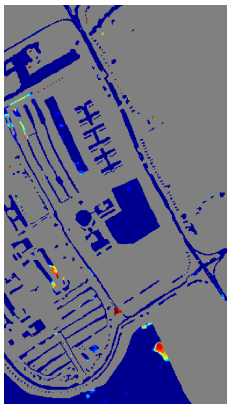
(d) Heatmap colorbar



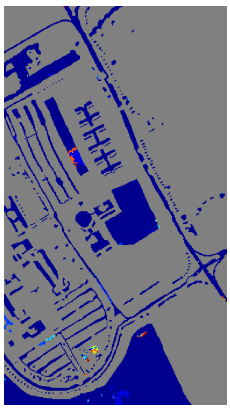
(e) ν-SVC [22, 23]



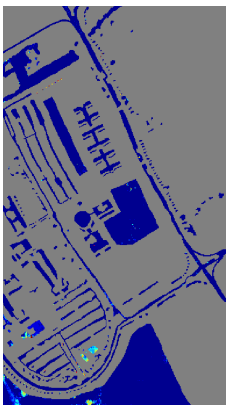
(f) SVM-CK [25]



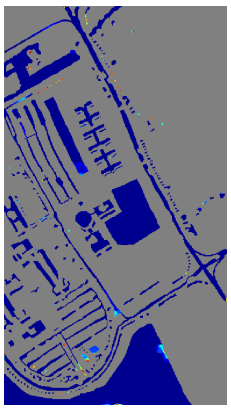
(g) EPF [28]



(h) SC-MK [26]

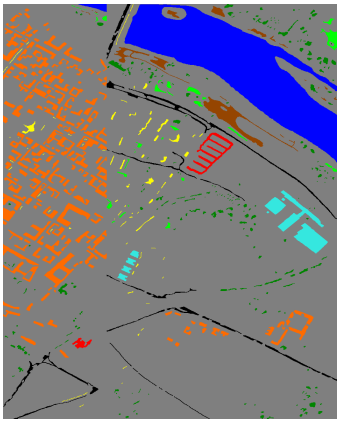


(i) MFASR [38]



(j) Our method

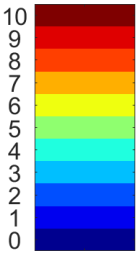
Fig. 5: University of Pavia data set. (a) ground-truth labels, (b) label color of the ground-truth labels, (c) false color image, (d) heatmap colorbar, (e)–(j) classification results by different methods.



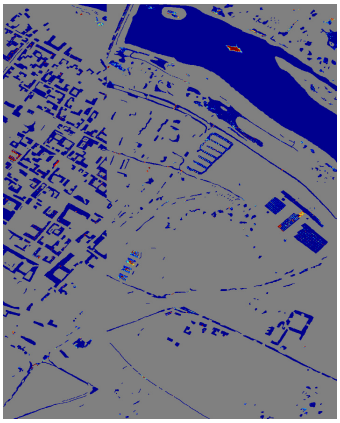
(a) Ground Truth

(b) Label color

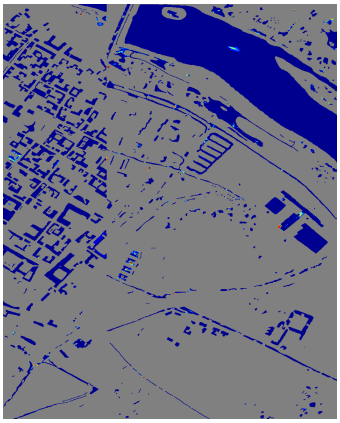
(c) False color image



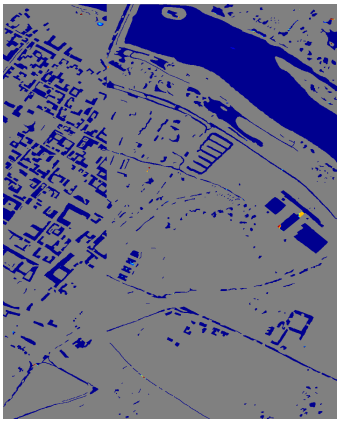
(d) Heatmap colorbar



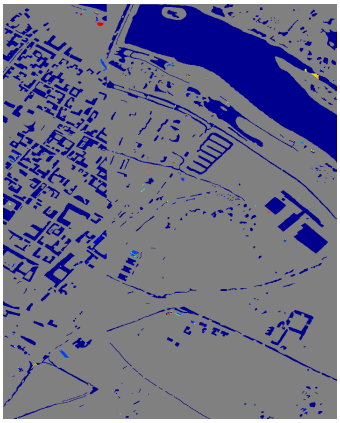
(e) ν-SVC [22, 23]



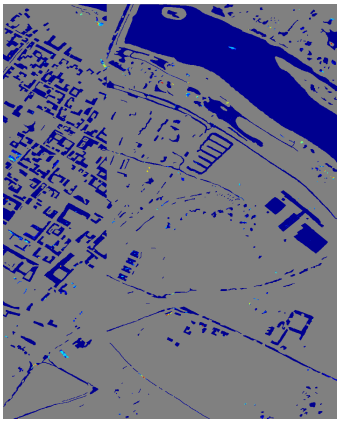
(f) SVM-CK [25]



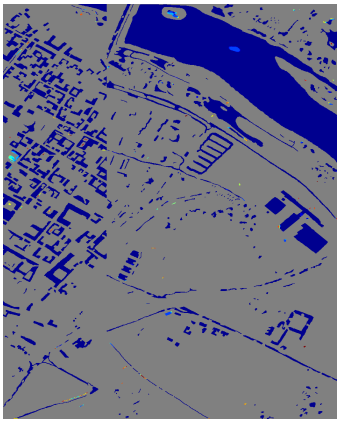
(g) EPF [28]



(h) SC-MK [26]



(i) MFASR [38]



(j) Our method

Fig. 6: Pavia Center data set. (a) ground-truth labels, (b) label color of the ground-truth labels, (c) false color image, (d) heatmap colorbar, (e)–(j) classification results by different methods.

Table 2: Number of training/testing pixels and classification accuracies for University of Pavia data set.

Class	train/test	ν-SVC	SVM-CK	EPF	SC-MK	MFASR	Our method
Asphalt	200/6431	84.65%	95.84%	98.84%	99.06%	**99.44%**	98.68%
Meadows	200/18449	89.96%	97.62%	99.62%	98.14%	98.52%	**98.78%**
Gravel	200/1899	83.59%	91.99%	95.50%	**99.98%**	99.80%	99.69%
Trees	200/2864	94.94%	97.95%	98.94%	**99.03%**	98.02%	96.56%
Metal Sheets	200/1145	99.59%	99.97%	99.03%	99.87%	99.91%	**100%**
Bare Soil	200/4829	90.69%	97.49%	92.95%	99.70%	99.78%	**100%**
Bitumen	200/1130	92.73%	98.41%	93.84%	100%	99.92%	**100%**
Bricks	200/3482	82.59%	92.71%	92.92%	99.05%	**99.41%**	99.02%
Shadows	200/747	99.60%	99.92%	99.30%	99.99%	**100%**	99.18%
OA		89.16%	96.80%	97.60%	98.83%	**99.02%**	98.89%
AA		90.93%	96.88%	96.77%	**99.42%**	**99.42%**	99.10%
kappa		0.857	0.957	0.968	0.984	**0.987**	0.985

Table 3: Number of training/testing pixels and classification accuracies for Pavia Center data set.

Class	train/test	ν-SVC	SVM-CK	EPF	SC-MK	MFASR	Our method
Water	150/65128	99.54%	99.82%	**100%**	99.86%	99.97%	99.66%
Trees	150/6357	94.22%	95.61%	**99.11%**	94.59%	95.52%	98.61%
Meadows	150/2741	95.14%	96.15%	97.16%	98.78%	98.54%	**98.84%**
Bricks	150/2002	92.56%	97.37%	90.08%	99.91%	99.62%	**99.98%**
Soil	150/6399	94.31%	96.51%	99.40%	**99.76%**	99.59%	98.69%
Asphalt	150/7375	95.94%	97.34%	98.86%	99.24%	98.76%	**99.60%**
Bitumen	150/7137	89.99%	94.75%	**99.79%**	98.64%	99.55%	97.86%
Tiles	150/2972	97.42%	99.33%	**99.97%**	99.32%	99.05%	99.52%
Shadows	150/2015	99.98%	**100%**	99.96%	99.85%	99.97%	99.27%
OA		97.54%	98.80%	**99.59%**	99.31%	99.33%	99.42%
AA		95.46%	97.43%	98.26%	98.88%	98.95%	**99.12%**
kappa		0.965	0.983	**0.994**	0.990	0.990	0.991

performs the other methods once the training percentage is reasonably high enough (6% for Example 1, 10% for Example 2, and 3% for Example 3). When it is not high, our method still gives a classification accuracy that is very close to the best method compared.

4.3.2 Model Complexity and Computation Time

Tables 7 and 8 show the computation time required and the number of parameters for all methods. We note that the reported timing does not count the time required to find the optimal set of parameters. The ν-SVC, SVM-CK and EPF methods have fast computation time because of the simplicity of their models. They have only a few parameters (2, 3 and 4 respectively). However, from the results in Sect. 4.2, they are worse than the other three methods. The SC-MK method is a good method in terms of accuracy and timing, but it has 9 parameters. The MFASR method has 10 parameters and the longest computation time. In comparison, our method has 5 parameters (2 parameters ν

Table 4: Classification results on the Indian Pines data with different levels of training pixels.

Method \Training percentage	6%	8%	10%	12%	14%
ν-SVC	75.24%	77.87%	79.78%	81.50%	82.40%
SVM-CK	87.92%	90.41%	92.11%	93.30%	94.28%
EPF	91.14%	92.35%	93.34%	94.64%	95.92%
SC-MK	97.39%	97.52%	97.83%	97.83%	97.94%
MFASR	96.59%	97.63%	97.88%	98.29%	98.47%
Our method	**97.51%**	**98.28%**	**98.83%**	**99.06%**	**99.26%**
Difference from the best	0.00 %	0.00 %	0.00 %	0.00 %	0.00%

Table 5: Classification results on the University of Pavia data with different levels of training pixels.

Method \Training percentage	4%	6%	8%	10%	12%
ν-SVC	89.16%	89.74%	91.19%	91.34%	91.80%
SVM-CK	96.80%	97.54%	97.93 %	98.24%	98.47%
EPF	97.60%	98.05%	98.37%	98.49%	98.56%
SC-MK	98.83%	99.24%	**99.67%**	99.45%	99.52%
MFASR	**99.02%**	**99.39%**	99.52%	99.60%	99.68%
Our method	98.89%	99.30%	99.58%	**99.63%**	**99.74%**
Difference from the best	0.13 %	0.09%	0.09%	0.00 %	0.00%

Table 6: Classification results on the Pavia Center data with different levels of training pixels.

Method \Training percentage	1%	2%	3%	4%	5%
ν-SVC	97.54%	98.01%	98.17%	98.28%	98.38%
SVM-CK	98.80%	99.46%	99.59%	99.67%	99.74%
EPF	**99.59%**	**99.76%**	99.73%	99.76%	99.86%
SC-MK	99.31%	99.59%	99.71%	99.75%	99.80%
MFASR	99.33%	99.64%	99.73%	99.87%	99.86%
Our method	99.42%	99.73%	**99.80%**	**99.90%**	**99.92%**
Difference from the best	0.17 %	0.03%	0.00 %	0.00 %	0.00%

and σ for the ν-SVC (1) and the RBF kernel (6) respectively in the first stage, 2 parameters β_1 and β_2 for the denoising model (10) in the second stage and 1 parameter μ for the ADMM algorithm (12)). It has much better (if not the best) classification accuracies with slightly longer computation time than those of ν-SVC, SVM-CK and EPF.

4.4 Effect of the Second-order Term

Here we examine empirically the importance of the term $\|\nabla \mathbf{u}\|_2^2$ in (10). Fig. 7 shows the heatmaps of mis-classifications on the Indian Pines data by using our method with and without $\|\nabla \mathbf{u}\|_2^2$ over ten runs. The training pixels are randomly selected and consist of 2.5% of the labeled pixels. Fig. 7 (a) shows the ground-truth labels. Figures 7 (b)–(d) show the heatmaps of mis-classifications of the ν-SVC classifier (i.e. the first stage of our method), the

Table 7: Comparison of number of parameters.

	ν-SVC	SVM-CK	EPF	SC-MK	MFASR	Our method
Number of parameters	2	3	4	9	10	5

Table 8: Comparison of computation times (in seconds)

Data	size/training %	ν-SVC	SVM-CK	EPF	SC-MK	MFASR	Our method
Indian Pines	$145 \times 145 \times 200/10\%$	5.98	6.32	6.92	9.44	119	8.24
University of Pavia	$610 \times 340 \times 103/4\%$	24.02	32.12	28.53	39.47	443	35.97
Pavia Center	$1096 \times 715 \times 102/1\%$	58.46	81.63	118	107	2599	145

second stage of our method without the $\|\nabla \mathbf{u}\|_2^2$ term, and the second stage of our method with the $\|\nabla \mathbf{u}\|_2^2$ term respectively. Recall the term $\|\nabla \mathbf{u}\|_2^2$ control the smoothness of the final probability maps and the final classification result is determined by taking the maximum over this map of each class. By choosing the parameter associated with the term appropriately, we can then control the level of shrinking or expanding the homogeneous regions in the final classification result. From Fig. 7 (c), when the term is dropped, the mis-classification regions at the top left and bottom left of the first stage result are not only still mis-classified, but the numbers of mis-classification increase. In contrast, when the term is kept, we see from Fig. 7 (d) that the numbers of mis-classification are significantly lowered. Moreover, most of the mis-classified regions of the first stage result are now correctly classified when the parameters are chosen appropriately.

5 Conclusions and Future Work

In this paper, we propose a novel two-stage hyperspectral classification method inspired by image denoising/segmentation. The method is simple yet performs effectively. In the first stage, a support vector machine method is used to estimate the pixel-wise probability map of each class. The result in the first stage has decent accuracy but is noisy. In the second stage, a convex variant of the Mumford-Shah model is applied to denoise and classify the hyperspectral image into different classes. Since both spectral and spatial information are effectively utilized, our method is very competitive when compared with state-of-the-art hyperspectral data classification methods. It also has a simpler framework with fewer numbers of parameters and faster computation times. It performs particularly well when the inter-class spectra are close or when the training percentage is high.

For future work, we plan to investigate the use of deep learning methods in the first stage [16–19]. We will also investigate the use of automated parameter selection [74–77] of the variational method in the second stage. Additionally, we plan on using our methods for classifying fused hyperspectral and LiDAR data [18, 78, 79].

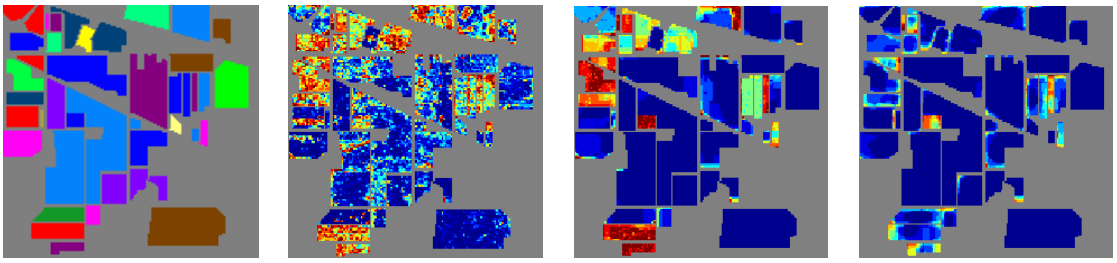
Acknowledgements The authors would like to thank the Computational Intelligence Group from the Basque University for sharing the hyperspectral data sets in their website[1], Prof. Leyuan Fang from College of Electrical and Information Engineering at Hunan University for providing the programs of the SC-MK and MFASR methods in his homepage[2] and Prof. Xudong Kang from College of Electrical and Information Engineering at Hunan University for providing the program of the EPF method in his homepage[3].

Raymond H. Chan's research is supported by HKRGC Grants No. CUHK14306316, CityU Grant: 9380101, CRF Grant C1007-15G, AoE/M-05/12.

[1] `http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes`

[2] `http://www.escience.cn/people/LeyuanFang`

[3] `http://xudongkang.weebly.com/`



(a) Ground Truth (b) ν-SVC [22, 23] (c) Our method without $\|\nabla \mathbf{u}\|_2^2$ (d) Our method with $\|\nabla \mathbf{u}\|_2^2$

Fig. 7: Heatmaps of mis-classifications on Indian Pines data. (a) ground-truth labels, (b) ν-SVC (the first stage), (c) and (d) our method without and with the second order term respectively.

Kelvin K. Kan's research is supported by US Air Force Office of Scientific Research under grant FA9550-15-1-0286. Mila Nikolova's research is supported by the French Research Agency (ANR) under grant No ANR-14-CE27-001 (MIRIAM) and by the Isaac Newton Institute for Mathematical Sciences for support and hospitality during the programme Variational Methods and Effective Algorithms for Imaging and Vision, EPSRC grant no EP/K032208/1. Robert J. Plemmons' research is supported by HKRGC Grant No. CUHK14306316 and US Air Force Office of Scientific Research under grant FA9550-15-1-0286.

References

1. N. Patel, C. Patnaik, S. Dutta, A. Shekh, and A. Dave, "Study of crop growth parameters using airborne imaging spectrometer data," *International Journal of Remote Sensing*, vol. 22, no. 12, pp. 2401–2411, 2001.
2. B. Datt, T. R. McVicar, T. G. Van Niel, D. L. Jupp, and J. S. Pearlman, "Preprocessing EO-1 hyperion hyperspectral data to support the application of agricultural indexes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 6, pp. 1246–1259, 2003.
3. M. Trierscheid, J. Pellenz, D. Paulus, and D. Balthasar, "Hyperspectral imaging or victim detection with rescue robots," in *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*, pp. 7–12, IEEE, 2008.
4. M. T. Eismann, A. D. Stocker, and N. M. Nasrabadi, "Automated hyperspectral cueing for civilian search and rescue," *Proceedings of the IEEE*, vol. 97, no. 6, pp. 1031–1055, 2009.
5. R. Lu and Y.-R. Chen, "Hyperspectral imaging for safety inspection of food and agricultural products," in *Pathogen Detection and Remediation for Safe Eating*, vol. 3544, pp. 121–134, International Society for Optics and Photonics, 1999.
6. A. Gowen, C. O'Donnell, P. Cullen, G. Downey, and J. Frias, "Hyperspectral imaging–an emerging process analytical tool for food quality and safety control," *Trends in Food Science & Technology*, vol. 18, no. 12, pp. 590–598, 2007.
7. D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE signal processing magazine*, vol. 19, no. 1, pp. 29–43, 2002.
8. D. W. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, "Anomaly detection from hyperspectral imagery," *IEEE signal processing magazine*, vol. 19, no. 1, pp. 58–69, 2002.
9. B. Hörig, F. Kühn, F. Oschütz, and F. Lehmann, "Hymap hyperspectral remote sensing to detect hydrocarbons," *International Journal of Remote Sensing*, vol. 22, no. 8, pp. 1413–1422, 2001.
10. G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing:

A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 3, pp. 247–259, 2011.
11. M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2013.
12. J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4085–4098, 2010.
13. J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 3, pp. 809–823, 2012.
14. J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 2, pp. 318–322, 2013.
15. J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 480–491, 2005.
16. J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral–spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sensing Letters*, vol. 6, no. 6, pp. 468–477, 2015.
17. K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, pp. 4959–4962, IEEE, 2015.
18. S. Morchhale, V. P. Pauca, R. J. Plemmons, and T. C. Torgersen, "Classification of pixel-level fused hyperspectral and LiDAR data using deep convolutional neural networks," in *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pp. 1–5, Aug 2016.
19. B. Pan, Z. Shi, and X. Xu, "R-vcanet: a new deep-learning-based hyperspectral image classification method," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 5, pp. 1975–1986, 2017.
20. B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, ACM, 1992.
21. C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
22. B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural computation*, vol. 12, no. 5, pp. 1207–1245, 2000.
23. F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on geoscience and remote sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.
24. G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.
25. G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 93–97, 2006.
26. 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 Transactions on Geo-*

science and Remote Sensing, vol. 53, no. 12, pp. 6663–6674, 2015.
27. Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, "Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2973–2987, 2009.
28. X. Kang, S. Li, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification with edge-preserving filtering," *IEEE transactions on geoscience and remote sensing*, vol. 52, no. 5, pp. 2666–2677, 2014.
29. Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "Svm-and mrf-based method for accurate classification of hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 4, pp. 736–740, 2010.
30. P. Ghamisi, J. A. Benediktsson, and M. O. Ulfarsson, "Spectral–spatial classification of hyperspectral images based on hidden markov random fields," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2565–2574, 2014.
31. T. Liu, Y. Gu, J. Chanussot, and M. Dalla Mura, "Multimorphological superpixel model for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 6950–6963, 2017.
32. M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using svms and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3804–3814, 2008.
33. A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.
34. Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, 2011.
35. Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 217–231, 2013.
36. L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification via multiscale adaptive sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7738–7749, 2014.
37. L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4186–4201, 2015.
38. L. Fang, C. Wang, S. Li, and J. A. Benediktsson, "Hyperspectral image classification via multiple-feature-based adaptive sparse representation," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 7, pp. 1646–1657, 2017.
39. S. Li, T. Lu, L. Fang, X. Jia, and J. A. Benediktsson, "Probabilistic fusion of pixel-level and superpixel-level hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7416–7430, 2016.
40. D. Mumford and J. Shah, "Boundary detection by minimizing functionals," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 17, pp. 137–154, San Francisco, 1985.
41. D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on pure and applied mathematics*, vol. 42, no. 5, pp. 577–685, 1989.
42. A. Morar, F. Moldoveanu, and E. Gröller, "Image segmentation based on active contours without edges," in *2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing*,

pp. 213–220, IEEE, 2012.
43. X. Cai, R. Chan, and T. Zeng, "A two-stage image segmentation method using a convex variant of the mumford–shah model and thresholding," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 368–390, 2013.
44. R. Chan, H. Yang, and T. Zeng, "A two-stage image segmentation method for blurry images with poisson or multiplicative gamma noise," *SIAM Journal on Imaging Sciences*, vol. 7, no. 1, pp. 98–127, 2014.
45. X. Cai, R. Chan, M. Nikolova, and T. Zeng, "A three-stage approach for segmenting degraded color images: Smoothing, lifting and thresholding (SLaT)," *Journal of Scientific Computing*, vol. 72, no. 3, pp. 1313–1332, 2017.
46. M. Pontil and A. Verri, "Support vector machines for 3d object recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 6, pp. 637–646, 1998.
47. I. El-Naqa, Y. Yang, M. N. Wernick, N. P. Galatsanos, and R. M. Nishikawa, "A support vector machine approach for detection of microcalcifications," *IEEE transactions on medical imaging*, vol. 21, no. 12, pp. 1552–1563, 2002.
48. E. Osuna, R. Freund, and F. Girosit, "Training support vector machines: an application to face detection," in *Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on*, pp. 130–136, IEEE, 1997.
49. F. E. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *omega*, vol. 29, no. 4, pp. 309–317, 2001.
50. K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1, pp. 307–319, 2003.
51. V. N. Vapnik and V. Vapnik, *Statistical learning theory*, vol. 1. Wiley New York, 1998.
52. L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
53. D. Mumford, "Elastica and computer vision," in *Algebraic geometry and its applications*, pp. 491–506, Springer, 1994.
54. T. Chan, A. Marquina, and P. Mulet, "High-order total variation-based image restoration," *SIAM Journal on Scientific Computing*, vol. 22, no. 2, pp. 503–516, 2000.
55. J. Shen, S. H. Kang, and T. F. Chan, "Euler's elastica and curvature-based inpainting," *SIAM journal on Applied Mathematics*, vol. 63, no. 2, pp. 564–592, 2003.
56. M. Hintermüller and G. Stadler, "An infeasible primal-dual algorithm for total bounded variation–based inf-convolution-type image restoration," *SIAM Journal on Scientific Computing*, vol. 28, no. 1, pp. 1–23, 2006.
57. K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.
58. T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on image processing*, vol. 10, no. 2, pp. 266–277, 2001.
59. X. Cai and G. Steidl, "Multiclass segmentation by iterated rof thresholding," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 237–250, Springer, 2013.
60. R. H. Chan, C.-W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Transactions on image processing*, vol. 14, no. 10, pp. 1479–1485, 2005.
61. H. Hwang and R. A. Haddad, "Adaptive median filters: new algorithms and results," *IEEE Transactions on image processing*, vol. 4, no. 4, pp. 499–502, 1995.
62. M. Nikolova, "A variational approach to remove outliers and impulse noise," *Journal of Mathematical Imaging and Vision*,

vol. 20, no. 1-2, pp. 99–120, 2004.
63. C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
64. H.-T. Lin, C.-J. Lin, and R. C. Weng, "A note on platt's probabilistic outputs for support vector machines," *Machine learning*, vol. 68, no. 3, pp. 267–276, 2007.
65. T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 975–1005, 2004.
66. C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
67. X.-L. Zhao, F. Wang, T.-Z. Huang, M. K. Ng, and R. J. Plemmons, "Deblurring and sparse unmixing for hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 4045–4058, 2013.
68. R. C. Gonzales and R. E. Woods, *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.
69. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
70. R. H.-F. Chan and X.-Q. Jin, *An introduction to iterative Toeplitz solvers*, vol. 5. SIAM, 2007.
71. P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.
72. R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, pp. 1137–1145, Montreal, Canada, 1995.
73. J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.
74. H. Liao, F. Li, and M. K. Ng, "Selection of regularization parameter in total variation image restoration," *JOSA A*, vol. 26, no. 11, pp. 2311–2320, 2009.
75. Y. Dong, M. Hintermüller, and M. M. Rincon-Camacho, "Automated regularization parameter selection in multi-scale total variation models for image restoration," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 82–104, 2011.
76. Y.-W. Wen and R. H. Chan, "Parameter selection for total-variation-based image restoration using discrepancy principle," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1770–1781, 2012.
77. K. Bredies, Y. Dong, and M. Hintermüller, "Spatially dependent regularization parameter selection in total generalized variation models for image restoration," *International Journal of Computer Mathematics*, vol. 90, no. 1, pp. 109–123, 2013.
78. P. Gader, A. Zare, R. Close, J. Aitken, and G. Tuell, "Muufl gulfport hyperspectral and LiDAR airborne data set," *Univ. Florida, Gainesville, FL, USA, Tech. Rep. REP-2013-570*, 2013.
79. C. Debes, A. Merentitis, R. Heremans, J. Hahn, N. Frangiadakis, T. van Kasteren, W. Liao, R. Bellens, A. Pižurica, S. Gautama, *et al.*, "Hyperspectral and LiDAR data fusion: Outcome of the 2013 grss data fusion contest," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2405–2418, 2014.
80. J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.
81. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipli-

ers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

82. M. K. Ng, R. H. Chan, and W.-C. Tang, "A fast algorithm for deblurring models with neumann boundary conditions," *SIAM Journal on Scientific Computing*, vol. 21, no. 3, pp. 851–866, 1999.