Efficient, Robust, and Similarity-constrained Algorithm for Subspace Learning and Clustering with Applications in Compressive Imaging

M.Sc. Carlos Alberto Hinojosa Montero

Doctoral thesis to qualify for the title of Doctor of Philosophy in Computer Science

Advisor

Henry Arguello

Ph.D. in Electrical and Computer Engineering

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática.

Bucaramanga

2022

Dedicatory

I dedicate my doctoral thesis to my mother, family, and friends, who have brought me unconditional support over all the years.

Content

page

List of Figures

List of Tables

List of Appendices

List of Abbreviations

SI Spectral Imaging

HSI Hyperspectral Imaging

CS Compressive Sensing

CSI Compressive Spectral Imaging

SSC Sparse Subspace Clustering

LRR Low Rank Representation

SC Subspace Clustering

PCA Principal Component Analysis

SVD Singular Value Decomposition

LSA Local Subspace Affinity

LLMC Locally Linear Manifold Clustering

SLBF Spectral Local Best-fit Flats

OMP Orthogonal Matching Pursuit

RIP Restricted Isometry Property

RP Random Projection

3D-CASSI Spatial-spectral Coded Compressive Spectral Imager

SP Single-pixel Camera

Resumen

Título: Algoritmo eficiente, robusto y con restricciones de similitud para el aprendizaje subespacial y la agrupación en clústeres con aplicaciones en imágenes comprimidas [*]

Autor: Carlos Alberto Hinojosa Montero [**]

Palabras Clave: Clasificación no supervisada de imágenes comprimidas, algoritmo de agrupación, aprendizaje no supervisado.

Descripción: El uso directo del algoritmo *Sparse Subspace Clustering* (SSC) en HSI presenta tres problemas principales: (1) Las HSI generalmente tienen ruido debido a su alta variabilidad de la información espectral y el ruido del sensor, lo que puede disminuir el rendimiento del agrupamiento si no se maneja adecuadamente; (2) Debido a la alta dimensionalidad de las HSI y la complejidad computacional de SSC, la mayoría de los algoritmos de agrupamiento trabajan en pequeñas regiones de interés seleccionadas; (3) dado que SSC ignora la información espacial en los HSI, su capacidad de discriminación es limitada, lo que dificulta la homogeneidad espacial de los resultados del agrupamiento. Esta tesis propone diseñar, desarrollar y simular un método eficiente, robusto y con restricciones de similitud para el agrupamiento subespacial, el cual pretende abordar estos tres problemas mencionados anteriormente. Específicamente, el capítulo 2 explora la inclusión de un término de regularización que reduce el ruido y mejora la precisión del agrupamiento. El capítulo 3 propone un algoritmo escalable que extrae características espaciales y realiza la agrupación de manera eficiente con alta precisión. El capítulo 4 adopta ideas de los dos capítulos anteriores para desarrollar un algoritmo rápido y robusto para HSI. Finalmente, los capítulos 5 y 6 proponen diferentes estrategias para realizar el agrupamiento directamente en las medidas comprimidas, evitando la reconstrucción.

---

[*] Tesis de doctorado

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Henry Arguello Fuentes, Doctorado en Ciencias de la Computación.

Abstract

Title: Efficient, Robust, and Similarity-constrained Algorithm for Subspace Learning and Clustering with Applications in Compressive Imaging [*]

Author: Carlos Alberto Hinojosa Montero [**]

Keywords: Compressive image clustering, spectral-spatial unsupervised classification, subspace clustering, unsupervised learning.

Description: Sparse Subspace Clustering (SSC) has become a popular tool for unsupervised learning because of its high performance. However, directly using SSC on HSIs poses three main problems: (1) HSIs typically have noise due to their high variability of the spectral information and sensor noise (acquisition), which may decrease the clustering performance if not handled appropriately; (2) Due to the high-dimensional nature of HSIs and computational complexity of SSC, most clustering algorithms can not be performed on the full HSIs but selected small regions of interest; (3) since SSC ignores the spatial information in the HSIs, their discrimination capability is limited, hampering the clustering results' spatial homogeneity. This thesis proposes designing, developing, and simulating an efficient, robust, and similarity-constrained method for subspace clustering to address these three relevant issues. Specifically, chapter 2 first explores the inclusion of a regularization term that reduces the noise and improves the clustering accuracy but still has high computational complexity. Then, chapter 3 proposes a scalable algorithm that extracts spatial features and performs clustering efficiently with high accuracy. Chapter 4 adopts ideas from the previous two chapters to develop a fast and robust algorithm. Finally, chapters 5 and 6 adopt the ideas developed in the previous chapters and propose different strategies to perform clustering and avoid expensive reconstruction.

---

[*] Doctoral Thesis

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Henry Arguello Fuentes, Doctorado en Ciencias de la Computación.

Introduction

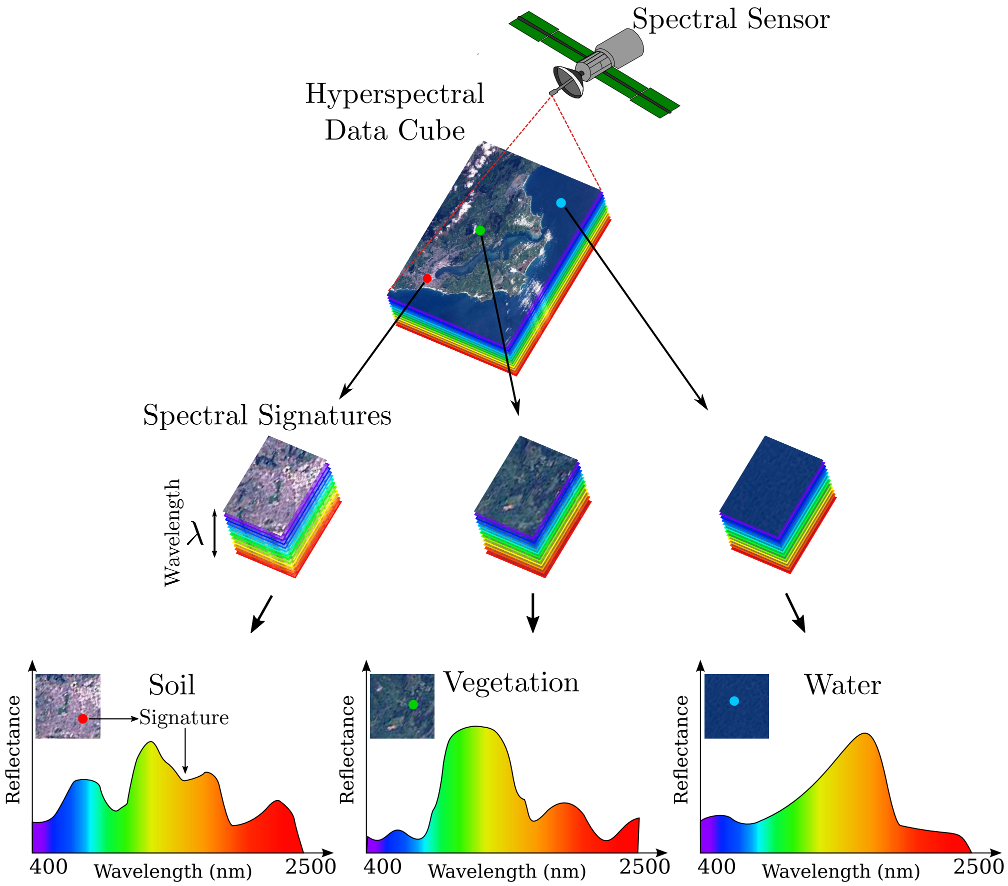


Figure 1. Hyperspectral images. Different materials usually reflect electromagnetic energy differently at specific wavelengths, the spectral signatures allow discriminating materials within an HSI.

Spectral remote sensing systems acquire the Earth's surface information by sensing a large amount of spatial data at different electromagnetic radiation frequencies. Hyperspectral images (HSI) are commonly regarded as three-dimensional (3D) datasets, or data cubes with two dimensions in the spatial domain (x,y) and one in the spectral domain (λ) (Shaw and Burke, 2003). As depicted in Fig. 1, every spatial location in an HSI is represented by a vector whose values correspond to the intensity at different spectral bands. These vectors are also known as the spectral signature of the pixels or spectral pixels. Since different materials usually reflect electromagnetic energy differently at specific wavelengths (Shaw and Burke, 2003), the information provided by the spectral signatures allows distinguishing different physical materials and objects within a scene.

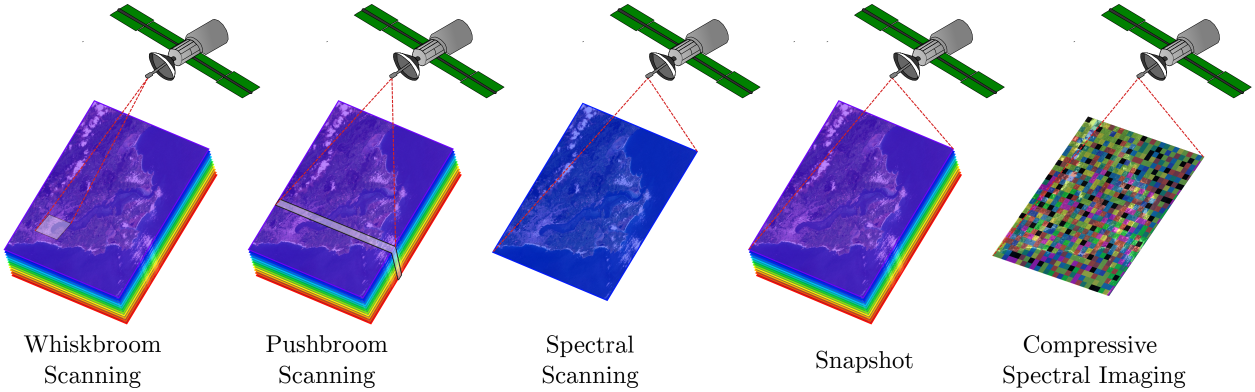


Figure 2. Spectral imaging techniques.

Many different techniques for spectral imaging have been developed over the years, see Fig. 2. For instance, traditional spectral imaging methods, such as Whiskbroom (Green et al., 1998), Pushbroom (Herrala et al., 1994), and tunable filter (spectral scanning) imagers (Morris et al., 1994), scan adjacent zones of the underlying spectral scene and merge the results to construct a spectral 3D data cube. On the other hand, snapshot spectral imaging captures the spatial and spectral information of a scene by mapping all the voxels of the spectral data cube into different regions of a large focal plane array (FPA) (Gao et al., 2010; Kerekes and Schott, 2007). In general, these traditional techniques require all voxels of the 3D scene to be measured. Then, as the spatial or spectral resolution increases, the number of voxels to be sensed increases proportionally, leading to an increment in the cost of sensing, storing and transmitting an spectral image acquired through these methods. However, there is a novel imaging approach that relies on compressive sensing theory (Donoho, 2006) to acquire a compressed version of an HSI and then reconstruct a full version if necessary. Specifically, compressive sprectral imaging (CSI) captures two-dimensional (2D) coded and dispersed projections of the 3D scene rather than direct measurements of the voxels. Therefore, a far less number of samples are acquired compared to traditional techniques, yielding an improvement in the sensing speed.

The research in hyperspectral remote sensing is mainly focused on developing new and automatic methods for hyperspectral imagery analysis, including spectral classification (Liu et al., 2021), image denoising (Shi et al., 2021b), and spectral clustering. In particular, the efficient clustering or unsupervised classification of hyperspectral images is an important task for many practical applications, such as precision agriculture (Lanthier et al., 2008), vegetation classification (Thenkabail and Lyon, 2016), urban land use mapping (He et al., 2021), change detection (Shi et al., 2021a), monitoring and management of the environment (Gessesse et al., 2015; Volpi and Ferrari, 2015), as well as security and defense issues (Briottet et al., 2006).

Accurate HSI clustering is challenging because of the high-dimensional feature space, and it has drawn widespread attention in remote sensing (Ghamisi et al., 2017b; Li et al., 2019). In the past decade, significant efforts have been made in the development of numerous HSI classification methods; however, most of them rely on supervised approaches (Sanchez et al., 2019a; Hinojosa et al., 2019b). More recently, with the blooming of deep learning techniques for extensive data analysis, several deep neural networks have been developed to extract high-level features of HSIs achieving state-of-the-art supervised classification performance (Paoletti et al., 2019). However, the success of such deep learning approaches hinges on a large amount of labeled data, which is not always available and often prohibitively expensive to acquire. As a result, the remote sensing community is currently focused on developing unsupervised methods that can adapt to new conditions without requiring a massive amount of data (Kolesnikov et al., 2019; Yue et al., 2021).

The sparse subspace clustering (SSC) algorithm is one of the most successful unsupervised learning methods in the literature. SSC exploits the fact that high-dimensional data can be well represented as the union of low-dimensional subspaces. Under this assumption, SSC captures the relationship among all data points by exploiting the *self-expressiveness* property (Elhamifar and Vidal, 2013). This property states that each data point in a union of subspaces can be written as a linear combination of other points from its own subspace. Then, the set of solutions is restricted to be sparse by minimizing the ℓ_1 norm. Finally, an affinity matrix is built using the obtained sparse coefficients, and the normalized spectral clustering algorithm (Von Luxburg, 2007a) is applied to achieve the final clustering map.

Assuming that spectral pixels with a similar spectrum approximately belong to the same low-dimensional structure, the SSC algorithm has been successfully applied to HSIs (Hinojosa et al., 2021c; Cai et al., 2020; Hinojosa et al., 2018a,a; Zhang et al., 2016; Zhai et al., 2016; Huang et al., 2019). Despite the great success of SSC in HSI clustering, two main problems are well-known: (1) under the context of HSI, the SSC model only captures the relationship of pixels by analyzing the spectral features without considering the spatial information. Indeed, the sparse coefficient matrix obtained by SSC should be piecewise smooth since spectral pixels belonging to the same land cover material are arranged in a common region; hence there is a spatial relation-

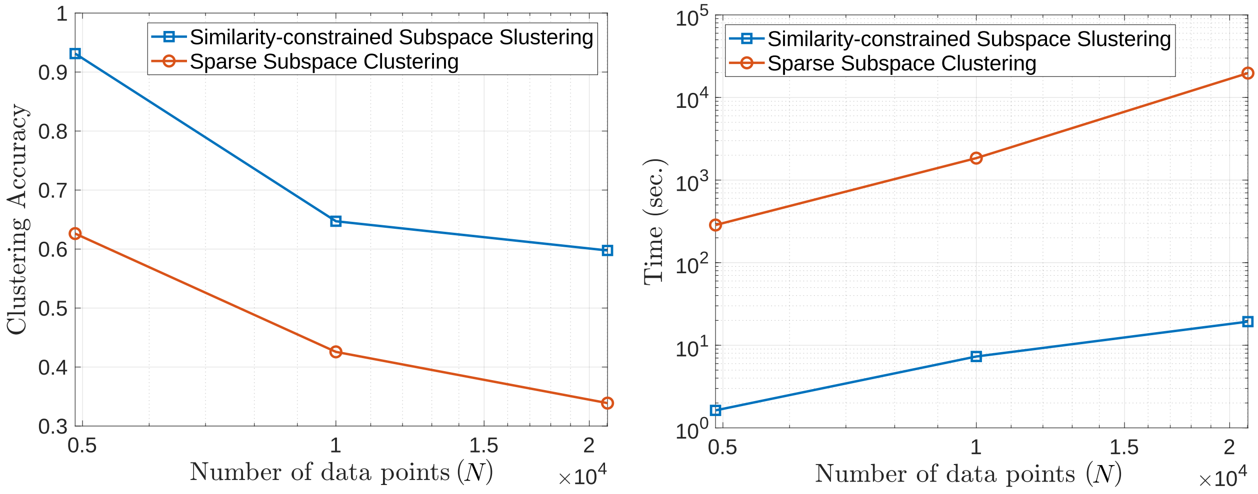


Figure 3. Clustering accuracy (left) and running time (right) of the SSC algorithm compared with the fastest method developed in this thesis for HSI clustering (in Chapter 4). In this example, the number of data points (N) is varied by performing the two subspace clustering algorithms on the full image and two regions of interest (ROIs) of the Indian Pines dataset (See Section 4.3). The first ROI has $N = 4900$ pixels and $k = 4$ classes; the second has $N = 10000$ pixels and $k = 12$ classes; the whole Indian Pines image has $N = 21025$ pixels and $k = 17$ classes.

ship between the representation coefficient vector of one pixel and its neighbors. (2) The overall computational complexity of SSC prohibits its usage on large HSI datasets. For instance, given a HSI with N_r rows, N_c columns, and L spectral bands, SSC needs to compute the $N \times N$ sparse coefficient matrix corresponding to $N = N_r N_c$ spectral pixels, whose computational complexity is $O(LN^3)$. Moreover, after building the affinity matrix, spectral clustering performs an eigenvalue decomposition over the $N \times N$ graph Laplacian matrix, which also has cubic time complexity, or quadratic using approximation algorithms (Chen et al., 2018) (see Fig. 3 right).

Thesis Contribution and Organization

This research thesis will address the problems introduced in chapter , by incorporating a similarity constraint in the self-representative dictionary during the optimization process. Also, will be explored a regularization constraint to enhance the robustness-to-noise of the algorithm. Finally, this thesis will explore similarity-constrained approaches with CSI measurements to identify and cluster low-dimensional structures by only accessing the compressed data without reconstructing the scene. Specifically, the main contributions of this thesis given by chapters are:

- **Chapter 2**: In this chapter, a regularization term is incorporated in the SSC model, which takes into account the neighboring spatial information of spectral pixels in the scene and enhance the robustness-to-noise of the algorithm. Specifically, the proposed method uses a 3D Gaussian filter to perform a 3D convolution on the sparse coefficients, obtaining a piecewise-smooth representation matrix that enforces an averaging constraint in the SSC optimization program. Extensive simulations demonstrate the effectiveness of the proposed method, achieving an overall accuracy of up to 99% in the selected hyperspectral remote sensing datasets.
- **Chapter 3**: This chapter proposes an efficient SSC-based method that significantly reduces the temporal and spatial computational complexity by splitting the HSI clustering task using similarity-constrained sampling. The author's proposed similarity-constrained sampling strategy considers both edge and superpixels information of the HSI to boost the clustering performance. This sampling strategy enables an intelligent selection of spectral signatures,

and then, the algorithm splits the clustering problem into multiples threads. Experimental results on widely used HSI datasets show that the efficiency of the proposed method outperforms baseline methods by up to 30% overall accuracy and up to 6 times in computing time.

- **Chapter 4**: Since SSC methods require to express each data point as a linear combination of all other points in the dataset, the main observation in this chapter is that it is possible to take advantage of the spatial-spectral properties of HSI to select the most representative spectral pixels over small regions and then constraint the remaining data points to be represented as a linear combination of them. This significantly reduces the clustering time and increases the accuracy. Specifically, this thesis proposes an efficient algorithm for selecting the most representative pixels at each subset by minimizing the maximum representation cost of the data. After this procedure, the most representative spectral pixels are concatenated to form a matrix. An optimization problem is solved using this matrix to attain the coefficients that encodes information about similarities between each subset's most representative spectral pixels and the whole HSI. Finally, the obtained coefficient matrix is enhanced via 2D smoothing convolution before applying a fast spectral clustering algorithm that provides the final clustering map in a significantly less amount of time. The motivation of using such 2D convolution is to enforce the piece-wise smoothness in the coefficient matrix. In essence, the author's proposed method enforces the connectivity in the affinity matrix and then efficiently obtains spectral embedding without the requirement to compute the eigenvalue decomposition, which has a computational complexity of $O(N^3)$.

Increasing the number of data points and the classes enlarges the computation time and make clustering more challenging. The author's proposed method, shown with the blue line in Fig. 3, can be up to three orders of magnitude faster than SSC and outperforms it in terms of accuracy when clustering more than 2×10^4 spectral pixels. This thesis evaluates and compares the proposed approach in this chapter on three real remote sensing hyperspectral images with different imaging environments and spectral-spatial resolution.

- **Chapter 5**: The spatial-spectral coded compressive spectral imager (3D-CASSI) is a CSI sensing scheme which modulates the spectral data cube in spatial and spectral dimensions using a 3D coded aperture (ensembles of 2D coded apertures) or a coding pattern array. Relying on compressive sensing, spectral image reconstruction is achieved by using nonlinear and relatively expensive optimization-based algorithms. In the CSI literature, several works have focused on improving reconstructions quality by properly designing the set of coding patterns. However, signal recovery is not actually necessary in many signal processing applications. For instance, assuming that compressed measurements with similar characteristics lie on the same subspace, unsupervised methods as subspace clustering can be used to separate them into the same cluster. Since the structure of compressed measurements is defined by the applied codification, it is possible to improve clustering performance. This chapter proposes to design an set of coding patterns such that inter-class and intra-class data structure is preserved after the CSI acquisition in order to improve clustering results directly on the compressed domain. To validate the coding pattern design, an algorithm based on sparse subspace clustering (SSC) is proposed to perform clustering on the compressed measurements. The proposed algorithm adds a 3D spatial regularizer to the SSC problem exploiting the spatial correlation of spectral images. In general, an overall accuracy up to 83.81% is obtained, when noisy measurements are assumed. In addition, a difference of at most 4% in terms of overall accuracy was observed when comparing the clustering results obtained by the full 3D data with those achieved using CSI measurements acquired with the proposed coding pattern design.

- **Chapter 6**: The single pixel camera (SPC) is a CSI system that acquire a single pixel measurement per spectral band of the HSI. SPC architecture provides reliable spectral imaging with lower-cost hardware, when acquiring a large number of spectral bands, compared with 2D sensor systems. In this chapter, the author is also interested in the unsupervised classification of SPC measurements which is difficult because of the extreme compression given by the SPC. Here the author uses the traditional spectral clustering algorithm instead of SSC to solve the classification problem. Specifically, in this chapter the author proposes a hierarchical adaptive approach to design a sensing matrix of the single pixel camera, such that pixel clustering can be performed in the compressed domain. At each step of the hierarchical model, a sensing matrix is designed such that clustering features can be directly extracted from the compressed measurements. Finally, the complete segmentation map is obtained with the majority voting method in the partial clustering results at each hierarchy step. In general, an overall accuracy of 78.94%, and 65.35% was obtained using the “Salinas”, and “Pavia University” spectral image datasets, respectively.

All the algorithms and methods developed in this research thesis were published in different international conferences and journals. In addition to the principal contributions listed above, extensions of the developed ideas in this thesis to other works are presented in the Appendix 1 and 2 and 5. The following section lists all the publications and authors’ contributions.

Publications and Author’s Contributions

International Journal Papers

1. **C. Hinojosa**, E. Vera and H. Arguello, "A Fast and Accurate Similarity-constrained Subspace Clustering Algorithm for Hyperspectral Image", in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. DOI: 10.1109/JSTARS.2021.3120071.

2. J. Lopez, **C. Hinojosa** and H. Arguello, "Efficient subspace clustering of hyperspectral images using similarity-constrained sampling", in Journal of Applied Remote Sensing, vol. 15, no. 3, pp. 01-16, 2021. DOI: 10.1117/1.JRS.15.036507.

3. **C. Hinojosa**, F. Rojas, S. Castillo, and H. Arguello, "Hyperspectral image segmentation using 3D regularized subspace clustering model", in Journal of Applied Remote Sensing, vol. 15, no. 1, pp. 01-17, 2021. DOI: 10.1117/1.JRS.15.016508.

4. **C. Hinojosa**, K. Sanchez, H. Garcia and H. Arguello, "C-3SPCD: Coded aperture similarity constrained design for Spatio-spectral Classification of Single-Pixel Measurements", in OPTICA (formerly OSA) Applied Optics (Accepted; to be published).

5. N. Diaz, **C. Hinojosa**, and H. Arguello, "Adaptive grayscale compressive spectral imaging using optimal blue noise coding patterns", in Optics & Laser Technology, vol. 117, pp. 147-157, 2019. DOI:10.1016/j.optlastec.2019.03.038.

6. K. Sanchez, **C. Hinojosa**, and Henry Arguello, "Supervised spatio-spectral classification of fused images using superpixels", in Applied Optics, vol. 58, no. 7, pp. B9-B18, 2019. DOI: 10.1364/AO.58.0000B9.

7. **C. Hinojosa**, J. Bacca and H. Arguello, "Coded Aperture Design for Compressive Spectral Subspace Clustering," in IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 6, pp. 1589-1600, 2018. DOI: 10.1109/JSTSP.2018.2878293.

International Conference Papers

1. **C. Hinojosa**, J. C. Niebles, and H. Arguello, "Learning Privacy-preserving Optics For Human Pose Estimation", in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

2. K. Sanchez, **C. Hinojosa**, H. Arguello, S. Freiss, N. Sans, D. Kouamé, O. Meyrignac, and A. Basarab. "Subspace-based Domain Adaptation Using Similarity Constraints for Pneumonia Diagnosis within a Small Chest X-ray Image Dataset", in Proceedings of the IEEE 18th International Symposium on Biomedical Imaging (ISBI), pp. 1232-1235, 2021. DOI: 10.1109/ISBI48211.2021.9434173.

3. J. Lopez, **C. Hinojosa**, and H. Arguello. "Fast Subspace Clustering Algorithm with Efficient

Similarity-Constrained Sampling for Hyperspectral Images", in Proceedings of the IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP), 2021. DOI: 10.1109/MLSP52302.2021.9596507.

4. J. Escobar, K. Sanchez, **C. Hinojosa**, and H. Arguello. "Accurate Deep Learning-based Gastrointestinal Disease Classification via Transfer Learning Strategy", in XXIII Symposium on Image, Signal Processing and Artificial Vision (STSIVA), 2021.
5. K. Sanchez, **C. Hinojosa**, H. Garcia, and H. Arguello. "Compressed-domain Classification Algorithm for Spectral Imaging Based on Designed Single-Pixel Camera Codification", in OSA Imaging and Applied Optics Congress, 2021. DOI: 10.1364/COSI.2021.CTu2F.5.
6. **C. Hinojosa**, J. Bacca, E. Vargas, S. Castillo and H. Arguello, "Single-Pixel Camera Sensing Matrix Design for Hierarchical Compressed Spectral Clustering," in Proceedings of the IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1-6, 2019. DOI: 10.1109/MLSP.2019.8918856.
7. **C. Hinojosa**, J. M. Ramirez and H. Arguello, "Spectral-Spatial Classification from Multi-Sensor Compressive Measurements Using Superpixels", in Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 3143-3147, 2019. DOI: 10.1109/ICIP.2019.8803266.
8. **C. Hinojosa**, J. Bacca, and H. Arguello, "Spectral Imaging Subspace Clustering with 3-D Spatial Regularizer", in Proceedings of the OSA Imaging and Applied Optics congress (3D,

AO, AIO, COSI, DH, IS, LACSEA, LS&C, MATH, pcAOP), paper JW5E.7, 2018.

9. K. Sanchez, **C. Hinojosa**, and H. Arguello, "Supervised Classification of Hyperspectral Images using Side Information", in Proceedings of the OSA Imaging and Applied Optics congress (3D, AO, AIO, COSI, DH, IS, LACSEA, LS&C, MATH, pcAOP), paper JW5E.5, 2018. DOI: 10.1364/3D.2018.JW5E.5.

1. Theoretical Background

1.1. General Notation

Throughout this thesis, $\mathbb{R}$ denotes the set of real numbers, and $\mathbb{R}^D$ denotes the D-dimensional linear space. Also, the lowercase boldface letters denote vectors, such as $\mathbf{x} \in \mathbb{R}^D$, and uppercase boldface letters denotes matrices, such as $\mathbf{X} \in \mathbb{R}^{D \times N}$. The transpose of the matrix $\mathbf{X} \in \mathbb{R}^{D \times N}$ is denoted as $\mathbf{X}^T \in \mathbb{R}^{N \times D}$. The uppercase calligraphic letters denote sets, such as $\mathscr{X} \subseteq \mathbb{R}^D$. For any vector $\mathbf{x} = [x_1, \cdots, x_D] \in \mathbb{R}^D$ and $p \geq 1$, the ℓ_p-norm is defined as $\|\mathbf{x}\|_p = \left(\sum_{i=1}^{D} |x_i|^p\right)^{1/p}$. As p approaches infinity, the infinity norm is defined as $\|\mathbf{x}\|_\infty = \max_{i=1}^{D} |x_i|$. Another particular interesting case for this thesis is when $p = 0$, in such case, $\|\mathbf{x}\|_0$ is defined as the number of nonzero entries in $\mathbf{x}$.

1.2. Spectral Imaging

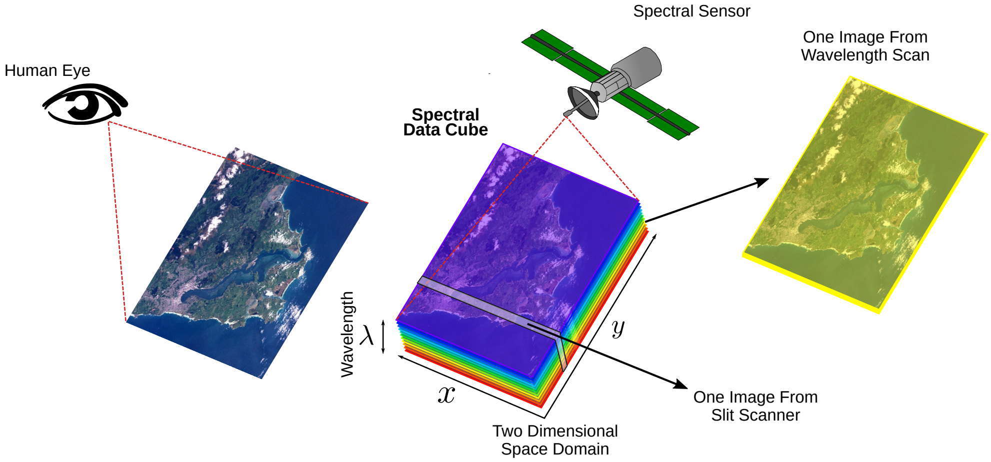


Figure 4. Human visual system and spectral imaging.

The human visual system builds a representation of the surrounding environment by detecting and interpreting the information from the visible range, roughly from 380 to 750 nm, of the electromagnetic spectrum. This limitation extends to traditional photo cameras which group all the spectral information within the visible spectral range into three broad spectral ranges roughly corresponding to the three primary colors, red, green and blue. Although such information allows perceiving the shape, surface texture and mutual spatial relation in the depth of 3D space, it is insufficient in applications where the spectral information of interest extends to other ranges of the electromagnetic spectrum.

Spectral imaging combines spectroscopy and two-dimensional imaging methodologies. Whereas imaging provides the intensity at each pixel of a 2D image, and a typical spectrometer measures a single spectrum, spectral imaging collects 2D images at specific wavebands across the electromagnetic spectrum. This is a three-dimensional (3D) data set and can be viewed as a cube of information, see Fig. 4.

Many different techniques for spectral imaging have been developed over the years. For instance, traditional spectral imaging methods, such as Whiskbroom (Green et al., 1998), Pushbroom (Herrala et al., 1994) and tunable filter imagers (Morris et al., 1994), scan adjacent zones of the underlying spectral scene and merge the results to construct a spectral 3D data cube. On the other hand, snapshot spectral imaging captures the spatial and spectral information of a scene by mapping all the voxels of the spectral data cube into different regions of a large focal plane array (FPA) (Gao et al., 2010; Kerekes and Schott, 2007). Furthermore, spectral imaging using Fabry-Perot filters or colored mosaic FPA detectors captures small subsets of spectral bands by assigning a particular spectral response to each FPA pixel such that a specific range of wavelengths is captured (Geelen et al., 2013; Eichenholz et al., 2010). In general, these traditional techniques require all voxels of the 3D scene to be sensed. Then, as the spatial or spectral resolution increases the number of voxels to be sensed increases proportionally, leading to an increment in the cost of sensing, storing and transmitting an spectral image acquired through these methods.

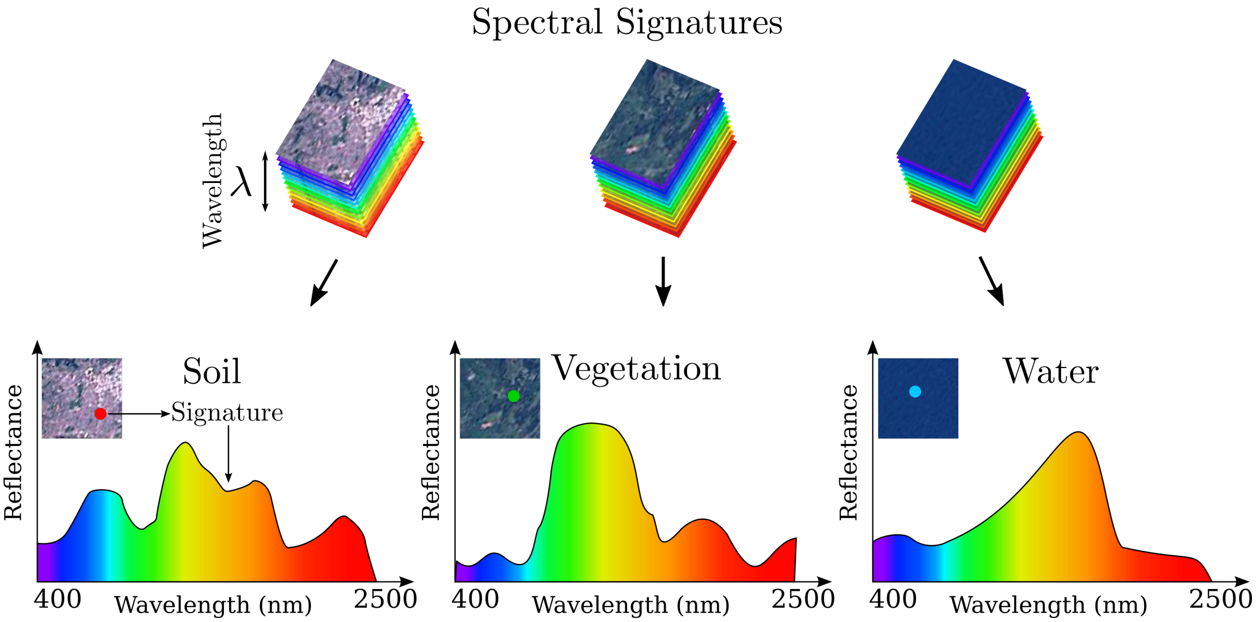


Figure 5. Spectral signatures in an HSI. Materials have a representative spectral signature.

Every spatial location in a spectral image is represented by a vector whose values correspond to the intensity at different spectral bands. These vectors are also known as the spectral signature of the pixels, see Fig. 5. All materials have unique spectral characteristics because they absorb, reflect, and emit radiation in a unique way. For instance, in the visible portion of the spectrum, a leaf appears green because it absorbs in the blue and red regions of the spectrum and reflects in the green region. These variations in absorption, reflection, and emission correspond to the material composition. Differences in spectral responses related to absorption, transmission, and reflection cause materials to have a unique spectral signature. Therefore, the information provided by the spectral signatures can be a valuable tool for the detection, identification, and classification of materials and objects with complex compositions (He et al., 2015). Particularly, the classification of spectral images consists of labeling individual spectral signatures to one of the classes based on its spectral characteristics. Spectral image classification has found many applications in various fields such as military (Manolakis and Shaw, 2002; Eismann et al., 2009), precision agriculture (Patel et al., 2001), and mineralogy (Hörig et al., 2001).

In general, there are two main approaches to the classification problem: supervised and unsupervised. Supervised techniques require the availability of a training set for learning the classifier. Among various supervised techniques, support vector machines (SVMs) (Boser et al., 1992; Vapnik, 2013) have shown a good performance for spectral image classification (Gualtieri and Cromp, 1999; Melgani and Bruzzone, 2004). In particular, a few spectral signatures are used as training samples to train a SVM classifier, and then, the remaining spectral signatures are classified. This process is depicted in Fig. 6. Variations of the SVM-based algorithms have also been proposed to improve the classification accuracy. These variations include semisupervised learning which exploits both labeled and unlabeled samples (Bruzzone et al., 2006), postprocessing of the individually labeled samples based on certain decision rules (Bovolo et al., 2006), and incorporating spatial information directly in the SVM kernels (Camps-Valls et al., 2006). More recent spectral imaging supervised classification techniques can be found in (Li et al., 2019; Yang et al., 2018).

Unsupervised methods, known also as clustering methods, perform classification just by exploiting information conveyed by the data, without requiring any training sample set. Supervised

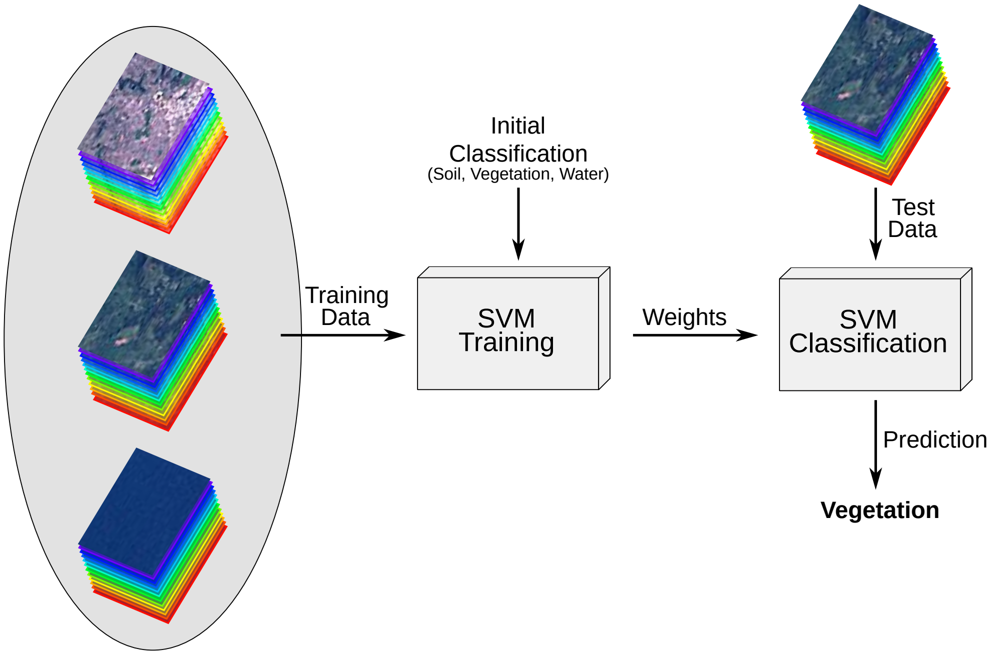


Figure 6. Supervised spectral image classification using SVM.

methods offer a higher classification accuracy compared to the unsupervised ones, but in some applications, it is necessary to resort to unsupervised techniques because training information is not available. When trying to cluster high dimensional data such as spectral images, the given set of data points, i.e., the spectral signatures, could be drawn from an arrangement of an unknown number of subspaces that have unknown and possibly different dimensions. Then, the goal is to simultaneously estimate these subspaces and cluster the points into their corresponding subspaces (Parsons et al., 2004).

1.3. Subspace Clustering Theory

In recent years, unprecedented technological advances have led to an increment in the availability and dimensionality of the data in all areas of science and engineering. These include machine learning, signal and image processing, computer vision, pattern recognition, bioinformatics, etc.

For instance, a conventional grayscale image consists of billions of pixels whereas a spectral image consists of hundreds of grayscale images which provide information from different wavelengths. This high dimensionality of the data leads to an increment in memory resources and computational cost for processing and data analysis. However, high dimensional data often lie in low dimensional structures instead of being uniformly distributed across the environmental space. Therefore, different techniques for finding a low-dimensional representation of a high dimensional data set have been development (Vidal, 2011).

A traditional technique is the principal component analysis (PCA). PCA assumes that a set of $\hat{N}$ points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\hat{N}}\} = \{\mathbf{x}_j\}_{j=1}^{\hat{N}}$ in high dimensional space $\mathbb{R}^L$ is drawn from a single low dimensional affine subspace $\mathscr{S}$ of dimension $l \ll L$. This technique is well established in the literature, and has become one of the most useful tools for data modeling, compression, and visualization (Vidal, 2011).

In practice, however, the data points could be drawn from the union of $\hat{n} \geq 1$ linear or affine subspaces $\{\mathscr{S}_i\}_{i=1}^{\hat{n}}$ of unknown dimensions $l_i = \dim(\mathscr{S}_i)$, $0 < l_i < L$. The subspaces can be described as

$$\mathscr{S}_i = \{\mathbf{x} \in \mathbb{R}^L : \mathbf{x} = \hat{\mu}_i + \mathbf{U}_i \mathbf{y}\}, \quad i = 1, \cdots \hat{n}, \tag{1}$$

where $\hat{\mu}_i \in \mathbb{R}^L$ is an arbitrary point in subspace $\mathscr{S}_i$, that can be chosen as $\hat{\mu}_i = \mathbf{0}$ for linear subspaces; $\mathbf{U}_i \in R^{L \times l_i}$ is a basis for subspace $\mathscr{S}_i$; and $\mathbf{y} \in \mathbb{R}^{l_i}$ is a low dimensional representation for point $\mathbf{x}$. Then, the goal of subspace clustering is to find the number of subspaces $\hat{n}$, their dimensions $\{l_i\}_{i=1}^{\hat{n}}$, the subspace bases $\{\mathbf{U}_i\}_{i=1}^{\hat{n}}$, the points $\{\hat{\mu}_i\}_{i=1}^{\hat{n}}$, and cluster the data points into their corresponding subspaces. Since data in a subspace is often distributed arbitrarily and not around a centroid, standard centroid-based clustering methods (Duda et al., 2012) that take advantage of the spatial proximity of the data in each cluster are not in general applicable to subspace clustering.

Different algorithms for subspace clustering which take into account the multi-subspace structure of the data have been proposed in the past two decades. These algorithms can be divided into four main categories: iterative, algebraic, statistical and spectral clustering-based methods (Elhamifar and Vidal, 2013). In particular, spectral clustering-based methods construct a weighted graph $\mathscr{G} = (\mathscr{V}, \mathscr{K}, \mathbf{W})$, where $\mathscr{V} = \{1, \cdots, \hat{N}\}$ is the set of nodes, $\mathscr{K} \subset \mathscr{V} \times \mathscr{V}$ is the set of edges, and $\mathbf{W} \in \mathbb{R}^{\hat{N} \times \hat{N}}$ is a symmetric nonnegative affinity matrix whose (j, j')-th entry, $W_{j,j'}$, measures the affinity between points $\mathbf{x}_j$ and $\mathbf{x}_{j'}$. Ideally, $W_{j,j'} = 1$ if points j and j' are in the same group and $W_{j,j'} = 0$ if points j and j' are in different groups. In practice, a typical affinity is given by

$$W_{j,j'} = \exp\left(-\frac{1}{2\sigma}\mathrm{dist}(\mathbf{x}_j, \mathbf{x}_{j'})^2\right), \tag{2}$$

where $\mathrm{dist}(\mathbf{x}_j, \mathbf{x}_{j'})$ is some measure of the distance between points j and j' and $\sigma > 0$ is a parameter. Let $\mathscr{D} = \mathrm{diag}(\mathbf{W}\mathbf{1})$, where $\mathbf{1} \in \mathbb{R}^{\hat{N}}$ is an all-ones vector, be a diagonal matrix whose jth diagonal entry gives the degree $\mathscr{D}_{j,j} = \sum_{j'} W_{j,j'}$ of node j, and let $\mathscr{L} = \mathscr{D} - \mathbf{W} \in \mathbb{R}^{\hat{N} \times \hat{N}}$ be the graph's Laplacian matrix. Spectral clustering obtains a clustering of the data by applying the K-means algorithm to the columns of the matrix $\mathscr{E} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{\hat{n}}]^T \in \mathbb{R}^{\hat{n} \times \hat{N}}$, where $\{\mathbf{e}_i\}_{i=1}^{\hat{n}}$ are the eigenvectors of $\mathscr{L}$ associated with its $\hat{n}$ smallest eigenvalues. However, the distance-based affinity described in Eq. (2) is not appropriate for subspace clustering since two points could be very close to each other but lie in different subspaces (e.g., near the intersection of two subspaces). Conversely, two points could be far from each other but lie in the same subspace. In general, the geometric relationships among multiple points must be considered in order to construct an effective affinity measure for subspace clustering (Vidal, 2011; Elhamifar and Vidal, 2013).

In fact, the construction of a good affinity matrix is one of the main challenges in applying spectral clustering to subspace clustering problem. Based on the representative method for designing an affinity matrix, the existing spectral clustering-based algorithms can be categorized in local and global methods. Local methods such as Local Subspace Affinity (LSA) (Yan and Pollefeys, 2006), Locally Linear Manifold clustering (LLMC) (Goh and Vidal, 2007) and Spectral Local Best-fit Flats (SLBF) (Zhang et al., 2012; Zelnik-Manor and Irani, 2003) compute an affinity between two points that only depends on the data points in a local neighborhood of each of the two points. These methods have difficulties in dealing with points near the intersection of two subspaces because the neighborhood of a point can contain points from different subspaces. Also, they are sensitive to the right choice of the neighborhood size to compute the local information at each element. On the other hand, global methods such as Spectral Curvature Clustering (SCC) (Chen and Lerman, 2009) and algebraic subspace affinity (Ng et al., 2002) compute an affinity between two points that depends on all the data points.

Following the same idea of writing a data point as a linear combination of neighboring data points, new subspace algorithms have emerged but, instead of choosing neighbors based on the angular or Euclidean distance between pairs of points, such algorithms consider that a point can be represented with any other elements in the data set. In this regard, the sparse subspace clustering

(SSC) is a well-known algorithm that uses the entire dataset as a dictionary and represent each datapoint as a linear combination of all other points in the dataset.

1.3.1. Sparse Subspace Clustering (SSC). Different works in processing complex high-dimensional data rely on the observation that their intrinsic dimension is often much smaller than the dimension of the ambient space (Elhamifar and Vidal, 2009). This led to the development of different techniques for finding a low-dimensional representation, or subspace, of high-dimensional data sets (Elhamifar and Vidal, 2013; Hinojosa et al., 2018a). In practice, all the high-dimensional data points can be modeled as being drawn from a union of multiple linear or affine subspaces where the membership of the data points to each subspace might be unknown. Then, subspace clustering refers to the problem of separating data according to their underlying subspaces. The subspaces can be linear or affine, which could be seen as a linear subspace with a constraint that the sum of all the coordinates is equal to one. One of the most popular subspace clustering algorithms is the sparse subspace clustering (SSC) (Elhamifar and Vidal, 2013). The key idea behind SSC is that among the infinitely many possible representations of a data point in terms of the others, a sparse representation corresponds to selecting a few points from the same subspace.

Denoting the matrix containing all data points as

$$\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\hat{N}}\right] = \left[\mathbf{X}_1, \cdots, \mathbf{X}_{\hat{n}}\right]\Gamma, \tag{3}$$

where $\mathbf{X}_i \in \mathbb{R}^{L\times\hat{N}_i}$ is a rank-l_i matrix of the $\hat{N}_i > l_i$ points that lie in $\mathscr{S}_i$, and $\Gamma \in \mathbb{R}^{\hat{N}\times\hat{N}}$ is an unknown permutation matrix. The SSC algorithm takes advantage of the self-expressiveness property of the data, i.e., each data point in a union of subspaces can be efficiently reconstructed by a combination of other points in the dataset. Then with the **X** matrix itself being used as the dictionary, the SSC algorithm constructs the sparse representation model as follows

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{E}} \quad & \|\mathbf{Z}\|_1 + \lambda \|\mathbf{E}\|_F^2 \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E},\ \operatorname{diag}(\mathbf{Z}) = 0, \end{aligned} \tag{4}$$

where the ℓ_1-norm regularization in this formulation suggests that a sparse representation of a data point finds points from the same subspace. Furthermore, the matrix **E** stands for the representation error and, the regularization parameter λ for the sparsity trade-off. The constraint $\operatorname{diag}(\mathbf{Z}) = 0$ is used to eliminate the trivial solution of writing a point as a linear combination of itself. Using the obtained sparse coefficient matrix **Z**, the affinity matrix **W** is constructed, which defines the weight on edge between the data nodes as follows

$$\mathbf{W} = |\mathbf{Z}| + |\mathbf{Z}|^T. \tag{5}$$

Finally, the clustering result is obtained by applying spectral clustering to the Laplacian $\mathscr{L}$ matrix induced by the affinity matrix **W**.

1.3.2. Efficient general-purpose SSC-based methods. Considering the self-expressiveness property (Elhamifar and Vidal, 2013), an early approach to address the SSC scalability issue assumes that a small number of data points can represent the whole dataset without loss of information. Then, authors in (Peng et al., 2013, 2015) proposed the *Scalable Sparse Subspace Clustering* (SSSC) algorithm to cluster a small subset of the original data and then classify the rest of the data based on the learned groups. However, this strategy is suboptimal since it sacrifices clustering accuracy for computational efficiency.

In (You et al., 2016b), authors replace the ℓ_1 optimization in the original SSC algorithm (Elhamifar and Vidal, 2013) with greedy pursuit, e.g., orthogonal matching pursuit (OMP) (Tropp et al., 2007), for sparse self-representation (Dyer et al., 2013). While SSC-OMP improves the time efficiency of SSC by several orders of magnitude, it significantly loses clustering accuracy (Chen et al., 2017). Besides, SSC-OMP also suffers from the connectivity issue presented in the original SSC algorithm. To solve this issue, authors in (You et al., 2016a) proposed to mixture the ℓ_1 and ℓ_2 norms to take advantage of subspace preserving of the ℓ_1 norm and the dense connectivity of the ℓ_2 norm. Specifically, this algorithm, named ORacle Guided Elastic Net solver (ORGEN), proposed to identify a support set for each sample. However, in this approach, a convex optimization problem is solved several times for each sample, limiting the algorithm's scalability.

More recent works use a different subset selection method for subspace clustering. In particular, the method named Scalable and Robust SSC (SR-SSC) (Abdolali et al., 2019) selects a few sets of anchor points using a randomized hierarchical clustering method. Then, within each set of anchor points, it solves the LASSO (Tibshirani, 1996) problem for each data point, allowing only anchor points to have non-zero weights.

Similar to the SSC-OMP paper, authors in (You et al., 2018) proposed an approximation algorithm (Williamson and Shmoys, 2011) to solve the SSC optimization problem. Specifically, instead of using all the dataset $\mathbf{X}$, the Exemplar-based Subspace Clustering (ESC-FFS) algorithm in (You et al., 2018) selects a small subset $\tilde{\mathbf{X}} \subseteq \mathbf{X}$ that represents all data points, and then each point is expressed as a linear combination of points in $\tilde{\mathbf{X}} \in \mathbb{R}^{D\times M}$, where $M < N$. In particular, the selection of $\tilde{\mathbf{X}}$ is obtained by using the Farthest first search (FFS) algorithm, which is a modified version of the Farthest-First Traversal (FarFT) algorithm where the main difference is the used distance metric (Williamson and Shmoys, 2011). The authors propose to construct $\tilde{\mathbf{X}}$ by first performing random sampling to select a base point and then progressively add new representative data points using the defined metric.

In general, the previously described algorithms provide an acceptable subspace clustering performance on large datasets. However, these general-purpose methods do not fully exploit the complex structure of remotely sensed HSIs, ignoring their rich spatial information, which could boost the accuracy of these algorithms.

1.3.3. HSI Subspace Clustering. Assuming that HSI's pixels are drawn from a union of multiple subspaces, and spectral pixels with a similar spectrum belong to the same land cover material, SSC can be effectively applied to perform unsupervised land cover segmentation. Denote a spectral pixel $\mathbf{x}_j$ as a D-dimensional vector after lexicographically reordering the $N_r \times N_c \times D$ 3D data cube into a 2D matrix as $\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_j, \cdots, \mathbf{x}_N\right] \in \mathbb{R}^{D}$, where $N = N_rN_c$, and D refers to the number of spectral bands (Hinojosa et al., 2018a). Note that each column of $\mathbf{X}$ now corresponds to a spectral pixel (signature), see Fig. 7. Under the context of subspace clustering, each column of $\mathbf{X}$ belongs to the union of k subspaces $\{\mathscr{S}\}_{i=1}^{k}$. By taking advantage of the subspace structure, the

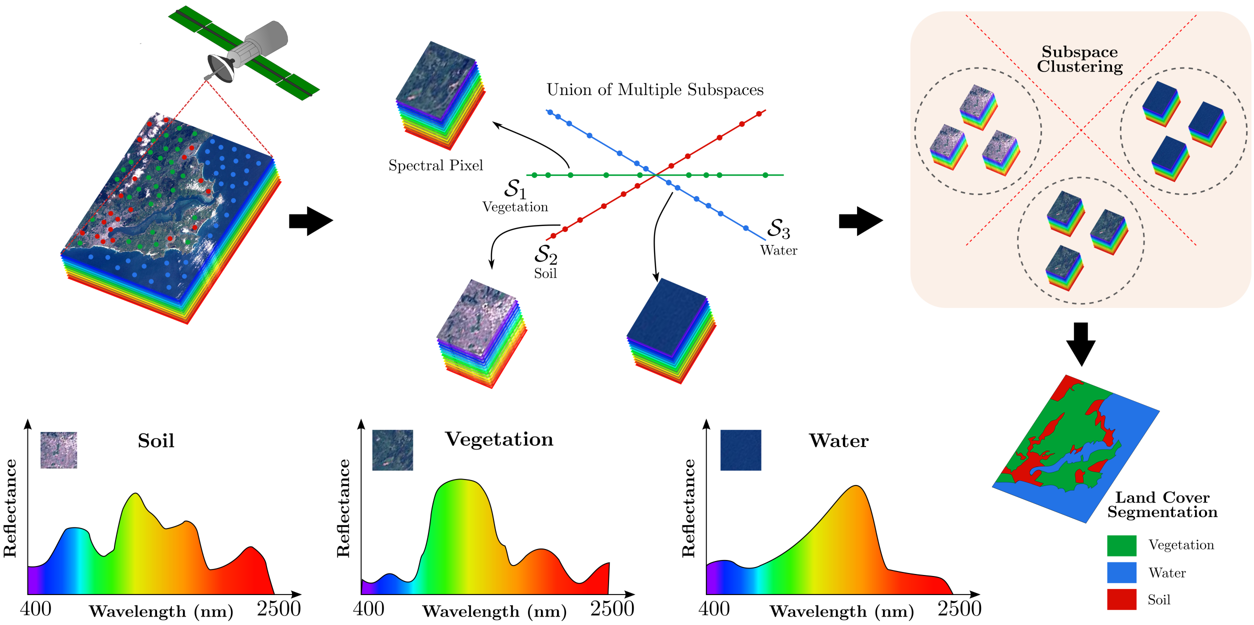


Figure 7. Hyperspectral image land cover classification via subspace clustering theory.

data points in $\mathbf{X}$ obey the so called *self-expressiveness property* (Elhamifar and Vidal, 2013), i.e., each data point in a union of subspaces can be well represented by a linear combination of other points in the dataset. This can be mathematically expressed as

$$\mathbf{X} = \mathbf{X}\mathbf{Z}, \tag{6}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times N}$ is the representation coefficient matrix. The choice of $\mathbf{Z}$ is usually not unique, and the goal is to find a certain $\mathbf{Z}$ that be discriminative for subspace clustering. The problem in Eq. (6) is that it may have many feasible solutions, and thus the regularization is necessary to produce the solution. Motivated by the observation that the block-diagonal solution is sparse, SSC finds a sparse $\mathbf{Z}$ by ℓ_1-norm minimizing. Here, the ℓ_1 norm is used as the convex surrogate of the ℓ_0 norm, which is not convex and, otherwise, would turn the problem NP-hard (Elhamifar and Vidal, 2013).

Then, the sparse representation model can be built as follows:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{R}} \|\mathbf{Z}\|_1 + \lambda \|\mathbf{R}\|_F^2 \\ &\text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{R}, \quad \text{diag}(\mathbf{Z}) = 0, \mathbf{Z}^T\mathbf{1} = \mathbf{1}, \end{aligned} \tag{7}$$

where $\mathbf{R}$ stands for the representation error, and the parameter λ is used to trade-off the relative contribution between the sparsity of the coefficients and the magnitude of the noise. Here, $\text{diag}(\mathbf{Z}) \in \mathbb{R}^N$ is the vector of the diagonal elements of $\mathbf{Z}$, and the constraint $\text{diag}(\mathbf{Z}) = 0$ is used to eliminate the trivial solution of writing a point as an affine combination of itself (Elhamifar and Vidal, 2013). In addition, the constraint $\mathbf{Z}^T\mathbf{1} = \mathbf{1}$ ensures that it is a case of an affine subspace.

The optimization problem in Eq. (7) can be solved by the alternating direction method of multipliers (ADMM), and its implementation can be found in (Elhamifar and Vidal, 2013). The solution obtained by Eq. (7) corresponds to subspace-sparse representation of the data points, which is used by spectral clustering to infer the clustering of the data. Specifically, the clustering result is obtained by applying the spectral clustering to the Laplacian matrix induced by the similarity matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$, defined as $\mathbf{W} = |\mathbf{Z}| + |\mathbf{Z}|^T$, where the operator $|\cdot|$ stands for the absolute value. This definition of $\mathbf{W}$ is adopted to ensure that the matrix is both ≥ 0 and symmetric (Elhamifar and Vidal, 2009). The complete SSC method is summarized in Algorithm. 1.

As observed in Eq. (7), the SSC model for HSI only captures the relationship of pixels by analyzing the spectral features without considering the spatial information. One objective of this thesis is to propose a robust subspace clustering approach that incorporates a noise-regularization

Algorithm 1 Sparse Subspace Clustering Algorithm (SSC) (Elhamifar and Vidal, 2013)

```
Input  : HSI Data $\mathbf{X} \in \mathbb{R}^{D \times N}$, containing a set of spectral pixels $\{\mathbf{x}_j\}_{j=1}^{N}$, in a union of $k$ affine subspaces $\{\mathscr{S}_i\}_{i=1}^{k}$;
         the number of subspaces $k$ and the regularization parameter $\lambda$
Output: Matrix $\hat{\mathbf{X}}$ with the obtained clustering labels of $\mathbf{X}$
Function SSC($\mathbf{X}, k, \lambda$)
    Solve the sparse optimization problem in Eq. (7).
    Normalize the columns of $\mathbf{Z}$ as $\mathbf{z}_j \leftarrow \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_\infty}$.
    Form a similarity matrix as $\mathbf{W} = |\mathbf{Z}| + |\mathbf{Z}|^T$, representing the relationship (weights) among data points.
    Apply Spectral Clustering (Von Luxburg, 2007b) to the similarity matrix.
    return Segmentation of the data: $\hat{\mathbf{X}}_1, \cdots, \hat{\mathbf{X}}_k$.
```

term in the subspace clustering optimization problem and also considers the spatial information to successfully improve the clustering results on HSI. This proposed algorithm is described in Chapter 2.

1.3.4. SSC-based methods for HSI clustering. Some SSC-based methods have been proposed for unsupervised HSI classification, which take advantage of the neighboring spatial information but still present the efficiency issue of SSC. Under the context of HSIs, the $N_r \times N_c \times L$ 3D image data cube can be rearranged into a 2D matrix $\mathbf{X} \in \mathbb{R}^{D \times N}$ to apply the SSC algorithm, where $N = N_r N_c$ and $D < L$ is the number of features extracted from the spectral signatures after using principal component analysis (PCA) (Sanchez et al., 2019a). Taking into account that the spectral pixels belonging to the same land cover material are arranged in common regions, different works (Zhang et al., 2016; Zhai et al., 2016, 2017; Bacca et al., 2017a; Hinojosa et al., 2018a,a, 2021c) aim at obtaining a piecewise smooth sparse coefficient matrix to incorporate such contextual dependence. In particular, S-SSC (Zhang et al., 2016) helps to guarantee spatial smoothness and reduce the representation bias by adding a regularization term in the SSC optimization problem, which enforces a local averaging constraint on the sparse coefficient matrix. More recently, authors in (Hinojosa et al., 2021c) propose the 3DS-SSC algorithm, which incorporates a 3D Gaussian filter in the optimization problem to perform a 3D convolution on the sparse coefficients, obtaining a piecewise-smooth representation matrix. Also, another recent work proposes a graph convolutional subspace clustering (GCSC) framework (Cai et al., 2020) that efficiently processes graph data by modeling information from neighbor samples (or nodes). Then, the graph is used as a dictionary for the subsequent affinity learning, and finally outputting classification results by a clustering model. In general, these methods are slow since they load and process all the data points hence having large memory dependency. Therefore, these methods commonly present clustering results on small regions of interest (ROI) from the full HSI in their original manuscripts.

1.4. Compressive Spectral Imaging

Most of the information acquired by traditional sampling methods of spectral images is discarded during the compression process before being stored. Due to the high dimensionality of the spectral images, recent works have focused on obtaining a compressed spectral image directly, avoiding the compression step.

Compressive spectral imaging (CSI) is a new approach which senses and reduces the data dimension in a single step. Specifically, CSI captures 2D coded and dispersed projections of the 3D scene rather than direct measurements of the voxels. Therefore, a far less number of samples are acquired compared to traditional techniques, leading to an improved sensing speed (Arce et al., 2014). CSI theory establishes that an estimation of the 3D data cube can be successfully recovered from the compressed measurements. To make this possible, CSI relies on two principles: sparsity, which characterizes the spectral scene of interest, and incoherence, which shapes the sensing structure (Donoho, 2006; Candès and Wakin, 2008).

A signal is K-sparse if at most K of its components are nonzero. However, many natural and man-made signals are not sparse but compressible in the sense that they can be well-approximated as a linear combination of just a few elements from a known basis or dictionary Ψ. Formally, denote $\mathscr{F}$ as the spatio-spectral input data cube, with $M \times N$ spatial dimensions, and L spectral bands. Then, a spectral image can be expanded in an orthonormal basis $\Psi = [\psi_1, \psi_2, \cdots, \psi_{MNL}]$ as

$$\bar{\mathbf{f}} = \sum_{i=1}^{MNL} \theta_i \psi_i, \tag{8}$$

where $\bar{\mathbf{f}} \in \mathbb{R}^{MNL}$ is a vector representation of the source $\mathscr{F}$, θ is the coefficient sequence of $\bar{\mathbf{f}}$, such that $\theta_i = \left\langle \bar{\mathbf{f}}, \psi_i \right\rangle$. Then, the spectral image $\bar{\mathbf{f}}$ can be expressed as $\bar{\mathbf{f}} = \Psi\theta$, where Ψ is an $MNL \times MNL$ matrix with $\psi_1, \psi_2, \cdots, \psi_{MNL}$ as columns.

On the other hand, the CSI incoherence property measures how correlated are the elements of $\mathbf{H}$ and Ψ where a low correlation is desired (Donoho and Huo, 2001). Specifically, let $\mathbf{H} \in \mathbb{R}^{MV \times MNL}$, where MV is the number of compressed measurements to acquire, and $\Psi \in \mathbb{R}^{MNL \times MNL}$ be two orthonormal bases. Following the compressive sensing theory, the first basis is used to measure/sense the signal $\bar{\mathbf{f}}$, and the second basis is used to represent $\bar{\mathbf{f}}$ in a sparse domain. The mutual coherence of the orthonormal basis $\mathbf{H}$ and Ψ is defined as the maximum absolute value of the inner product between any two columns of the basis, given by

$$\mu(\mathbf{H}, \Psi) = \sqrt{MN} \max_{1 \leq i,j \leq MN} |\langle \boldsymbol{H}_i, \Psi_i \rangle|, \tag{9}$$

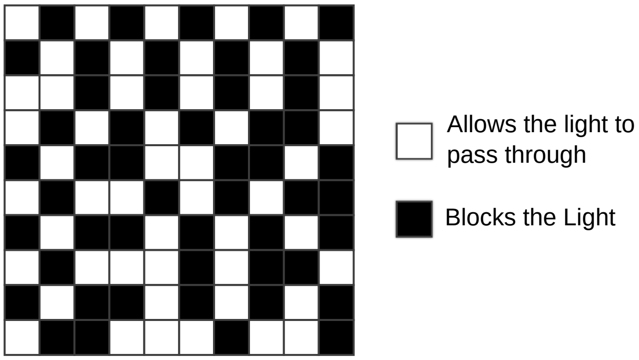


Figure 8. Example of a coded aperture. The black elements represent the zero values which block the light whereas the white elements represent the one values which allow the light to pass through.

where $\boldsymbol{H}_i$ represents the i-th column of the sensing matrix $\mathbf{H}$, whose structure is determined by the sensing architecture. The coherence measures the maximum correlation between any two elements of $\mathbf{H}$ and Ψ. In CSI a minimum coherence among the orthonormal base is desired.

CSI-based systems can capture a spectral image with a single snapshot, without the need to linearly scan all the adjacent zones (Cao et al., 2016). In order to acquire the 2D compressed projections, CSI devices usually employ optical elements like focal plane arrays; optical coding elements, e.g., coded apertures; and dispersive elements, e.g., prisms.

Coded Apertures or Coded-Aperture Masks are grids, gratings, or other patterns of materials opaque to various wavelengths of light. A coded aperture can be represented as a binary matrix, where the one-valued elements allow the electromagnetic radiation to pass through and the zero-valued elements block the light, generating a coding pattern in the object of interest. Figure 8 shows an example of a coded aperture whose elements were generated at random.

Another fundamental optical component of the CSI devices is the dispersive element, whose function is to decompose the light in its spectral components. A commonly used optical dispersive prism in CSI systems is the double Amici prism shown in Fig. 9. Such prism is composed by the union of three prisms with different refraction indices (n_1, n_2, n_3). The prisms at the ends have the same refractive index while the prism at the center has a higher refractive index and, therefore, a greater capacity for dispersion. This prism is specially designed to increase the angular dispersion of the set of waves, and also has the property that, when refracted again, the spectral component with central wavelength forms a straight line with the trajectory of the incident ray.

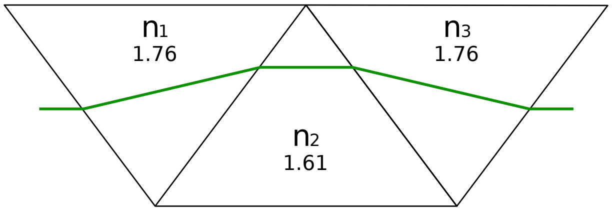


Figure 9. Double Amici prism.

In the CSI mathematical model, the spectral image $\mathscr{F}$ with $M \times N$ spatial dimensions and L spectral bands is discretized as $\mathscr{F}_{m,n,k}$, where m and n index the spatial coordinates, and k determines the k-th spectral band. $\mathscr{F}$ can also be represented in vector form as $\mathbf{f} = \left[\mathbf{f}_0^T, \cdots, \mathbf{f}_{L-1}^T\right]^T$, where each spectral band $\mathbf{f}_k$ can be expressed as $\mathbf{f}_k = \left[\mathscr{F}_{0,0,k}, \mathscr{F}_{1,0,k}, \cdots, \mathscr{F}_{(M-1),0,k}, \cdots, \mathscr{F}_{0,1,k}, \mathscr{F}_{1,1,k}, \cdots, \mathscr{F}_{(M-1),1,k}, \cdots, \mathscr{F}_{(M-1),(N-1),k}\right]^T$.

In general, CSI projections can be written in matrix notation as

$$\mathbf{y} = \mathbf{H}\mathbf{f} = \mathbf{H}\Psi\theta, \tag{10}$$

where $\mathbf{y}$ is the measurement set in vector form and, $\mathbf{H}$ is known as the system sensing matrix whose entries are determined by the CSI optical setup (Cao et al., 2016; Arce et al., 2014). The spectral image **f** is said to have a sparse representation θ in a basis Ψ, as depicted in Fig. 10.

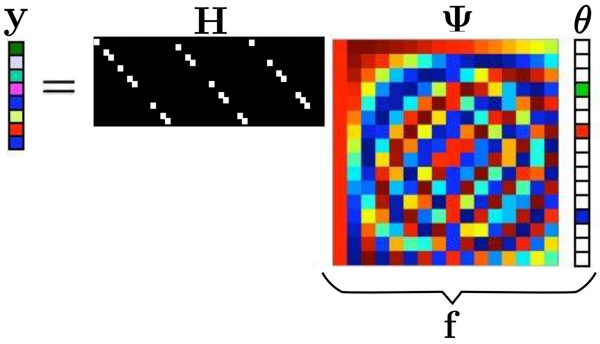


Figure 10. CSI linear system representation. A measurement vector **y** is obtained by sensing a spectral image **f** using the sensing matrix **H**. The spectral image **f** has a sparse representation θ in a representation basis Ψ.

Although CSI allows reconstructing a spectral image from a considerably smaller number of samples than those required by the Shannon-Nyquist theorem, there is a minimum number of necessary measurements that must be acquired to successfully reconstruct a scene. Considering a spectral image of size MNL which is K-sparse in a basis Ψ, the CSI theory states that $u \geq K\log(MNL)$ measurements are necessary in order to achieve a correct reconstruction. However, for spectrally rich scenes or very detailed spatial scenes, a single shot CSI measurement may not provide a sufficient number of compressed measurements to satisfy the above constraint. This leads to an excessively undetermined inverse problem hence failing to obtain an adequate reconstruction. Increasing the number of measurement shots will multiply the number of measurements, thus rapidly overcoming such limitations (Correa et al., 2016; Kittle et al., 2010; Arguello and Arce, 2011).

In CSI it is possible to acquire $S \ll L$ measurement shots employing a different coded aperture at each snapshot, such that different measurements of the spectral data cube are acquired each time. Mathematically, the compressed measurements acquired with the s-th snapshot can be expressed in vector notation as

$$\mathbf{y}^s = \mathbf{H}^s\mathbf{f}, \tag{11}$$

where $\mathbf{y}^s \in \mathbb{R}^{MV}$ is the vector form representation of the acquired measurements, and $\mathbf{H}^s$ is the sensing matrix of the system for each snapshot s. Specifically, $\mathbf{H}^s$ is a $MV \times MNL$ sparse matrix whose nonzero entries are determined by the coded aperture. The measurement vectors $\mathbf{y}^s$ acquired at each snapshot can be succinctly expressed in vector form as $\mathbf{y} = \left[\left(\mathbf{y}^0\right)^T, \cdots, \left(\mathbf{y}^{S-1}\right)^T\right]^T$. Therefore, Eq. 11 can be rewritten in the standard form of an underdetermined system, described in Eq. 10, where $\mathbf{H} = \left[\left(\mathbf{H}^0\right)^T, \cdots, \left(\mathbf{H}^{S-1}\right)^T\right]^T$ is the concatenation of all sensing matrices $\mathbf{H}^s$.

Once acquired the compressed measurements $\mathbf{y}$, the subsequent procedure is spectral image recovery. Given that the amount of the acquired compressed measurements SMV (generally $V \leq N+L-1$), is far less than the number of 3D data cube entries to be estimated MNL, the reconstruction problem to be solved becomes ill-posed. Therefore, it cannot be solved by directly inverting the system in Eq. 10. In particular, CSI spectral image recovery consists on finding a K-sparse representation of $\mathbf{f}, \theta$ in a given basis Ψ. The sparse representation can be recovered by minimizing the $\ell_2 - \ell_1$ cost function given by $||\bar{\mathbf{y}} - \mathbf{A}\theta||_2 + \tau||\theta||_1$, where τ is a regularization constant (Arce et al., 2014; Donoho, 2006). In other words, it seeks a sparse approximation of the spectral data cube. Formally, the reconstruction optimization problem is written as

$$\tilde{\mathbf{f}} = \Psi \left\{ \underset{\theta \in \mathbb{R}^{MNL}}{\arg\min} \|\bar{\mathbf{y}} - \mathbf{A}\theta\|_2 + \tau \|\theta\|_1 \right\}, \tag{12}$$

where $\mathbf{A} = \mathbf{H}\Psi$.

CSI reconstruction algorithms can be classified into five computational approaches (Tropp et al., 2010). The *greedy* algorithms obtain a sparse estimation of θ in an iterative manner by identifying the components which provide the best reconstruction at each iteration. Algorithms such as OMP (orthogonal matching pursuit) (Tropp et al., 2007), StOMP (Donoho et al., 2006) and CoSaMP (*Compressive Sampling Matching Pursuit*) (Needell and Tropp, 2009) implement this approach. The second approach obtains a sparse representation of the spectral image by solving the optimization problem described in Eq. 12. Algorithms within this approach are SpaRSA (Wright et al., 2009), TwIST (Bioucas-Dias and Figueiredo, 2007) and GPSR (Figueiredo et al., 2007). The third category of algorithms employ a Bayesian framework and assume a prior distribution for the unknown coefficients that favors sparsity (Ji et al., 2008). The fourth approach consists of relaxing the ℓ_0 problem to a related nonconvex problem and attempts to identify a stationary point (Chartrand, 2007). Finally, in the last category are the brute force algorithms which search through all possible support sets, possibly using cutting-plane methods to reduce the number of possibilities (Miller, 2002). The complete analysis and review of these algorithms is found in (Tropp et al., 2010).

The matrix $\mathbf{A}$ plays a crucial role in the mathematics of the inverse CSI problem. Indeed, the Restricted Isometry Property (RIP) of $\mathbf{A}$ must be satisfied to achieve a reliable estimation of the original spectral image. The RIP establishes the conditions necessary for $\mathbf{A}$ such that the ℓ_2 norm of the underlying 3D spectral image is approximately preserved under the transformation $\mathbf{A}\theta$. Furthermore, the RIP determines the minimum number of compressed projections needed for a correct reconstruction (Arce et al., 2014). Formally, assuming that $|\theta| = K$, the restricted isometry property of the CSI matrix $\mathbf{A}$ of order K is defined as the smallest δ_s such that

$$(1-\delta_s)\|\theta\|_2^2 \leq \|\mathbf{A}\theta\|_2^2 \leq (1+\delta_s)\|\theta\|_2^2, \tag{13}$$

where the constant δ_s is given by

$$\delta_s = \max_{\mathscr{T}\subset[MNL],|\mathscr{T}|\leq K} \|\mathbf{A}_{|\mathscr{T}|}^T \mathbf{A}_{|\mathscr{T}|} - \mathbf{I}\|_2^2, \tag{14}$$

the operator $\|\cdot\|_2^2$ is the squared norm from ℓ_2 into ℓ_2, $\mathbf{A}_{|\mathscr{T}|}$ is a $KV \times |\mathscr{T}|$ matrix whose columns are equal to $|\mathscr{T}|$ columns of $\mathbf{A}$ indexed by the set $\mathscr{T}$, and $\mathbf{I}$ is an identity matrix (Rauhut, 2010).

Since the coded aperture directly determines the structure of the matrix $\mathbf{A}$, different works have proposed to improve the spectral image reconstruction quality by designing a set of coded apertures such that the RIP is better satisfied (Arguello and Arce, 2014; Rueda et al., 2015, 2014). For instance, in (Arguello and Arce, 2014) the traditional block-unblock coded apertures are replaced with a set of colored coded apertures. Then, an optimal design of such coded apertures is developed based on the RIP.

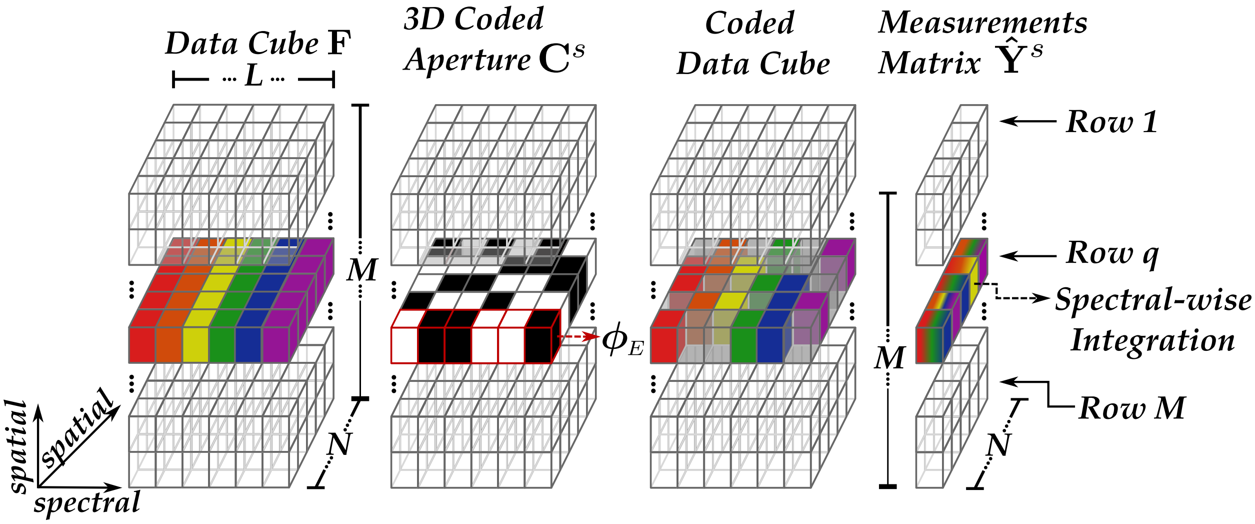


Figure 11. CSI sensing approach for compressed measurements acquisition at snapshot s.

1.5. Spatial-spectral coded compressive spectral imager (3D-CASSI)

The spatial-spectral coded compressive spectral imager (3D-CASSI) is a CSI sensing scheme which modulates the spectral data cube in spatial and spectral dimensions using a 3D coded aperture (ensembles of 2D coded apertures) or a coding pattern array, see Fig. 12. Then, the coded spectral data cube is integrated along the spectral dimension such that each spatial position of the acquired measurements contains the compressed information of a single coded spectral signature (Cao et al., 2016). As shown in Fig. 11, the 3D-CASSI first modulates the scene using a 3D coded aperture $\mathbf{C}$, whose entries are indexed as $C_{m,n,k}$ and then, the coded spectral scene is integrated along the spectral axis. The output of the sensing process, at a specific snapshot s, can be expressed as

$$\hat{Y}^{s}_{m,n} = \sum_{k=0}^{L-1} \mathscr{F}_{m,n,k} C^{s}_{m,n,k}, \tag{15}$$

where $\hat{\mathbf{Y}}^{s}$ is the matrix containing the compressed information of all spectral signatures

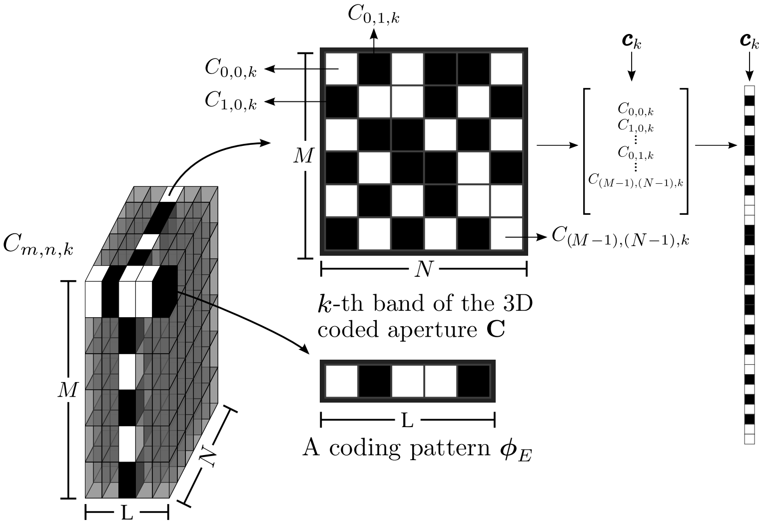


Figure 12. Representation of the 3D coded aperture. This ensemble of block-unblock 2D coded apertures can be seen as a set of different coding patterns ϕ_E.

at a specific spatial position (m,n) and snapshot s. Also, observe that in the 3D-CASSI sensing approach $V = N$, i.e., the number of acquired compressed measurements is $MV = MN$.

Note that each voxel of the spectral data cube is coded by one voxel of the 3D coded aperture at the same position (m,n,k). More specifically, each spatial location $C_{m,n}$ contains a coding pattern $\phi_E \in \mathbb{R}^L$, with $\phi_k \in \{0,1\}$, that modulates a spectral pixel in that particular position (m,n), see Fig. 12.

A snapshot of the 3D-CASSI can be described in vector form using Eq. 11, where $\mathbf{y}^s$ is the vectorization of the matrix $\hat{\mathbf{Y}}^s$ and $\mathbf{H}$ is a $MN \times MNL$ matrix whose structure is determined by the 3D coded aperture. Formally, the j-th row entries of the sensing matrix $\mathbf{H}^s$ can be written as

$$\left(\mathbf{h}_j\right)_\ell^s = \begin{cases} (\phi_E)_{\lfloor \ell/MN \rfloor} & \text{if} \quad j = \ell - \lfloor \ell/MN \rfloor MN \\ 0 & \text{otherwise,} \end{cases} \tag{16}$$

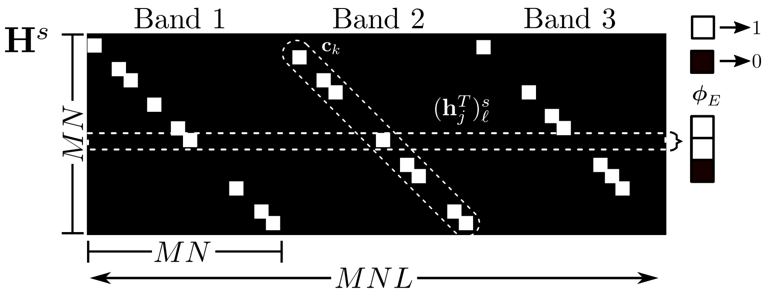


Figure 13. Sensing matrix $\mathbf{H}$ for $N = 4, M = 4, L = 3$ and $S = 2$.

for $j = 0, \cdots, MN-1$, $\ell = 0, \cdots, MNL-1$ and $s = 0, \cdots, S$, where $\phi_E \in \{\phi_0, \cdots, \phi_{P-1}\}$ is selected at random among P coding patterns, such that a different one is assigned for each snapshot s at a specific row j. An example of the $\mathbf{H}^s$ matrix is shown on Fig. 13 for $N = 4, M = 4$ and $L = 3$.

Considering that one spectral signature is coded by a different coding pattern at each measurement shot, there are essentially three cases to analyze. The first case is when the number of measurement shots is greater than the number of coding patterns, i.e., $S > P$. In this case, some pixels are oversampled; thus redundant information is acquired. The second case is when $S < P$. In such a case, the pixels are coded by a subset of coding patterns. Since a specific pattern ϕ_E encodes a determined group of spectral bands, every subset must be designed such that all the spectral bands are sensed to avoid losing or discarding important information. Furthermore, clustering two spectral signatures become harder if they were coded by different subsets of coding patterns with no elements in common. Finally, in the case $S = P$, all spectral signatures are coded by the same set of coding patterns and each ϕ_E can be designed such that no redundant information is acquired. Through this thesis, the case $S = P$ is assumed for simplicity. Therefore, I denote ϕ_E=ϕ_s in Chapter 5.

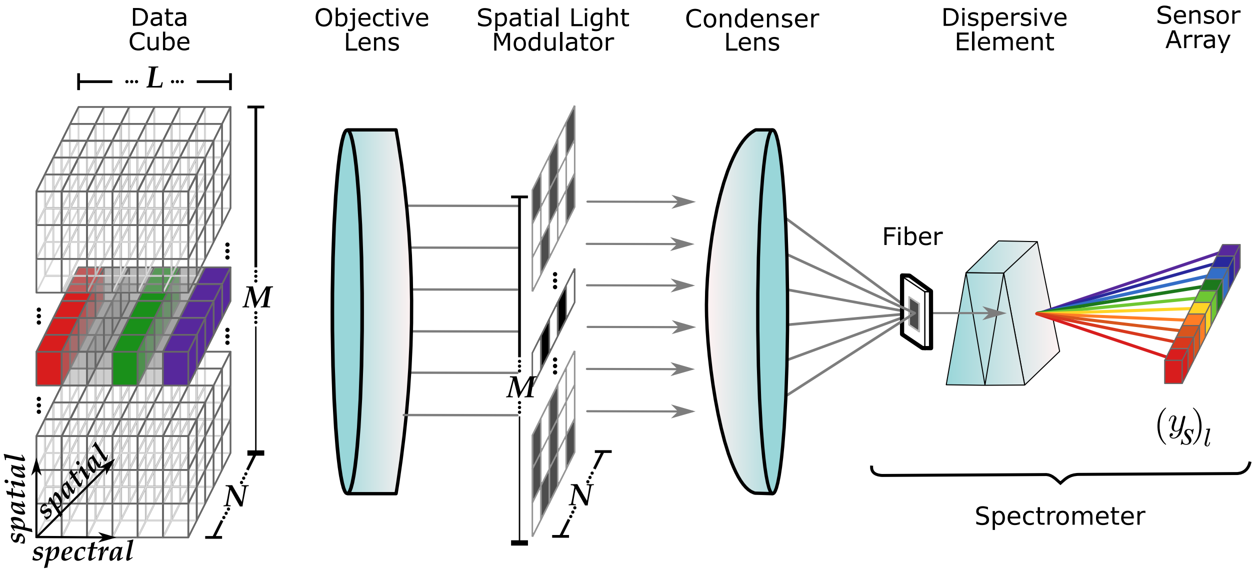


Figure 14. Sensing process of the single pixel camera (SPC) with a point spectrometer.

1.6. Single Pixel Camera

Another CSI architecture used in this thesis is the single-pixel camera (SPC) which compress each spectral band in a single pixel measurement. SPC employs a point spectrometer to obtain the spectral information (Duarte et al., 2008). Figure 14 depicts the components and sensing process of the adopted SPC architecture. Specifically, the objective lens focuses the input 3D scene $\mathscr{F}$, with L spectral bands and $M \times N$ spatial pixels, onto the coded aperture $\mathbf{T} \in \mathbb{R}^{M \times N}$, that spatially modulates each spectral pixel. The coded aperture $\mathbf{T}$ can be modeled as a binary pattern, that blocks the light or lets it pass through each pixel, see section 1.4. Then, the coded scene passes through the condenser lens, which concentrates the light to a single spatial point, that contains all the encoded data. Finally, a single point Whisk-broom spectrometer is used as a detector to obtain the spectral information. Mathematically, the discrete sensing process can be modeled as

$$y_l = \sum_{m=1}^{M} \sum_{n=1}^{N} T_{m,n} \mathscr{F}_{m,n,l}, \tag{17}$$

for $l = 1 \cdots L$, where y_l represents the measurement corresponding to the l-th band. Note that one coded aperture modulates all spectral bands uniformly, hence, for each band, the sensing process can be rewritten in the standard form of linear equations as

$$y_l = \mathbf{h}^T \mathbf{f}_l, \tag{18}$$

where $\mathbf{h} \in \{1, -1\}^{MN}$ is the vectorization of the coded aperture, given by $\mathbf{h} = [T_{1,1}, T_{2,1}, \cdots, T_{M,N}]$, and $\mathbf{f}_l$ is the vector form of the l-th spectral band of $\mathscr{F}$. Furthermore, it is possible to capture several snapshots by employing a different coded aperture pattern each time. Then, the multi-shot scheme for each band is expressed in matrix form as

$$\mathbf{y}_l = \mathbf{H}\mathbf{f}_l, \tag{19}$$

where $\mathbf{H} \in \mathbb{R}^{K \times MN}$, K is the number of shots, each row of $\mathbf{H} = \left[\mathbf{h}_1^T, \cdots, \mathbf{h}_K^T\right]$ is the vector form of the coded aperture used on that particular shot, and $\mathbf{y}_l = [y_1, \cdots y_K]^T$.

In general, the sensing model for all spectral bands can be stacked in a single vector as

$\mathbf{y} = [\mathbf{y}_1^T \cdots \mathbf{y}_L^T]^T$, such that the sensing model can be rewritten as

$$\mathbf{y} = (\mathbf{I}_{(L)} \otimes \mathbf{H})[\mathbf{f}_1^T, \cdots, \mathbf{f}_L^T]^T = \hat{\mathbf{H}}\mathbf{f}, \tag{20}$$

where $\hat{\mathbf{H}} = \mathbf{I}_L \otimes \mathbf{H}$, is a block diagonal matrix, $\mathbf{I}_L$ represents an $L \times L$ identity matrix, and $\otimes$ denotes the Kronecker product. The compression ratio in this model is given by $\gamma = \frac{K}{MN}$, where $\gamma \in [0,1]$.

1.7. CSI and Random Projection

A dimensionality reduction technique that is able to reduce the data into a lower-dimensional model, while preserving the reconstructive or discriminative properties of the original data can be marked as ideal. However, in practice information is lost as the dimensionality is reduced. Therefore, a method which efficiently reduces dimensionality, while preserving as much as possible information from the original data is needed. One solution is to reduce the dimensionality of data by projecting it onto a lower-dimensional subspace (Lin and Gunopulos, 2003).

Principal component analysis (PCA) is a classic dimensionality reduction (DR) method that finds the low-dimensional linear subspace that minimizes the mean-squared error between the original data and the data projected onto the subspace. A low-dimensional representation of the data is constructed in such a way that it describes as much of the variance in the data as possible. This is achieved by finding a linear basis of reduced dimensionality for the data (a set of eigenvectors) in which the variance of the data is maximal (Van Der Maaten et al., 2009).

PCA, however, is based on a linear subspace model that is generally not capable of capturing the geometric structure of real-world datasets (Baraniuk et al., 2010). Besides, most of the computationally efficient approaches to computing the principal components assume ready access to the stored full data samples. Nevertheless, this full data access is not always possible in modern data settings. For instance, in the traditional spectral imaging approaches, the whole data is obtained by gradually scanning adjacent zones of the spectral scene; hence the PCA analysis must be strictly performed after the sensing process. Furthermore, considering a spectral image with $M \times N$ spatial dimension and L spectral bands, computing the Q principal components takes $O(MNL^2 + L^3 + MNQL) = O(MNL^2)$ which is computationally expensive when the spatial resolution or the number of spectral bands grow exponentially (Du and Fowler, 2008).

One promising strategy to reduce the data dimensionality efficiently, which also allows for rigorous theoretical analysis, is to use Random Projections (RP). In RP, the data dimensionality reduction is computationally simple: the original high-dimensional data is projected onto a low-dimensional subspace using a random matrix, thereby saving memory and computation. For example, reducing the dimension of a $M \times N \times L$ spectral image using RP will only take $O(MNLS)$, which is a significant computation time reduction in comparison to PCA.

There are theoretical results supporting that RP preserves volumes and affine distances (Magen, 2002) or the data structure; hence it is possible to learn from the low dimensional data (Calderbank et al., 2009) directly. Based on the concept of affinity (Soltanolkotabi et al., 2012), which characterizes the similarity between two subspaces, it has been theoretically proved and numerically verified that several dominant subspace clustering algorithms could successfully perform clustering on the compressed data (Heckel et al., 2015; Tschannen and Bölcskei, 2016; Wang et al., 2016; Li and Gu, 2017; Meng et al., 2018). Also, recent works have shown that RP improves the algorithm performance when dealing with high dimensional data. Specifically, compression reduces the dimension of environment signal space; hence the computational cost of finding the similarity representation in subspace clustering can be efficiently reduced (Xu et al., 2020; Meng et al., 2018).

Generally, RP uses a random matrix with elements generated by a normal distribution of $N(0,1)$ because of its simplicity in terms of analysis. The problem of this type of RP matrix is the computational complexity due to the dense nature of the projection matrix and hardware implementation difficulties (Vempala, 2005). However, different authors have suggested the use of simpler distributions that generate sparse projections matrices (Achlioptas, 2003; Li et al., 2006). The introduction of sparsity has shown to significantly reduce computational requirements and to allow an efficient hardware implementation (Fox et al., 2016).

Given a high dimensional data set, the random projection is essentially performed by applying the same realization of a random projection matrix to each data point to obtain the set of dimensionality reduced or compressed measurements. In this sense, the CSI sensing mechanism can be rewritten in a RP scheme. Particularly, the measurement vectors $\mathbf{y}^s$ acquired with each snapshot, can be arranged in a matrix $\mathbf{Y} = \left[\mathbf{y}^0, \mathbf{y}^1, \cdots, \mathbf{y}^{S-1}\right]^T$ where each column contains different spectral signature codifications, acquired with distinct coding patterns. If the case $S = P$ is assumed, it is possible to rearrange the entries of $\mathbf{Y}$ such that each row contains a compressed spectral signature acquired with a specific coding pattern ϕ_s. Formally, this rearrangement can be expressed as

$$Y_{sj} \leftrightarrow Y_{s'j} \quad \text{if } Y_{s'j} = \phi_s^T f_j,$$

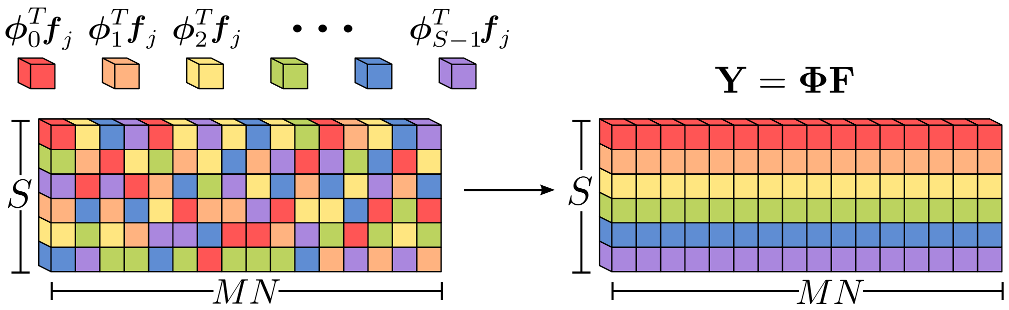


Figure 15. Rearrangement of the matrix **Y** such that the s-th row contains the compressed measurements acquired with the s-th coding pattern ϕ_s. In this figure, colors represent a specific codification, e.g, red pixels denote the compressed measurements acquired with the ϕ_0 coding pattern.

for $s, s' = 0, \cdots, S-1$ $(s' \neq s)$ and $j = 0, \cdots, MN-1$, where $\leftrightarrow$ stands for the swap operation and f_j is the j-th spectral signature. The described acquisition model and the rearrangement is depicted in Fig. 15. Alternatively, define the matrix of S coding patterns as $\Phi = \left[\phi_0^T, \phi_1^T, \cdots \phi_{S-1}^T\right]^T$, the problem of acquiring and rearranging the measurements $\mathbf{y}$ can be succinctly expressed as

$$\mathbf{Y} = \mathbf{\Phi F}, \tag{21}$$

where $\mathbf{F} = [\mathbf{f}_0, \mathbf{f}_1, \cdots, \mathbf{f}_{L-1}]^T$ is a $L \times MN$ matrix whose columns, denoted as f_j, are the spectral signatures of the data cube.

Spectral image recovery in Eq. 12 is typically achieved using nonlinear and relatively expensive optimization-based or iterative algorithms. However, signal recovery is not necessary for many signal processing applications (Davenport et al., 2010). In particular, as shown in chapters 5 and 6, it is possible to apply the designed clustering algorithms directly on the compressed measurements; thus the cost of recovering all the data is avoided.

2. Hyperspectral image segmentation using 3D regularized subspace clustering model

Part of this chapter has been adapted from the journal paper (Hinojosa et al., 2021c) (published) and the conference paper (Hinojosa et al., 2018b) (published).

Table 1
Notation used in Chapter 2.

Notation	**Definition**
M	Number of rows in the HSI.
N	Number of columns in the HSI.
L	Number of spectral bands of the HSI.
$\mathbf{X} \in \mathbb{R}^{L\times MN}$	Points in-sample data.
$\mathbf{Z} \in \mathbb{R}^{MN\times MN}$	Sparse coefficient matrix.
$\hat{\mathbf{Z}} \in \mathbb{R}^{M\times N\times MN}$	Sparse coefficients ($\mathbf{Z}$) in 3D form.
$\bar{\mathbf{Z}} \in \mathbb{R}^{MN\times MN}$	Enhanced coefficient matrix.
$\mathbf{U} \in \mathbb{R}^{MN\times MN}$	Auxiliary variable in the ADMM Algorithm.
$\mathbf{G} \in \mathbb{R}^{h\times h\times h}$	3D Gaussian Kernel with window size h.
$\mathbf{W} \in \mathbb{R}^{N\times N}$	Similarity matrix.

2.1. Introduction

In general, the accurate unsupervised classification of the spectral pixels is challenging due to the high-dimensional feature space. Many different clustering methods for HSI land-cover segmentation have been proposed in the literature. The Fuzzy c-mean (Bezdek, 2013) (FCM) is one the early and most used methods for image segmentation, and its success relies on the introduction of fuzziness for the belonging of each image pixel. Even though FCM is efficient for images with simple texture and background, it fails to segment images with complex structures, e.g., HSI or images corrupted by noise since it does not take advantage of spatial information. In this respect, authors in (Chen and Zhang, 2004) proposed the Spatial Fuzzy c-mean (FCM_S1), which employee average filtering and median filtering to obtain the spatial neighborhood information in advance.

Among recent works on HSI unsupervised classification, the Robust Manifold Matrix Factorization (RMMF) is based on unified low-rank matrix factorization and performs dimensionality reduction in conjunction with data clustering (Zhang et al., 2019). The authors also designed an Augmented Lagrangian method-based procedure to find the optimal local solution of the proposed optimization. Moreover, in (Nalepa et al., 2020), authors presented the 3D-CAE, an end-to-end approach to segment HSI and offer high-quality segmentation in an unsupervised way. In particular, the paper introduces a deep learning technique that uses a 3-D convolutional autoencoder to learn embedded features with later undergo clustering, which is performed during the network training with a clustering-oriented loss.

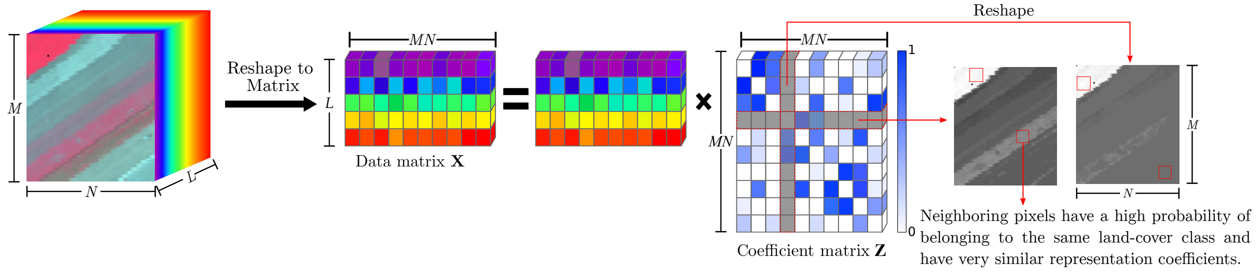
On the other hand, the sparse subspace clustering (SSC), described in section 1.3.3, has successfully been applied to HSI to achieve unsupervised spectral signature classification. Some recent subspace clustering algorithms proposed to select the most representative data points before building the affinity matrix instead of learning the underlying subspaces by expressing each point as a linear combination of all other points. For instance, the Scalable and Robust SSC (SR-SSC) (Abdolali et al., 2019) selects a few sets of anchor points using a randomized hierarchical clustering method. Then, within each set of anchor points, it solves the LASSO (Tibshirani, 1996) problem for each data point, only allowing anchor points to have non-zero weights. Similarly, the Exemplar-based Subspace Clustering (ESC-FFS) (You et al., 2018) proposes to select a small subset that represents all data points by using the Farthest first search (FFS) (You et al., 2018) algorithm, which is a modified version of the Farthest-First Traversal (FFT) algorithm (Williamson and Shmoys, 2011). However, these general-purpose methods do not fully exploit the complex structure of remotely sensed hyperspectral images. Furthermore, they do not consider the rich spatial information of the spectral images, which could boost the accuracy of these algorithms on HSI imagery.

The spectral signatures corresponding to a specific land-cover material would generally lie in the same non-linear manifold. However, the manifold clustering represents a more complex problem than subspace (flat manifolds) clustering (Elhamifar and Vidal, 2011). On the other hand, different previous works have been successfully modeling the hyperspectral image segmentation problem using the subspace clustering theory (Zhang et al., 2016; Zhai et al., 2016). Those works relax the problem by assuming that the spectral variability of the signatures is approximately linear; hence the spectral signatures with similar spectral characteristics have a high probability of lying in the same affine low-dimensional subspace. For instance, the SSC algorithm has been successfully used for land cover segmentation (Guo et al., 2014; Mehta and Dikshit, 2017; Hinojosa et al., 2018a; Liang et al., 2019; Huang et al., 2019). However, the traditional SSC model only captures the relationship between pixels by analyzing the spectral features and does not take advantage of spatial information. Considering that spectral pixels belonging to the same land cover material are arranged in common regions within the HSI, the sparse representation matrix of the SSC model should also account for the spatial relationship between neighboring pixels. Indeed, it is expected that the obtained representation coefficients of two adjacent pixels would be very similar since they capture the geometric relationship among pixels. Recently, few works have proposed to take into account the neighboring spatial information by incorporating a regularization term in the SSC optimization problem, which enforces a local averaging constraint on the sparse coefficient matrix (Zhang et al., 2016; Zhai et al., 2016; Hinojosa et al., 2018a). However, the regularization term proposed in such works assigns the same sparse basis to adjacent pixels by performing independent 2D smoothing operations over one column vector of the SSC representation matrix without taking advantage of the information among columns.

Chapter contribution. This chapter presents an efficient integration of the HSI's spatial-contextual information in the SSC model by performing a 3D convolution operation with a 3D Gaussian filter over the representation coefficient matrix. This procedure provides a more accurate coefficient matrix by considering the information among adjacent representation coefficient column vectors of the SSC matrix to enforce an averaging constraint, preserving the similarity and improving the clustering accuracy. Incorporating the 3D convolution into the SSC model is done by adding a regularization term to the optimization problem. Experimental results demonstrate that the proposed approach improves the clustering performance in visual and quantitative evaluations.

2.2. Sparse Subspace Clustering with 3D Spatial Regularization

In the SSC model, the j-th column of the sparse representation matrix $\mathbf{Z}$ contains the representation coefficient distribution of the whole image with respect to a single atom. Alternatively, using the spectral unmixing jargon, this can be thought of as a specific fractional abundance concerning an endmember in the unmixing domain. Similarly, the j-th row of $\mathbf{Z}$ should present a similar



Neighboring pixels have a high probability of belonging to the same land-cover class and have very similar representation coefficients.

Figure 16. Graphical interpretation of each column and row of the sparse coefficient matrix $\mathbf{Z}$.

coefficient distribution since the symmetric nature of $\mathbf{Z}$. The meaning of each row or column of $\mathbf{Z}$ is better appreciated by reshaping the corresponding MN vector into a $M \times N$ matrix, as shown in Fig. 16.

Notice that the spectral pixels belonging to the same land cover material are arranged in common regions, i.e., two spatially neighboring pixels in an HSI usually have a high probability of belonging to the same class. Hence, according to the SSC mechanism, their representation coefficients should also be very close concerning the same sparse basis. Then, in general, the obtained $\mathbf{Z}$ should be piecewise smooth since there is a spatial relationship between the sparse representation vector of every pixel and its neighbors. This observation suggests that it is reasonable to apply a spatial constraint to $\mathbf{Z}$ in the optimization problem. In this chapter, the representation coefficient matrix is improved by enforcing a 3D spatial regularization, incorporating the pixels' contextual dependence.

2.2.1. Problem Formulation. As previously discussed, the spatial neighborhood information of an HSI can be incorporated into the SSC model by constraining $\mathbf{Z}$ to be piecewise smooth. One approach to achieve it is by using a smoothing convolution that assigns similar sparse coefficients to adjacent pixels. Since it is desired that the sparse coefficient matrix obtained by the SSC

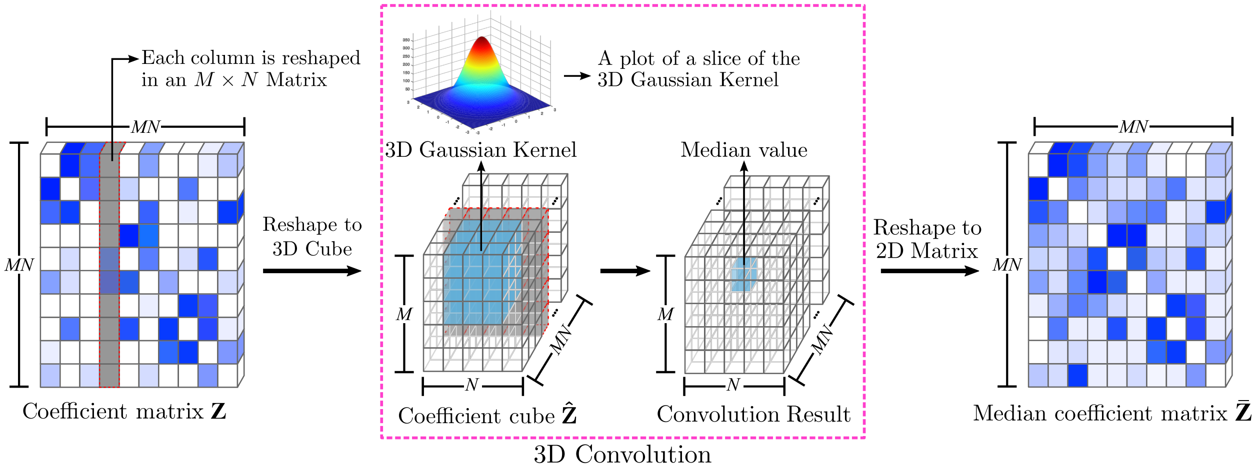


Figure 17. The proposed methodology to enhance the sparse representation coefficients matrix using a 3D convolution.

to be symmetric, and hence rows and columns of $\mathbf{Z}$ present a similar coefficient distribution, it is proposed to use a 3D convolution to perform the smoothing on $\mathbf{Z}$.

Specifically, first the 2D coefficient matrix $\mathbf{Z}$ is rearrange into a 3-D cube $\hat{\mathbf{Z}} \in \mathbb{R}^{M\times N\times MN}$, such that each column of $\mathbf{Z}$ corresponds to a slice of $\hat{\mathbf{Z}}$. Then, a 3D smoothness convolution is performed by opening a 3D window to each spatial location of $\hat{\mathbf{Z}}$. The aim of using a 3D Gaussian kernel to perform the smoothness operation is to simultaneously take advantage of the information among adjacent columns/rows of $\mathbf{Z}$, obtaining a more accurate sparse coefficient matrix. The filtering is performed using the isotropic 3D Gaussian kernel given by

$$G_{i,j,k} = -\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{i^2+j^2+k^2}{2\sigma^2}}, \tag{22}$$

with a specific window size h and standard deviation σ. After the filtering process, the cube $\hat{\mathbf{Z}}$ is rearranged into the filtered coefficient matrix $\bar{\mathbf{Z}}$. Figure 17 depicts the proposed methodology described above.

Instead of directly imposing a smoothness constraint over $\mathbf{Z}$, the enhanced coefficient matrix $\bar{\mathbf{Z}}$ can be used to regularize the solution of the SSC optimization problem described in Eq. (7). Therefore, the proposed subspace clustering model for HSI segmentation is expressed as follows

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{R}} \quad & \|\mathbf{Z}\|_1 + \frac{\lambda}{2}\|\mathbf{R}\|_F^2 + f_{3DS}(\mathbf{Z}) \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{R},\ \operatorname{diag}(\mathbf{Z}) = 0,\ \mathbf{Z}^T\mathbf{1} = \mathbf{1}, \end{aligned} \tag{23}$$

where

$$f_{3DS}(\mathbf{Z}) = \frac{\alpha}{2}\|\mathbf{Z} - \bar{\mathbf{Z}}\|_F^2 \tag{24}$$

corresponds to the 3D spatial regularization term.

2.2.2. Optimization Algorithm. To solve the optimization problem in Eq. (23), an algorithm is developed based on the alternating direction method of multipliers (ADMM) scheme (Boyd et al., 2011). First, an auxiliary matrix $\mathbf{U} \in \mathbb{R}^{N\times N}$ with the same size as the sparse coefficient matrix $\mathbf{Z}$ is defined to separate the variables. In this way, the following optimization problem needs to be solved:

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{U},\bar{\mathbf{Z}}} \quad & \|\mathbf{Z}\|_1 + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{U}\|_F^2 + \frac{\alpha}{2}\|\bar{\mathbf{Z}} - \mathbf{U}\|_F^2 \\ \text{s.t.} \quad & \mathbf{U}^T\mathbf{1} = \mathbf{1}, \mathbf{U} = \mathbf{Z} - \operatorname{diag}(\mathbf{Z}), \end{aligned} \tag{25}$$

where the regularization term $f_{3DS}(\mathbf{Z})$ of Eq. 24 is directly replaced in Eq. (23). Note that, for convenience, in this section, $\operatorname{diag}(\mathbf{Z})$ denotes both a vector whose elements are the diagonal entries of $\mathbf{Z}$ and a diagonal matrix whose diagonal elements are the diagonal entries of $\mathbf{Z}$.

Following the ADMM scheme, and using a parameter $\rho > 0$, two penalty terms are added to the objective function of Eq. (25) corresponding to the constraints $\mathbf{U^T 1} = \mathbf{1}$ and $\mathbf{U} = \mathbf{Z} - \mathrm{diag}(\mathbf{Z})$. Then, the following optimization problem is obtained:

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{U},\bar{\mathbf{Z}}} \quad & \|\mathbf{Z}\|_1 + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{U}\|_F^2 + \frac{\alpha}{2}\|\bar{\mathbf{Z}} - \mathbf{U}\|_F^2 \\ & + \frac{\rho}{2}\|\mathbf{U}^T\mathbf{1} - \mathbf{1}\|_2^2 + \frac{\rho}{2}\|\mathbf{U} - (\mathbf{Z}{-}\mathrm{diag}(\mathbf{Z}))\|_F^2 \\ \text{s.t.} \quad & \mathbf{U}^T\mathbf{1} = \mathbf{1}, \mathbf{U} = \mathbf{Z}{-}\mathrm{diag}(\mathbf{Z}). \end{aligned} \tag{26}$$

It can be easily proved that adding the penalty terms to Eq. (25) do not change its optimal solution, then, the solutions to Eq. (23) and (25) coincide with that of (26) (Boyd et al., 2011). Next, let a vector $\bar{\delta} \in \mathbb{R}^{MN}$ and a matrix $\bar{\Delta} \in \mathbb{R}^{MN \times MN}$ be the Lagrange multipliers for the two equality constraints in (26) to obtain the Lagrange function

$$\begin{aligned} \mathscr{L}(\mathbf{Z},\mathbf{U},\bar{\mathbf{Z}},\bar{\delta},\bar{\Delta}) = & \|\mathbf{Z}\|_1 + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{U}\|_F^2 + \frac{\alpha}{2}\|\bar{\mathbf{Z}} - \mathbf{U}\|_F^2 \\ & + \frac{\rho}{2}\|\mathbf{U}^T\mathbf{1} - \mathbf{1}\|_2^2 + \frac{\rho}{2}\|\mathbf{U} - (\mathbf{Z}{-}\mathrm{diag}(\mathbf{Z}))\|_F^2 \\ & + \bar{\delta}^T(\mathbf{U}^T\mathbf{1} - \mathbf{1}) + \mathrm{tr}(\bar{\Delta}^T(\mathbf{U} - \mathbf{Z} + \mathrm{diag}(\mathbf{Z}))), \end{aligned} \tag{27}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operator of a given matrix.

The optimization problem in Eq. (27) can then be divided into four subproblems:

1. **U update:** $\mathbf{U}^{(t+1)}$ is obtained by minimizing $\mathscr{L}$ with respect to $\mathbf{U}$, while $(\mathbf{Z}^{(t)},\bar{\mathbf{Z}}^{(t)},\bar{\delta}^{(t)},\bar{\Delta}^{(t)})$ are fixed. The derivative of $\mathscr{L}$ is calculated with respect to $\mathbf{U}$ and set it to zero to obtain the calculation formula of $\mathbf{U}$ as follows:

$$\begin{aligned}(\lambda\mathbf{X}^T\mathbf{X}+\alpha\mathbf{I}+\rho\mathbf{1}\mathbf{1}^T+\rho\mathbf{I})\mathbf{U}^{(t+1)} &= \lambda\mathbf{X}^T\mathbf{X}+\alpha\bar{\mathbf{Z}}^{(t)} \\ &+\rho(\mathbf{1}\mathbf{1}^T+\mathbf{Z}^{(t)})-1\delta^{(t)^T}-\bar{\Delta}^{(t)}.\end{aligned} \tag{28}$$

2. **Z update:** $\mathbf{Z}^{(t+1)}$ is computed by minimizing $\mathscr{L}$ with respect to $\mathbf{Z}$, while $(\mathbf{U}^{(t+1)},\bar{\mathbf{Z}}^{(t)},\bar{\delta}^{(t)},\bar{\Delta}^{(t)})$ are fixed, obtaining

$$\begin{aligned}\mathbf{Z}^{(t+1)} &= \mathbf{J}-\operatorname{diag}(\mathbf{J}), \\ \mathbf{J} &\triangleq \Gamma_{\frac{1}{\rho}}\left(\mathbf{U}^{(t+1)}+\frac{\bar{\Delta}^{(t)}}{\rho}\right),\end{aligned} \tag{29}$$

where $\Gamma_{1/\rho}(\cdot)$ is a shrinkage-thresholding operator, $\Gamma_{1/\rho}(v)=(|u|-(1/\rho))_{+}\operatorname{sgn}(v)$, and the operator $(\cdot)_{+}$ returns its arguments if it is nonnegative and zero otherwise.

3. **Enhanced coefficient matrix $\bar{\mathbf{Z}}$ update:** Once $\mathbf{Z}^{(t+1)}$ is obtained in the previous step, the actual matrix $\mathbf{Z}$ is used to follow the procedure described in Fig. 17 to update $\bar{\mathbf{Z}}^{(t+1)}$. This thesis proposes to use the Gaussian Kernel given by Eq. (22).

4. **The Lagrange multipliers update:** $\bar{\delta}^{(t+1)}$ and $\bar{\Delta}^{(t+1)}$ are obtained through a gradient ascent update with step size $\rho = 300$, as follows

$$\begin{aligned}\bar{\delta}^{(t+1)} &= \bar{\delta}^{(t)}+\rho\left(\mathbf{Z}^{(t+1)}\mathbf{1}-\mathbf{1}\right) \\ \bar{\Delta}^{(t+1)} &= \bar{\Delta}^{(t)}+\rho\left(\mathbf{U}^{(t+1)}-\mathbf{Z}^{(t+1)}\right).\end{aligned} \tag{30}$$

These four steps are repeated until convergence is achieved or the number of iterations exceeds the maximum fixed. Specifically, the iteration is terminated when $\|\mathbf{U}^{(t)^T}\mathbf{1}-\mathbf{1}\|_\infty \leq \bar{\varepsilon}$, $\|\mathbf{U}^{(t)}-$

Algorithm 2 Sparse Subspace Clustering with 3D Regularization (3DS-SSC)

```
Input  : HSI Data $\mathbf{X} \in \mathbb{R}^{D \times N}$, containing a set of spectral pixels $\{\mathbf{x}_j\}_{j=1}^{N}$, in a union of $k$ affine subspaces $\{\mathscr{S}_i\}_{i=1}^{k}$;
         the number of subspaces $k$ and the regularization parameter $\lambda$; the window size $h$ and standard deviation $\sigma$
         of the 3D Gaussian kernel; ADMM maximum number of iteration $MAXITER$
Output: Matrix $\hat{\mathbf{X}}$ with the obtained clustering labels of $\mathbf{X}$
Function 3DS-SSC($\mathbf{X}, k, \lambda$)
    ▷ Solving the optimization problem in Eq.  (23)
    $\mathbf{U}^{(0)} = 0,\ \mathbf{Z}^{(0)} = 0,\ \bar{\mathbf{Z}}^{(0)} = 0,\ \bar{\delta}^{(0)} = 0,\ \bar{\Delta}^{(0)} = 0,\ \rho = 300$
    for $t = 0, \cdots, MAXITER - 1$ do
        $\mathbf{U}^{(t+1)} \leftarrow \arg\min_{\mathbf{U}} \mathscr{L}(\mathbf{Z}^{(t)}, \mathbf{U}, \bar{\mathbf{Z}}^{(t)}, \bar{\delta}^{(t)}, \bar{\Delta}^{(t)})$
        $\mathbf{Z}^{(t+1)} \leftarrow \arg\min_{\mathbf{Z}} \mathscr{L}(\mathbf{Z}, \mathbf{U}^{(t+1)}, \bar{\mathbf{Z}}^{(t)}, \bar{\delta}^{(t)}, \bar{\Delta}^{(t)})$
        $\bar{\mathbf{Z}}^{(t+1)} \leftarrow conv3D(\mathbf{Z}^{(t+1)}, h, \sigma)$
        $\bar{\delta}^{(t+1)} \leftarrow \bar{\delta}^{(t)} + \rho\left(\mathbf{Z}^{(t+1)}\mathbf{1} - \mathbf{1}\right)$
        $\bar{\Delta}^{(t+1)} \leftarrow \bar{\Delta}^{(t)} + \rho\left(\mathbf{U}^{(t+1)} - \mathbf{Z}^{(t+1)}\right).$
    ▷ Obtaining the segmentation using Spectral Clustering
    Normalize the columns of $\mathbf{Z}$ as $\mathbf{z}_j \leftarrow \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_\infty}$.
    Form a similarity matrix as $\mathbf{W} = |\mathbf{Z}| + |\mathbf{Z}|^T$, representing the relationship (weights) among data points.
    Apply Spectral Clustering (Von Luxburg, 2007b) to the similarity matrix.
    return Segmentation of the data: $\hat{\mathbf{X}}_1, \cdots, \hat{\mathbf{X}}_k$.
Function conv3D($\mathbf{Z}, h, \sigma$)
    ▷ Perform 3D Convolution (see Fig.  17)
    $\hat{\mathbf{Z}} \leftarrow reshape(\mathbf{Z})$ ▷ Reshapes each column of Z into a matrix using columnwise order
    $G_{i,j,k} = -\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{i^2+j^2+k^2}{2\sigma^2}}$ ▷ 3D Gaussian Kernel $\mathbf{G} \in \mathbb{R}^{h \times h \times h}$
    $\hat{\mathbf{Z}} \leftarrow \hat{\mathbf{Z}} * \mathbf{G}$ ▷ 3D Convolution
    $\bar{\mathbf{Z}} \leftarrow reshape(\hat{\mathbf{Z}})$ ▷ Reshapes the cube back to a matrix
    return $\bar{\mathbf{Z}}$
```

$\mathbf{Z}^{(t)}\|_\infty \leq \bar{\varepsilon}, \|\mathbf{U}^{(t)} - \mathbf{U}^{t-1}\|_\infty \leq \bar{\varepsilon}$, where $\bar{\varepsilon}$ denotes the error tolerance for the primal and dual residuals.

Once the sparse coefficient matrix $\mathbf{Z}$ is obtained by solving the optimization problem in Eq. (23), the segmentation of the data points is attained by applying spectral clustering to the Laplacian matrix induced by the similarity matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$, which is defined as $\mathbf{W} = |\mathbf{Z}| + |\mathbf{Z}|^T$ (Elhamifar and Vidal, 2013). The complete SSC with 3D regularization (3DS-SSC) is summarized in Algorithm 2.

2.3. Experimental Evaluation

This section shows the performance of 3DS-SSC for land cover segmentation. The ADMM algorithm for solving the sparse optimization problem in Eq. (23) was implemented in MATLAB. The MATLAB implementation code of the complete algorithm (3DS-SSC) can be found in Appendix 4. The implementation of SSC, described in Algorithm 1, was provided by authors in (Elhamifar and Vidal, 2013). All the experiments were executed on an Intel Core i7 CPU (2.60GHz, 6 cores), with 32 GB of RAM.

2.3.1. Experiment Setup.

Databases. The proposed subspace clustering approach (3DS-SSC) was tested on three well-known hyperspectral images[1], with different imaging environments: Indian Pines, Salinas, and University of Pavia dataset. The **Indian Pines** hyperspectral data set has 145×145 pixels and 200 spectral bands in the range $0.4 - 2.5 \mu m$. The second scene, **Salinas**, has 512×217 pixels and 204 spectral bands in the range $0.24 - 2.40 \mu m$. The third scene, **University of Pavia**, comprises 610×340 pixels and has 103 spectral bands with spectral coverage ranging from $0.43 - 0.84 \mu m$. In the first experiments, one frequently used region of interest (ROI) in hyperspectral image clustering is selected for each image, as shown in Fig. 18. The Indian Pines ROI has a size of 70×70 pixels, which includes four main land-cover classes: corn-no-till, grass, soybeans-no-till, and soybeans-min-till. The Salinas ROI comprises 83×83 pixels and includes six classes: broccoli-1, corn-senesced, lettuce-4wk, lettuce-5wk, lettuce-6wk, and lettuce-7wk. The

---

[1] The datasets can be downloaded from `http://www.ehu.eus/ccwintco/index.php`.

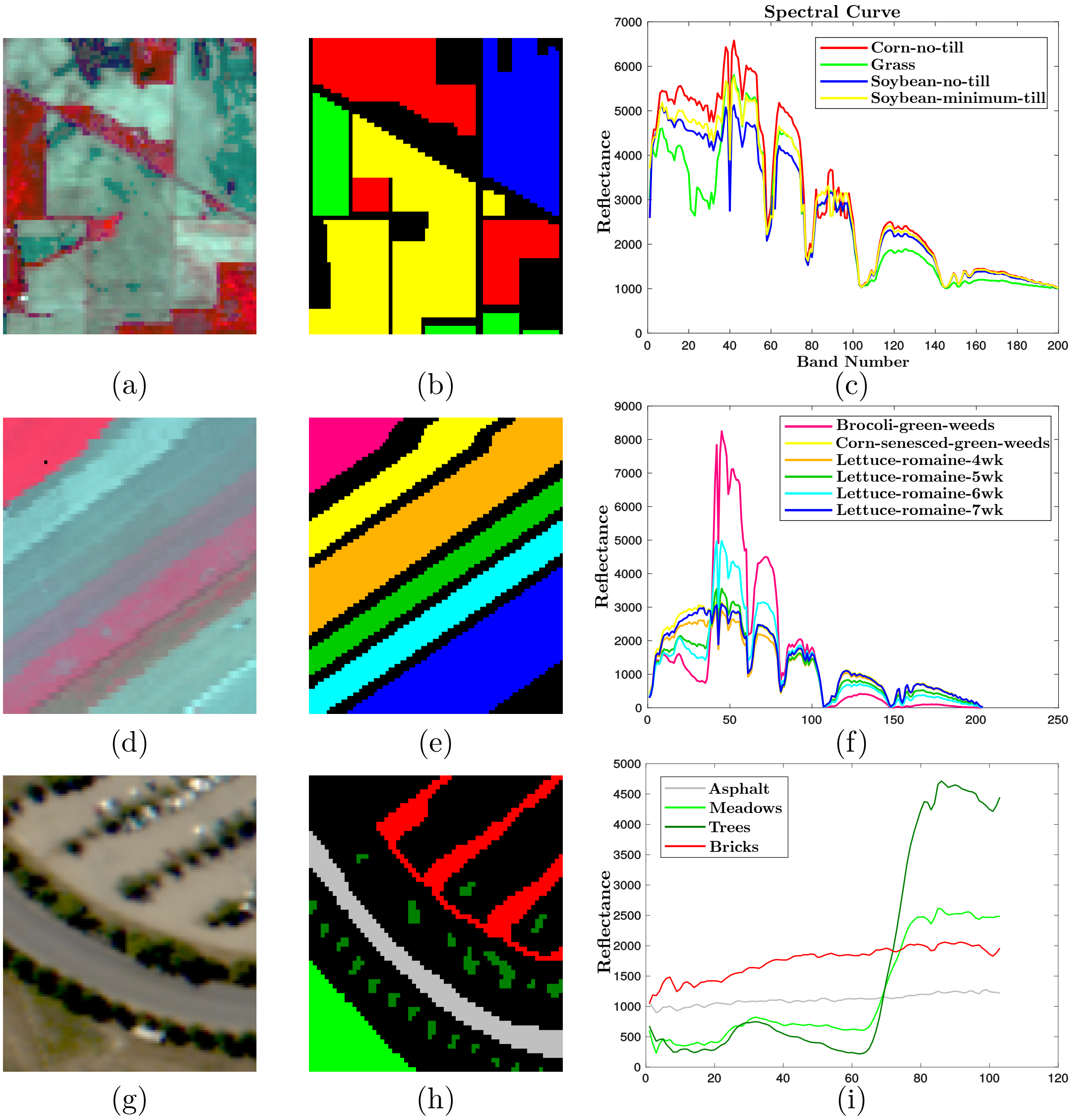


Figure 18. False-color images, ground truth images and spectral curves of each land-cover classes for Indian Pines (a)-(c), Salinas (d)-(f), and University of Pavia (g)-(i) ROIs, respectively.

University of Pavia ROI comprises 64×64 pixels and includes four classes: asphalt, meadows, trees, and bricks. Figure 18 presents a false-color image, the ground truth image, and the spectral signatures of each land cover class within the Indian pines (a)-(c), Salinas (d)-(f), and University of Pavia (g)-(i) selected ROIs, respectively. In addition, the performance of the proposed method is also evaluated using the full hyperspectral images in Section 2.3.2.4.

Baselines and Evaluation Metrics. The proposed approach in this chapter is compared with three non SSC-based algorithms for hyperspectral land cover segmentation: the FCM_S1 (Chen and Zhang, 2004) , the RMMF (Zhang et al., 2019) , and the 3D-CAE (Nalepa et al., 2020) , which use 3D convolutional autoencoders to perform the clustering; with two SSC-based algorithms for hyperspectral land-cover segmentation: the TV-KSSC-SMP (Bacca et al., 2017a) and S-SSC (Zhang et al., 2016) ; and with two state-of-the-art general-purpose subspace clustering algorithms: ESC-FFS (You et al., 2018) and SR-SSC (Abdolali et al., 2019) . Besides, the results obtained with SSC (Elhamifar and Vidal, 2013) are shown as a reference. For the sake of fairness, ten experiments were performed for the ESC-FFS method, which relies on random initialization. Therefore, the presented results in the tables are the average of all run trials, and the best results were selected to show the land cover maps.

To compare the clustering performance of the proposed model in this chapter, six standard metrics were used: producer's accuracy (PA), user's accuracy (UA), average accuracy (AA), overall accuracy (OA), Kappa coefficient, and normalized mutual information (NMI) (Lillesand et al., 2015; Strehl and Ghosh, 2002). PA and UA represent the clustering accuracy of each class. UA, AA, and OA values are presented in percentage, while Kappa coefficients and NMI values range from 0 to 1, where 1 stands for perfect classification.

Parameters Setting.

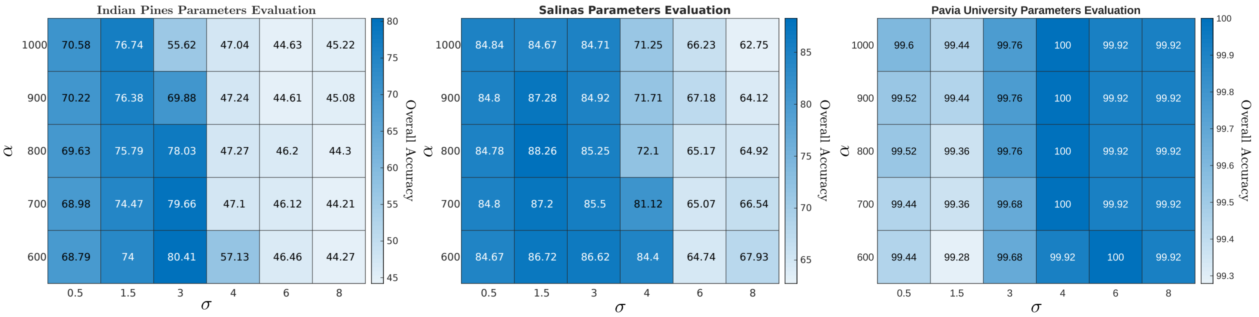
The number of clusters was manually set for all the subspace clustering algorithms in the experiments. The regularization parameter λ, in Eq. (23), which acts as the trade-off between the sparsity of the coefficient matrix and the magnitude of the noise, was set using the following formulation (Elhamifar and Vidal, 2013):

$$\lambda = \frac{\beta}{\gamma}, \quad \gamma = \min_{j} \max_{j \neq j'} |\mathbf{x}_{j'}^T \mathbf{x}_j|, \tag{31}$$

where β is the adjustment coefficient, γ is a parameter related to the data set, which can be explicitly determined, and $\mathbf{x}_j, \mathbf{x}_{j'}$ are columns of $\mathbf{X}$. The regularization parameter α in Eq. (24) denotes the weight of the spatial information in 3DS-SSC. Furthermore, the window size h of the Gaussian kernel and the standard deviation σ in Eq. (22) also contribute to the incorporation of spatial information to the subspace clustering model. In this section, the window size h is defined as a function of the standard deviation σ as follows:

$$h = 2\lceil 2\sigma \rceil + 1. \tag{32}$$

Different experiments were conducted varying α and σ parameters to analyze the performance of the proposed method with each hyperspectral image. The change in the overall accuracy (OA) of the 3DS-SSC algorithm, corresponding to α values ranging from 600 to 1000 and σ values ranging from 0.5 to 8, is shown in Fig. 19. It is observed that the clustering precision can significantly change when using different values of α and σ, which suggests that spatial infor-



(a) (b) (c)

Figure 19. Analysis of the parameters σ and α in terms of Overall Accuracy (OA) for (a) Indian pines, (b) Salinas, and (c) University of Pavia datasets.

mation plays a very important role in the clustering process. In order to make a fair comparison, multiple simulations with the other methods were performed to select the best parameters in their configurations.

2.3.2. Visual and Quantitative Results. This section presents the visual and quantitative results obtained for the three hyperspectral images described in Section 2.3.1. The clustering was a challenging task because the spectral signatures of the land-cover classes in the selected ROIs are very similar and some of the spectral curves are mixed, as shown in Fig. 18 (c), (f), and (i). Table 2 summarizes the selected parameters for each SSC-based clustering method and hyperspectral image. For the other methods (FCM_S1, RMMF, and 3D-CAE), the same parameter settings reported by the authors in the corresponding literature were used. Please note that, in the next subsections, the quantitative evaluations shown in Tables 3, 4, and 5 present the best results in bold font and the second-best are underlined.

2.3.2.1. Indian Pines ROI. The clustering performance of each method with the Indian Pines ROI is visually and quantitative reported in Fig. 20, and Table 3, respectively. The re-

Table 2

Selected Parameters for Each Testing Hyperspectral Images.

	Parameters		
SC-Based Method	**Indian Pines**	**Salinas**	**University of Pavia**
SSC	$\lambda = 7.76 \times 10^{-7}$	$\lambda = 1.17 \times 10^{-6}$	$\lambda = 3.90 \times 10^{-6}$
S-SSC	$\lambda = 7.76 \times 10^{-7}, \alpha = 2.1 \times 10^{5}$	$\lambda = 1.17 \times 10^{-6}, \alpha = 1.12 \times 10^{6}$	$\lambda = 3.90 \times 10^{-6}, \alpha = 8.8 \times 10^{5}$
TV-KSSC-SMP	$\lambda = 5.4, nv = 3, nf = 150$	$\lambda = 7.4, nv = 6, nf = 50$	$\lambda = 3.6, nv = 8, nf = 15$
ESC-FFS	$\lambda = 10, k = 700, t = 10$	$\lambda = 20, k = 700, t = 10$	$\lambda = 10, k = 1000, t = 20$
SR-SSC	$\tau = 200, nGraph = 5, Ns = 10$	$\tau = 100, nGraph = 8, Ns = 10$	$\tau = 700, nGraph = 15, Ns = 10$
3DS-SSC	$\lambda = 7.76 \times 10^{-7}, \alpha = 1.18 \times 10^{5}, \sigma = 3$	$\lambda = 1.17 \times 10^{-6}, \alpha = 8.4 \times 10^{5}, \sigma = 1.5$	$\lambda = 3.90 \times 10^{-6}, \alpha = 6.6 \times 10^{5}, \sigma = 6$

sults show that the two methods that perform better in this HSI are the TV-KSSC-SMP and the proposed method (3DS-SSC). Specifically, the 3DS-SSC achieved an OA of 80.41%, and the TV-KSSC-SMP obtained 76.72%. As observed in the quantitative evaluations, the most challenging classes to correctly classify were the Soybeans-no-till and Soybeans-min-till. However, the 3DS-SSC method achieves the best Producer's Accuracy with the Soybeans-no-till and the best User's Accuracy with Soybeans-min-till. The visual results show that the TV-KSSC-SMP assigns more pixels to the wrong classes; hence, it is more penalized by the used accuracy metrics. On the other hand, it is observed that the proposed method can take advantage of the spatial contextual information better, thus providing a smoother result in comparison with the SSC and S-SSC methods.

2.3.2.2. Salinas ROI. Figure 21 depicts the obtained land cover maps on the Salinas ROI. The quantitative evaluations corresponding to the PA, UA, AA, OA, Kappa, and NMI with all the clustering methods are reported in Table 4. As observed, the RMMF algorithm achieves the best performance with an OA of 98.20%. The second-best result was acquired by the proposed method (3DS-SSC) with an OA of 88.26%. In general, the most difficult land-cover class to classify was the *Corn-senesced* according to the Producer's accuracy and the *Lettuce-4wk* according to the

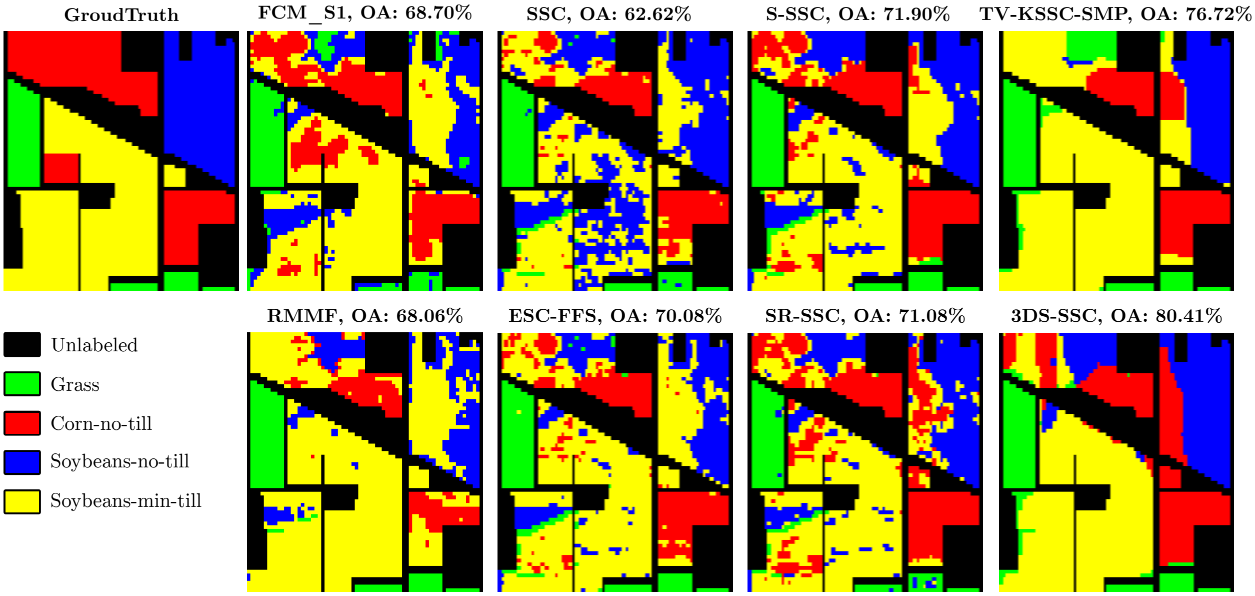


Figure 20. Land cover maps on Indian Pines ROI. The proposed method (3DS-SSC) is compared with the methods FCM_S1, SSC, S-SSC, TV-KSSC-SMP, RMMF, ESC-FFS, and SR-SSC.

Table 3

Clustering Performance of the Different Algorithms for the Indian Pines ROI.

	Metric	FCM_S1	RMMF	SSC	S-SSC	TV-KSSC-SMP	ESC-FFS	SR-SSC	3DS-SSC
Producer's Accuracy	Corn-no-till	**63.28**	35.02	48.96	63.18	45.37	51.08	62.09	59.90
	Grass	93.29	**100.00**	98.60	**100.00**	**100.00**	99.92	94.69	**100.00**
	Soybean-no-till	46.99	45.22	70.63	62.30	68.44	49.00	63.11	**77.05**
	Soybeans-min-till	77.20	94.17	59.23	75.78	**96.54**	84.12	75.44	91.32
User's Accuracy	Corn-no-till	80.30	**100.00**	97.43	83.99	84.29	93.98	68.87	73.24
	Grass	80.87	**97.02**	89.14	87.32	67.80	86.98	91.13	83.64
	Soybean-no-till	53.42	64.39	41.36	53.84	**97.85**	55.15	58.19	73.73
	Soybeans-min-till	66.16	59.47	61.57	71.79	71.58	63.26	74.28	**86.62**
	AA	70.19	68.60	69.35	75.31	77.59	71.03	73.83	**82.07**
	OA	68.70	68.06	62.62	71.90	76.72	69.20	71.08	**80.41**
	Kappa	0.55	0.52	0.48	0.60	0.66	0.55	0.59	**0.72**
	NMI	0.37	0.45	0.39	0.42	0.53	0.42	0.38	**0.57**

User's Accuracy. Although the 3DS-SSC method did not obtain the best performance on these land-cover classes, the obtained results were not the worst reported in Table 4. Although TV-

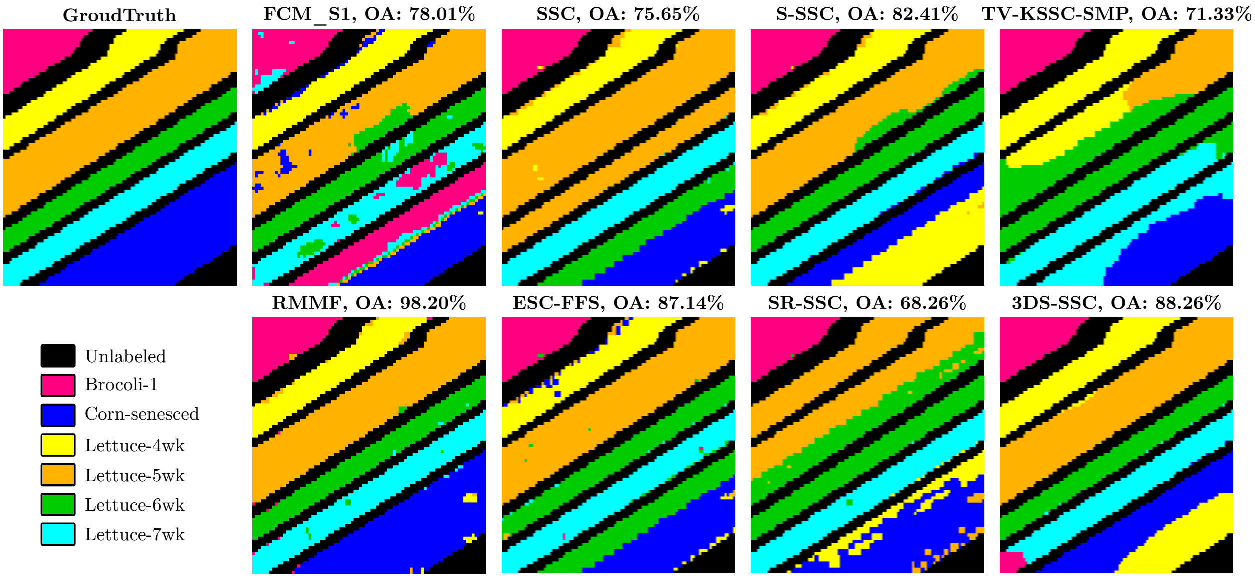


Figure 21. Land cover maps on Salinas ROI. The proposed method (3DS-SSC) is compared with the methods FCM_S1, RMMF, SSC, S-SSC, TV-KSSC-SMP, ESC-FFS, and SR-SSC.

Table 4

Clustering Performance of the Different Algorithms for the Salinas ROI.

Metric		**FCM_S1**	**RMMF**	**SSC**	**S-SSC**	**TV-KSSC-SMP**	**ESC-FFS**	**SR-SSC**	**3DS-SSC**
Producer's Accuracy	Brocoli-1	89.00	98.72	99.49	99.74	**100.00**	99.51	99.74	**100.00**
	Corn-senesced	47.15	**97.31**	55.62	42.01	62.82	55.09	71.28	58.31
	Lettuce-4wk	91.23	95.45	93.51	95.29	**100.00**	90.99	0.00	**100.00**
	Lettuce-5wk	88.36	99.53	99.53	91.46	36.79	**99.80**	56.29	99.06
	Lettuce-6wk	94.62	98.92	0.00	99.85	**100.00**	29.48	99.85	99.08
	Lettuce-7wk	78.47	98.44	**99.87**	97.41	90.27	98.20	98.31	92.09
User's Accuracy	Brocoli-1	35.44	97.23	**100.00**	**100.00**	**100.00**	99.62	**100.00**	87.28
	Corn-senesced	85.63	**100.00**	**100.00**	96.37	**100.00**	95.90	**100.00**	99.46
	Lettuce-4wk	**99.65**	96.39	95.21	44.57	62.22	94.97	0.00	53.24
	Lettuce-5wk	95.91	97.69	68.23	97.63	**100.00**	76.39	55.10	**100.00**
	Lettuce-6wk	74.73	97.87	0.00	83.63	50.35	16.17	49.77	**100.00**
	Lettuce-7wk	81.32	98.57	99.23	**99.87**	59.69	98.64	**99.87**	99.16
AA		81.47	**98.06**	74.67	87.63	81.65	78.85	70.91	91.42
OA		78.01	**98.20**	75.65	82.41	71.33	78.76	68.26	88.26
Kappa		0.73	**0.98**	0.69	0.79	0.66	0.73	0.61	0.86
NMI		0.68	**0.94**	0.83	0.83	0.70	0.80	0.75	0.87

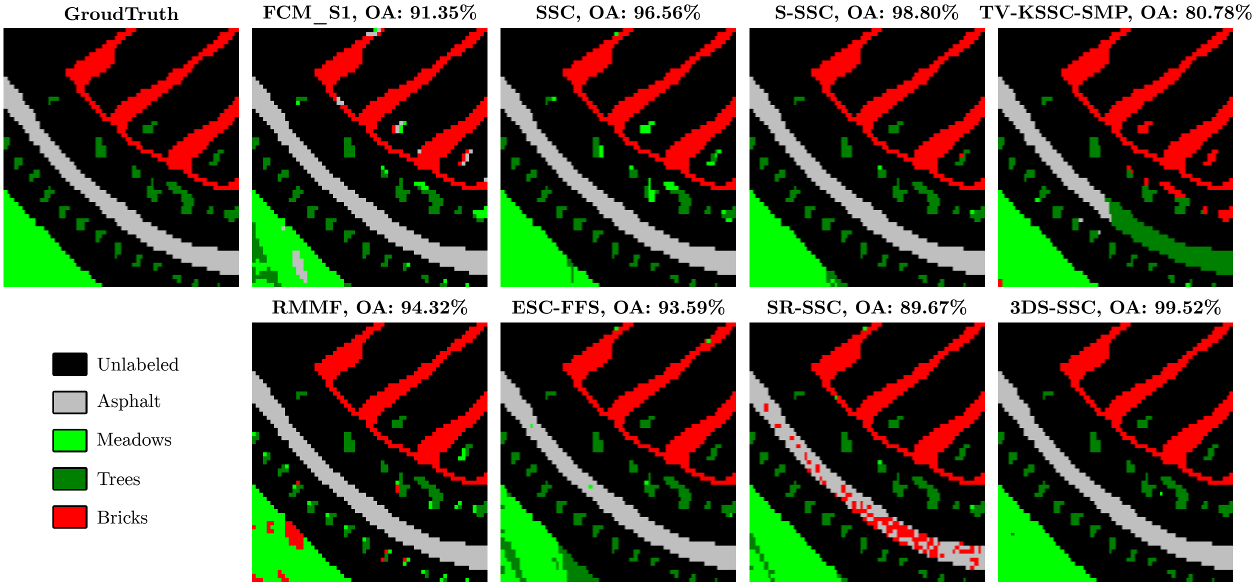


Figure 22. Land cover maps on University of Pavia ROI. The proposed method (3DS-SSC) is compared with the methods FCM_S1, RMMF, SSC, S-SSC, TV-KSSC-SMP, ESC-FFS, and SR-SSC.

KSSC-SMP achieves the best performance in three different land-cover classes, its performance on the other classes is worst compared to 3DS-SSC. Conversely, although the RMMF algorithm only obtains the best performance on one land cover class (Corn-senesced), it achieves high and similar performance on the other classes, hence getting the best overall accuracy. The visual results show that the SR-SSC assigns more pixels to the wrong classes; therefore, it is more penalized by the used accuracy metrics. On the other hand, it is observed that the visual results provided by the proposed method are smooth, which suggests that 3DS-SSC takes advantage of the spatial contextual information in comparison with the other methods.

2.3.2.3. University of Pavia ROI. Finally, Figure 22 presents the obtained land cover maps on the University of Pavia ROI, and the quantitative evaluations corresponding to the PA, UA, AA, OA, Kappa, and NMI with all the clustering methods are reported in Table 5. As observed,

Table 5

Clustering Performance of the Different Algorithms for the University of Pavia ROI.

Metric		**FCM_S1**	**RMMF**	**SSC**	**S-SSC**	**TV-KSSC-SMP**	**ESC-FFS**	**SR-SSC**	**3DS-SSC**
Producer's Accuracy	Asphalt	**100.00**	**100.00**	**100.00**	**100.00**	50.26	99.02	72.49	**100.00**
	Meadows	80.37	87.85	97.82	96.26	99.38	67.73	93.46	**99.82**
	Trees	77.71	79.62	77.71	98.09	68.15	99.11	98.73	**99.25**
	Bricks	97.46	**100.00**	99.75	**100.00**	**100.00**	98.98	99.49	**100.00**
User's Accuracy	Asphalt	91.53	**100.00**	**100.00**	**100.00**	98.96	**100.00**	**100.00**	**100.00**
	Meadows	91.81	91.56	89.71	99.36	**99.38**	96.69	98.68	99.12
	Trees	73.49	**100.00**	94.57	92.77	36.27	62.72	88.07	99.93
	Bricks	98.46	89.73	**100.00**	99.75	89.12	**100.00**	78.99	**100.00**
AA		88.88	91.87	93.82	98.59	79.45	91.21	91.04	**99.76**
OA		91.35	94.32	96.56	98.80	80.78	90.98	89.67	**99.52**
Kappa		0.88	0.92	0.95	0.97	0.74	0.88	0.86	**0.99**
NMI		0.78	0.86	0.92	0.96	0.77	0.87	0.80	**0.99**

the proposed method (3DS-SSC) obtains an OA of 99.52%, which corresponds to the highest classification score of the land-cover classes within the hyperspectral scene. The S-SSC method achieved the second-best result with an OA of 98.80%. It is important to notice that this image was the easiest to cluster since its spectral signatures are not critically mixed, as in the other hyperspectral scenes, see Fig. 18 (i). From the results, it is observed that the TV-KSSC-SMP achieves the worst performance in this particular image. Although it also provides smooth results, TV-KSSC-SMP assigns more pixels from the class *Tree* to the other classes; hence it is penalized by the used accuracy metrics. On the other hand, it can be observed that the proposed method better takes advantage of the spatial contextual information, providing a smoother result in comparison with RMMF, SSC, and S-SSC, which assign some pixels to the wrong classes.

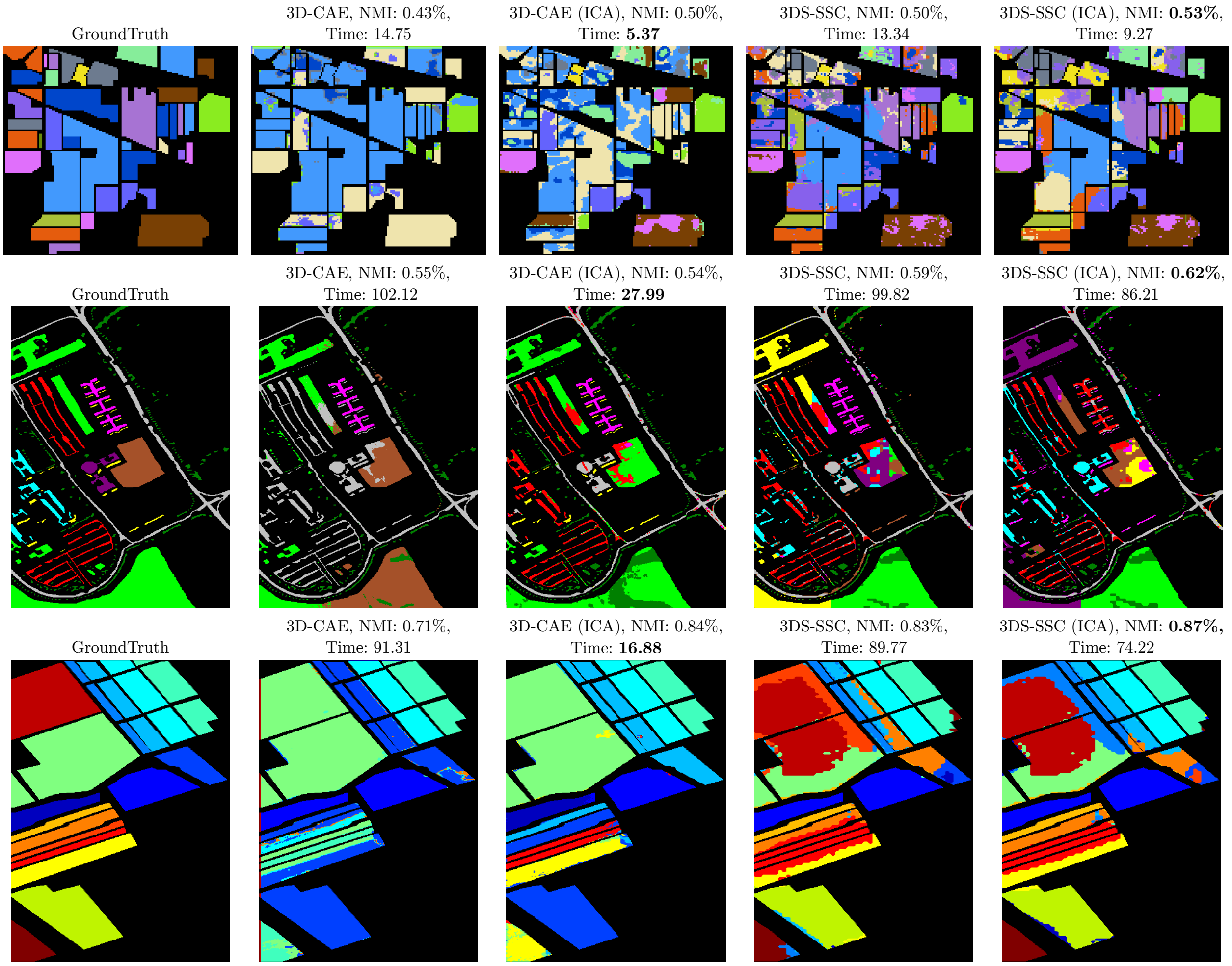


Figure 23. Land cover maps on the Indian Pines, University of Pavia and Salinas full images. The proposed method (3DS-SSC) is compared with the 3D-CAE. The best results for NMI and Time are shown in bold font.

2.3.2.4. Full Hyperspectral Images Comparison. For the sake of completeness, the proposed method presented in this chapter is compared with a recent deep learning-based approach (3D-CAE) (Nalepa et al., 2020) that relies on 3D convolutional autoencoders to perform the clustering. The proposed 3DS-SSC method was executed on the full hyperspectral images described in Section 2.3.1, and the obtained results are presented in Fig. 23. The presented land cover maps of the 3D-CAE method, as well as the given execution time (in minutes) and NMI score, were obtained from its original manuscript[2]. The figure presents the results for the full image after applying dimensionality reduction via independent component analysis (ICA) and without any preprocessing. Besides, the shown execution time only reflects the segmentation time and does not consider the preprocessing time. As observed, the proposed method (3DS-SSC) achieves the best performance over the test images.

Moreover, the execution time of the proposed method 3DS-SSC is similar to 3D-CAE only when no applying dimensionality reduction. However, if ICA is applied to the HSI, the 3D-CAE obtains the clustering results faster than 3DS-SSC. Such behavior is expected since the theoretical complexity analysis of the SSC-based method is $O(DM^3N^3)$; hence the computational burden is dominated by the number of pixels (MN), which remains equal after ICA preprocessing (Elhamifar and Vidal, 2013).

2.4. Conclusions

This chapter proposed a subspace clustering method for hyperspectral image land-cover segmentation. Precisely, the proposed approach consists of using a 3D Gaussian filter to enhance the sparse coefficients matrix, and then the final segmentation is obtained using spectral clustering. The proposed method significantly improves when compared with other subspace clustering methods used for hyperspectral image land cover segmentation. Specifically, an OA of 80.41%, 88.26%, and

---

[2] The authors provide the land-cover maps in the project repository at `https://gitlab.com/jnalepa/3d-cae`. The original land-cover maps are retrieved for comparison purposes but changed the colormap to compare it with the 3DS-SSC's results.

99.52% was achieved when performing the proposed 3DS-SSC method on a selected region of interest of the Indian Pines, Salinas, and University of Pavia hyperspectral images.

3. Efficient Subspace Clustering of Hyperspectral Images Using Similarity-constrained Sampling

Part of this chapter has been adapted from the journal paper (Lopez et al., 2021b) (published) and the conference paper (Lopez et al., 2021a) (published).

Table 6
Notation used in Chapter 3.

Notation	**Definition**
ρ	Number of points belonging to a segment.
p	Number of in-sample. data points in a segment.
k	Number of subspaces to be estimated.
M	Number of rows in the HSI.
N	Number of columns in the HSI.
L	Number of spectral bands of the HSI.
n	Number of data points (spatial pixels in the image).
N_v	Number of segments.
S_i	i-th subspace in the data.
$[D]_i$	Data points belonging to the subspace S_i.
$\mathbf{H} \in \mathbb{R}^{M\times N\times L}$	Hyperspectral image.
$\mathbf{S_p} \in \mathbb{R}^{M\times N}$	HSI represented with super pixels.
$\mathbf{B} \in [0,1]^{M\times N}$	Binary image resulting from edge detection.
$\mathbf{D} \in \mathbb{R}^{n\times L}$	Dataset.
$\mathbf{X} \in \mathbb{R}^{p\times L}$	Points in-sample data.
$\mathbf{Y} \in \mathbb{R}^{(\rho-p)\times L}$	Points out-sample data.
$\Gamma \in \mathbb{R}^{p\times p}$	Sparse representation of the points in-sample.
$\mathbf{A} \in \mathbb{R}^{p\times p}$	Affinity matrix obtained from Γ.
$\mathbf{L} \in \mathbb{R}^{p\times p}$	Laplacian matrix.
$\mathbf{V} \in \mathbb{R}^{p\times k}$	Matrix of eigenvalues.
$\mathbf{k} = [k_1, k_2, \dots, k_{N_v}]$	k_i is the number of clusters present in each segment.

3.1. Introduction

In the previous chapter, the spatial-contextual information of the HSI was incorporated as a regularization constraint in the SSC optimization model. However, the main limitation of the proposed 3DS-SSC method presented in Chapter 2 is the overall computational burden. For instance, given a HSI with M rows, N columns, and L spectral bands, 3DS-SSC needs to compute the $n \times n$ sparse coefficient matrix, where $n = NM$ spectral pixels, whose computational complexity is $O(Ln^3)$ (Hinojosa et al., 2021c). Recently, different works have proposed more efficient approaches to address the scalability issue of SSC (Peng et al., 2015; You et al., 2016a,b; Gauraha, 2018; You et al., 2018; Abdolali et al., 2019) . For instance, the authors in (Peng et al., 2015) proposed to address the subspace clustering problem by randomly dividing the dataset into two subsets to alleviate the SSC computational burden. Although such methods achieve a good performance when clustering large datasets, they do not fully exploit the information and spectral-spatial dependence of HSIs.

Chapter contribution. Considering that an efficient and careful selection of data subsets could speed up the unsupervised learning process, and the spatial-contextual information of the HSI could increase the clustering accuracy, this chapter presents an efficient subspace clustering algorithm for HSI using similarity-constrained sampling. Specifically, the proposed approach first cluster the spectral pixels into high spatially correlated blocks using edge and superpixel information; then, it separates the data points within each segment into two subsets. The proposed method employs sparse subspace clustering on the first subset to learn the underlying subspace structure. Finally, the spectral pixels belonging to the second subset are projected on the learned structure, and their cluster membership is computed considering the smallest projection error. The obtained results show that an efficient and careful selection of data subsets speeds up the unsupervised learning process while achieving high clustering performance in terms of overall accuracy. Table 6 summarizes the notation used throughout the chapter, where matrices and vectors are highlighted in bold, and their corresponding dimensions are shown.

3.2. Fast Similarity-constrained Sampling Sparse Subspace Clustering (F4SC)

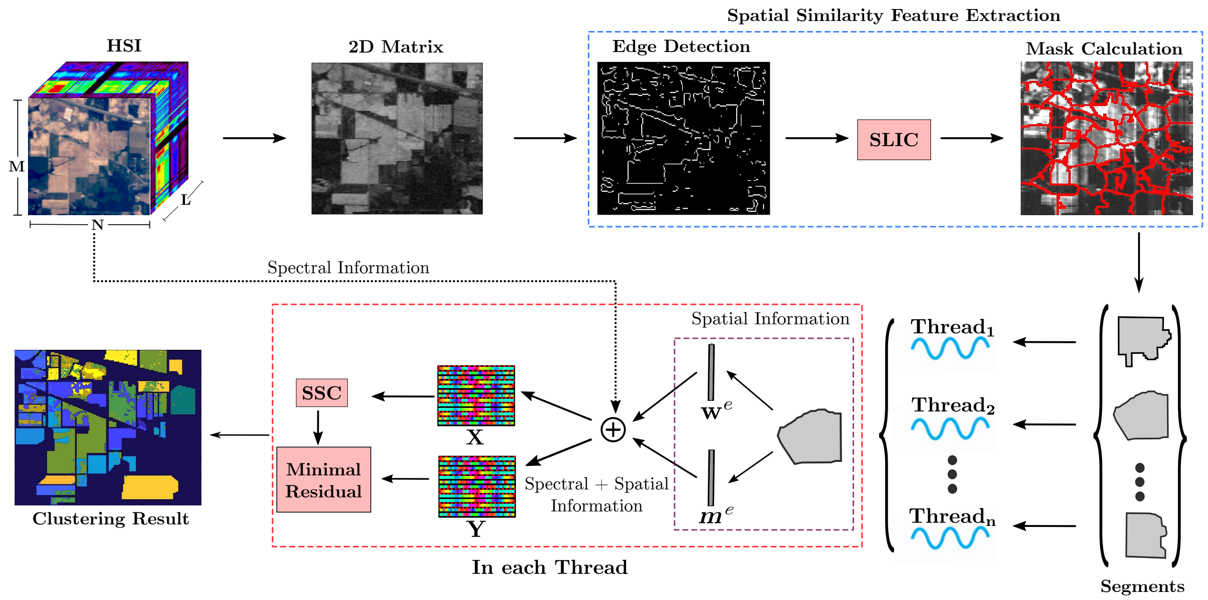


Figure 24. Fast similarity sampling sparse subspace clustering (F4SC). Initially, a 2D matrix is obtained from the HSI by averaging all the spectral bands. Then, the algorithm extracts spatial similarities by fusing the edge information and superpixel. Using the spectral information of the HSI and the extracted spatial information, the proposed F4SC algorithm performs similarity-constrained sampling to obtain segments and obtain two subsets from each of them $\mathbf{X}, \mathbf{Y}$. Finally, F4SC performs simultaneously subspace clustering on $\mathbf{X}$ and use the learned subspaces structure to clustering $\mathbf{Y}$ in the threads configured in the proposed method execution environment.

Most subspace clustering methods have problems assigning a point to a subspace when they are close to the intersection of two or more subspaces (Elhamifar and Vidal, 2013). This chapter presents a sampling restricted by spatial similarity to separate data that are possibly close to the intersection of multiple subspaces. To do this, the HSI is initially divided into high spatially correlated segments containing points that possibly belong to the intersection of subspaces. After, efficient sampling will be applied to each segment to select in-sample and out-sample points. Therefore, SSC will be applied individually at points in-sample points of each segment. The assignment of the out-sample points to the clusters is carried out using the minimum residual obtained by projecting each point on already clustered in-sample points. An overview of the proposed method, named Fast similarity-constrained sampling sparse subspace clustering (F4SC), is shown in Fig 24.

3.2.1. Edge detection and superpixels. Generally, SSC is used to cluster HSI by only taking advantage of the spectral information. However, these algorithms do not take into account the spatial relationship among pixels of the HSI when deciding the class to which each point belongs. In contrast, here edge detection and superpixels are used to extract the spatial information present in the HSI since, with the combination of these two methods, the generated segments are better adapted to the structure of the possible subspaces present in the HSI, as shown in Fig 25. Note that in subfigure 25-(c), the structure of the segments tends to be regular geometric figures, and a possible structure of the possible clusters present in the scene is not obtained. In contrast, in subfigure 25-(b), it is achieved.

Edge detection in grayscale images has been thoroughly studied and it is a well established (Trucco and Verri, 1998) since its great use in different areas, such as computer vision. However, for multichannel images, such as HSI, this topic is less developed since defining borders for these

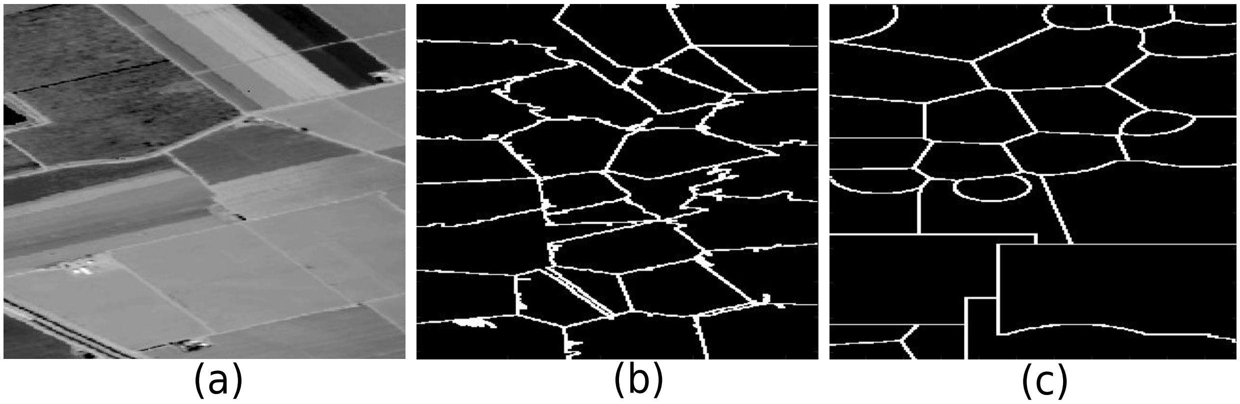


Figure 25. (a) 2D matrix obtained from the HSI. (b) Result of edge detection by averaging all HSI spectral bands and then applying superpixels. (c) Averaging all HSI spectral bands and calculate superpixels.

images is a challenge due to the high dimensionality of the data. Therefore, all the spectral bands of a hyperspectral image are averaged to obtain a two-dimensional image as:

$$\mathbf{H}'(i,j) = \frac{1}{L}\sum_{k=1}^{L}\mathbf{H}(i,j,k) \quad \forall(i,j) \in (M,N). \tag{33}$$

It is worth mentioning that there are other methods to obtain a 2D matrix from the HSI, such as Principal Component Analysis (Abdi and Williams, 2010) (PCA) or random projection (Bingham and Mannila, 2001). Once the 2D image is obtained, the Sobel edge detection operator Ω (Vincent et al., 2009) is applied to obtain the edge image as:

$$\mathbf{B} = \Omega(\mathbf{H}') \in \{\mathbf{0},\mathbf{1}\}^{M\times N}. \tag{34}$$

Considering the edge information, the SLIC (Achanta et al., 2012) algorithm is applied to obtain a segmentation of neighboring pixels. The first row of Fig 24 depicts the procedure

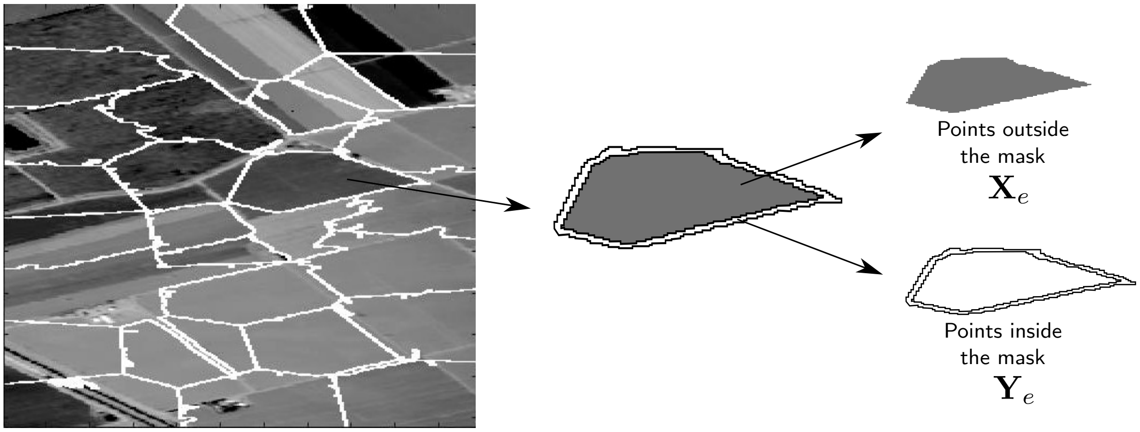


Figure 26. Point selection inside and outside the mask.

mentioned above, which allows discriminating the points present at the intersection of subspaces. It is important to note that, in this step, the algorithm only extracts the spatial information from the HSI using edge detection and a superpixel algorithm, as shown in Fig 26, where the extraction of the spatial information of the points inside and outside the mask of a particular segment is shown. Specifically, the mask is formed for each segment by taking pixels around its edge. In this sense, the mask allows defining a region limit of each segment. Notice that I refer to “points outside the mask” to those points within the segment edge and to “points inside the mask” to those belonging to the segment but outside its edge. In the following section, the proposed algorithm uses the information given by these masks to perform a similarity-constrained sampling of the spectral signatures before subspace clustering.

3.2.2. Similarity-constrained Sampling. The superpixel map is represented as an image $\mathbf{S} \in \mathbb{R}^{M \times N}$, which is calculated on the image **B**. Using the information contained in **S**, the proposed method performs a similarity-constrained sampling within each segment considering that the pixels of the superpixels maintain a high spatial correlation, i.e., the correlation is greater between the points near the centroid and decreases at the boundaries of the segment (superpixel). Specifically, the proposed method first calculates a mask for each segment $e = 1, \cdots, N_v$ by taking pixels around the border to improve the separation, where N_v is the total amount of segments. Denote $\mathbf{w}^e \in \mathbb{R}^{u_e}$ as the vector of size u_e containing the indices of all pixels belonging to the superpixel e but lie outside the mask, see Fig. 26. Similarly, denote $\mathbf{m}^e \in \mathbb{R}^{n_e}$ as the vector of size n_e containing the indices of all pixels belonging to the superpixel e that lie within the mask. Then, the algorithm performs the sampling on the HSI data for each segment e as follows:

$$\begin{aligned} \mathbf{X}_e &= \left\{ \mathbf{H}_p : p \in \mathbf{w}^e \right\} \in \mathbb{R}^{L \times u_e}, \\ \mathbf{Y}_e &= \left\{ \mathbf{H}_p : p \in \mathbf{m}^e \right\} \in \mathbb{R}^{L \times n_e} \end{aligned} \tag{35}$$

where, $\mathbf{H}_p \in \mathbb{R}^L$ denotes the spectral pixel of $\mathbf{H}$ given by the spatial position index p. Through the chapter, I will refer to as in-sample points ($\mathbf{X}_e$) to those spectral pixels from e that lie outside the mask, and out-of-sample points ($\mathbf{Y}_e$) to those spectral pixels from e that only lie within the border mask. For ease of the notation, the subscript e was removed in the $\mathbf{X}_e$ and $\mathbf{Y}_e$ matrices; nevertheless, note that all the following operations are performed at each segment e.

3.2.3. Subspace Clustering Approach. Once defining the masks, the proposed method learns the underlying subspace structure given by the in-sample points $\mathbf{X}$. In particular, SSC solves the following optimization problem:

$$\begin{aligned}&\min_{\Gamma,\mathbf{Z}} \|\Gamma\|_1 + \frac{\lambda_Z}{2}\|\mathbf{Z}\|_F^2\\ &\text{s.t. } \mathbf{X} = \mathbf{X}\Gamma + \mathbf{Z},\ \operatorname{diag}(\Gamma) = 0,\end{aligned} \tag{36}$$

where $\Gamma \in \mathbb{R}^{L\times u_e}$ is the sparse representation of the data, $\mathbf{X} \in \mathbb{R}^{L\times u_e}$ are the spectral signatures of the HSI, $\mathbf{Z}$ denotes reconstruction errors for limited rendering ability, and λ_Z corresponds to a regularization parameter. Once the sparse representation matrix Γ is obtained, SSC constructs the similarity graph by solving $\mathbf{A} = |\Gamma|^T + |\Gamma|$ and applies spectral clustering (Ng et al., 2002). The optimization problem in Eq. 36 is solved by using the alternating direction method of multipliers (ADMM) described in (Annergren et al., 2012).

3.2.4. Minimal Residual. After learning the underlying subspace structure of $\mathbf{X}$, F4SC uses such knowledge to cluster the out-of-sample points $\mathbf{Y}$. Specifically, each point belonging to $\mathbf{Y}$ is projected on each of the computed subspaces from $\mathbf{X}$. Then, the sparse representation of $\mathbf{Y}$ over $\mathbf{X}$ is computed and assigned each $\mathbf{y}_i$ to the nearest subspace on sparse representation-based classification (SRC) (Wright et al., 2009). Then, for each out-of-sample data point $\mathbf{y}_i$, the following optimization problem is solved:

$$\min_{\mathbf{c}_i} \|\mathbf{y}_i - \mathbf{X}\mathbf{c}_i\|_2^2 + \gamma\|\mathbf{c}_i\|_2^2, \tag{37}$$

where $\gamma > 0$ is the tolerance error, and $\mathbf{y}_i$ denotes an out-of-sample point. Once the optimal $\mathbf{c}_i$ is obtained, $\mathbf{y}_i$ is assigned to the nearest subspace, by solving:

Algorithm 3 F4SC algorithm with input parameters → $\mathbf{H}$, N_v, $\mathbf{k}$

```
Result: Assigning clusters of $\mathbf{H}$
1  $\mathbf{H}^* \leftarrow \sum_{\mathbf{i=1}}^{\mathbf{L}} \mathbf{H}[:,:,\mathbf{i}]$
2  $\mathbf{B} \leftarrow \Omega(\mathbf{H}^*)$
3  $\mathbf{S_p} \leftarrow algorithmSlic(\mathbf{B}, \mathbf{N_v})$
4  $\mathbf{Bw} \leftarrow calculateMask(\mathbf{S_p})$
5  $[\mathbf{IDX}, \mathbf{IDY}] \leftarrow getIndex(\mathbf{Bw})$
6  for $i \leftarrow 1$ to $N_v$ do
7      $[\mathbf{idxx}, \mathbf{idxy}] \leftarrow \mathbf{IDX}[i]$
8      $[\mathbf{idyx}, \mathbf{idyy}] \leftarrow \mathbf{IDY}[i]$
9      $\mathbf{X} \leftarrow \mathbf{H}[\mathbf{idxx}, \mathbf{idxy}, :]$
10     $\mathbf{Y} \leftarrow \mathbf{H}[\mathbf{idyx}, \mathbf{idyy}, :]$
11     $k_i \leftarrow \mathbf{k}[i]$
12     if $k_i = 0$ then
13         $k_i \leftarrow estimateNumberClusters(X)$
14     $\mathbf{G}^{(i)} \leftarrow \begin{cases} \mathbf{R} \leftarrow SSC(\mathbf{X}, k_i) \\ minimalResidual(\mathbf{X}, \mathbf{Y}, \mathbf{R}) \end{cases}$
```

$$f\left(\mathbf{y}_i\right) = \underset{j}{\operatorname{argmin}} \left\{ \left\| \mathbf{y}_i - \mathbf{X}\delta_j\left(\mathbf{c}_i\right) \right\|_2 \right\}, \tag{38}$$

where $f(\mathbf{y}_i)$ denotes the assignment of $\mathbf{y}_i$, and the nonzero entries of $\delta_j(c_i)$ are the elements in c_i associated with the j-th subspace (Wright et al., 2009; Peng et al., 2015) . The term δ is used to avoid over-fitting.

In addition to Fig 24, all the steps of F4SC are also shown in the Algorithm 14. Lines 1-5 present the spatial information extraction by doing the similarity-restricted sampling. In lines 7-10, the algorithm obtains the spectral signatures for each segment's in-sample and out-sample points. If the number of clusters present in a particular segment is unknown, these are estimated in line 13. Finally, in line 14, the algorithm allocates clusters for each segment.

Table 7
Complexity of the proposed algorithm, where t_1 y t_2 are the iterations required to solve the optimization problem shown in Eq. (36).

Term	**Description**
$O(N_v t_1 L \psi^3)$	Construction of the similarity graph
$O(N_v t_2 \psi k_i^2)$	Spectral clustering (Ng et al., 2002)
$O(N_v n \psi^2)$	Projection of clustered points within the sample
$O(N_t)$	SLIC
$O(N_t)$	Edge detection

3.2.5. Complexity analysis. The computational complexity of the proposed method Algorithm 14, is $O(N_v t_1 L \psi^3) + O(N_v t_2 \psi k_i^2) + O(N_v n \psi^2) + O(2n)$, where ψ is the number of in-sample points that belongs to a segment. In order to compute the computational complexity, assume that the value of ψ is the same for all segments. However, such value could change in practice and depends on the size of the segments. Table 7 details the complexity of each step of the proposed algorithm. Using the Big O notation properties, the computational complexity of F4SC can be summarized as $O(N_v t_1 L \psi^3)$. Since F4SC uses SSC for assigning clusters at points in-sample of each segment, the computational complexity remains cubic, but as $\psi \ll n$, F4SC is faster than other subspace clustering-based approaches; further, it achieves high clustering accuracy because of the proposed similarity-constrained sampling technique.

3.3. Experimental Results

This section presents a comparison of the F4SC method with the following subspace clustering-based algorithms: SSSC (Peng et al., 2015) , SR-SSC (Abdolali et al., 2019) , SSC-OMP (You et al., 2016b) , EnSC (You et al., 2016a) , ESC-FFS (You et al., 2018) and 3DS-SSC (Hinojosa et al., 2021c). All the experiments were conducted using MATLAB vR2020a with 16 workers in parallel pool configured by default to execute the proposed method on a computer with Intel Xeon Processor E5-2697 v3 and 180 GB RAM.

3.3.1. Hyperspectral Datasets and Used Metrics.

The proposed method was tested on three real hyperspectral datasets (Graña et al., 2020) with different imaging environments. These images were acquired by the Airborne Visible / Infrared Imaging Spectrometer (**AVIRIS**) and the Reflecting Optics System Imaging Spectrometer (**ROSIS**). The first HSI is the Indian Pines with a spatial resolution of 145×145 and 203 spectral bands. The second image is Salinas with a spatial resolution of 512×217 and 204 spectral bands. The third image is Pavia University with a spatial resolution of 610×340 and 103 spectral bands. In the experiments, the number of k_i classes within a segment was set as manual input for the proposed subspace clustering Algorithm 14. The parameters shown in Eqs. 36 and 37, were fixed as $\lambda_Z = 2 \cdot 10^{-4}$ and $\gamma = 10^{-6}$, respectively. To evaluate the clustering precision, I used: Overall Accuracy (OA), Average Accuracy (AA), Cohen's kappa coefficient (kappa), and normalized mutual information (NMI) (Lillesand et al., 2015) .

3.3.2. Ablation Study.

First, four ablation experiments were conducted and investigated different options in the proposed workflow shown in Fig. 24:

- **Experiment 1:** only used superpixels for similarity-restricted sampling.
- **Experiment 2:** used another superpixel segmentation method, known as *Linear Spectral Clustering* (LSC) (Li and Chen, 2015), instead of SLIC to segment the 2D image.
- **Experiment 3:** used the Canny (Canny, 1986) algorithm for edge detection instead of Sobel.
- **Experiment 4:** used a different method to get a 2D matrix from the HSI before applying

Table 8
Results of ablation experiments using the Indian Pines HSI.

Experiment	N_v	**OA**	**AA**	**Kappa**	**NMI**	**Time [s]**
1	750	77.58	75.90	0.748	0.632	2.68
2	1050	81.78	79.25	0.795	0.683	6.96
3	400	75.07	68.73	0.719	0.591	2.83
4	1000	79.11	74.40	0.764	0.636	7.22
F4SC	1250	85.08	81.55	0.832	0.711	4.69

edge detection. In particular, PCA was used for these experiments to obtain such a 2D matrix.

In the Table 8, are shown the results obtained by performing different experiments are shown, which led us to establish the workflow shown in Fig 24 and Algorithm 14. From experiment 1, it is possible to conclude that using the superpixels technique only is not enough to extract the spatial information from the HSI and the edge information helps to better discriminate among pixels within the intersection of the clusters. In the second experiment, F4SC also achieves good performance by using LSC, suggesting that any superpixels segmentation algorithm can be adopted in the proposed framework. However, as shown in the last row of Table 8, the F4SC algorithm achieves the best clustering accuracy and computational time when using the SLIC algorithm to obtain the superpixels. In the third experiment, the Canny algorithm allows achieving a better clustering of the subspaces than the previous experiment using the default parameters of the method. However, the Canny algorithm requires more parameters to tune in comparison with the Sobel operator. Finally, in experiment 4, it was decided to apply PCA to obtain a 2D matrix from the HSI for edge detection using the Sobel operator. In this experiment, a better clustering

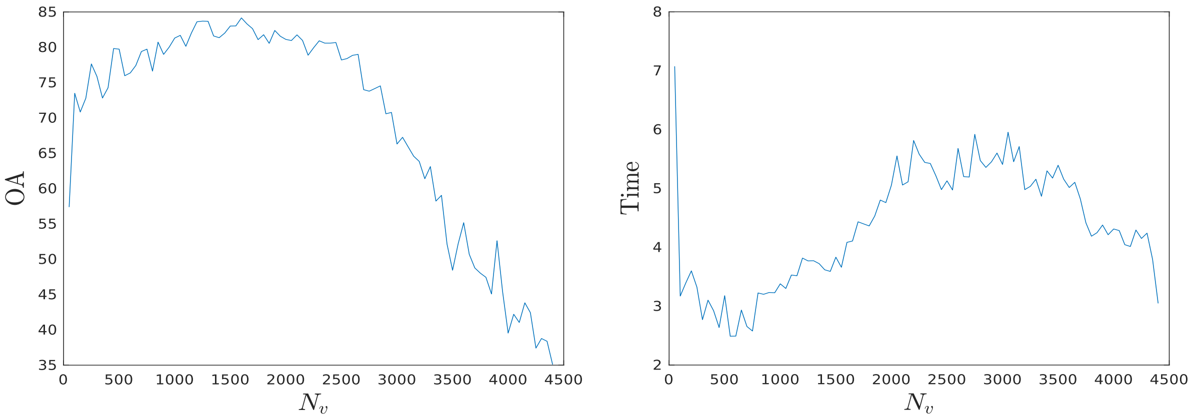


Figure 27. Evolution of overall accuracy (OA) (left) and execution time (right) as a function of the number of segments (N_v) in the Indian Pines hyperspectral image.

performance was obtained compared with previous experiments; however, the PCA algorithm is slow, and its computational complexity grows exponentially with the dimensions of the HSI. It is worth mentioning that the number of segments shown in Table 8 leads to the best OA within each experiment configuration.

Through the development of all these four experiments, it is possible to conclude that the best way to approach the cluster of subspaces in hyperspectral images using the spatial information for the proposed approach is the steps described in Fig. 24 and Algorithm 14. The next step is to analyze the impact on precision and computational time in the number of segments obtained on image **B**. Then, in Figs. 27, 28 and 29 illustrated the evolution of the OA and execution time in terms of the number of segments (N_v) for each dataset previously mentioned.

As shown in the previous plots, Figs. 27-29 regardless of the HSI, when the number of segments is small, the computation time is high, and the precision is low because most of the spectral signatures are taken as points within the sample are clustered by SSC. When the number

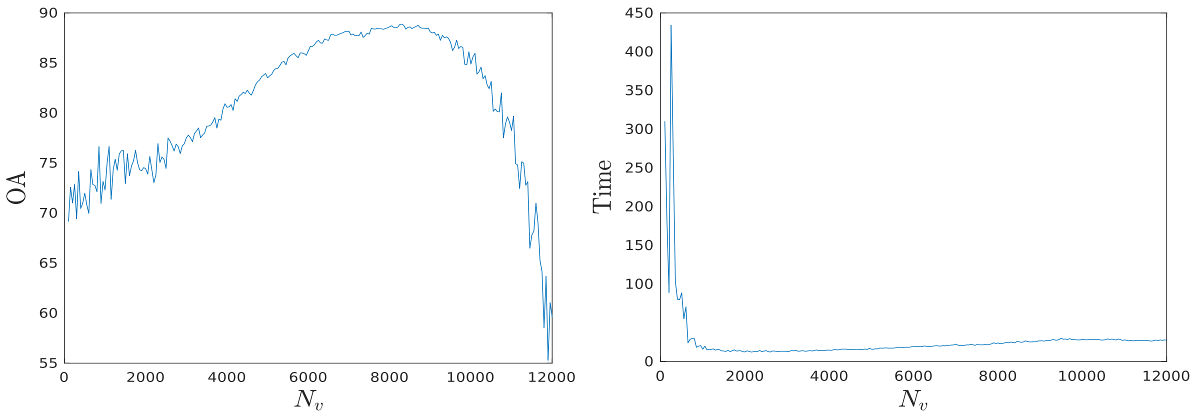


Figure 28. Evolution of overall accuracy (OA) (left) and execution time (right) as a function of the number of segments (N_v) in the Salinas hyperspectral image.

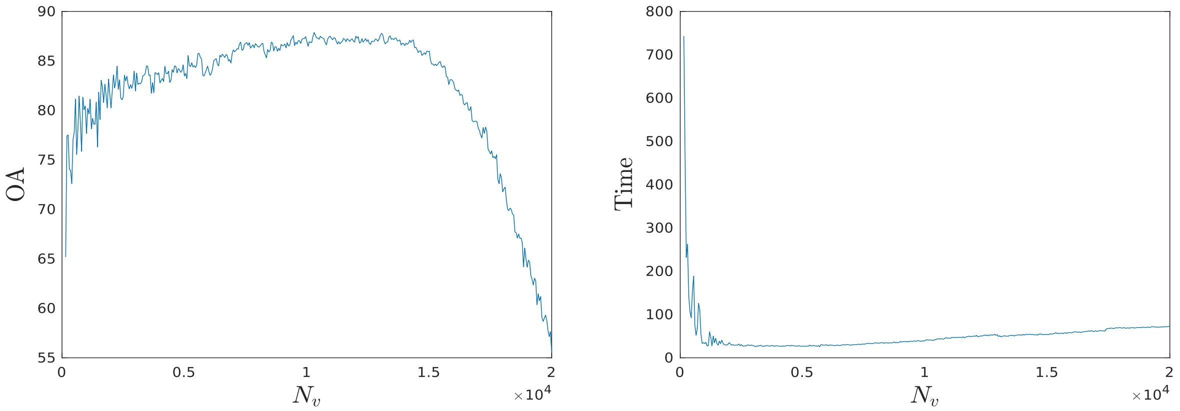


Figure 29. Evolution of overall accuracy (OA) (left) and execution time (right) as a function of the number of segments (N_v) in the Pavia University hyperspectral image.

of segments is large, the precision decreases, and the computation time is optimal; is because not enough points are taken within the sample to learn the structure of the clusters present, and therefore, the clustering that is carried out using minimal residual is imprecise. It is important to remark that the highest precision is obtained when the number of points belonging per segment $\rho \in [15, 35]$.

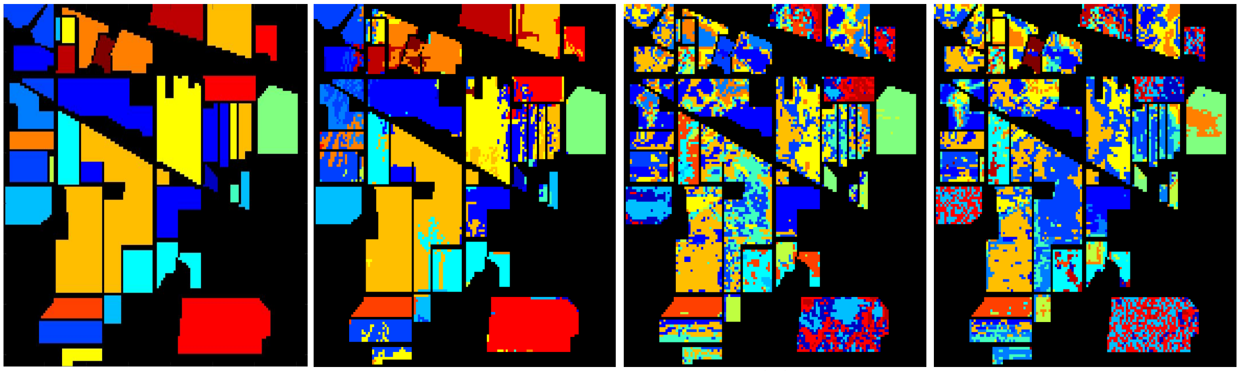

(a) (b) (c) (d)

Figure 30. Visual maps for Indian Pines. (a) Ground truth. (b) F4SC. (c) ESC-FFS. (d) SR-SSC.

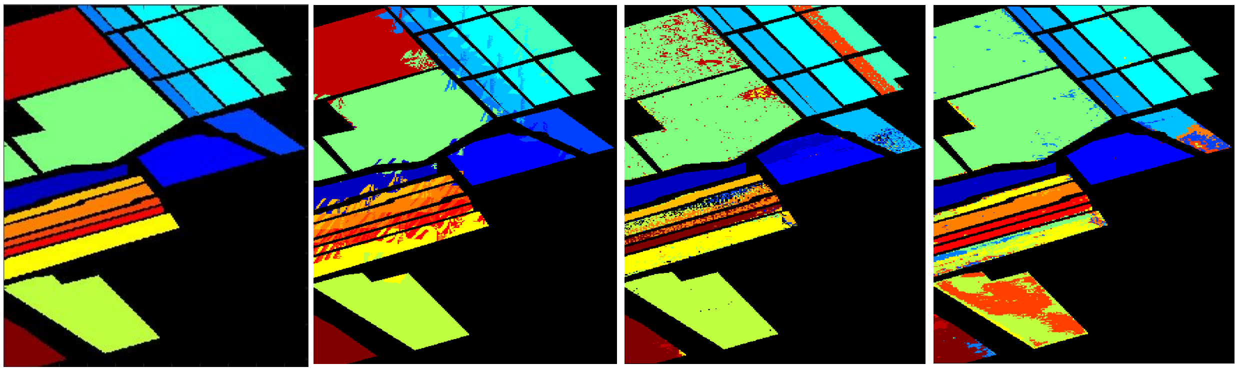

(a) (b) (c) (d)

Figure 31. Visual maps for Salinas. (a) Ground truth. (b) F4SC. (c) SSSC. (d) SR-SSC.

3.3.3. Visual Maps and Quantitative results. Below, Tables 9, 10 and 11 present quantitative results of the proposed approach (F4SC) and the baseline algorithms for the Indian pines, Salinas, and Pavia University images, respectively. For the state-of-the-art algorithms, the results shown in the tables are the average of running this method 10 times. Additionally, the optimal value for each metric is shown in bold, and the second-best result is underlined. Following each table, the visual maps of the three methods that obtained the best overall accuracy are shown, ordered from highest to lowest, in Figs. 30, 31 and 32.

Table 9
Quantitative results for Indian Pines, where $N_v = 1250$ *in the proposed method.*

Class	SR-SSC	EnSC	SSC-OMP	ESC-FFS	3DS-SSC	SSSC	F4SC
Alfalfa	00.00	00.00	00.00	00.00	00.00	00.00	**89.10**
Corn-notill	53.20	59.77	16.70	44.53	26.82	62.20	**86.41**
Corn-mintill	14.00	19.68	04.00	37.71	36.87	19.00	**87.73**
Corn	07.00	0.880	00.00	08.86	13.08	04.40	**83.97**
Grass-pasture	22.30	34.10	**94.10**	52.58	57.14	35.30	83.02
Grass-trees	82.90	81.39	51.60	28.21	57.67	71.20	**88.36**
Grass-pasture-mowed	00.00	4.73	00.20	00.00	00.00	00.00	**60.71**
Hay-windrowed	85.10	85.67	27.60	85.98	**97.49**	86.80	83.26
Oats	04.80	04.70	12.10	50.00	00.00	03.70	**65.00**
Soybean-notill	32.40	37.87	37.50	30.04	49.07	29.60	**81.89**
Soybean-mintill	53.90	56.36	35.70	34.82	50.55	52.90	**85.54**
Soybean-clean	15.30	15.68	32.10	15.68	25.13	13.20	**80.27**
Wheat	62.30	32.95	47.40	**94.63**	92.20	29.90	87.81
Woods	72.80	87.90	50.00	42.76	56.92	84.10	**89.86**
Buildings-Grass-Trees-Drives	28.40	29.43	4.300	15.54	49.22	12.60	**82.84**
Stone-Steel-Towers	73.60	13.77	00.00	10.57	**86.02**	00.00	73.12
Overall Accuracy (OA)	36.84	33.21	20.32	38.01	48.09	35.44	**85.08**
Average Accuracy (AA)	39.06	40.64	11.54	34.51	43.64	32.36	**81.55**
Kappa	0.303	0.272	0.095	0.315	0.432	0.291	**0.832**
NMI	0.423	0.425	0.132	0.422	0.505	0.4155	**0.711**
Time [seg]	16.85	1640	17.50	61.20	753.4	56.77	**4.69**

3.3.4. Comparison with deep learning methods. In this section, a comparison between F4SC and some works in which deep learning is used for subspace clustering in hyperspectral images is shown. Below, tables 12 and 13 compare F4SC results with the following works: fused 3-D deep neural networks (F3D) (Sellami et al., 2020) , fast 3D-CNN (FST) (Ahmad, 2020) , graph convolutional neural network (GCN) (Qin et al., 2018) and Multi-scale 3D convolutional neural network (MSC) (He et al., 2017).

Table 10

Quantitative results for Salinas, where $N_v = 7000$ in the proposed method.

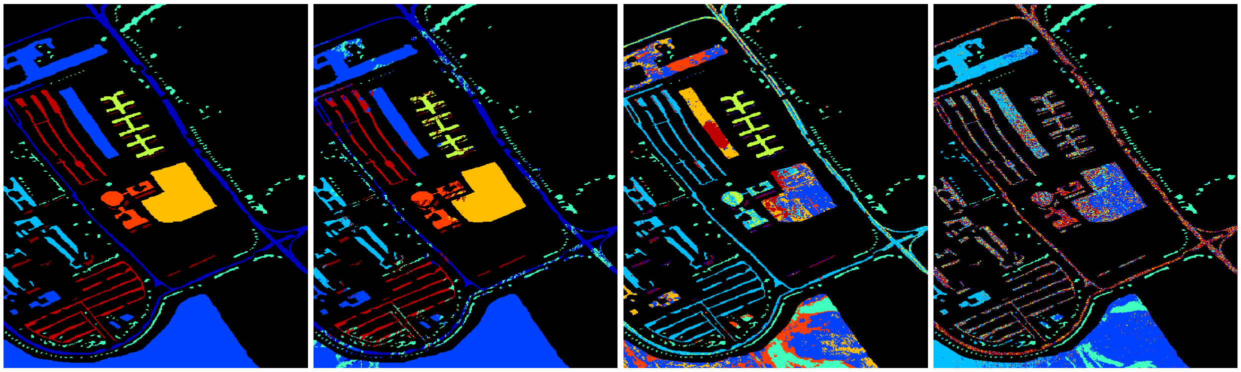
Class	SR-SSC	EnSC	SSCOMP	ESC-FFS	3DS-SSC	SSSC	F4SC
Brocoli_green_weeds_1	00.00	00.00	04.50	00.00	00.00	00.00	**89.50**
Brocoli_green_weeds_2	51.90	64.30	29.30	99.40	**100.0**	62.20	89.16
Fallow	93.30	**100.0**	55.00	00.00	26.19	19.00	86.54
Fallow_rough_plow	96.10	03.70	99.00	**99.13**	85.06	04.40	89.45
Fallow_smooth	74.30	61.40	85.60	**94.39**	91.23	35.30	88.76
Stubble	99.80	99.90	62.20	90.47	**100.0**	71.20	91.89
Celery	64.00	98.00	18.40	**99.55**	98.66	00.00	90.89
Grapes untrained	63.90	72.90	64.50	62.07	42.17	86.80	**88.68**
Soil vinyard develop	82.70	00.60	71.80	91.01	**91.93**	03.70	89.54
Corn senesced green weeds	63.60	64.20	18.50	56.92	61.14	29.60	**88.77**
Lettuce romaine 4wk	4.90	00.00	01.80	00.20	**98.51**	52.90	84.83
Lettuce romaine 5wk	91.40	39.10	22.50	**95.58**	94.17	13.20	85.47
Lettuce romaine 6wk	00.00	84.00	00.00	**91.37**	00.00	29.90	85.26
Lettuce romaine 7wk	38.70	93.40	11.10	92.24	**100.0**	84.10	85.61
Vinyard untrained	50.00	51.00	45.30	59.75	74.72	12.60	**86.87**
Vinyard vertical trellis	98.50	96.6	01.20	10.84	**100.0**	00.00	86.81
OA	56.98	56.18	33.36	69.24	72.20	70.76	**88.55**
AA	54.79	56.23	29.45	65.20	72.80	60.05	**87.99**
Kappa	0.523	0.519	0.279	0.656	0.695	0.660	**0.873**
NMI	0.71	0.744	0.527	0.739	**0.812**	0.809	0.752
Time [seg]	102.83	5987	248.9	894.3	4453	2339.1	**25.71**

As can be noticed, in the previous tables, the methods based on deep learning obtain the highest precision when assigning the clusters belonging to an HSI. However, even with their high precision, these methods have a high inference time compared to F4SC. The proposed method can be seen as a trade-off between precision and computation time, being efficient in both aspects.

Table 11

Quantitative results for Pavia University, where $N_v = 10300$ *in the proposed method.*

Class	SR-SSC	EnSC	SSC-OMP	ESC-FFS	3DS-SSC	SSSC	F4SC
Asphalt	62.10	70.00	52.90	64.75	76.90	61.07	**87.27**
Meadows	83.90	76.00	**87.70**	24.01	48.01	77.09	86.71
Gravel	00.00	00.00	27.20	00.28	74.71	01.21	**90.14**
Trees	48.40	66.00	69.60	71.63	**91.53**	58.11	87.24
Painted metal sheets	80.90	58.00	00.00	**99.47**	00.00	00.00	91.08
Bare Soil	37.70	23.00	20.30	30.62	45.54	20.23	**84.73**
Bitumen	00.00	08.00	03.10	91.42	00.00	13.23	**91.50**
Self-Blocking Bricks	37.80	05.00	27.30	88.51	87.32	13.72	**90.47**
Shadows	78.30	00.00	00.90	**99.04**	00.00	03.75	92.08
OA	43.46	32.98	39.36	44.03	55.98	42.70	**87.86**
AA	51.13	37.18	30.11	66.30	47.12	34.30	**88.18**
Kappa	0.346	0.224	0.284	0.370	0.479	0.245	**0.843**
NMI	0.519	0.477	0.413	0.515	0.623	0.368	**0.708**
Time [seg]	264.1	5185	730.3	4169	5172	5272	**40.59**



(a) (b) (c) (d)

Figure 32. Visual maps for Pavia University. (a) Ground truth. (b) F4SC. (c) SR-SSC. (d) SSSC.

3.4. Conclusions

This chapter presented an efficient subspace clustering algorithm for hyperspectral images, which can handle large-scale datasets and take advantage of spectral image's neighboring spatial infor-

Table 12
Results of deep learning approaches in the HSI Indian Pines.

	F3D	FST	GCN	MSC	F4SC
OA	**96.98**	95.83	95.42	94.28	85.05
AA	**96.85**	95.68	95.28	94.08	81.55
Kappa	**0.968**	0.957	0.954	0.941	0.832
Time [s]	274.3	321.7	322.4	389.5	**4.69**

Table 13
Results of deep learning approaches in the HSI Salinas.

	F3D	FST	GCN	MSC	F4SC
OA	**97.65**	96.75	96.84	96.42	88.55
AA	**97.52**	96.89	96.63	96.24	87.99
Kappa	**0.974**	0.969	0.967	0.963	0.873
Time [s]	289.6	364.1	210.9	394.4	**25.71**

mation to boost the clustering accuracy. The proposed method considers the spatial information present in the scene by detecting edges and superpixels and dividing the hyperspectral image into multiple segments. Next, F4SC performs the sparse subspace clustering algorithm and minimal residual within each segment. Finally, the method merges the individual results to obtain the full HSI clustering. The experimental results with real datasets show that F4SC is remarkably efficient compared with other evaluated algorithms. Specifically, the proposed algorithm (F4SC) achieved an increase in the overall accuracy of 36%, 32%, and 16% for the Indian Pines, Pavia University, and Salinas, respectively, compared to the state-of-the-art algorithms. Furthermore, F4SC can be up to 50 times faster than some methods based on deep learning as F4SC does not need long training schedules.

4. A Fast and Accurate Similarity-constrained Subspace Clustering Algorithm for Hyperspectral Image

Part of this chapter has been adapted from the journal paper (Hinojosa et al., 2021b) (published).

Table 14
Notation used in this Chapter 4.

Notation	**Definition**
N_r	Number of rows in the HSI.
N_c	Number of columns in the HSI.
$N = N_r Nc$	Total amount of spectral pixels (data points).
k	number of classes.
L	Number of spectral bands of the HSI.
$D < L$	New Dimension after dimensionality reduction.
$\mathbf{X} \in \mathbb{R}^{L \times N}$	Points in-sample data.
$\mathbf{x}_j \in \mathbb{R}^N$	j-th spectral pixel.
$\tilde{\mathbf{m}} \in \mathbb{R}^N$	Region maps obtained by superpixels algorithm.
E	Number of segments (superpixels).
$\mathbf{p}_e \in \mathbb{R}^{N_e}$	Indices of the N_e most similar spectral pixels from subset e.
$M_e = \lfloor \rho N_e \rfloor$	Amount of pixels from each subset, where $\rho \in (0,1)$.
$\hat{\mathbf{X}} \in \mathbb{R}^{D \times EM}$	Most representative spectral pixels from all subsets.
$\mathbf{C} \in \mathbb{R}^{M \times N}$	Sparse coefficient matrix, where $M = \lfloor \rho N \rfloor$.
$\mathbf{X}_e \in \mathbb{R}^{D \times M} \subseteq \mathbf{X}$	Most representative pixels from subset e.
$\mathbf{I}_{K_s} \in \mathbb{R}^{K_s \times K_s}$	Blur kernel matrix with window size K_s.
$\mathscr{G}(\cdot)$	Convolution operation.

4.1. Introduction

This chapter combines the ideas from Chapter 2 and Chapter 3 to develop a fast and accurate similarity-constrained subspace clustering algorithm to enhance both the clustering accuracy and execution time in unsupervised HSI classification. As shown in previous chapters, two main problems are well-known in HSI subspace clustering: (1) The overall computational complexity of

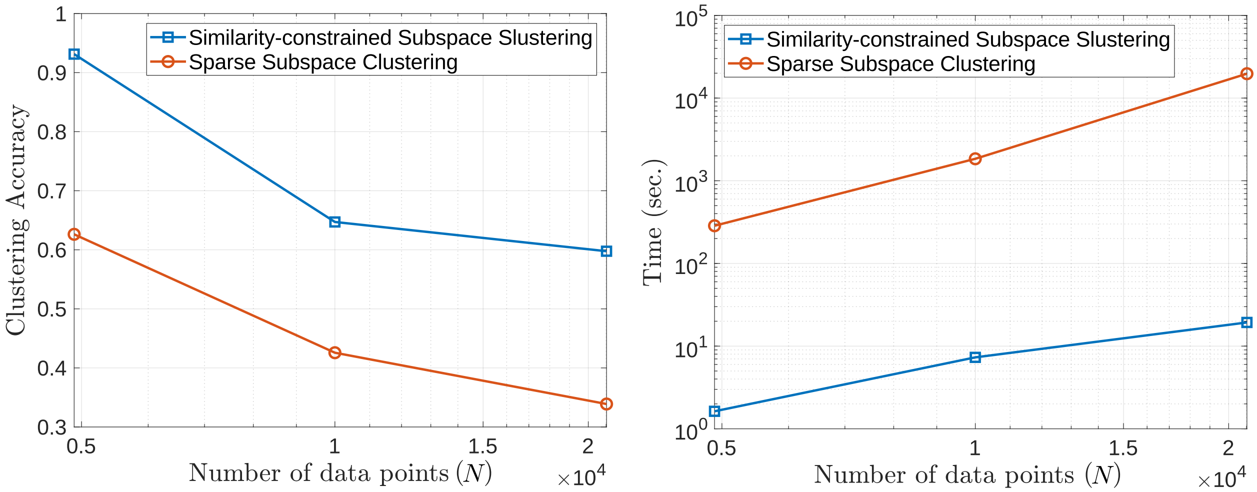


Figure 33. Clustering accuracy (left) and running time (right) of the SSC algorithm compared with the proposed method for HSI clustering in this Chapter. In this example, the number of data points (N) is varied by performing the two subspace clustering algorithms on the full image and two regions of interest (ROIs) of the Indian Pines dataset (See Section 3.3). The first ROI has $N = 4900$ pixels and $k = 4$ classes; the second has $N = 10000$ pixels and $k = 12$ classes; the whole Indian Pines image has $N = 21025$ pixels and $k = 17$ classes.

SSC prohibits its usage on large HSI datasets. For instance, given a HSI with N_r rows, N_c columns, and L spectral bands, SSC needs to compute the $N \times N$ sparse coefficient matrix corresponding to $N = N_r N_c$ spectral pixels, whose computational complexity is $O(LN^3)$. Moreover, after building the affinity matrix, spectral clustering performs an eigenvalue decomposition over the $N \times N$ graph Laplacian matrix, which also has cubic time complexity, or quadratic using approximation algorithms (Chen et al., 2018) (see Fig. 33 right). (2) Under the context of HSI, the SSC model only captures the relationship of pixels by analyzing the spectral features without considering the spatial information. Indeed, the sparse coefficient matrix obtained by SSC should be piecewise smooth since spectral pixels belonging to the same land cover material are arranged in a common region; hence there is a spatial relationship between the representation coefficient vector of one pixel and its neighbors.

Chapter contribution. Since SSC methods require to express each data point as a linear combination of all other points in the dataset, the main observation is that one can take advantage of the spatial-spectral properties of HSI to select the most representative spectral pixels over small regions and then constraint the remaining data points to be represented as a linear combination of them. This significantly reduces the clustering time and increases the accuracy. Specifically, here, an efficient algorithm is proposed for selecting the most representative pixels of each subset by minimizing the maximum representation cost of the data. After this procedure, the proposed algorithm concatenates the obtained most representative spectral pixels to form a matrix. Then, an optimization problem is solved using this matrix to get the coefficients that encodes information about similarities between each subset's most representative spectral pixels and the whole HSI. Finally, it is proposed to enhance the obtained coefficient matrix via 2D smoothing convolution before applying a fast spectral clustering algorithm that provides the final clustering map in a significantly less amount of time. As shown in Chapter 2, the motivation of using such convolutions is to enforce the piece-wise smoothness in the coefficient matrix. In contrast with Chapter 2, here, a 2D convolution is used to obtain faster results. In essence, the proposed method enforces the connectivity in the affinity matrix and then efficiently obtains spectral embedding without the need to compute the eigenvalue decomposition, which has a computational complexity of $O(N^3)$.

Increasing the number of data points and the classes enlarges the computation time and make clustering more challenging. The proposed method, shown with the blue line in Fig. 33, can be up to three orders of magnitude faster than SSC and outperforms it in terms of accuracy when

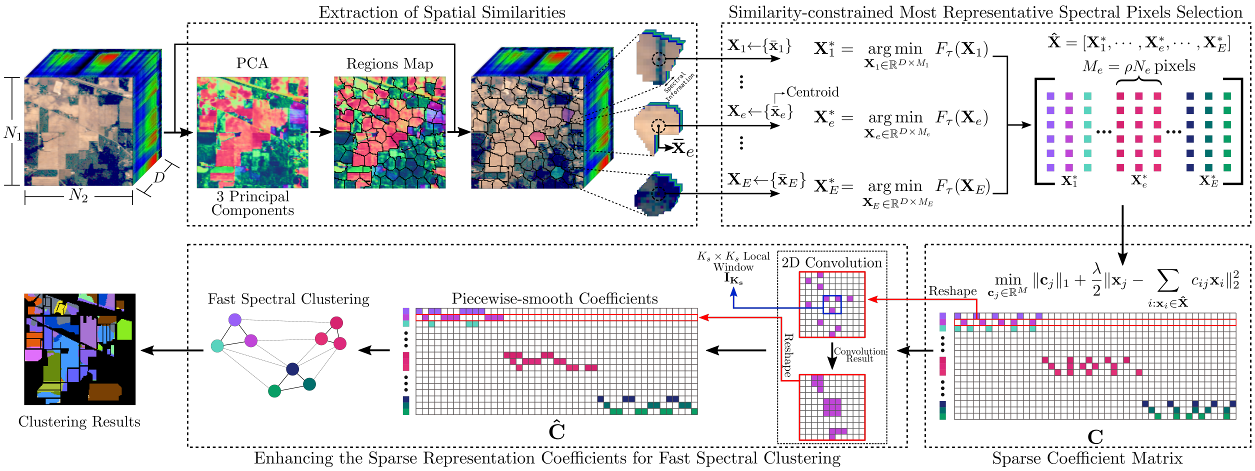


Figure 34. The proposed workflow is composed of four stages. In the first stage, spectral pixels with similar values are grouped into regions. For this step, PCA is applied to obtain the three principal components of the hyperspectral image; then, the SLIC algorithm is adopted to generate a guidance map and group similar spectral pixels into regions. In the second stage, the Algorithm 4 is used to select the most representative spectral pixels from each region or subset and then stacked them as columns in the matrix $\hat{\mathbf{X}}$. In the third stage, Eq. (42) is solved to obtain the $\{\mathbf{c}_j\}$ vectors. To further improve the connectivity of matrix $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_N]$, a 2D convolution is performed with a $K_s \times K_s$ kernel hence obtaining a piecewise-smooth coefficient matrix. The final data segmentation is obtained via fast spectral clustering. The computational complexity of the overall algorithm is $O(\rho^2 N^3)$, see Section 4.2.3.2.

clustering more than 2×10^4 spectral pixels. This Chapter evaluates and compares my proposed approach on three real remote sensing hyperspectral images with different imaging environments and spectral-spatial resolution.

4.2. Fast and Accurate Similarity-constrained Subspace Clustering (SC-SSC)

This section presents the proposed subspace clustering algorithm for unsupervised HSI classification that incorporates both properties: it is efficient and takes advantage of the spatial information of HSIs to boost the clustering accuracy. In general, the proposed algorithm exploits the self-representation property within subsets of neighboring similar pixels to select the most representative data points of the whole hyperspectral image. Then, it enhances the sparse representation and performs fast spectral clustering to obtain the segmentation result. The complete workflow of the proposed method is shown in Fig. 34.

4.2.1. Similarity-constrained Spectral Pixels Selection. As neighboring spatial pixels commonly belong to the same land cover material, the proposed method aims to select a small subset of pixels that best represent their neighborhood. In this regard, first, all the spectral pixels are grouped with a similar spectrum into small spatial regions. In general, this preprocessing step can be done in different ways. In this Chapter, the simple linear iterative clustering (SLIC) (Achanta et al., 2012) algorithm is adopted to generate the guidance regions map due to its efficiency. In particular, the proposed algorithm performs PCA to retrieve the three principal components of $\mathbf{X}$ and form the matrix $\mathbf{X}_{PCA} \in \mathbb{R}^{3\times N}$. Then, it use the SLIC algorithm (Achanta et al., 2012) to obtain a regions map $\tilde{\mathbf{m}} \in \mathbb{R}^{N}$ from $\mathbf{X}_{PCA}$, such that $\tilde{m}_j \in \{1, \cdots, E\}$, where E is the number of segments. For instance, if $\tilde{m}_j = e$ means that the spectral pixel $\mathbf{x}_j$ belongs to the segment e. Note that PCA is only performed to obtain $\tilde{\mathbf{m}}$ from $\mathbf{X}_{PCA}$ via SLIC; then, the proposed algorithm uses $\tilde{\mathbf{m}}$ to select the most representative spectral pixels $\mathbf{x}_j$ from $\mathbf{X}$ within each segment.

Let $\mathbf{p}_e \in \mathbb{R}^{N_e}$ be the vector containing the indices of the N_e most similar spectral pixels belonging to the subset e. We are interested in selecting the $M_e = \lfloor \rho N_e \rfloor$ most representative pixels from each subset, where $\rho \in (0,1)$. Taking advantage of the self-expressiveness property, the selection of the pixels within each neighborhood e is obtained by searching for a subset $\mathbf{X}_e^* \subseteq \mathbf{X}$

that minimizes

$$\mathbf{X}_e^* = \underset{\mathbf{X}_e \in \mathbb{R}^{D \times M}}{\arg\min}\, F_\tau(\mathbf{X}_e), \tag{39}$$

where F_τ is the *self-representation* cost function defined as

$$F_\tau(\mathbf{X}_e) := \sup_{\mathbf{x}_j \in \mathbf{X}\,:\, j \in \mathbf{p}_e} f_\tau(\mathbf{x}_j, \mathbf{X}_e). \tag{40}$$

The metric function $f_\tau(\mathbf{x}_j, \mathbf{X}_e)$ geometrically measures how well a data point $\mathbf{x}_j \in \mathbf{X} : j \in \mathbf{p}_e$ can be represented by the subset $\mathbf{X}_e$, and define it as

$$f_\tau(\mathbf{x}_j, \mathbf{X}_e) := \min_{\mathbf{c}_j \in \mathbb{R}^N} \|\mathbf{c}_j\|_1 + \frac{\tau}{2} \|\mathbf{x}_j - \sum_{i:\mathbf{x}_i \in \mathbf{X}_e} c_{ij}\mathbf{x}_i\|_2^2, \tag{41}$$

where $\tau \in (1, \infty)$ is a parameter. Note that Eq. (40) constrains Eq. 39 to search only for pixels $\mathbf{x}_j$ within the subset e, using the vector $\mathbf{p}_e$. To efficiently solve Eq. (39) for each subset e, then, the proposed algorithm described in Algorithm 4. Note that, instead of using a random initialization, the proposed algorithm selects the centroid spectral pixel $\bar{\mathbf{x}}_e$ as the initialization since it is the most similar point, in the Euclidean distance, to all other data points in e. The search space constraint, given by dividing the HSI into subsets and selecting the centroid spectral pixel, speeds up the acquisition of the representative spectral pixels.

4.2.2. Enhancing the sparse representation coefficients for fast spectral clustering. Once obtained the most representative spectral pixels from each subset, the matrix $\hat{\mathbf{X}}$ is built by stacking the results as columns, i.e., $\hat{\mathbf{X}} = [\mathbf{X}_1, \cdots, \mathbf{X}_E]$. Then, the sparse coefficient matrix $\mathbf{C}$ of

Algorithm 4 Similarity-constrained spectral pixels selection.

```
Input  : Data $\mathbf{X} \in \mathbb{R}^{D\times N}$, Indices vector $\mathbf{p}_e \in \mathbb{R}^{N_e}$, Parameters $0 < \rho < 1$, and $\tau > 1$.
Output: $\mathbf{X}_e \in \mathbb{R}^{D\times \lfloor \rho N_e \rfloor}$.
Function Data_Selection($\mathbf{X}, \mathbf{p}_e, \rho, \tau$)
    $\bar{\mathbf{x}}_e \leftarrow \mathrm{centroid}(\{\mathbf{x}_j \in \mathbf{X} : j \in \mathbf{p}_e\})$
    $\mathbf{X}_e^{(1)} \leftarrow \{\bar{\mathbf{x}}_e\}$ ▷ $(\mathbf{p}_e)_k$ gets the $k$ element of the vector $\mathbf{p}_e$.
    Compute $b_k = f_\tau(\mathbf{x}_j, \mathbf{X}_e^{(1)})$ for $k = 1, \cdots, N_e$, and $j = (\mathbf{p}_e)_k$.
    $M_e \leftarrow \lfloor \rho N_e \rfloor$
    for $i = 1, \cdots, M_e - 1$ do
        Let $o_1, \cdots, o_{N_e}$ be an ordering of $1, \cdots, N_e$ such that $b_{o_p} \geq b_{o_q}$ when $p < q$.
        Initialize $max\_cost = 0$.
        for $k = 1, \cdots, N_e$ do
            Set $b_{o_k} = f_\tau(\mathbf{x}_{o_k}, \mathbf{X}_e^{(i)})$.
            if $b_{o_k} > max\_cost$ then
                Set $max\_cost = b_{o_k}$, and $new\_index = o_k$.
            if $k = N_e$ or $max\_cost \geq b_{o_{k+1}}$ then
                break
        $\mathbf{X}_e^{(i+1)} = \mathbf{X}_e^{(i)} \cup \{\mathbf{x}_{new\_index}\}$
    return $\mathbf{X}_e$
```

size $M \times N$, with $M = \lfloor \rho N \rfloor$, can be obtained by solving the following optimization problem

$$\min_{\mathbf{c}_j \in \mathbb{R}^M} \|\mathbf{c}_j\|_1 + \frac{\tau}{2} \|\mathbf{x}_j - \sum_{i:\mathbf{x}_i \in \hat{\mathbf{X}}} c_{ij} \mathbf{x}_i\|_2^2, \quad \forall\, \mathbf{x}_j \in \mathbf{X}. \tag{42}$$

Note that $\mathbf{C}$ encodes information about the similarities between $\hat{\mathbf{X}}$ and $\mathbf{X}$. Besides, each row of $\mathbf{C}$ contains the representation coefficients distribution of the whole image with respect to a single representative pixel. Considering that the spectral pixels belonging to the same land cover material should be regionally distributed in the image, i.e., two spatially neighboring pixels in an HSI usually have a high probability of belonging to the same class. Then, according to the self-expressiveness property, their representation coefficients should also be very close concerning the same basis; hence, each row of $\mathbf{C}$ should be piecewise smooth. Therefore, to further improve the

Algorithm 5 The proposed algorithm for HSI clustering (SC-SSC).

```
Input  : The hyperspectral image in matrix form $\mathbf{X} \in \mathbb{R}^{D \times N}$, parameters $\tau > 1$, $0 < \rho < 1$, $E > 1$,
         $K_s > 1$.
Output: The segmentation of $\mathbf{X}$.
$\mathbf{X}_{PCA} \in \mathbb{R}^{3 \times N} \leftarrow PCA(\mathbf{X})$
$\tilde{\mathbf{M}} \leftarrow Extract\_Regions\_Map(\mathbf{X}_{PCA})$
$\tilde{\mathbf{m}} \leftarrow \mathrm{vec}(\tilde{\mathbf{M}})$
$\hat{\mathbf{X}}^{(e)} \leftarrow \emptyset$
for $e \leftarrow 1$ to $E - 1$ do
    $\mathbf{p}_e = \left\{ j : \tilde{\mathbf{m}}_j = e, \forall j \in \{1, \cdots, N\} \right\}$
    $\mathbf{X}_e \leftarrow Data\_Selection(\mathbf{X}, \mathbf{p}_e, \rho, \tau)$
    $\hat{\mathbf{X}}^{(e+1)} = \hat{\mathbf{X}}^{(e)} \cup \mathbf{X}_e$
Compute $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_N]$ by solving Eq. (42).
$\mathbf{I}_{K_s} = (1/K_s^2) \cdot \mathbf{1}$   ▷ $\mathbf{1}$:  $K_s \times K_s$ all-ones matrix.
$\hat{\mathbf{C}} = \mathscr{G}(\mathbf{C}, \mathbf{I}_{K_s})$ ▷ 2D Convolution Operation
$\tilde{\mathbf{C}} = [\hat{\mathbf{c}}_1 / \|\hat{\mathbf{c}}_1\|_2, \cdots, \hat{\mathbf{c}}_N / \|\hat{\mathbf{c}}_N\|_2]$
$\alpha = \sum_{j=1}^{N} \tilde{\mathbf{c}}_j$
$\mathbf{D} = \mathrm{diag}(\tilde{\mathbf{C}}^T \alpha)$
Run $k$-means clustering algorithm on the top $k$ right singular vectors of $\tilde{\mathbf{C}}\mathbf{D}^{-1/2}$ to obtain the
    segmentation of $\mathbf{X}$.
return The cluster assignments of $\mathbf{X}$
```

structure of matrix $\mathbf{C}$, a 2D smoothing convolution is applied on its coefficients. This is a post-processing procedure, and it is performed before spectral clustering. Given a blur kernel matrix $\mathbf{I}_{K_s}$ of size $K_s \times K_s$, let denote the 2D convolution process as $\hat{\mathbf{C}} = \mathscr{G}(\mathbf{C}, \mathbf{I}_{K_s})$. Specifically, as depicted in Fig. 34 within dashed blue line, the $\mathscr{G}$ operation is performed after reshaping each row of $\mathbf{C}$ to a window of size $N_r \times N_c$, which corresponds to the spatial dimensions of the HSI, and then conducting the convolution with $\mathbf{I}_{K_s}$. Finally, the convolution result is rearranged back as a row vector of the piecewise-smooth coefficient matrix $\hat{\mathbf{C}} = [\hat{\mathbf{c}}_1, \cdots, \hat{\mathbf{c}}_N] \in \mathbb{R}^{M \times N}$.

Finally, fast spectral clustering is used to efficiently obtain the spectral embedding of the input data. Specifically, let us consider the columns of $\tilde{\mathbf{C}} = [\tilde{\mathbf{c}}_1, \cdots, \tilde{\mathbf{c}}_N] \in \mathbb{R}^{M \times N}$, where $\tilde{\mathbf{c}}_j =$

$|\hat{\mathbf{c}}_j|/\|\hat{\mathbf{c}}_j\|_2$, and compute the i-th element of $\mathbf{D}$ as

$$(\mathbf{D})_i = \sum_{j=1}^{N} A_{ij} = \sum_{j=1}^{N} \tilde{\mathbf{c}}_i^T \tilde{\mathbf{c}}_j = \tilde{\mathbf{c}}_i^T \sum_{j=1}^{N} \tilde{\mathbf{c}}_j = \operatorname{diag}(\tilde{\mathbf{C}}^T \alpha)_i, \tag{43}$$

where $\alpha = \sum_{j=1}^{N} \tilde{\mathbf{c}}_j \in \mathbb{R}^M$. Next, the eigenvalue decomposition of $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is obtained by computing the singular value decomposition (Golub and Reinsch, 1971) of $\tilde{\mathbf{C}}\mathbf{D}^{-1/2} \in \mathbb{R}^{M \times N}$. Finally, the segmentation of the data can be obtained by running the k-means algorithm on the top k right singular vectors for $\tilde{\mathbf{C}}\mathbf{D}^{-1/2} = \mathbf{U}\Sigma\mathbf{P}^T$. As a result, the computational complexity of spectral clustering in the framework is linear with respect to the size of the data N. The proposed SC-SSC method is summarized in Algorithm 5.

4.2.3. Analysis of the Proposed Method. Now, let's analyze how the proposed method optimizes the sparsity (subspace-preserving property) and the connectivity in the representation coefficient matrix. Furthermore, the computational complexity of Algorithm 5.

4.2.3.1. Subspace-preserving Property and Connectivity. One of the main requirements for the success of subspace clustering methods is that the optimization process recovers a subspace-preserving solution. Specifically, the non-zero entries of the sparse representation vector $\mathbf{c}_j$ should be related only to the intra-subspace samples of $\mathbf{x}_j$. Indeed, as the following definition states, the representation coefficients among intra-subspace data points are always larger than those among inter-cluster points.

Definition 1 (**Intra-subspace projection dominance, IPD (Peng et al., 2016)**)**.** The IPD property of a coefficient matrix $\mathbf{C}$ indicates that for all $\mathbf{x}_u, \mathbf{x}_v \in \mathscr{S}$ and $\mathbf{x}_q \notin \mathscr{S}$, where $u, v, q \in \{1, \cdots, N\}$, and

$\mathscr{S}$ is a subspace of $\mathbf{X}$, we have $C_{uv} \geq C_{uq}$.

Since the proposed method selects the most representative spectral pixels for each subset e based on the self-representation property, it is expected that each subset is subspace-preserving, i.e., c_{ij} is nonzero only if $\mathbf{x}_i$ and $\mathbf{x}_j$, for $i, j \in \mathbf{p}_e$, belong to the same subspace $\mathscr{S}$. Furthermore, note that it is very probable that a subset e has more spectral pixels from the same class due to the spatial dependence in HSI; then, the resulting coefficients vector will have large values for those spectral pixels within e. Therefore, the strategy adopted in the proposed method will improve the structure of the vectors $\mathbf{c}_j$ obtained by Eq. (42) and will improve the probability that $\mathbf{c}_j$ satisfies the IPD.

Besides, using the 2D smoothing convolution procedure $\mathscr{G}(\mathbf{C}, \mathbf{I}_{K_s})$, the proposed method improves the connectivity of the data points by preserving the most significant values in the coefficient matrix $\mathbf{C}$ and reducing the small or noisy isolated values, based on the IPD property (Peng et al., 2016). Then, the resulting matrix $\hat{\mathbf{C}}$ will have localized neighborhoods in the sparse codes making the representation coefficients of spatially neighboring pixels very close as well, following the main assumption in section 4.2.2.

4.2.3.2. Computational Complexity Analysis. As shown in Fig. 34, the proposed method mainly involves four stages: the extraction of spatial similarities, the selection of similarity-constrained representative spectral pixels, the sparse coefficient matrix estimation by solving Eq. (42), and enhancing the representation coefficients for fast spectral clustering. Given a hyperspectral image in matrix form $\mathbf{X} \in \mathbb{R}^{D \times N}$ and E subsets $\mathbf{X}_e \subseteq \mathbf{X}$ of dimensions $D \times M_e$, with $M_e = \rho N_e$, the complexity of each stage is computed before establishing the total complexity of Algorithm 5.

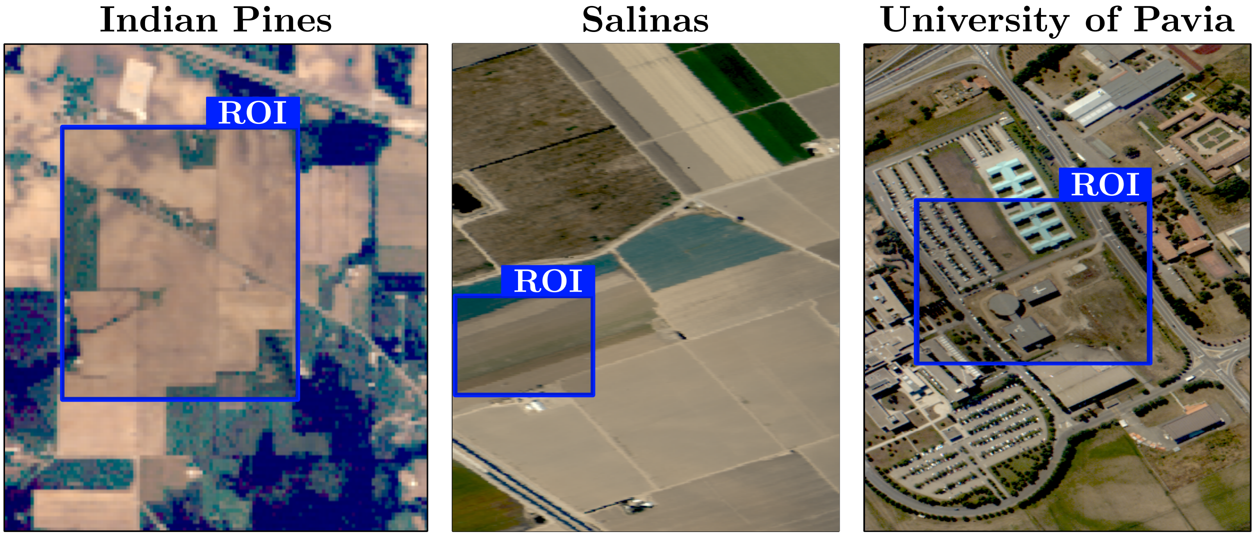


Figure 35. False-color images and regions of interest (ROI) for the three real remote sensing images used in the experiments.

Specifically, in the first stage, we acquire the segmentation map $\tilde{\mathbf{m}}$ for an HSI. Such procedure involves computing PCA over $\mathbf{X}$ to retrieve only the three principal components, which takes $O(N)$, and performing SLIC superpixels (Achanta et al., 2012) which also has linear time complexity $O(N)$. The second stage requires to execute Algorithm 4, which has $O(\rho N_e^2)$ time complexity over E subsets, then the overall complexity of this stage will be $O(\rho \max(N_1^2, \cdots, N_E^2))$. The third stage entails solving Eq. (42) which is a LASSO problem that can be efficiently computed in $O(M^2 N)$ using the LARS algorithm (Efron et al., 2004). Finally, in the last stage, the 2D convolution takes $O(N)$ as $K_s \ll N$ and, since for the spectral clustering we only need the k largest singular values, we can use the truncated singular value decomposition (SVD), which takes $O(k^2 N)$. Thus, the overall complexity of this stage is $O(k^2 N)$. Therefore, the complexity of Algorithm 5 will be dominated by the complexity of the third stage, hence it will run in $O(M^2 N) = O(\rho^2 N^3)$, where $\rho \in (0,1)$.

4.3. Experimental Evaluation

In this section, the performance of SC-SSC is shown for unsupervised HSI classification. The sparse optimization problem in Eq. (42) is solved by the LASSO version of the LARS algorithm (Efron et al., 2004) implemented in the SPAMS package (Mairal et al., 2010). All the experiments were run on an Intel Core i7 9750H CPU (2.60GHz, 6 cores), with 32 GB of RAM. Please refer to the repository list in Appendix 4 to found a Matlab implementation of SC-SSC Algorithm.

4.3.1. Setup. **Datasets**. The proposed subspace clustering approach (SC-SSC) was tested on three well-known hyperspectral images[3] with different imaging environments (see Fig. 2). 35. The **Indian Pines** hyperspectral data set has 145×145 pixels and 200 spectral bands in the range of $0.4 - 2.5\mu m$ and includes sixteen land-cover classes. The second scene, **Salinas**, has 512×217 pixels and 204 spectral bands in the range of $0.24 - 2.40\mu m$ and includes sixteen land-cover classes. The third scene, **University of Pavia**, comprises 610×340 pixels and has 103 spectral bands with spectral coverage ranging from $0.43 - 0.84\mu m$ and includes nine classes.

As mentioned in Section 1.3.2, most SSC-based methods for HSI are inefficient and have a large memory dependency; hence they suffer from out-of-memory problems during their execution. In order to make a fair comparison with such non-scalable methods, I separately take a frequently used region of interest (ROI) of these datasets for evaluation, as done in several previous works (Hinojosa et al., 2021c; Cai et al., 2020; Zhang et al., 2016). The selected ROIs are

---

[3] `https://link.carloshinojosa.me/hsi_scenes`.

illustrated in Fig. 35. The Indian Pines ROI has a size of 70×70 pixels, which includes four main land-cover classes: corn-no-till, grass, soybeans-no-till, and soybeans-min-till. The Salinas ROI comprises 83×83 pixels and includes six classes: broccoli-1, corn-senesced, lettuce-4wk, lettuce-5wk, lettuce-6wk, and lettuce-7wk. Finally, the University of Pavia ROI is composed of 200×200 pixels and includes all the classes (nine) as in the full image: asphalt, meadows, gravel, trees, metal sheets, bare soil, bitumen, bricks, and shadows. For all experiments, The spectral dimensions of each image are reduced by using PCA to $D = 0.25L$, where L is the number of spectral bands. Then, the data cube is rearranged to form a matrix $\mathbf{X} \in \mathbb{R}^{D \times N}$ and normalize the columns to have unit ℓ_2 norm.

Baselines and Evaluation Metrics. The proposed approach is compared with several methods and separate them into four categories: *Scalable SSC-based methods*, *Non-scalable SSC-based methods*, *Unsupervised Deep-learning-based methods*, and *Traditional clustering methods*. Specifically, under the *Scalable SSC-based methods*, the proposed approach is compared to the algorithms: SSC-OMP (You et al., 2016b), SSSC (Peng et al., 2013), ESC-FFS (You et al., 2018), and SR-SSC (Abdolali et al., 2019). Under the *Non-scalable SSC-based methods*, the proposed approach is compared with the algorithms: S-SSC (Zhang et al., 2016), ORGEN (You et al., 2016a), 3DS-SSC (Hinojosa et al., 2021c), and EGCSC (Cai et al., 2020). The results with SSC as an additional reference are shown. For the sake of completeness, the proposed approach is compared to VAE (Tulczyjew et al., 2020), AE-GRU (Tulczyjew et al., 2020), AE-LSTM (Tulczyjew et al., 2020), and the 3D-CAE (Nalepa et al., 2020), which fall under the *Unsupervised Deep-learning-based methods*, and the K-Means (Lloyd, 1982), FCM (Bezdek, 2013), FCM_S1 (Chen and Zhang,

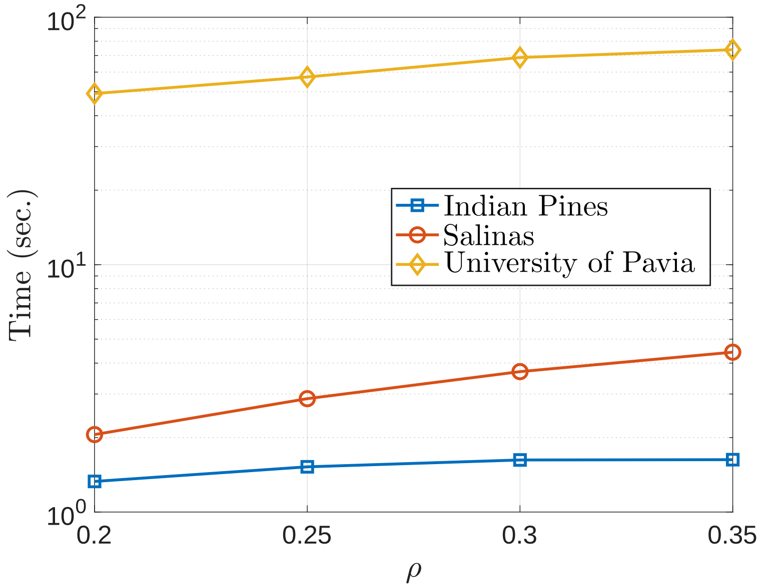


Figure 36. Running time (in seconds) as a function of the ρ parameter.

2004), and FCM_S2 (Chen and Zhang, 2004), which are *Traditional clustering methods*.

To compare the clustering performance of the proposed model, five standard metrics were used: user's accuracy (UA), average accuracy (AA), overall accuracy (OA), Kappa coefficient, and normalized mutual information (NMI) (Lillesand et al., 2015; Strehl and Ghosh, 2002). In particular, UA, AA, OA, and Kappa coefficient can be obtained employing an error matrix (a.k.a, confusion matrix) (Lillesand et al., 2015). UA represents the clustering accuracy of each class, while AA is the mean of UA, and OA is computed by dividing the number of correctly classified pixels by the total number of reference pixels. UA, AA, and OA values are presented in percentage, while Kappa coefficients and NMI values range from 0 (poor clustering) to 1 (perfect clustering). The methods are compared in terms of clustering time.

4.3.2. Parameters analysis and tuning. In this section, the impact of the parameters ρ, E, and K_s is analyzed in Algorithm 5. I conduct different experiments varying each parameter,

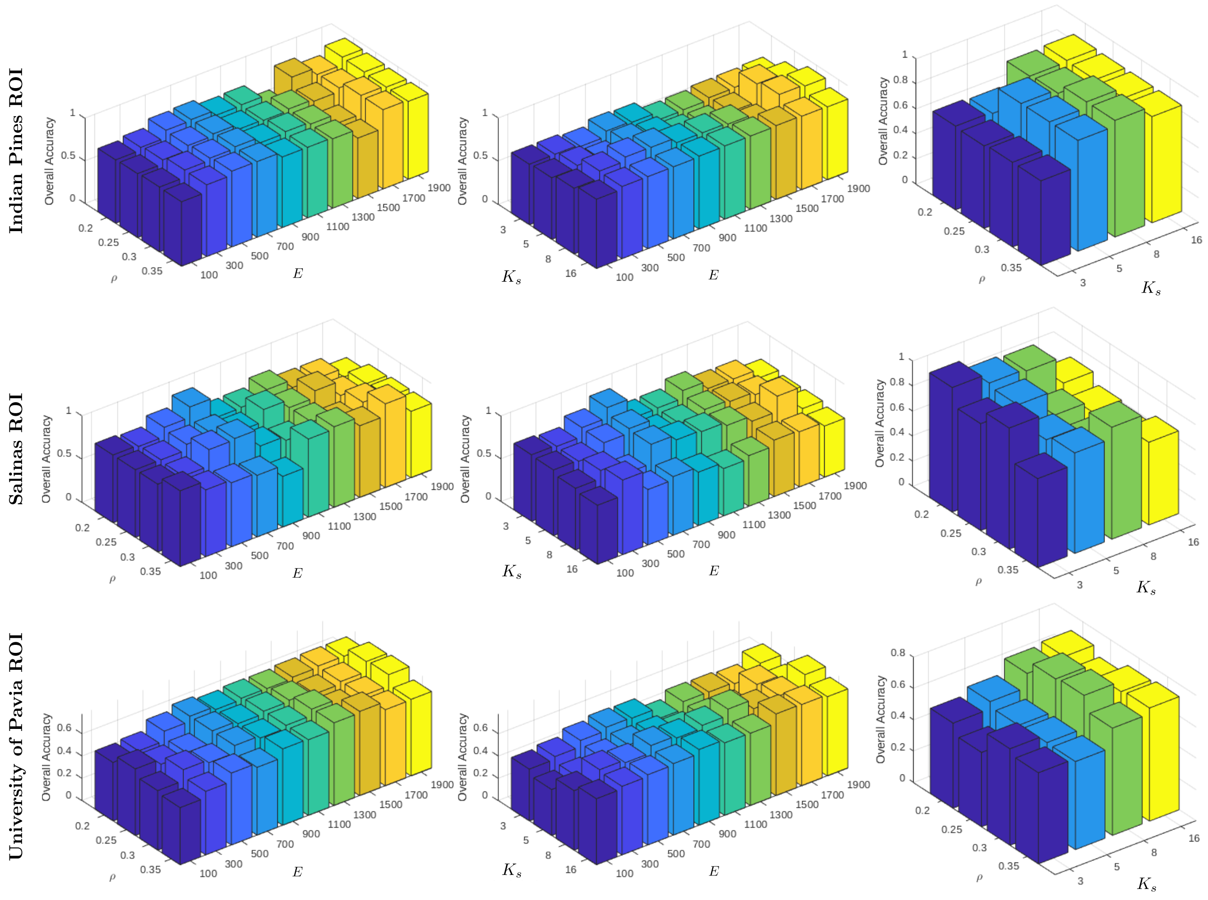


Figure 37. Analysis of influence of parameters ρ, E, and K_s in Algorithm 5. Each row presents the 3D bar plot of ρ vs. E, K_s vs. E, and ρ vs. K_s for each dataset, and the evaluation is given by the overall accuracy with values between 0 and 1. The plot ρ vs. E shows how the OA changes when the number of selected representative data points varies concerning the number of segments E. K_s vs. E depicts how the OA is affected by the number of spectral segments and the kernel size used in the 2D convolution to enhance the sparse coefficient matrix. Finally, ρ vs. K_s shows the change in OA when the number of selected representative data points varies, and a specific kernel size is used in the 2D convolution.

with the others fixed. During simulations, I observed that the parameter ρ directly impacts the execution time of the proposed method. Figure 36 presents the running time of SC-SSC for all the datasets. As shown, increasing ρ directly increases the running time; however, the most significant increment in time is given by the number of spectral pixels N, i.e., the size of the HSI, as observed

Table 15

Selected parameters in Algorithm 5 for each HSI.

Parameter	**Indian Pines**	**Salinas**	**University of Pavia**
ρ	0.35	0.2	0.3
E	1700	900	1900
K_s	8	3	8

with the differences in time between the curves. As presented in Section 4.2.3.2, this behavior is expected since the computational complexity of the algorithm is $O(\rho^2 N^3)$.

In the experiments, the parameters were varied between the following values: $\rho \in \{0.2, 0.25, 0.3, 0.35\}, E \in \{100, 300, 500, 700, 900,, 1100, 1300, 1500, 1700, 1900\}, K_s \in \{3, 5, 8, 16\}$. The parameter ρ determines the number of the selected most-representative data points within each of the E segments, and K_s is the kernel size used in the 2D convolution. Figure 37 shows the performance of the proposed method with a different combination of the parameters for all the datasets, where the overall accuracy is shown between 0 and 1. By analyzing Figure 37, I noticed that the precision mainly changes with values of ρ and E. Also, $K_s = 8$ provided good results for the three images. In practice, an adequate balance when selecting the parameters ρ, E, and K_s is crucial to obtain the best performance. The selection of the best parameters often varies with the dataset's size; however, an effective way to select them is by using a grid search. Empirically, I found that the suitable values of ρ and E can be found in the range $[0.2, 0.3]$ and $[500, 1000]$, respectively. Further, $\rho = 0.3, E = 700, K_s = 8$ are good starting points. According to the empirical study, I provide a group of the best parameter setting in Table 15.

4.3.3. Ablation Studies. We conduct six ablation experiments to investigate different configurations for the proposed workflow in Fig. 34. Specifically, the SC-SSC's performance is compared when incorporating/excluding PCA, superpixels, and the 2D convolution. Table 16 present the results obtained from the different combinations in terms of OA and NMI for the three tested images. It is important to highlight that using superpixels to extract spatial similarities improves the clustering performance for the three tested images in all the cases, which evidences the importance of the neighboring spatial information in the proposed workflow. Also, using superpixels and the 2D convolution (Experiment II) leads to the second-best result, while only using 2D convolution (Experiment III) does not lead to a significant clustering improvement. Finally, Experiment VI corresponds to the proposed approach where the best results are achieved in terms of OA and NMI when using the three operations as described in the workflow in Fig. 34 or Algorithm 5.

Table 16
Ablation study. The configuration shown in bold (Experiment VI) corresponds to the SC-SSC.

				Indian Pines		Pavia		Salinas	
Experiment	PCA	Superpixels	2D Conv	OA	NMI	OA	NMI	OA	NMI
I	✓	✓		64.36	0.36	37.78	0.49	74.26	0.65
II		✓	✓	80.53	0.63	67.04	0.81	84.63	0.86
III			✓	41.72	0.29	38.11	0.46	43.33	0.42
IV	✓		✓	41.81	0.3	38.21	0.47	49.53	0.41
V		✓		65.84	0.37	47.60	0.49	73.29	0.77
VI	✓	✓	✓	**93.14**	**0.8**	**77.57**	**0.82**	**99.42**	**0.98**

4.3.4. Comparison with scalable methods. The performance of SC-SSC is compared with the scalable approaches: SSC-OMP, SSSC, ESC-FFS, and SR-SSC. Figure 38 and Table 20 present the visual and quantitative evaluation results respectively on the full HSIs shown in Fig. 35. In the table, the best results are shown in bold, and the second-best is underlined. It is important to mentioning that the proposed SC-SSC method outperforms the other approaches in terms of OA, Kappa, and NMI score from both qualitative and quantitative results. Although the proposed method is not the fastest one, it provides high clustering performance in a shorter amount of time than other methods.

4.3.5. Comparison with non-scalable methods. The obtained land cover maps are presented on the Indian Pines, Salinas, and the University of Pavia ROIs in Fig. 39, where the performance of SC-SSC is compared with the non-scalable methods: SSC, S-SSC, ORGEN, and 3DS-SSC. The quantitative evaluations corresponding to the OA, Kappa, NMI, and Time with the non-scalable clustering methods are reported in Table 17. From Table 17, it can be observed that, in general, the proposed SC-SSC method performs better than others. Specifically, SC-SSC achieves an OA of 93.14% and 99.42%, in only 1.63 and 2.06 seconds, for the Indian Pines and Salinas dataset, respectively, which are remarkable results for unsupervised learning settings. Similarly, for the University of Pavia ROIs, it is observed from Table 17 that the proposed SC-SSC achieves the best clustering performance in all the accuracy evaluation metrics, among all the other algorithms.

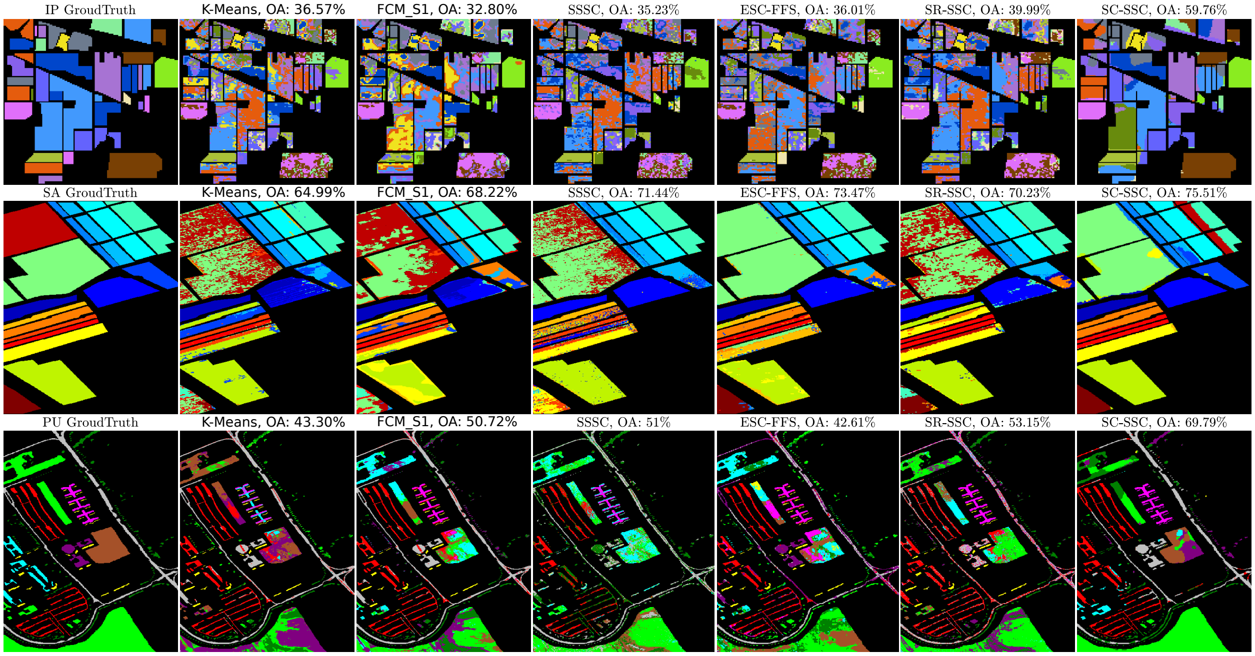


Figure 38. Land cover maps on the Indian Pines (IP), Salinas Valley (SA), and University of Pavia (PU) Full images. The proposed method is compared with traditional clustering algorithms (K-Means and FCM_S1) and scalable SSC-based state-of-the-art methods.

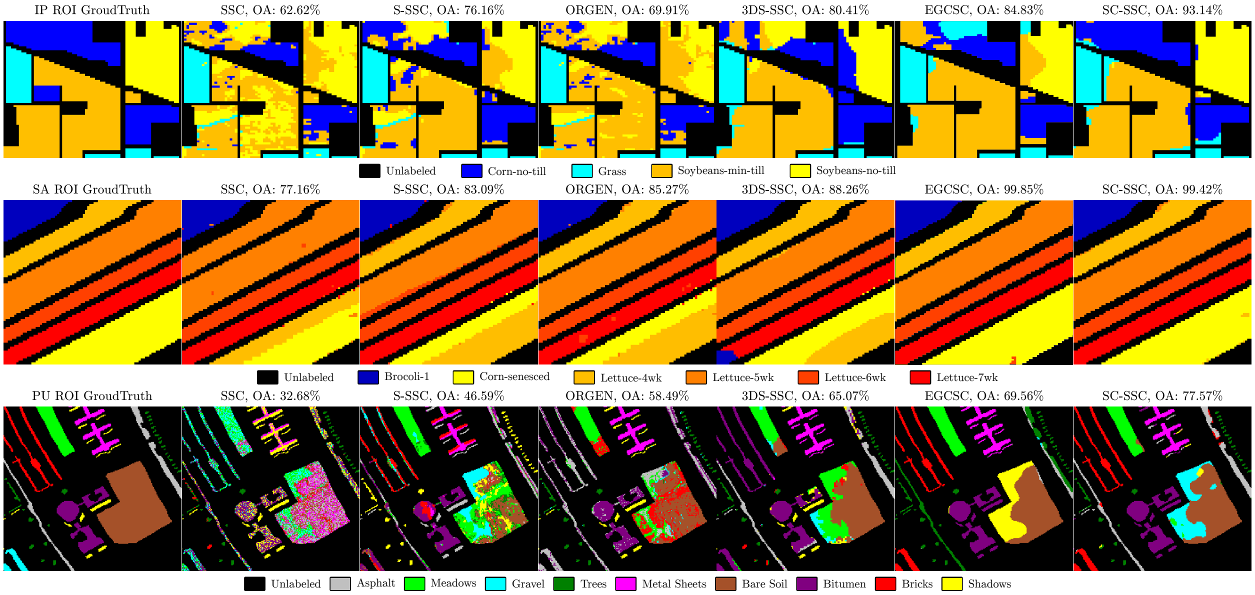


Figure 39. Land cover maps of (first row) Indian Pines ROI, (second row) Salinas ROI, and (last row) the University of Pavia ROI. The proposed method is compared with non-scalable methods.

Table 17

Clustering performance of the compared methods on the Indian Pines, Salinas, and the University of Pavia ROIs.

Dataset	**Class**	**SSC**	**S-SSC**	**ORGEN**	**3DS-SSC**	**EGCSC**	**SC-SSC**
Indian Pines ROI	OA	62.62	76.16	69.91	80.41	84.83	**93.14**
	Kappa	0.48	0.66	0.56	0.72	0.64	**0.90**
	NMI	0.39	0.47	0.42	0.57	0.64	**0.79**
	Time [s]	285.57	301.41	668.36	341.21	69.37	**1.63**
Salinas ROI	OA	77.16	83.09	85.27	88.26	**99.85**	99.42
	Kappa	0.71	0.79	0.82	0.86	**0.99**	**0.99**
	NMI	0.85	0.84	0.87	0.87	0.99	**0.98**
	Time [s]	319.42	327.66	1355.79	377.11	92.95	**2.06**
University of Pavia ROI	OA	32.68	46.59	58.49	65.07	69.56	**77.57**
	Kappa	0.23	0.38	0.50	0.56	0.63	**0.72**
	NMI	0.41	0.49	0.59	0.65	**0.82**	**0.82**
	Time [s]	17821.25	10195.89	551.02	10501.29	171.21	**68.72**

4.3.6. Comparison with unsupervised deep-learning-based methods. For the sake of completeness, the proposed SC-SSC method is compared with unsupervised deep-learning-based methods based on autoencoders (AE) for hyperspectral image clustering. Three of them were proposed in (Tulczyjew et al., 2020) (VAE, AE-GRU, and AE-LSTM), and the 3D-CAE method was proposed in (Nalepa et al., 2020) which is based on a 3D convolutional AE. Note that the SC-SSC was only compared with totally unsupervised deep learning approaches to make a fair comparison. Table 18 shows the quantitative results in terms of the NMI score. In the table, the best result is shown in bold font, and the second-best is underlined. As observed, SC-SSC obtains an NMI score of 0.601, 0.892, and 0.643 on Indian Pines, Salinas, and University of Pavia full hyperspectral images, respectively, corresponding to the highest clustering scores.

Table 18

Quantitative comparison with unsupervised deep learning-based methods in terms of NMI score.

Methods Datasets	VAE	3D-CAE	AE-GRU	AE-LSTM	SC-SSC
Indian Pines	0.429	0.504	0.515	0.478	**0.601**
Salinas	0.722	0.839	0.825	0.830	**0.892**
University of Pavia	0.505	0.639	0.524	0.569	**0.643**

4.3.7. Comparison with traditional clustering methods. I further compare SC-SSC with traditional centroid-based clustering methods, such as K-Means (Lloyd, 1982), fuzzy c-means (FCM) (Bezdek, 2013), and FCM with spatial information (FCM_S1 and FCM_S2) (Chen and Zhang, 2004). Such methods are based on the fact that similar data points generate clusters in the feature space. These algorithms optimize the clusters by giving initial clustering centers and continuously updating their location until minimizing the sum of squared errors. Table 19 shows the quantitative evaluation results. Furthermore, Fig. 38 shows the clustering map obtained by the K-Means and the FCM_S1 algorithms. The table and figure show that such traditional clustering methods do not perform well on the real HSIs used in this Chapter. In general, such methods are sensitive to noise and get easily stuck in a local optimum. On the other hand, SC-SSC is more robust to the noise present in the real HSI and achieves higher clustering performance.

4.4. Conclusions

In this Chapter, a new subspace clustering algorithm is proposed for unsupervised classification of HSIs, which is efficient and takes advantage of hyperspectral images' neighboring spatial information to boost the clustering accuracy. SC-SSC algorithm considers the spatial similarity among

Table 19

Quantitative comparison with traditional clustering algorithms in terms of OA.

Methods / Datasets	K-means	FCM	FCM_S1	FCM_S2	SC-SSC
Indian Pines	36.57	31.41	32.80	36.54	**59.76**
Salinas	64.99	55.93	68.22	64.38	**76.17**
Pavia University	43.30	47.98	50.72	49.60	**69.79**

spectral pixels to select the most representative ones, such that all other adjacent points can be well-represented by those pixels in terms of the sparse representation cost. Then, the obtained sparse coefficients matrix is enhanced by filtering the coefficients, and a fast spectral clustering algorithm gives the segmentation. Through simulations using traditional test hyperspectral images, the effectiveness of the proposed SC-SSC method was demonstrated for fast HSI classification, obtaining remarkable high clustering performance compared with state-of-the-art SSC algorithms and even novel unsupervised-deep-learning-based methods.

Table 20

Clustering performance on the Indian Pines, Salinas, and the University of Pavia Full Images.

Dataset	Class	SSC-OMP	SSSC	ESC-FFS	SR-SSC	SC-SSC
Indian Pines	Alfalfa	**7.69**	6.90	0.60	0.00	0.00
	Corn-no-till	37.04	34.33	57.79	47.95	**66.96**
	Corn-min-till	20.83	17.46	17.45	17.84	**55.25**
	Corn	22.73	15.56	11.34	8.85	**27.25**
	Grass-pasture	21.05	35.53	34.81	32.39	**90.52**
	Grass-trees	57.14	**84.91**	77.94	70.11	77.05
	Grass-mowed	**0.45**	0.00	0.29	0.00	0.00
	Hay-windrowed	25.81	88.29	85.91	76.91	**90.53**
	Oats	0.00	3.36	0.02	**4.36**	0.00
	Soybean-no-till	18.18	30.89	38.54	36.11	**64.15**
	Soybean-min-till	28.67	52.25	55.12	48.52	**62.23**
	Soybean-clean	6.32	22.17	17.44	15.76	**32.10**
	Wheat	8.24	37.60	37.53	56.35	**66.13**
	Woods	11.54	88.22	81.03	89.42	**91.49**
	Building-grass	5.56	17.33	25.51	22.15	**67.34**
	Stone-stell-towers	0.00	45.20	18.53	38.14	**49.73**
	AA	9.64	40.55	33.94	40.26	**56.37**
	OA	12.84	35.23	36.01	39.99	**59.76**
	Kappa	0.03	0.29	0.30	0.33	**0.55**
	NMI	0.03	0.42	0.43	0.44	**0.60**
	Time [s]	37.93	32.35	47.15	**16.39**	19.33
Salinas	Brocoli 1	11.34	0.00	**100.00**	0.00	88.39
	Brocoli 2	28.88	57.41	**99.68**	62.98	99.57
	Fallow	9.70	78.98	72.16	0.00	**80.25**
	Fallow Plow	7.43	91.56	89.52	**94.35**	64.12
	Fallow Smooth	22.58	58.11	71.98	71.82	**99.89**
	Stubble	21.54	99.05	99.70	**99.91**	97.04
	Celery	22.08	97.12	97.78	85.62	**100.00**
	Grapes	3.47	70.19	58.44	**70.68**	59.22
	Soil	24.68	91.18	**95.91**	85.58	91.11
	Corn	3.91	61.92	26.70	58.04	**89.95**
	Lettuce 4	3.97	**79.10**	24.01	0.00	0.00
	Lettuce 5	4.42	**81.09**	65.14	51.99	63.49
	Lettuce 6	1.90	**41.17**	0.00	0.00	0.00
	Lettuce 7	0.00	**58.92**	55.09	51.18	51.13
	Vineyard	11.57	**56.92**	25.00	49.22	0.00
	Vineyard trellis	3.05	0.00	98.53	98.89	**100.00**
	AA	10.95	64.76	73.15	61.57	**75.11**
	OA	10.74	71.44	73.47	70.23	**75.51**
	Kappa	0.05	0.68	0.70	0.67	**0.73**
	NMI	0.16	0.75	0.83	0.78	**0.85**
	Time [s]	78.18	**58.29**	615.40	168.18	580.33
University of Pavia	Asphalt	20.67	60.00	**95.60**	68.64	73.52
	Meadows	48.11	85.58	71.91	77.98	**96.81**
	Gravel	**13.73**	0.15	0.06	1.25	11.73
	Trees	9.54	31.28	18.03	**72.63**	32.48
	Metal sheets	81.71	**97.01**	36.74	60.15	86.22
	Bare soil	0.00	5.75	24.15	15.04	**97.53**
	Bitumen	0.00	4.26	**27.80**	0.00	0.00
	Bricks	0.00	55.27	**60.70**	39.11	54.54
	Shadows	0.00	43.32	62.99	**96.45**	0.00
	AA	16.48	39.76	**56.87**	53.50	52.23
	OA	35.34	51.00	42.61	53.15	**69.79**
	Kappa	0.07	0.39	0.33	0.42	**0.61**
	NMI	0.07	0.39	0.49	0.52	**0.64**
	Time [s]	201.52	**73.45**	1821.58	148.55	913.67

5. Applications in CSI (I): Coded Aperture Design for Compressive Spectral Subspace Clustering

Part of this chapter has been adapted from the journal paper (Hinojosa et al., 2018b) (published).

Table 21
Notation used in Chapter 5.

Notation	**Definition**
M	Number of rows in the HSI.
N	Number of columns in the HSI.
L	Number of spectral bands of the HSI.
S	Number of measurement shots.
Δ	Coding pattern bandwidth.
$\mathscr{F} \in \mathbb{R}^{M\times N\times L}$	Full spectral input data cube.
$\mathbf{C}^s \in \mathbb{R}^{M\times N\times L}$	3D Coded aperture for the s-th measurement shot.
$\hat{\mathbf{Y}}^s \in \mathbb{R}^{M\times N}$	Measurement matrix acquired in the s-th shot.
$\mathbf{Y}$	Sorted version of $\hat{\mathbf{Y}}^s$ for all s using Eq. 45.
$\mathbf{F} \in \mathbb{R}^{L\times MN}$	Matrix form of $\mathscr{F}$; the columns are spectral signatures $\mathbf{f}_j$.
$\Phi \in \mathbb{R}^{S\times L}$	Projection (encoding) matrix to be designed.
λ_1^s, λ_2^s	Cutoff wavelengths used in the design of Φ.
$\varphi^s \in \{0,1\}^L$	L-long vector defined by λ_1^s, and λ_2^s.
δ_x	Kronecker delta function, i.e., $\delta_x = 1$ only when $x = 0$.
$\mathbf{Z} \in \mathbb{R}^{MN\times MN}$	Sparse coefficient matrix in subspace clustering.
$\mathscr{Z} \in \mathbb{R}^{M\times N\times MN}$	Coefficient matrix $\mathbf{Z}$ in 3D form.
$\mathscr{W} \in \mathbb{R}^{3\times 3\times 3}$	3D moving window used in the 3D median filtering of $\mathscr{Z}$.
$\bar{\mathbf{Z}} \in \mathbb{R}^{MN\times MN}$	Enhanced coefficient matrix used for regularization.

5.1. Introduction

In the previous chapters, different subspace clustering algorithms were proposed for HSI that consider spatial similarity to improve the accuracy and efficiency of the clustering. In this chapter and the following chapters, I adopted similar strategies to perform clustering directly on compressed

HSI measurements acquired using the compressive spectral imaging (CSI) theory. Here, the key observation is that the careful design of the compressive sensing mechanism allows the extraction of useful features that enable the clustering directly on compressed measurements. In this thesis, I particularly focus on designing the coded aperture which determines how spectral signatures are encoded. Refer to section 1.4 for details on CSI components.

CSI has emerged as a new spectral imaging approach which acquires compressed 2D projections of the entire data cube rather than direct measurements of all voxels (Arce et al., 2014). This approach simultaneously allows sensing and reducing the data dimensionality without further processing steps (Martín and Bioucas-Dias, 2016; Xing et al., 2012). Therefore, the cost of sensing, storing, transmitting and processing spectral images using this method is significantly reduced. Until now, different CSI sensing methods have been proposed to compress spectral images (Martín and Bioucas-Dias, 2016; Xing et al., 2012; Cao et al., 2016; Arguello and Arce, 2014; Correa et al., 2016). Particularly, in this chapter I use the spatial-spectral coded compressive spectral imager (3D-CASSI). This CSI sensing scheme modulates the spectral data cube in spatial and spectral dimensions using a 3D coded aperture (ensembles of 2D coded apertures) or a coding pattern array. Then, the coded spectral data cube is integrated along the spectral dimension such that each spatial position of the acquired measurements contains the compressed information of a single coded spectral signature (Cao et al., 2016). Please refer to section 1.5 and the works in (Gehm et al., 2007; Lin et al., 2014a,b) for more details about the 3D-CASSI scheme.

Since the coded aperture determines how the spectral pixels are encoded, it is possible to define the general structure of the acquired compressed data by properly designing the 3D coded aperture ensemble. Assuming that the compressed measurements with similar characteristics lie on the same subspace, the aim is to perform subspace clustering directly (compressed spectral pixels) without recovering the original spectral scene. Specifically, the main contribution in this chapter is to provide a coded aperture design such that the information, and the similarity between the spectral signatures, are approximately preserved after the sensing process, enhancing the spectral pixel clustering directly in the compressed measurements. The proposed design is based on three criteria: considering that surface-emitted spectral signatures are, in general, relatively smooth functions of the wavelength, only neighboring spectral bands are sensed; the cosine of the angle between each pair of spectral pixels should be approximately preserved in the compressed domain; each 3D coded aperture realization should extract new information from the scene. Once the designed set of coded apertures is obtained, 2D projections of the spectral image data are acquired using the 3D-CASSI sensing approach. Finally, a sparse subspace clustering-based algorithm is proposed in order to perform spectral image clustering. Such an algorithm incorporates a 3D spatial regularizer in the SSC problem to exploit the spatial correlation of spectral images. The overall CSI subspace clustering workflow is depicted in Fig. 40. Despite recent works that have successfully performed subspace clustering on compressed measurements (Liu et al., 2019; Mao and Gu, 2014), this chapter particularly designs a binary coded aperture ensemble to improve clustering results directly on CSI measurements without recovering the underlying spectral scene.

5.2. Compressed Measurements Acquisition

Denote $\mathscr{F}$ as the spatio-spectral input data cube, with $M \times N$ spatial dimensions, L spectral bands and entries denoted as $\mathscr{F}_{m,n,k}$, where m and n index the spatial coordinates, and k determines the

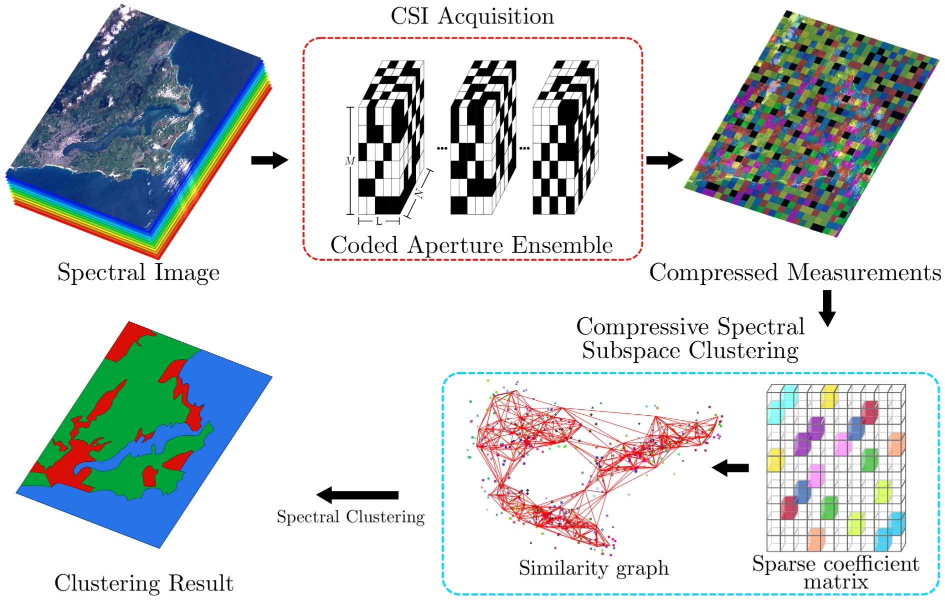


Figure 40. CSI subspace clustering workflow. Spectral signatures of the scene are encoded by the coded aperture ensemble before the spectral-wise integration. Proper coded aperture design to preserve spectral similarities, allows to perform clustering directly on the compressed measurements.

k-th spectral band. As shown in Fig. 11, the 3D-CASSI first modulates the voxels of the spectral scene using a 3D coded aperture $\mathbf{C}^s$, whose entries are indexed as $C^s_{m,n,k}$. Then, the coded spectral scene is integrated into the focal plane array (FPA) detector along the spectral axis. In CSI, it is possible to acquire $S \ll L$ measurement shots, each one employing a different coded aperture, such that different measurements of the spectral data cube are acquired each time. Therefore, the output of the sensing process, at the (m,n)-th detector pixel and a specific snapshot s, can be expressed as

$$\hat{Y}^s_{m,n} = \sum_{k=0}^{L-1} \mathscr{F}_{m,n,k} C^s_{m,n,k}. \tag{44}$$

Note that to each spatial location of $\mathbf{C}^s$ is assigned a coding pattern $\phi^s \in \mathbb{R}^L$, with entries $(\phi^s)_k \in \{0,1\}$, that modulates a spectral pixel in that particular position before being integrated at the FPA detector. Further, observe that there is a finite number of coding patterns randomly distributed in $\mathbf{C}^s$.

At each measurement shot, the spectral pixels are encoded differently using distinct coding patterns, hence new information is acquired from the underlying data cube. Denoting P as the number of different coding patterns distributed on $\mathbf{C}^s$, if the number of measurement shots S is greater than P, then some pixels are oversampled, thus redundant information is acquired. On the other hand, if $S < P$ clustering becomes hard since the similarity among two spectral pixels in the same cluster may decrease if they are encoded (projected) differently. Finally, if $S = P$ the 3D coded aperture for each measurement shot can be designed such that a particular pixel is modulated once by a different coding pattern in a particular snapshot, acquiring new information each time.

The set of compressed measurements from (44) can be arranged in a $S \times MN$ matrix $\hat{\mathbf{Y}} = \left[[\hat{Y}^0_{0,0}, \hat{Y}^0_{1,0}, \cdots, \hat{Y}^0_{0,1}, \cdots, \hat{Y}^0_{(M-1),(N-1)}]^T, \cdots, [\hat{Y}^{(S-1)}_{0,0}, \cdots, \hat{Y}^{(S-1)}_{(M-1),(N-1)}]^T\right]^T$, where each column value corresponds to a compressed spectral signature. Note that, each row of $\hat{\mathbf{Y}}$ contains the compressed information (spectral responses) of the pixels acquired in the s-th snapshot. However, column vectors of $\hat{\mathbf{Y}}$ may contain the measurement acquired with a specific coding pattern ϕ^s in different rows since, in a particular snapshot, all the spectral pixels are not necessarily encoded by the same coding pattern. Then, the matrix $\hat{\mathbf{Y}}$ is not convenient for SSC as its structure makes difficult to discriminate among compressed measurements. Therefore, using $S = P$, the entries of $\hat{\mathbf{Y}}$ are rearranged to form a new matrix $\mathbf{Y}$, such that each row contains the compressed information acquired with a specific coding pattern ϕ^s. Note that this rearrangement is possible only when $S = P$ since in this case it can be guaranteed that, at a specific snapshot, one pixel is encoded only once by a different pattern and, at the end of the sensing procedure, all pixels were encoded by the whole set of S coding patterns. Formally, the rearrangement can be expressed as

$$Y_{s,j} = \hat{Y}_{s',j} \quad \text{if } \hat{Y}_{s',j} = (\phi^s)^T \mathbf{f}_j \quad \forall s', \tag{45}$$

for $s, s' = 0, \cdots, S-1$, where $\mathbf{f}_j \in \mathbb{R}^L$ denotes the j-th spectral signature for $j = 0, \cdots, MN-1$. This rearrangement, depicted in Fig. 15, preserves the structure of the underlying high dimensional data improving the subspace clustering results. Alternatively, define the matrix of S coding patterns as $\Phi = \left[\phi^0, \phi^1, \cdots \phi^{S-1}\right]^T$ then, the problem of acquiring and rearranging the measurements $\hat{\mathbf{Y}}$ can be succinctly expressed as the random projection

$$\mathbf{Y} = \Phi \mathbf{F}, \tag{46}$$

where $\mathbf{F} = \left[[\mathscr{F}_{0,0,0}, \mathscr{F}_{1,0,0}, \cdots, \mathscr{F}_{0,1,0}, \cdots, \mathscr{F}_{(M-1),(N-1),0}]^T, \cdots, [\mathscr{F}_{0,0,(L-1)}, \cdots, \mathscr{F}_{(M-1),(N-1),(L-1)}]^T\right]^T$ is a $L \times MN$ matrix whose columns are the spectral signatures $\mathbf{f}_j$ of the data cube and Φ can be viewed as the projection matrix.

A typical procedure after the compressed measurements acquisition is signal recovery, which is achieved using nonlinear and relatively expensive optimization-based or iterative algorithms (Figueiredo et al., 2007; Bacca et al., 2017b). CSI signal recovery procedure is explained in (Arce et al., 2014). The proposed method in this chapter aims to avoid the computational cost of recovering all the data cube by performing all the spectral image clustering directly on the compressed measurements using a set of designed coded apertures. Since the coding patterns ϕ^s

determine the structure of $\mathbf{C}^s$, designing the set of 3D coded apertures is equivalent to the coding pattern design.

5.3. Coding Pattern Design for CSI Subspace Clustering

Based on the concept of affinity (Soltanolkotabi et al., 2012), which characterizes the similarity between two subspaces, it has been theoretically proven and numerically verified that several dominant subspace clustering algorithms could successfully perform clustering on the compressed data (Heckel et al., 2015; Tschannen and Bölcskei, 2016; Wang et al., 2016). These works employ the random projection method (Vempala, 2005), using Gaussian random matrices, in order to acquire the dimensionality-reduced or compressed data. Such random projection matrices preserve the structure of high dimensional data; hence it is possible to directly learn from the low dimensional data (Calderbank et al., 2009). In addition, recent works in (Li and Gu, 2017; Jiao et al., 2017) have theoretically proved that, with high probability, the Euclidean distance and principal angles between two subspaces remain almost unchanged after Gaussian random projections, which allows performing sparsity related signal processing tasks directly on compressed measurements. However, in CSI, the measurements are typically acquired using sensing or projection matrices whose entries are not Gaussian. In addition, although work in (Wang et al., 2016) provides theoretical results for other compression matrices which are not necessarily Gaussian, these do not take into account the structure of hyperspectral data. In this section, a coding pattern design for CSI, which considers the hyperspectral data structure, is presented. The proposed design considers how the spectral scene information is encapsulated in the CSI measurements. Further, such design allows one to learn from the compressed data and preserves its ℓ_2 norm and the spectral signature similarities among vectors on the compressed domain. Additionally, an algorithm to generate the coding matrix Φ is developed in this section.

5.3.1. Coding Pattern Design. Recent works in CSI have focused on properly designing the coding patterns in order to better reconstruct the underlying spectral scene (Arguello and Arce, 2014, 2013). These designs use the restricted isometry property (RIP) as the main optimization criterion. On the other hand, because the proposed method in this chapter aims to perform classification on the compressed measurements, the design of the coding patterns must preserve the similarity among the spectral signatures. In order to design the coding patterns matrix Φ, the following three design criteria are considered.

5.3.1.1. Sensing Scheme. The entries of the matrix Φ can be chosen from a Bernoulli distribution $(\Phi)_{s,k} \sim Be(p)$. Therefore, the entries of the s-th coding pattern can be expressed as

$$(\phi^s)_k = \begin{cases} 1, & \text{with probability } p \\ 0, & \text{with probability } q, \end{cases} \tag{47}$$

for $k = 0, 1, \cdots, L-1$, where $q = 1 - p$. A projection matrix with this structure simply carries out a random sampling on the data vectors, across all the spectral bands, before performing element-wise addition. Considering that surface-emitted spectral signatures are, in general, relatively smooth functions of wavelength (Gu et al., 2000), acquiring the information of different sets of adjacent spectral bands at each snapshot will preserve the original signal structure. Therefore, the intuition is to perform the sampling of neighboring spectral bands instead of randomly sampling all the spectral data vectors, which could add outliers to the measurements.

For each coding pattern ϕ^s, two cutoff wavelengths $\lambda_1^s, \lambda_2^s \in \{0, 1, \cdots, L-1\}$ are selected, such that $\lambda_1^s < \lambda_2^s$ and $\lambda_2^s - \lambda_1^s + 1 = \Delta$, where Δ is defined as the coding pattern bandwidth. Then, the band-structured matrix can be expressed as

$$(\phi^s)_k = \begin{cases} \varphi_k^s, \iff \lambda_1^s \leq k \leq \lambda_2^s \\ \\ 0, \quad \text{otherwise,} \end{cases} \tag{48}$$

where $\varphi^s \in \{0,1\}^L$ is a L-long vector whose nonzero elements, within the region delimited by λ_1^s and λ_2^s, represent the sampled spectral bands of the pixels. Equation (48) can be alternatively written as

$$(\phi^s)_k = \delta_{\lfloor \lambda_1^s / k \rfloor} \delta_{\lfloor k / \lambda_2^s \rfloor} \varphi_k^s, \tag{49}$$

where δ_x is the Kronecker delta function, i.e., δ_x is equal to 1 only when $x = 0$.

5.3.1.2. Preserving Similarities. The success of subspace clustering on the compressed measurements depends fundamentally on how the coding matrix Φ affects the mutual similarities of the spectral signatures. A usual measure of similarity among two vectors is the cosine of the angle between them. Then, assuming that the vectors have unit length, the similarity between two compressed measurements $\mathbf{y}_j = \Phi \mathbf{f}_j$, $\mathbf{y}_{j'} = \Phi \mathbf{f}_{j'}$ is defined as

$$\text{sim}(\mathbf{y}_j, \mathbf{y}_{j'}) = \mathbf{y}_j^T \mathbf{y}_{j'} = \mathbf{f}_j^T \Phi^T \Phi \mathbf{f}_{j'} \quad j \neq j', \tag{50}$$

where $\mathbf{y}_j \in \mathbb{R}^S$ and $\mathbf{f}_j \in \mathbb{R}^L$ correspond to the j-th column of the matrices $\mathbf{Y}$ and $\mathbf{F}$, respectively. If the columns of Φ are normalized, it is possible to decompose the matrix $\Phi^T\Phi$ as

$$\Phi^T\Phi = \mathbf{I} + \Theta, \tag{51}$$

where

$$\Theta_{kk'} = (\mathit{\Phi}_k)^T \mathit{\Phi}_{k'} \quad k \neq k', \tag{52}$$

$\mathit{\Phi}_k$ denotes the k-th column of Φ, and $\Theta_{kk} = 0$. Observe that the matrix Θ collects all the entries outside the diagonal of $\Phi^T\Phi$. Therefore, if $\Theta_{jj'} = 0 \ \forall j, j'$, the matrix $\Phi^T\Phi$ would be equal to $\mathbf{I}$ and the similarities of the spectral signatures would be exactly preserved in the compressed measurements. However, because the matrix Φ has more columns than rows, all the entries of Θ could be mostly small but not equal zero (Kaski, 1998). Considering that a linear mapping such as that in (46) can cause significant distortions in the compressed measurements if $\Phi^T\Phi$ is not approximately $\mathbf{I}$, the proposed coded aperture design should minimize the entries of Θ.

5.3.1.3. Information Acquisition. In order to better discriminate among the classes, new information from the underlying spectral scene should be acquired in each measurement shot, hence the coding patterns should be as different as possible, and the matrix Φ should be full rank. Note that if all coding patterns (rows of Φ) are different, i.e. a spectral band is only sensed once by a unique coding pattern, the values outside the diagonal of $\Phi\Phi^T$ should be zero and the diagonal values will be constants related to the number of nonzero elements in each coding pattern (Arguello and Arce, 2014). Since it is desired that all coding patterns have the same bandwidth Δ, $\Phi\Phi^T$ will approximate to the matrix $\Delta\mathbf{I}$. Then, the matrix $\Phi\Phi^T$ can be decomposed as

$$\Phi\Phi^T = \mathbf{I} + \Lambda, \tag{53}$$

where

$$\Lambda_{ss'} = \phi^s(\phi^{s'})^T \quad s \neq s', \tag{54}$$

and $\Lambda_{ss} = \Delta - 1$. Therefore, the minimization of the entries of Λ should be considered in the coded aperture design.

5.3.2. Optimization Algorithm for Coding Patterns Design. Taking into account the previous considerations, the proposed coding pattern design can be succinctly expressed as the following optimization problem

$$\begin{aligned} \underset{\{\Phi,\lambda_1,\lambda_2,\varphi^s\}}{\arg\min} \quad & f(\Phi) = \|\Phi^T\Phi - \mathbf{I}\|_F^2 + \|\Phi\Phi^T - \mathbf{I}\|_F^2 \\ \text{subject to} \quad & (\phi^s)_k = \delta_{\lfloor \lambda_1^s/k \rfloor}\delta_{\lfloor k/\lambda_2^s \rfloor}\varphi_k^s, \\ & \lambda_2^s = \lambda_1^s + \Delta - 1, \\ & \text{Rank}(\Phi) = S, \end{aligned} \tag{55}$$

for $s = 0, \cdots, S-1$, $k = 0, \cdots, L$ and where $\lambda_1 = [\lambda_1^0, \cdots, \lambda_1^{S-1}]$, $\lambda_2 = [\lambda_2^0, \cdots, \lambda_2^{S-1}]$. This optimization problem can be efficiently solved with the procedure summarized in Algorithm 6. Specifically, lines 2 to 4 generate the first coding pattern ϕ^0 , which has a band structure with a predefined

Algorithm 6 Coding Patterns Design

```
Input  : Number of bands $L$, number of shots $S$, bandwidth
         $\Delta > 0$.
Output: $\Phi = \left[\phi^0, \phi^1, \cdots \phi^{S-1}\right]^T$
1  Initialization :
2  $\Phi^0 \leftarrow \mathbf{1}_{S,L}$
3  Randomly select $\lambda_1^0, \lambda_2^0$ such that $\lambda_2^0 > \lambda_1^0$ with $\lambda_2^0 - \lambda_1^0 + 1 = \Delta$
4  Select $\varphi^s \in \mathbb{R}^L$such that $\varphi_k^s \sim Be(\frac{1}{2})$
5  $\left(\phi^0\right)_k \leftarrow \delta_{\lfloor \lambda_1^0/k \rfloor} \delta_{\lfloor k/\lambda_2^0 \rfloor} \varphi_k^0$ ▷ Banded Structure
6  for $i_t \leftarrow 1$ to $S-1$ do
7      for $i \leftarrow 0$ to $(L-\Delta)$ do
8          $\mathbf{u}_i \leftarrow \sum_{s'=0}^{i_t} \sum_{k=i}^{i+\Delta-1} (\phi^{s'})_k$ ▷ Count sensed spectral bands
9      $A \leftarrow \arg\min_i \mathbf{u}_i$ ▷ $\Lambda$ minimizer set
10     $\lambda_1^{i_t} \sim \mathscr{U}[A]$ ▷ $\lambda_1^{i_t}$ is sampled uniformly at random from set $A$
11     $\lambda_2^{i_t} = \lambda_1^{i_t} + \Delta - 1$
12     $\tilde{\ell} = 0$
13     for $i \leftarrow \lambda_1^{i_t}$ to $\lambda_2^{i_t}$ do
14         $\varphi_i^{i_t} \leftarrow 0$ ▷ zero values in the bandwidth
15         $\mathbf{b}_{\tilde{\ell}} \leftarrow \sum_{s'=0}^{i_t} \prod_{k=(i-1)}^{i} (\phi^{s'})_k$ ▷ Counter of inner products
16         $\tilde{\ell} = \tilde{\ell} + 1$
17     for $\tilde{i} \leftarrow 0$ to $\lfloor \frac{1}{2}\Delta \rfloor$ do
18         $B \leftarrow \arg\min_{\tilde{\ell}} \mathbf{b}_{\tilde{\ell}}$ ▷ $\Theta$ minimizer set
19         $l \sim \mathscr{U}[B]$ ▷ $l$ is sampled uniformly at random from set $B$
20         $\varphi^{i_t}_{\lambda_1^{i_t}+l-1} \leftarrow 1$
21         $\mathbf{b}_l \leftarrow \infty$
22     $\left(\phi^{i_t}\right)_k \leftarrow \delta_{\lfloor \lambda_1^{i_t}/k \rfloor} \delta_{\lfloor k/\lambda_2^{i_t} \rfloor} \varphi_k^{i_t}$
```

bandwidth Δ. Then, lines 6-9 are intended to minimize the number of times a spectral band is sensed. Specifically, the algorithm counts how many spectral bands have been sensed in a certain bandwidth and then λ_1^s is chosen uniformly at random from the set of banded sections with less information A, complying with the criteria of subsection 5.3.1.3. Finally, the algorithm chooses the position in which the inner products are minimized. This is attained by minimizing the elements outside the diagonal, in detail, lines 13-17 count the inner products in the neighborhood. Then, l is sampled uniformly at random from the set that minimizes the inner product (B) to assign a 1 in the coding pattern position l. As observed in Fig. 41, a random design of Φ entries may lead to

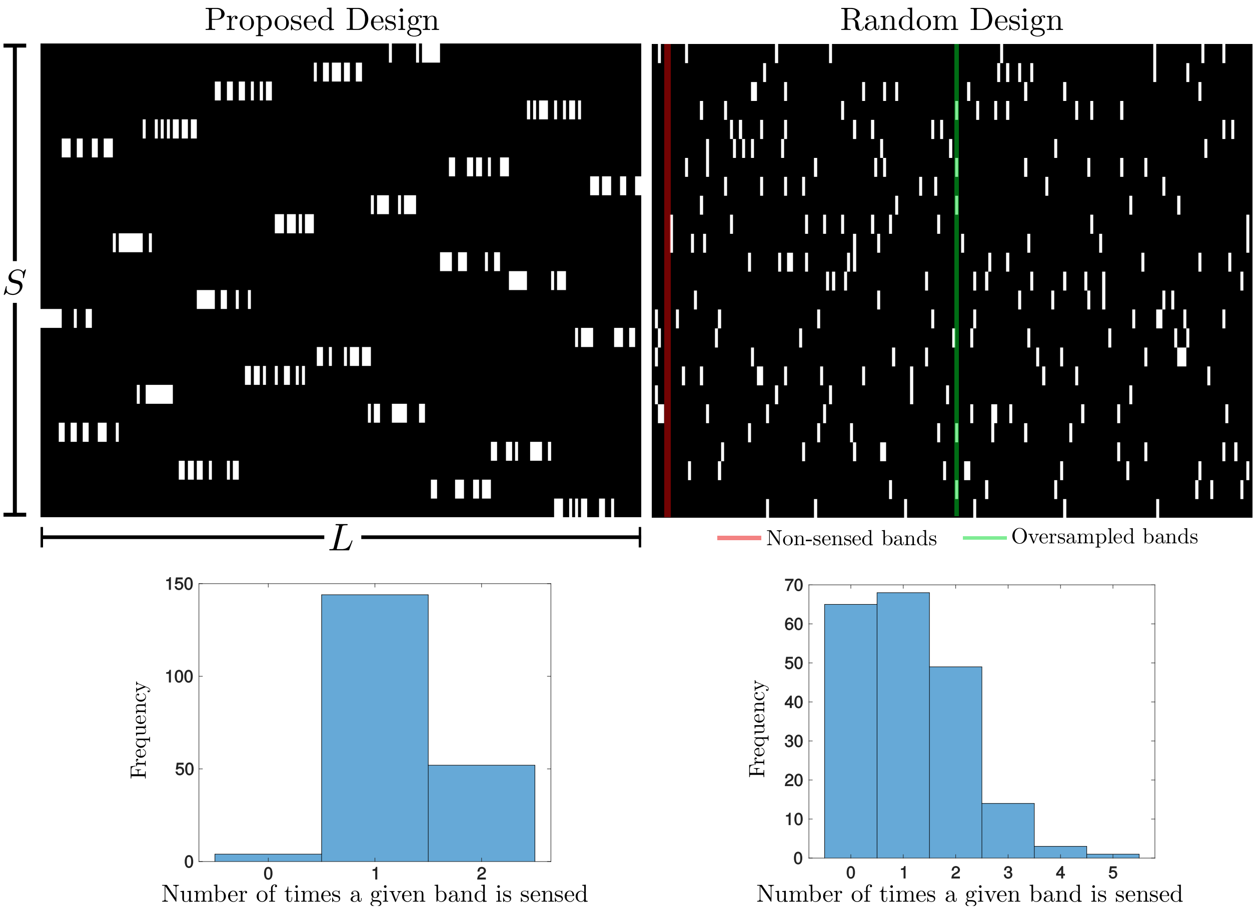


Figure 41. Examples of coding patterns generated by the proposed (left) and random (right) design, respectively.

oversample a subset of spectral bands (green line) while leaving some spectral bands unsampled (red line).

Note that, Algorithm 6 is a greedy algorithm since local choices are made at each stage with the intent of minimizing the objective function. Theorem 1 provides theoretical guarantees that the final result of Algorithm 6 tends to a stationary point.

Theorem 1. Assuming that $\Phi^0 = \mathbf{1}$, where $\mathbf{1}$ is an all-ones matrix, the sequence $\{\Phi^{i_t}\}_{i_t \in \mathbb{N}}$ generated by Algorithm 6 converge to a stationary point. Further, this stationary point satisfies the three equality constraints in (55).

Proof. The proof can be found in Appendix 3. □

5.3.3. Theoretical Results. In the previous section, the optimization algorithm for coding pattern design, proposed in (55), seeks at improving the Φ matrix orthogonality by minimizing $f(\Phi)$. Then, it is expected that, with high probability, the main angles between the projected subspaces are preserved, i.e., the separability among subspaces is preserved. In this section, it is shown that if the SSC ℓ_1 minimization program (described in the next section, see (59)) recovers a subspace-sparse solution for the spectral pixels $\mathbf{F}$, then it will also recover a subspace-sparse solution for the compressed pixels $\mathbf{Y}$.

First observe that the matrix Φ can be decomposed as the product of two matrices as $\Phi = \hat{\Phi}\mathbf{J}$, where $\hat{\Phi} \in \mathbb{R}^{S\times S\Delta}$ is a block rectangular matrix with vectors $\hat{\varphi}^s \in \{0,1\}^{\Delta}$ placed in diagonal form, where $\hat{\varphi}^s$ correspond to the values in the select bandwidth of φ^s for each s. Mathematically, the entries of the s-th row of $\hat{\Phi}$ can be written as

$$(\hat{\Phi}^s)_{\hat{r}} = \begin{cases} (\hat{\varphi}^s)_{\hat{r} \bmod \Delta}, & \text{if } \; s = \lfloor \hat{r}/\Delta \rfloor \\ \\ 0, & \text{otherwise}, \end{cases} \tag{56}$$

for $\hat{r} = 0, \cdots, S\Delta - 1$. Moreover, the matrix $\mathbf{J}$ has one-valued diagonals where each diagonal chooses the position of a specific vector $\hat{\varphi}^s$ within the s-th row of Φ. Taking this into account,

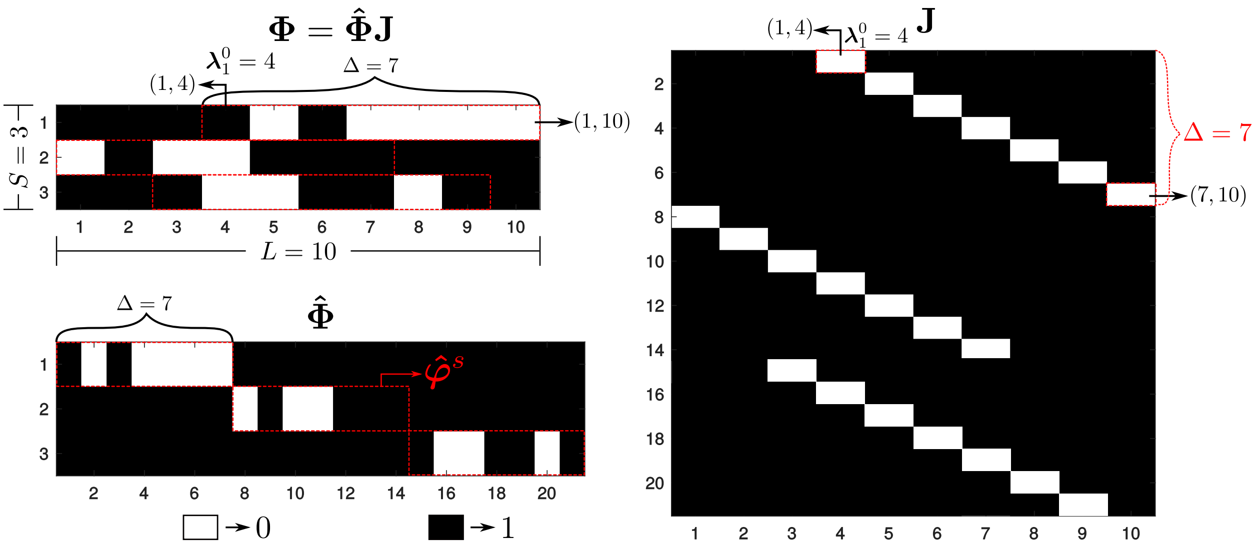


Figure 42. Example of $\Phi, \hat{\Phi}$ and $\mathbf{J}$ structure, using $S = 3$, $L = 10$ and $\Delta = 7$. Note that the first one-valued diagonal in $\mathbf{J}$ will place the first random vector, of length $\Delta = 7$, within columns 4-10 of Φ.

the entries of the $\hat{r}$-th row of $\mathbf{J}$ can be written as

$$(\mathbf{J}^{\hat{r}})_k = \begin{cases} 1, & \text{if } k = \lambda_1^{\lfloor \hat{r}/\Delta \rfloor} + \hat{r} \bmod \Delta \\ \\ 0, & \text{otherwise}, \end{cases} \tag{57}$$

for $k = 0, \cdots, L-1$. Note that the position of one-valued diagonals in $\mathbf{J}$ is determined by the selected cutoff wavelengths $\lambda_1^{\lfloor \hat{r}/\Delta \rfloor}$ obtained using Algorithm 6. In particular Fig.42 illustrates the structure of the matrices $\Phi, \hat{\Phi}$ and, $\mathbf{J} \in \mathbb{R}^{S\Delta \times L}$ for $S = 3, L = 10$ and $\Delta = 7$. Additionally, note that $\mathbf{J}$ can be viewed as a band-selection matrix which selects neighboring bands on the spectral pixel $\mathbf{f}_j$ and groups them in blocks, as $\mathbf{J}\mathbf{f}_j = \bar{\mathbf{f}}_j = \left[(\bar{\mathbf{f}}_j^0)^T, (\bar{\mathbf{f}}_j^1)^T, \cdots, (\bar{\mathbf{f}}_j^S)^T \right]$, with $\bar{\mathbf{f}}_j \in \mathbb{R}^{S\Delta}$ and $\bar{\mathbf{f}}_j^s \in \mathbb{R}^{\Delta}$, hence if $S\Delta \geq L$, then $\text{Rank}(\mathbf{F}) = \text{Rank}(\bar{\mathbf{F}})$, where $\bar{\mathbf{F}} = \mathbf{J}\mathbf{F}$.

In (Elhamifar and Vidal, 2013), the authors of SSC provide recovery conditions under which, for data points $\mathbf{f}_j$ that lie in a union of linear subspaces, the optimization program in (59) recovers subspace-sparse representation of the data, where nonzero elements correspond to points belonging to the same subspace. Particularly, denote $\bar{\mathbf{F}}_d$ as the matrix containing all data points $\bar{\mathbf{f}}_j$ from the subspace $\mathscr{S}_d$ with dimension Q_d. Similarly, denote $\bar{\mathbf{F}}_{-d}$ as the matrix containing data points in all subspaces except $\mathscr{S}_d$. Further, let $\mathbb{W}_d$ be the set of all full-rank submatrices $\tilde{\mathbf{F}}_d \in \mathbb{R}^{S\Delta \times Q_d}$ of $\bar{\mathbf{F}}_d$. From (Elhamifar and Vidal, 2013, Theorem 3), if the condition

$$\max_{\tilde{\mathbf{F}}_d \in \mathbb{W}_d} \tilde{\sigma}_{Q_d}(\tilde{\mathbf{F}}_d) > \sqrt{Q_d} \max_{d \neq d'} \cos(\theta_{d,d'}) \tag{58}$$

for a column-normalized data holds, then for every $\bar{\mathbf{f}}_j$ in the subspace $\mathscr{S}_q$, the ℓ_1-minimization in (59) recovers a subspace-sparse solution. In (58), $\theta_{j,j'}$ is the first principal angle between $\mathscr{S}_d$ and $\mathscr{S}_{d'}$ and $\tilde{\sigma}_{Q_d}(\tilde{\mathbf{F}}_d) = 1/\|(\tilde{\mathbf{F}}_d^T\tilde{\mathbf{F}}_d)^{-1}\ \tilde{\mathbf{F}}_d^T\|_{2,2}$ denotes the Q_d-th largest singular value of $\tilde{\mathbf{F}}_d$. Now, in the following theorem proves that the condition in (58) still holds for $\tilde{\mathbf{Y}}_d = \hat{\Phi}\tilde{\mathbf{F}}_d$.

Theorem 2. Consider a collection of measurements given by (46) drawn from n subspace $\{\mathscr{S}\}_{d=1}^n$ of dimensions $\{Q_d\}_{d=1}^n$. Let $\mathbb{W}_d$ be the set of all full-rank submatrices of the column-normalized data $\tilde{\mathbf{F}}_d \in \mathbb{R}^{S\Delta \times Q_d}$, and $\hat{\Phi}$ the decomposition given by (56) of the matrix obtained from the Algorithm 6. If the number of shots $S \geq Q_d$ then, the following condition holds

$$\tilde{\sigma}_{Q_d}(\tilde{\mathbf{Y}}_d) \geq \frac{(1-\rho)}{k} \sqrt{Q_d} \max_{d \neq d'} \cos(\theta_{d,d'})$$

for some $\rho \in (0,1)$ with probability at least $1 - 2e^{-\rho^2/2}$, where $\tilde{\mathbf{Y}}_d = \hat{\Phi}\tilde{\mathbf{F}}_d$ is a full column rank matrix and $k \geq 1$ is it condition number.

Proof. The proof can be found in Appendix 3. □

Notice that Theorem 2 essentially establishes that the sparse solution can be obtained from the compressed measurements with high probability. In addition, note that because one of the $\hat{\mathbf{\Phi}}$ design criteria is to preserve the similarity (a.k.a, cosine of the angle between two vectors), it is expected that $\theta_{j,j'}$ is also preserved. Therefore, it is possible to infer that if the condition in (58) holds for the spectral pixels $\mathbf{F}$, it will also hold for the compressed pixels $\mathbf{Y}$ with high probability.

5.4. Compressed Sparse Subspace Clustering with Spatial Regularizer

Assuming that compressed pixels of the same land-cover class lie in one independent subspace, subspace clustering methods can be used in order to separate them into the same group or cluster. In particular, SSC builds the similarity matrix, which describes the data points membership, by finding a sparse representation for each compressed pixel whose nonzero elements ideally correspond to points from the same subspace. Given the designed matrix $\mathbf{\Phi}$ and the compressive measurements $\mathbf{Y} = \mathbf{\Phi}\mathbf{F}$, the SSC sparse representation model is formulated as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{R}} \quad & \|\mathbf{Z}\|_1 + \frac{\lambda}{2}\|\mathbf{R}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{Y}\mathbf{Z} + \mathbf{R},\ \operatorname{diag}(\mathbf{Z}) = 0,\ \mathbf{Z}^T\mathbf{1} = \mathbf{1}, \end{aligned} \tag{59}$$

where $\mathbf{1}$ is a one-valued vector, $\mathbf{Z} \in \mathbb{R}^{MN \times MN}$ refers to the representation coefficient matrix and the ℓ_1-norm regularization in this formulation suggests that a sparse representation of a data point finds points from the same subspace. The matrix $\mathbf{R}$ stands for the representation error, and the regularization parameter λ for the sparsity trade-off. The constraint $\text{diag}(\mathbf{Z}) = 0$ is used to eliminate the trivial solution of writing a point as an affine combination of itself and the constraint $\mathbf{Z}^T\mathbf{1} = \mathbf{1}$ ensures that it is a case of affine subspaces (Elhamifar and Vidal, 2009, 2013).

Taking into account that neighboring pixels in a spectral image usually consist of similar materials, a smoothing filter can be applied to the sparse coefficient matrix, in order to reduce the representation error, being able to extract more information from the data (Zhang et al., 2016). Specifically, the smoothing filter will reduce the noise trying to assign the same representation value to neighboring pixels. In this chapter, such spatial information is effectively incorporated into the similarity matrix by first rearranging the 2D sparse coefficient matrix $\mathbf{Z} \in \mathbb{R}^{MN\times MN}$ into a 3D cube $\mathscr{Z} \in \mathbb{R}^{M\times N\times MN}$, treating each coefficient vector as a "pixel" in the cube. Then, considering the method proposed in chapter 2, the smooth filtering was performed by using a 3D median filter with a 3D moving window $\mathscr{W} \in \mathbb{R}^{3\times3\times3}$. Specifically, $\mathscr{W}$ is moved through $\mathscr{Z}$, on each band, pixel by pixel and replacing each value with the median value of neighboring pixels. Finally, the filtered cube $\mathscr{Z}$ is rearranged back to form the matrix $\bar{\mathbf{Z}} \in \mathbb{R}^{MN\times MN}$. Figure 43 depicts the filtering process described above. Using $\bar{\mathbf{Z}}$, a regularization term $\|\mathbf{Z}-\bar{\mathbf{Z}}\|_F^2$ is incorporated in the original SSC optimization program shown in (59). Then, the problem of finding a sparse representation coefficient matrix exploiting the spatial information of the scene is formulated as the following optimization program

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{R},\bar{\mathbf{Z}}} \quad & \|\mathbf{Z}\|_1 + \frac{\lambda}{2}\|\mathbf{R}\|_F^2 + \frac{\alpha}{2}\|\mathbf{Z}-\bar{\mathbf{Z}}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{Y}\mathbf{Z} + \mathbf{R},\ \text{diag}(\mathbf{Z}) = 0,\ \mathbf{Z}^T\mathbf{1} = \mathbf{1}, \end{aligned} \tag{60}$$

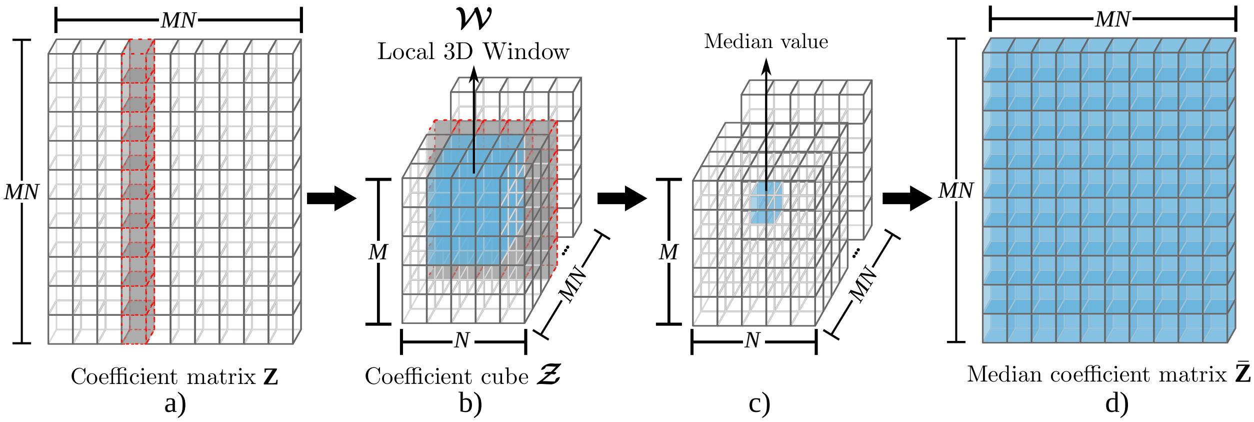


Figure 43. Visual representation of the median filter step, a) Sparse Coefficient matrix **Z**, then it is reshaped as in b) and a median filter is applied to obtain the new values c) and finally it is reshaped to its initial size d).

where α is a regularization parameter denoting the weight of the spatial information in the subspace clustering algorithm. In the subsequent sections, I also refer to the optimization problem in (60) as S-SSC. The minimization in (60) can be efficiently solved by the alternating direction method of multipliers (ADMM), which is described in detail in the Appendix 3. The solution of (60) corresponds to subspace-sparse representation of the data points, which is used by spectral clustering (SC) to infer the clustering of the data. Specifically, the clustering result is obtained by applying SC to the Laplacian matrix induced by the similarity matrix $\mathbf{W} \in \mathbb{R}^{MN \times MN}$ which is defined as $\mathbf{W} = |\mathbf{Z}| + |\mathbf{Z}|^T$ (Elhamifar and Vidal, 2009, 2013). The complete CSI subspace clustering algorithm (CSI-SSC) is summarized in Algorithm 7.

5.5. Experimental Results

The proposed compressed spectral image clustering approach was tested on two real hyperspectral data sets, with different imaging environments. The **Indian Pines** hyperspectral data set was acquired by the AVIRIS sensor from the Northwestern Indian Pines test site in June 1992. The

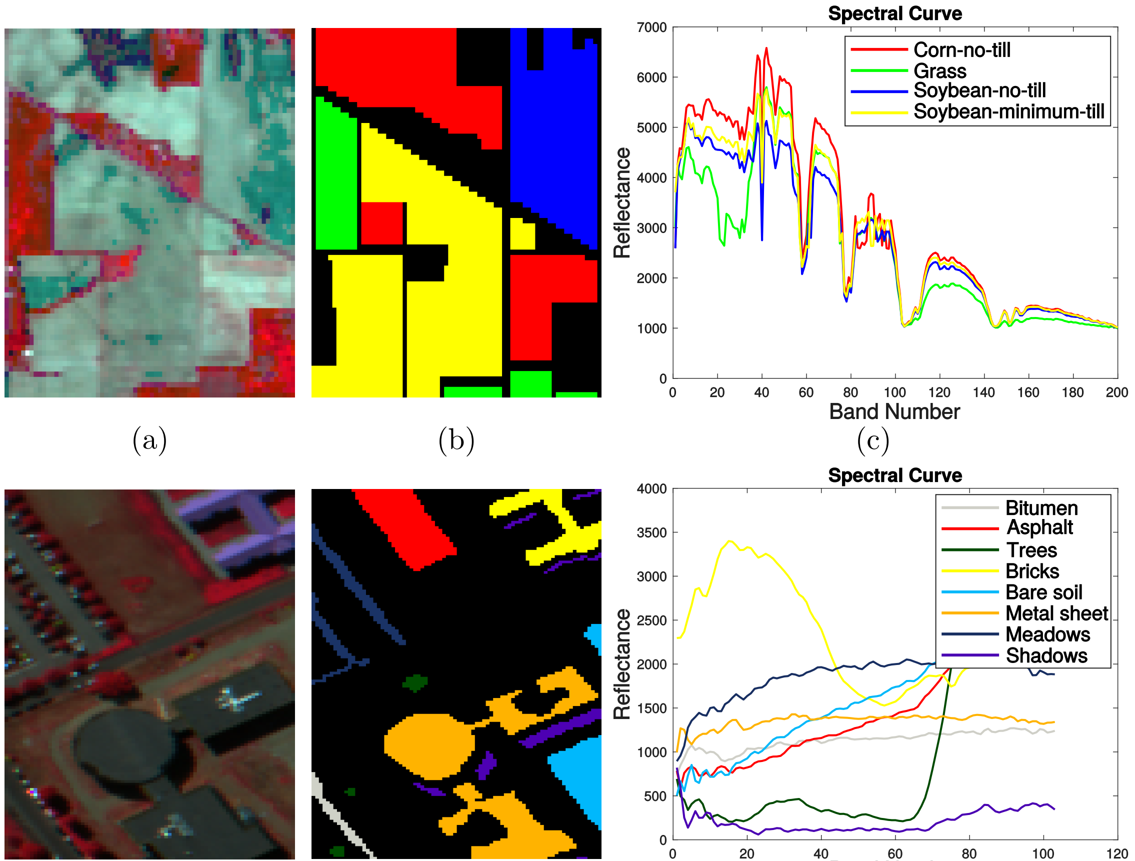
Algorithm 7 Compressive Spectral Subspace Clustering

```
Input  : A set of CSI measurements acquired as $\mathbf{Y} = \Phi\mathbf{F}$, where the coding pattern matrix $\Phi$ is obtained
         with Algorithm 6.
Output: $Y_1, \cdots, Y_\ell$
Solve the sparse optimization problem in (60).
Normalize the columns of $\mathbf{Z}$ as $\mathbf{z}_j \leftarrow \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_\infty}$
Form a similarity graph representing the data points. Set the weights on the edges between the nodes as
    $\mathbf{W} = |\mathbf{Z}| + |\mathbf{Z}|^T$.
Apply SC (Ng et al., 2002) to the similarity graph.
return Segmentation of the data: $Y_1, \cdots, Y_\ell$
```

spatial dimensions of this image are 145 × 145 pixels. A total of 20 water absorption and noisy bands were removed from the original 220 bands, leaving 200 spectral features for the experiment (Zhang et al., 2016). Considering the computational efficiency, a subimage with size 70×70 pixels, which includes four main land-cover classes: corn-no-till, grass, soybeans-no-till, and soybeans-minimum-till, was used in the experiments.

The second scene, **University of Pavia**, was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over Pavia, Northern Italy. The spatial dimensions of the image are 610 × 340 pixels, with 103 bands used in the experiments. A typical area for the test data with a size of 140 × 80 pixels, containing eight main land-cover classes: Bitumen, asphalt, trees, bricks, bare soil, metal sheet, meadows and shadows, was used.

Clustering of these selected images is a challenging task because the spectral signatures of the land-cover classes are very similar and some of the spectral curves are mixed, as observed in Fig 44. (c) and (f). A false-color image and the ground truth for the Indian Pines and Pavia University are also provided in Fig. 44 (a)-(b) and (d)-(e), respectively. In the experiments, the number of clusters was set as a manual input for the subspace clustering algorithm. Furthermore,



(a) (b) (c)

(d) (e) (f)

Figure 44. False-color images, ground truth images and spectral curves of each land-cover classes for AVIRIS Indian Pines (a)-(c) and ROSIS Pavia University (d)-(f), respectively.

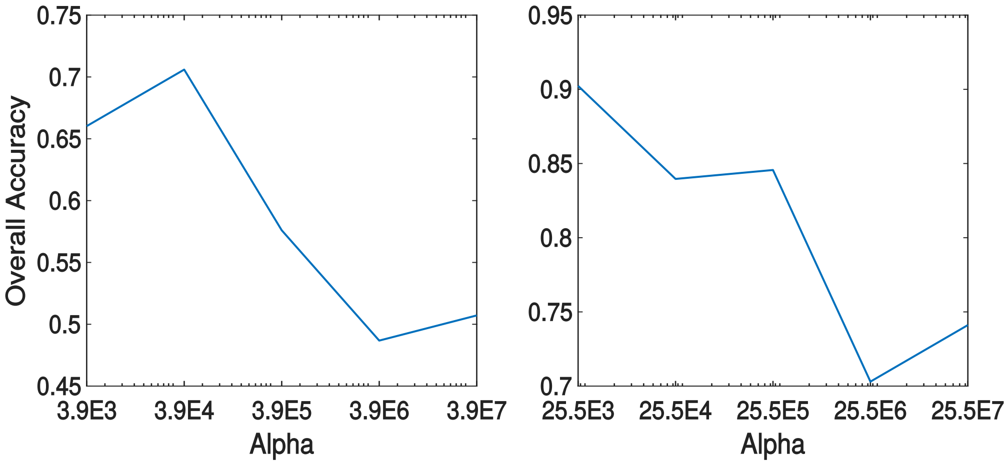
the parameters of the algorithm were manually adjusted. Specifically, the regularization parameter λ, which acts as the trade-off between the sparsity of the coefficient matrix and the magnitude of the noise, was set using the following formulation (Elhamifar and Vidal, 2013):

$$\lambda = \frac{\beta}{\gamma}, \quad \gamma = \min_{j} \max_{j \neq j'} |\mathbf{y}_{j'}^{T} \mathbf{y}_{j}|, \tag{61}$$

where β is the adjustment coefficient, γ is a parameter related to the data set, which can be explicitly determined, and $\mathbf{y}_j$,$\mathbf{y}_{j'}$ are columns of $\mathbf{Y}$. The regularization parameter α in (60) denotes the weight of the spatial information in S-SSC. In order to analyze the sensitivity of α, experiments for each data set were conducted. In these experiments, the coding patterns Φ were generated using Algorithm 6 with $\Delta = 20$ and $S = 25$. Further, white Gaussian noise with a signal-to-noise ratio (SNR) of 25 dB was added to the acquired compressed measurements, simulating the CSI acquisition system noise.

The change in the overall accuracy of the proposed S-SSC algorithm corresponding to different α values, with other parameters fixed, is shown in Fig. 45. It can be seen that the precision changes significantly with different values of α, which suggests that the spatial information plays a very important role in the clustering process. For simplicity this parameter was fixed for all experiments: $\alpha = 3.9 \times 10^4$ for the Indian Pines image and $\alpha = 25.5 \times 10^5$ for the University of Pavia image. Similarly, the parameter λ for all the experiments is calculated using (61) with $\beta = 1000$. Since the structure of the acquired compressed measurements is determined by the generated coding pattern, the γ parameter is determined at the beginning of each experiment using (61).

In the next experiments, the random coding patterns are generated from a Bernoulli distribution $\Phi \sim B_e(p)$, using (47) with $p = \Delta/L$ in order to use a similar transmittance (the number of nonzero elements) to the designed coding patterns. The results presented throughout the document are the average of five experiments, each with a different set of coding patterns. A MatLab implementation of Algorithm 6 and some experiments presented in the next sections can be found in the repository list, see Appendix 4.



(a) (b)

Figure 45. Analysis of parameter α: Change in the overall accuracy with various values of α. (a) Indian Pines image, (b) University of Pavia image.

5.5.1. Similarity Preservation. In this experiment, the performance of the proposed coding pattern design is tested. Specifically, this experiment is intended to show how well the similarity between two spectral signatures is approximately preserved when the designed coding patterns are used. For this experiment, 100 spectral pixels from each image are randomly chosen and then are compressed using both random and the designed coding pattern matrix Φ, generated with parameters $\Delta = 20$ and $S = 25$. Then, all the spectral signatures and their compressed versions are normalized to have unit ℓ_2 norm, i.e., $||\mathbf{f}||_2 = 1$ and $||\mathbf{y}||_2 = 1$. Using the definition of similarity, presented in (50), the absolute error is calculated as

$$|\mathrm{sim}(\mathbf{f}_{j_k}, \mathbf{f}_{j'_k}) - \mathrm{sim}(\mathbf{y}_{j_k}, \mathbf{y}_{j'_k})| = |\mathbf{f}_{j_k}^T \mathbf{f}_{j'_k} - \mathbf{y}_{j_k}^T \mathbf{y}_{j'_k}|, \tag{62}$$

where $j, j' \in \{0, 1, \cdots, MN\}$, index a spectral signature chosen at random among MN possibilities, and $k = 1, \cdots, 100$. Fig. 46 shows the obtained results for the two spectral images. As observed, the absolute errors obtained with the designed coding patterns are significantly smaller than those obtained with a random-designed matrix Φ. Therefore, the proposed coding pattern design approximately preserves the similarities among the spectral signatures after the scene projection.

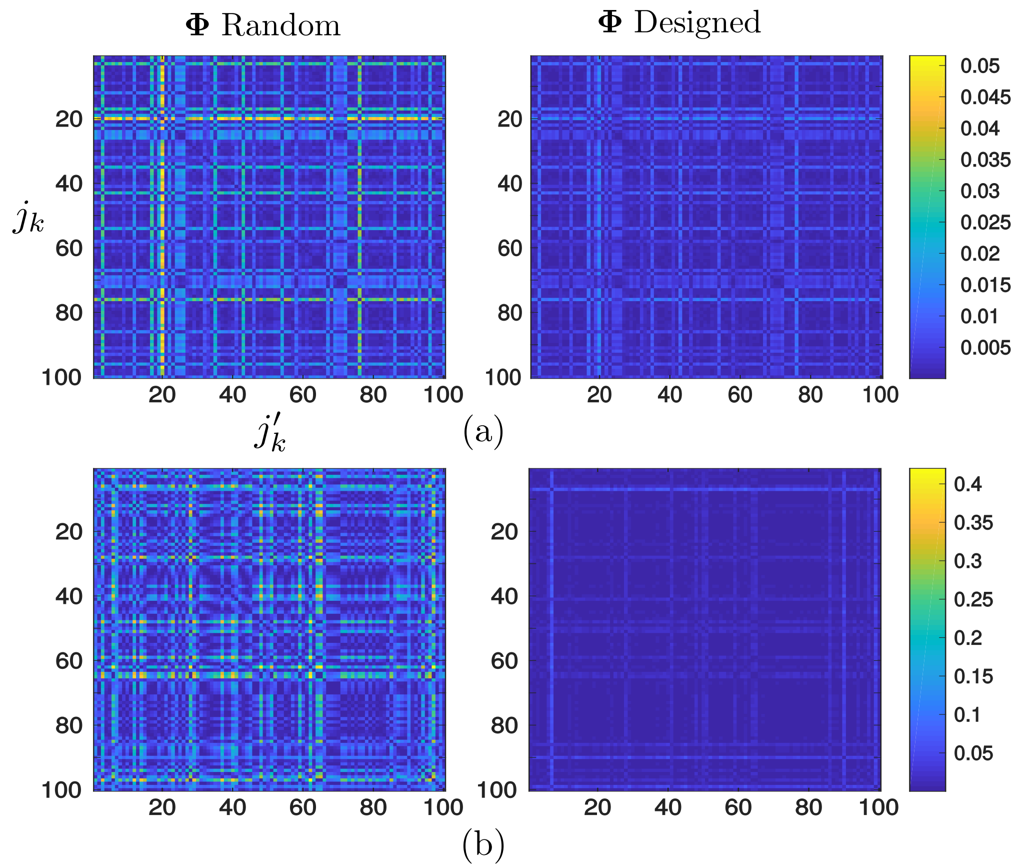
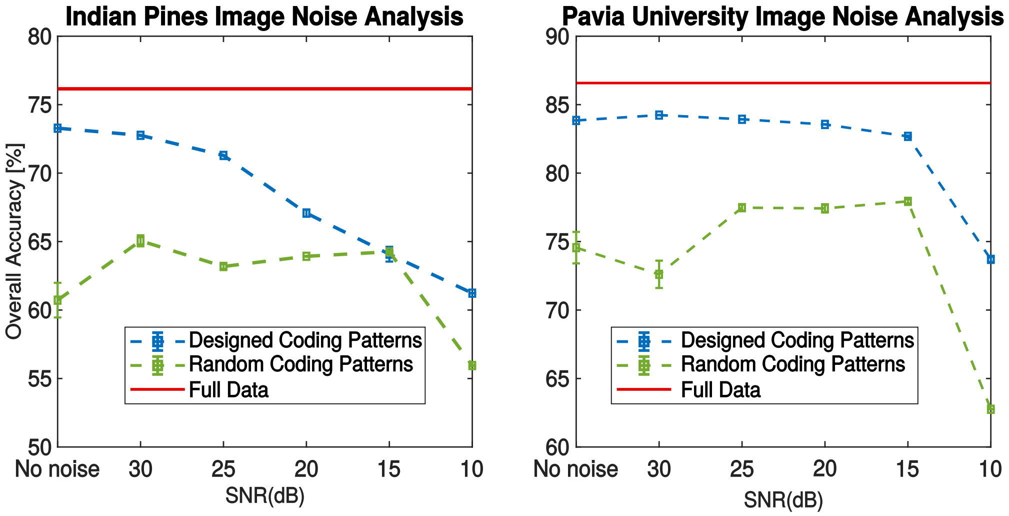


Figure 46. Absolute error between the spectral signatures similarities and the compressed measurements similarities acquired with the random and designed coding patterns for (a) Indian Pines and (b) Pavia University datasets.

5.5.2. Noise Analysis. It is important to note that the acquired measurements described by (46) are noise free. However, in real CSI architectures, the acquired compressed measurements



(a) (b)

Figure 47. Overall clustering accuracy as a function of the aggregated noise using the two types of coding patterns.

are contaminated with noise due to the physical limitations of the sensor. Therefore, (46) should be rewritten as $\mathbf{Y} = \Phi\mathbf{F} + \Omega$, where $\Omega \sim N(0, \sigma^2)$ represents the noise of the system. In order to analyze the impact of noise, different experiments varying SNR are performed. Fig. 47 presents the classification accuracy results obtained with the designed and the random coding patterns for the two hyperspectral data sets. Additionally, the overall clustering accuracy achieved when using the spectral image data cube (Full data) as input for the optimization problem in (60) is shown as reference. The designed patterns are generated with the fixed parameters $\Delta = 20$ and $S = 25$.

The results show that the proposed coding pattern design outperforms the random-generated patterns even when white Gaussian noise is added to the CSI measurements. As expected, when SNR decreases the clustering overall accuracy is affected. However, the accuracy curve, obtained with the proposed coding patterns, achieves a slow decrease rate in comparison with the random coding patterns.

5.5.3. Analysis of the Coding Pattern Design Parameters . The parameters S and Δ determine the structure of the proposed coding pattern design. In order to analyze how the accuracy is affected by those parameters, experiments for each pair (Δ, S) were performed varying each parameter. In this experiment, the noise added to the system was fixed to 25 dB of SNR. Fig. 48 (a) and (c) present the obtained overall accuracy and Fig. 48 (b) and (d) show the variance of the obtained accuracy for each spectral image, respectively.

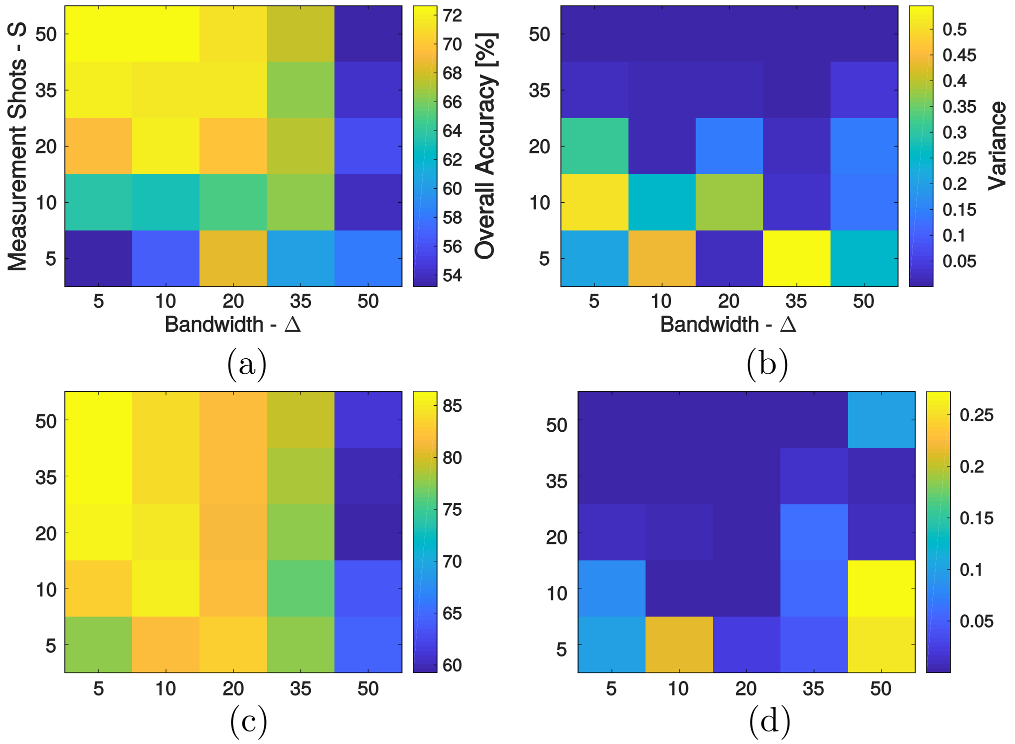


Figure 48. Analysis of the coded aperture design parameters. (a) and (c) show the overall accuracy varying the bandwidth Δ and the measurement shots S for the hyperspectral datasets Indian Pines and Pavia University, respectively. (b) and (d) present the variance of the obtained accuracy.

Note that the number of shots S determines the ambient space of the projected vectors. It can be shown that, when the ambient dimension after random projection is sufficiently large, the distance between two subspaces almost remains unchanged after random projection (Li and Gu, 2017). This behavior is observed in the presented results where increasing the measurement shots leads to better classification accuracies. In addition, notice that the sparsity of the projection matrix Φ and the selected bandwidth Δ are directly related. Specifically, denote the sparsity of the matrix Φ as $K = 1/\kappa$, where $\kappa \approx \frac{2L}{\Delta}$. As described in (Li et al., 2006), sparse random projections are robust for $\kappa \approx \sqrt{L}$. However, as κ increases (decreasing Δ), variances for sparse random projections will also increase and large errors are expected. On the other hand, by decreasing κ (increasing Δ), the entries outside diagonal of Θ will increase leading to significant distortions in the acquired compressed measurements. In all the other experiments I fixed $S = 25$ and $\Delta = 10$ since, as observed in Fig. 48, the OA and variance stabilize when $S > 20$ in both hyperspectral datasets. Note that $S = 25$ yields a compression of 87.5 % and 75.7% for Indian Pines and University of Pavia data set, respectively.

5.5.4. Visual Maps and Quantitative Results. In order to validate the clustering performance of the proposed coding pattern design, cluster maps and quantitative results are presented for the two hyperspectral scenes. In all the experiments, the coding patterns were generated for parameters $\Delta = 20$ and $S = 25$. Further, white Gaussian noise with 25 dB of SNR was added to the acquired compressed measurements. The S-SSC algorithm described in Section 5.4 was used to perform the clustering on both compressed measurements and the complete spectral data cube (Full-data). Additionally, the results obtained with the original sparse subspace clustering algorithm (SSC), when the complete spectral data cube is used as input (Full-data-SSC), are also shown for comparison purposes. Note that, in the experiments, “Proposed-design” refers to the clustering results obtained when the S-SSC algorithm is directly applied on the compressed measurements acquired with the proposed pattern designs. Similarly, “Random-design” corresponds to the results obtained when the S-SSC is directly applied to the compressed measurements acquired with the random patterns. Finally, “Full-data” refers to using the proposed S-SSC method with the complete spectral image, i.e., no compression was performed.

Fig. 49 presents the obtained visual clustering results on Indian Pines. The quantitative evaluations corresponding to the accuracy for each class, overall accuracy (OA), average accuracy(AA) and Kappa coefficients are shown in Table 22, where all values are given in percentage. Similarly, Fig. 50 and Table 23 present the visual clustering results and quantitative evaluation on the Pavia University, respectively. In the tables, the optimal value of each row is shown in bold and the second-best result is underlined. From Tables 22 and 23, it can be clearly observed that the proposed clustering approach (S-SSC), using the proposed coding patterns, provides comparable results to applying clustering directly on the full spectral data cube. Furthermore, it is observed from the visual clustering maps that, although the reconstruction is avoided, the results obtained with the proposed coding patterns are very similar to the results obtained with the Full-data-S-SSC. This behavior was expected since the proposed coding patterns approximately preserve the similarities among spectral pixels, as it was theoretically shown in section 5.3.3 and experimentally verified in section 5.5.1.

5.5.5. Clustering Time and Spectral Image Reconstruction. In this section, the effectiveness of applying clustering directly on the compressed domain is evaluated. For this purpose, a subimage of ROSIS Pavia university dataset with the size of 64×64 pixels, which contains four

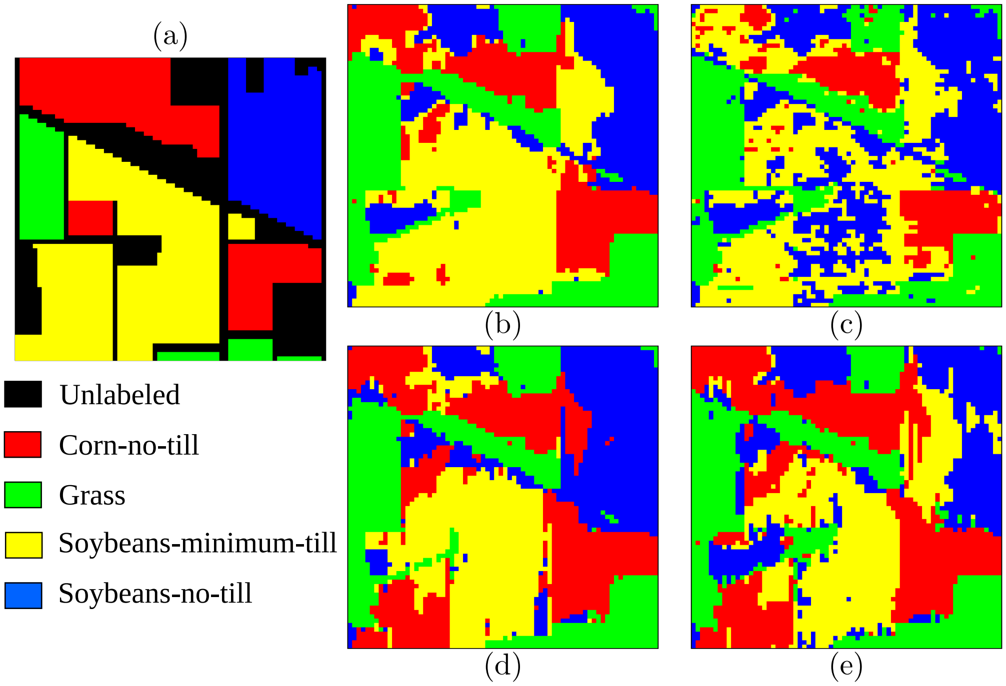


Figure 49. Visual clustering results on AVIRIS Indian Pines image. (a) Ground truth. (b) Full-data, (c) Full-data-SSC, (d) Proposed-design and (e) Random-design.

Table 22

Quantitative evaluation of the different clustering results for the AVIRIS Indian Pines Image.

Class	Random-design	Proposed-design	Full-data-SSC	Full-data
Corn-no-till	**73.13**	70.45	48.96	66.77
Grass	95.25	**100**	98.60	**100**
Soybeans-no-till	52.87	**88.80**	70.63	69.54
Soybeans-minimun-till	55.29	60.52	59.23	**80.05**
OA	63.83	73.07	62.62	**76.16**
AA	69.14	**79.94**	69.35	79.09
Kappa	49.26	62.65	47.58	**65.89**

land-cover classes: asphalt, meadows, trees, and bricks was used, see Fig. 51 (a). CSI measurements were acquired using the random and proposed coding patterns. Then, 300 iterations of the gradient projection for sparse reconstruction algorithm (GPSR) (Figueiredo et al., 2007) were used

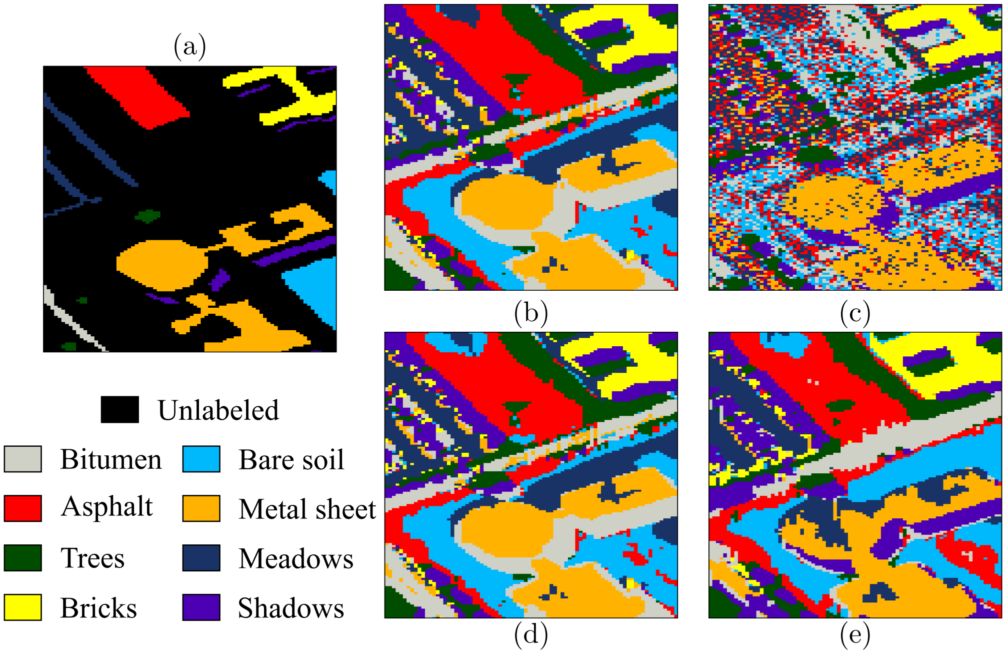


Figure 50. Visual clustering results on ROSIS Pavia University image. (a) Ground truth. (b) Full-data, (c) Full-data-SSC, (d) Proposed-design and (e) Random-design.

Table 23

Quantitative evaluation of the different clustering results with the AVIRIS Pavia University Image.

Class	Random-design	Proposed-design	Full-data-SSC	Full-data
Bitumen	18.60	88.37	0	**90.70**
Asphalt	71.37	67.25	33.84	**80.26**
Trees	90.38	88.46	**100**	90.38
Bricks	**100**	99.68	99.68	99.68
Bare Soil	46.78	61.40	36.26	**66.67**
Metal sheet	82.90	**97.73**	91.00	**97.73**
Meadows	91.16	**100**	55.02	**100**
Shadows	**99.48**	24.35	98.45	24.35
OA	78.72	83.81	71.45	**86.58**
AA	75.09	78.41	64.28	**81.22**
Kappa	72.63	78.89	62.95	**82.50**

to reconstruct the underlying spectral scene. Fig. 51 presents the obtained visual clustering results on the selected subimage. In Table 24, the time, quality of the reconstruction and the result of

Table 24

Time and classification accuracy when clustering the reconstructed spectral image and the CSI measurements

	Reconstruction		**No Reconstruction**	
	Random Patterns	Proposed Patterns	Random Patterns	Proposed Patterns
CSI Recovery				
PSNR [dB]	27.92	**34.38**	-	-
Time [s]	**28.56**	26.23	-	-
Subspace Clustering				
Asphalt	**100**	**100**	**100**	**100**
Meadows	71.65	**99.07**	69.78	97.76
Trees	56.05	63.06	18.85	**65.35**
Bricks	88.04	**99.24**	98.68	98.83
OA	83.43	**94.88**	81.70	94.65
AA	78.94	90.34	71.89	**90.45**
Kappa	77.35	**92.90**	74.74	92.60
Solving (60) [s]	16.62	**15.78**	16.28	10.15
SC Time [s]	118.25	**101.65**	106.11	93.55
Total Time [s]	163.43	143.66	122.39	**103.70**

clustering for the types of coding patterns, are shown. From this table it is possible to observe that the proposed coding pattern design shows a gain of up to 6 dB in terms of peak signal-to-noise ratio (PSNR) in comparison with the random patterns. Further, it can be seen that the designed coding pattern not only improves the reconstruction quality but also the clustering result for the two scenarios, i.e., when the subspace clustering is applied after reconstruction and when it is applied directly on the compressed data. Note that, the total clustering time of the reconstructions is greater than the time of directly applying clustering on the compressed measurements because it takes into account the reconstruction time. In the simulations, when using the proposed coding pat-

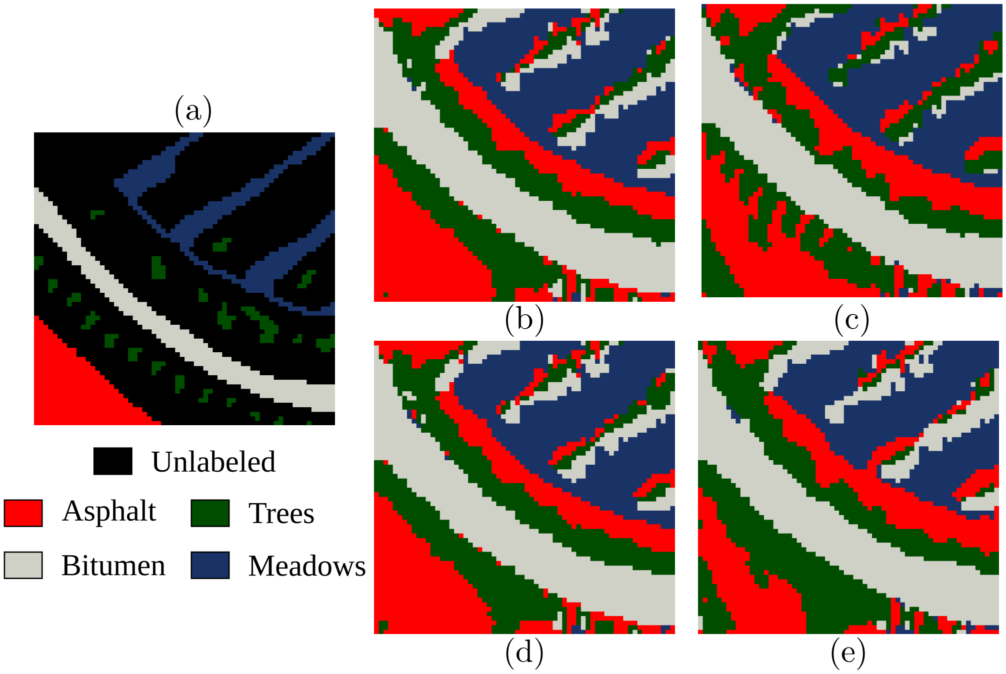


Figure 51. Visual clustering results on a 64×64 region of Pavia University. (a) Ground truth. (b), (c) clustering results on reconstructed images using the random and proposed coding patterns, respectively. (d),(e) clustering results by directly using the compressed measurements acquired with the random and proposed coding patterns respectively.

terns, the total clustering time of the reconstructed spectral image was 143.66 [s], while applying clustering directly on the compressed measurements takes only 103.70 [s], obtaining very similar classification results.

5.5.6. Comparison with other CSI sensing approaches. In this section, the proposed CSI sensing design was compared with two different CSI approaches described in (Martín and Bioucas-Dias, 2016) and (Xing et al., 2012). In order to perform such comparison, some considerations were taken into account. Specifically, (Martín and Bioucas-Dias, 2016) uses the measurement matrices $\mathbf{A} \in \mathbb{R}^{S_1 \times L}$ and $\mathbf{B}_k \in \mathbb{R}^{S_2 \times L}$ whose elements are independently drawn at random from Gaussian, Rademacher, or Bernoulli distribution. The measurements $\mathbf{Y}_a = \mathbf{AF}$ obtain S_1 projections per pixel and $\mathbf{y}_{b,k} = \mathbf{B}_k\mathbf{f}_{i_k}$ for $k = 1, \cdots, n_v$, obtains S_2 measurements per sample pixel n_v. Because the main goal of this paper is to apply clustering directly from compressive measurements and, as explained in section 5.2, the similarity among two spectral pixels decrease when they are encoded (projected) differently, here it is only considered $\mathbf{Y}_a$ with $S_1 = S$ in order to apply clustering such that fair comparisons were performed. Furthermore, Gaussian distribution was chosen, since Φ with $0,1$ entries was already considered in previous sections.

To perform a comparison with the work in (Xing et al., 2012), which assumes that only a few voxels of the image are known (incomplete hyperspectral imagery), I considered that all spectral signatures had the same number of missing voxels in order to be able to apply subspace clustering directly on the compressed measurements. Then, two different scenarios were evaluated for incomplete images: the first is when some voxels are randomly removed in each spectral signature; the second is when complete spectral bands are removed from the image i.e. $\mathbf{y}_i = \mathbf{A}_i\mathbf{f}_i$, where $\mathbf{A}_i \in \{0,1\}^{S\times L}$ and the rows of $\mathbf{A}_i$ are all zero except one corresponding to the observed voxels. In the results, I refer to these scenarios as *randomly-removed-voxels (RRV)* and *randomly-removed-bands (RRB)*, respectively.

In the experiments I used $S = 25$ and white Gaussian noise with 25 dB of SNR was added to the measurements. The results shown in Table 25 and 26 are the average of five experiments. All the clustering results were obtained using the proposed subspace clustering method (S-SSC). At each experiment, different measurements were obtained. From Tables 25 and 26 it can be clearly observed that the adopted 3D-CASSI sensing scheme, using the designed coding patterns, outperforms the CSI compression schemes proposed in (Martín and Bioucas-Dias, 2016)-(Xing et al., 2012).

Table 25

Comparison of clustering results for the AVIRIS Indian Pines Image when using the proposed CSI sensing design and other CSI sensing approaches.

Class	Gaussian	RRV	RRB	Proposed-design
Corn-no-till	27.36	45.01	52.76	**70.45**
Grass	57.21	86.70	**100**	**100**
Soybeans-no-till	34.18	42.76	72.65	**88.80**
Soybeans-minimun-till	47.48	36.31	53.66	**60.52**
OA	40.06	45.14	61.95	**73.07**
AA	41.56	52.70	69.77	**79.94**
Kappa	16.34	24.02	47.43	**62.65**

Table 26

Comparison of clustering results for the AVIRIS Pavia University Image when using the proposed CSI sensing design and other CSI sensing approaches.

Class	Gaussian	RRV	RRB	Proposed-design
Bitumen	57.91	12.10	48.37	**88.37**
Asphalt	43.99	68.63	**72.62**	67.25
Trees	41.54	84.62	87.69	**88.46**
Bricks	96.66	99.29	99.55	**99.68**
Bare Soil	34.15	46.08	34.85	**61.40**
Metal sheet	68.82	97.47	97.67	**97.73**
Meadows	88.92	99.12	99.76	**100**
Shadows	**100**	75.65	44.87	24.35
OA	66.78	82.73	81.66	**83.81**
AA	66.50	71.36	73.17	**78.41**
Kappa	59.22	77.35	76.09	**78.89**

5.6. Conclusions

This chapter presented a coding pattern design, which aims at preserving the spectral signatures separability as much as possible after the scene projection. This design allows to improve clustering results on the compressive domain. Furthermore, a subspace clustering algorithm which takes into account the spatial information of the spectral images in order to correct the representation bias and obtain a more accurate representation coefficient matrix was also developed. The coding pattern design and the spectral image subspace clustering approach were validated trough several experiments. The Indian pines hyperspectral remote sensing scene, from the AVIRIS sensor, and the Pavia University, from the ROSIS sensor, were used in the experiments. In general, the results show that performing the clustering directly with the compressed measurements provides similar accuracy results, in a lesser time, compared with those provided by performing the clustering on the full 3D spectral image, when a properly designed coding pattern was used. Particularly, a maximum difference of just 4% in terms of overall accuracy was observed when comparing the clustering results obtained by the full 3D data with those achieved using CSI measurements acquired with the proposed coding pattern design.

6. Applications in CSI (II): Single-Pixel Camera Sensing Matrix Design for Hierarchical Compressed Spectral Clustering

Part of this chapter has been adapted from the conference paper (Hinojosa et al., 2019a) (published).

Table 27
Notation used in Chapter 6.

Notation	**Definition**
M	Number of rows in the HSI.
N	Number of columns in the HSI.
L	Number of spectral bands of the HSI.
K	Number of measurement shots.
N_{seg}	Number of measurement segments.
$\mathbf{H} \in \mathbb{R}^{K\times MN}$	Sensing matrix.
$\mathbf{W} \in \{-1,1\}^{K\times K}$	Hadamard matrix.
$\Delta \in \mathbb{R}^{K\times MN}$	Decimation matrix.
$\mathbf{f}_l \in \mathbb{R}^{MN}$	l-th spectral band in vector form.
$\mathbf{y}_l \in \mathbb{R}^{K}$	Compressed measurement of band l.
$\bar{\mathbf{f}}_l \in \mathbb{R}^{K}$	Features from the l-th spectral band.
$\mathbf{p}^e \in \mathbb{Z}^{n_e}$	Vector with indices of all pixels from the e-th segment.
$\bar{\mathbf{F}} \in \mathbb{R}^{N_{seg}\times L}$	Feature matrix.
$\mathbf{G} \in \mathbb{R}^{N_{seg}\times N_{seg}}$	k-nearest neighbor graph (Von Luxburg, 2007a).
$\mathbf{C} \in \mathbb{Z}^{M\times N}$	Final clustering map obtained via spectral clustering (Von Luxburg, 2007a).

6.1. Introduction

In this chapter, I adopted the strategies proposed in the previous chapters to develop a method for coding patterns design and subspace clustering of compressed measurements acquired with a single pixel camera (SPC). The SPC architecture compress each spectral band in a single pixel measurement, see section 1.6 for details. Consequently, the cost of storing, and processing SPC measurements is significantly reduced in comparison with the 3D-CASSI employed in the previous chapter. As described in section 1.4, recovering the compressed measurements requires to solve an expensive optimization problem that presents slow convergence, which limits CSI in terms of time (Hinojosa et al., 2015; Bacca et al., 2019b).

However, as discussed in the previous chapter and in Appendix 1, post-processing tasks such as classification (Xu et al., 2018), clustering (Hinojosa et al., 2018a), and unmixing (Vargas et al., 2018) can be performed directly on the compressive domain bypassing the signal recovery stage. For instance, the previous chapter (5) shows that it is possible to employ sparse subspace clustering directly from 3D-CASSI measurements acquired with a carefully designed matrix, which preserves the subspace structure.

In this chapter, a hierarchical approach is proposed to design a sensing matrix of the single pixel camera (SPC) such that clustering features are extracted directly from the acquired compressed measurements. Specifically, at each level of the hierarchy, a sensing matrix is designed as the product of a Hadamard matrix and a decimation matrix. This decomposition allows obtaining a set of features directly from the compressed measurements exploiting the properties of the Hadamard matrix.

In the proposed approach, the decimation matrix at a given level is designed to group similar features of the previous level in segments. Therefore, the composite sensing matrix has less sampling vectors and it is intended to provide fewer and more compact features than those obtained in the previous level. In this thesis, the decimation matrix is obtained using the k-means algorithm, however, other alternatives can be employed. Finally, a complete segmentation map is obtained by performing majority voting on the partial clustering results obtained using the set of features of each hierarchy level.

6.2. Sensing Matrix Design

As described in section 1.6, the SPC multi-shot scheme for each band l is expressed in matrix form as

$$\mathbf{y}_l = \mathbf{H}\mathbf{f}_l, \tag{63}$$

where $\mathbf{H} \in \mathbb{R}^{K \times MN}$, K is the number of shots, each row of $\mathbf{H}$ is the vector form of the coded aperture used on that particular shot, and $\mathbf{y}_l = [y_1, \cdots y_K]^T$.

Since several applications require fast spectral image reconstructions, recent works have focused on developing fast, although with low precision, recovery methods (Garcia et al., 2018; Sankaranarayanan et al., 2015). In particular, taking into account the structure of Hadamard matrices, the work in (Sankaranarayanan et al., 2015) proposes to design the sensing matrix for each band $\mathbf{H}$ as

$$\mathbf{H} = \mathbf{W}\Delta, \tag{64}$$

where $\mathbf{W} \in \{-1, 1\}^{K \times K}$ is a Hadamard matrix, and $\Delta \in \mathbb{R}^{K \times MN}$ is a decimation matrix.

Recently, a fast spectral image recovery method was introduced in (Garcia et al., 2018), where authors proposed to design Δ by obtaining superpixels from an RGB image which was acquired as side information. Specifically, the method named *FMR* takes advantages of the fact that the inverse of a Hadamard matrix is its transposes and perform a fast low-resolution reconstruction from the compressed measurements $\mathbf{y}_l$ and for each spectral band ($\forall \mathbf{f}_l$) as

$$\tilde{\mathbf{f}}_l = (1/K)\Delta^T \mathbf{W}^T \mathbf{y}_l = (1/K)\Delta^T \mathbf{W}^T W \Delta f_l \approx \mathbf{f}_l. \tag{65}$$

Note that, instead of performing the complete reconstruction, it is possible to directly extract features from the compressed measurements. In particular, features from the l-th band can be obtained as

$$\bar{\mathbf{f}}_l = \mathbf{W}^T \mathbf{y}_l = \Delta f_l, \tag{66}$$

where $\bar{\mathbf{f}}_l$ contains the average spectral information of pixels grouped in segments given by the structure of the downsampling matrix Δ. It is important to note that, similar as in previous chapter, in the following sections I assume that $K = N_{seg}$.

6.3. Proposed CSI Clustering

Taking into account, the sensing matrix construction approach presented in Eq. 64, it is possible to design the downsampling matrix Δ to efficiently extract clustering features from the compressed measurements. This section presents an unsupervised approach to perform both, Δ matrix design and clustering of the spectral image pixels by directly using the compressed measurements. The complete workflow of the proposed approach is depicted in Fig. 52.

6.3.1. Downsampling Matrix Design. In general, the binary matrix $\Delta \in \mathbb{R}^{N_{seg} \times MN}$ groups the $M \times N$ spectral pixels in N_{seg} segments, such that each component of the vector $\bar{\mathbf{f}}_l = \Delta f_l$ contains the average spectral information of pixels grouped in one segment. More formally, denote $\mathbf{p}^e$ as the vector of size n_e containing the indices of all pixels belonging to the e-th segment. Then,

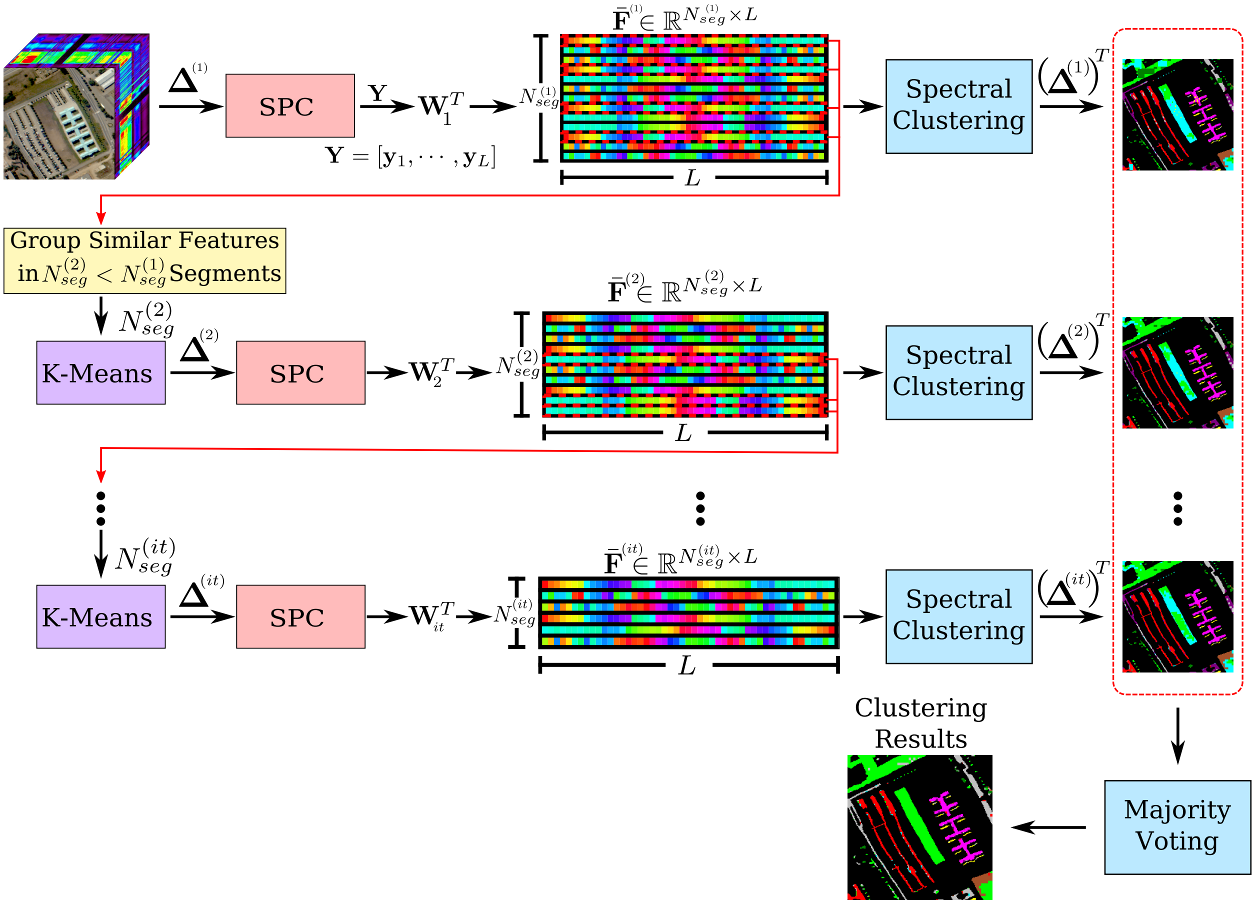


Figure 52. Workflow of the proposed SPC sensing matrix design and compressed measurements clustering.

the nonzero values of the $e-th$ row of Δ, denoted in vector form as $(\delta_e)^T$, are determined by the entries of $\mathbf{p}^e$ and the value of n_e as follows:

$$(\delta_e)^T_{(\mathbf{p}^e)_j} = \frac{1}{n_e}, \quad \text{for } j = 1, \cdots, n_e, \tag{67}$$

where $(\delta_e)^T_{(\mathbf{p}^e)_j}$ denotes the position in δ_e indexed by the $j-th$ entry of the vector $\mathbf{p}^e$.

The main idea of the proposed design of Δ is to group pixels such that similar spectral infor-

Algorithm 8 Downsampling Matrix Design

```
Input  : $N_{seg}$, $\bar{\mathbf{F}}$
Output: $\Delta$
$k_{idx} \leftarrow k\text{-means}(\bar{\mathbf{F}}, N_{seg})$ ▷ $k_{idx}$ contains the segment labels
$\Delta \leftarrow$ zeros($N_{seg}$, length($k_{idx}$))
for $e \leftarrow 1$ to $N_{seg}$ do
    $\mathbf{p}^e \leftarrow$ find($k_{idx} = e$)
    $n_e \leftarrow$ length($\mathbf{p}^e$)
    for $j \leftarrow 1$ to $n_e$ do
        $(\delta_e)^T_{(\mathbf{p}^e)_j} = \frac{1}{n_e}$ ▷ Update each row of $\Delta$
```

mation is taken into account. As only the compressed measurements are available, it is proposed to design Δ in an iterative hierarchical fashion such that N_{seg} decreases (more pixels are grouped in one segment) in each iteration and the previous design of Δ is used to redefine the new segments, see Algorithm 8.

In the first iteration, Δ is designed such that all the pixels are grouped in ${N_{seg}}^{(1)}$ square segments, which are determined without prior information. Once the compressed measurements are acquired, the feature vector $\bar{\mathbf{f}}_l$ is obtained for each spectral band l, hence the feature matrix $\bar{\mathbf{F}}$ is constructed as

$$\bar{\mathbf{F}} = \left[\bar{\mathbf{f}}_1, \cdots, \bar{\mathbf{f}}_L\right] \in \mathbb{R}^{N_{seg} \times L}, \tag{68}$$

where the rows contain the average spectral information of each segment. For the next *it* iterations, N_{seg} is selected as ${N_{seg}}^{(it)} < {N_{seg}}^{(it-1)} < \cdots < {N_{seg}}^{(1)}$ and the k-means algorithm is used to find the new segments with inputs $\bar{\mathbf{F}}$ and N_{seg}, as the data matrix and the number of desired clusters, respectively. Then, the $\mathbf{p}^e$ vectors are built for each segment e using the output of k-means, and the new Δ matrix is obtained using the Eq. 67.

6.3.2. Data Clustering. At each iteration of the main algorithm, the downsampling matrix Δ is constructed and it is used to obtain a partial clustering of the pixels using the Spectral Clustering method. Since, at each iteration, the number of segments N_{seg} is decreased, this approach can be seen as a multiscale clustering of pixels. Furthermore, denoting N_s as the number of scales or levels in the hierarchy, the compression ratio given by using the SPC architecture and the proposed clustering approach can be determined as

$$\tilde{\gamma} = \frac{1}{MN} \sum_{it=1}^{N_s} N_{seg}^{(it)}. \tag{69}$$

In order to perform the data clustering, the similarity graph $\mathbf{G} \in \mathbb{R}^{N_{seg} \times N_{seg}}$ is constructed using the k-nearest neighbor approach described in (Von Luxburg, 2007a). Then, the cluster indices $\bar{\mathbf{C}}$ are obtained by applying the spectral clustering to the similarity graph. Finally, the cluster membership of all the spectral pixels in the full image are obtained by applying the upsampling operator Δ^T onto $\bar{\mathbf{C}}$, see Algorithm 9. Note that both, the similarity graph construction and the spectral clustering computation are performed on the feature matrix $\bar{\mathbf{F}}$. Hence, the proposed approach boosts the computational performance.

Algorithm 9 Data Clustering

```
Input : $\bar{\mathbf{F}} \in \mathbb{R}^{N_{seg} \times L}$, $\Delta$ downsampling matrix, $k$ clusters
Segmentation of the spectral pixels: $\mathbf{F}_1, \cdots, \mathbf{F}_k$
$\mathbf{G} \leftarrow$ Build_Sim_Graph($\bar{\mathbf{F}}$) ▷ $k$-nearest neighbor graph
$\bar{\mathbf{C}}_{idx} \leftarrow$ Spectral_Clustering($\mathbf{G}, k$) ▷ Spectral Clustering (Von Luxburg, 2007a)
$\mathbf{C}_{idx} \leftarrow \Delta^T \bar{\mathbf{C}}_{idx}$ ▷ Upsampling
```

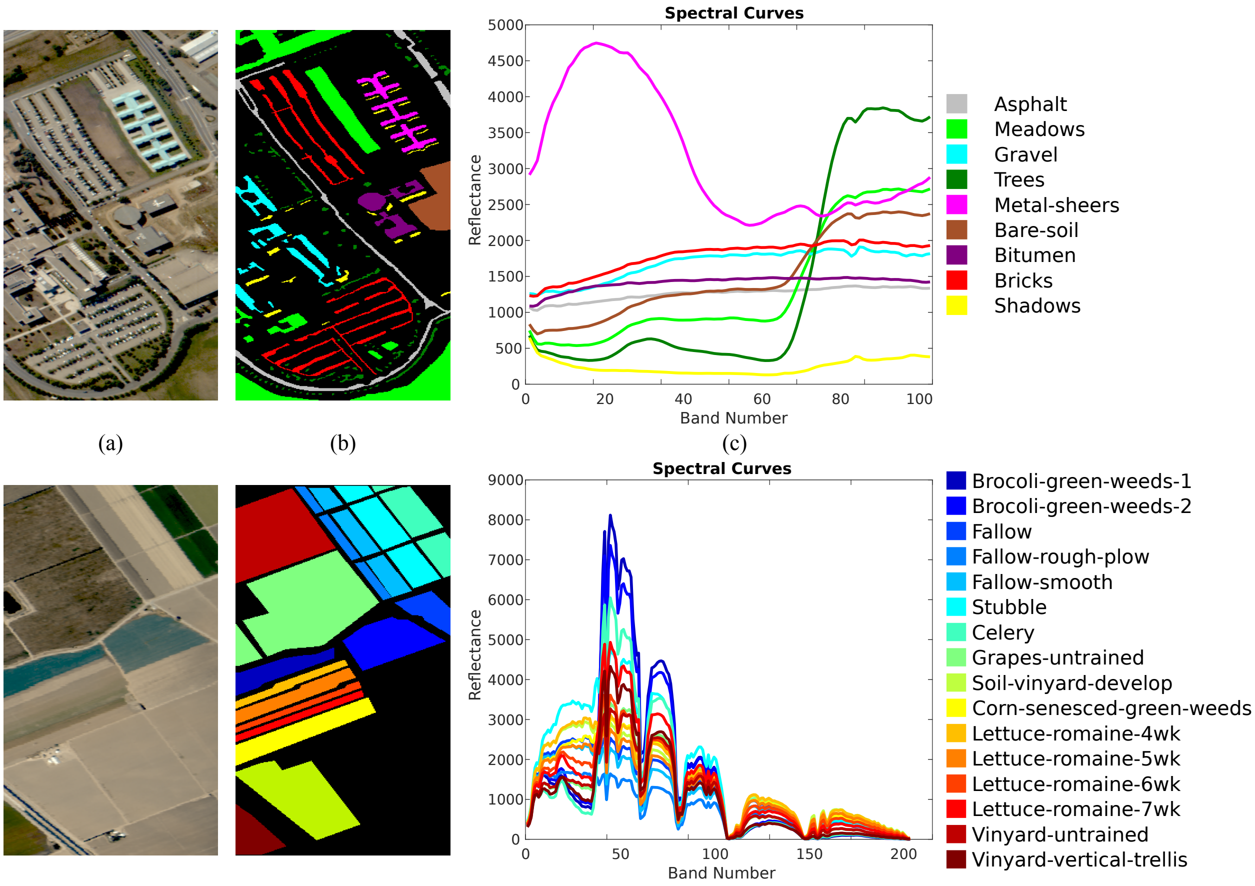
6.4. Simulations and Results

In this section, the performance of the proposed image classification method is evaluated. In particular, Pavia University and the Salinas Valley data sets were used.

The Pavia University hyperspectral data set was sensed over an urban area in northern Italy, by the Reflective Optics System Imaging Spectrometer (ROSIS-03) airborne sensor. The reference image is a section of 512×192 pixels and 103 spectral bands, with a spatial resolution of 1.3 meters per pixel and a spectral coverage ranging from 0.43 to 0.84 μm wavelengths. Pavia University contains nine main land-cover classes: asphalt, meadows, gravel, trees, metal sheets, bare soil, bitumen, bricks, and shadows. Figure 53 (a), (b) and (c) shows a false-color version, the ground truth, and the spectral signatures of the dataset, respectively.

The Salinas dataset was collected by the airborne visible/infrared imaging spectrometer (AVIRIS) on Salinas Valley, California, USA. The reference image is also a section of 512×192 pixels and 204 spectral bands in the range of 0.24 to 2.40 μm. The Salinas ground truth contains 16 land-cover classes: broccoli green weeds 1, broccoli green weeds 2, fallow, fallow rough plow, fallow smooth, stubble, celery, grapes untrained, soil vineyard develop, corn senesced green weeds, lettuce romaine 4 wk, lettuce romaine 5 wk, lettuce romaine 6 wk, lettuce romaine 7 wk, vineyard untrained, and vineyard vertical trellis. The ground truth is shown in Fig. 53 (e), where the spectral signatures of each class are presented in Fig. 53 (f).

The first two experiments were performed to show the sensitivity of the main parameters of the proposed method, i.e., the number of neighbors in the data clustering step and the number



(a) (b) (c)

(d) (e) (f)

Figure 53. False-color images (a) and (d), ground truth maps (b) and (e), and land-cover spectral signatures (c) and (f), for Pavia University and Salinas Valley, respectively.

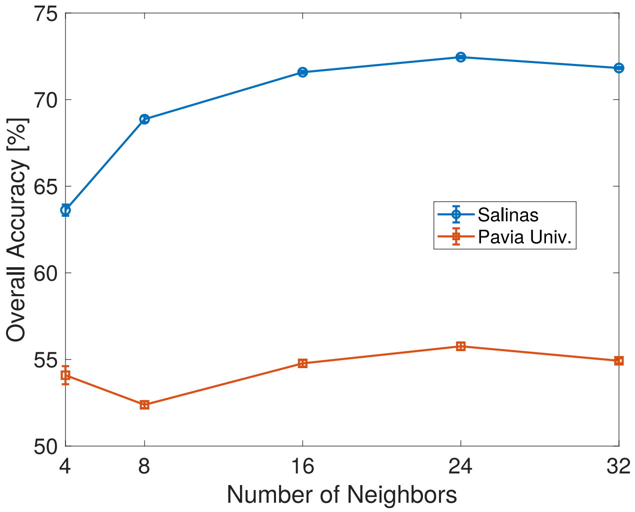


Figure 54. Overall clustering accuracy as a function of the number of Neighbors in the proposed method.

of scales or levels in the hierarchy sensing model. In both experiments, the number of segments were fixed as $N_{seg}^{(1)} = 8192$, $N_{seg}^{(2)} = 6144$, $N_{seg}^{(3)} = 4096$, $N_{seg}^{(4)} = 3072$, $N_{seg}^{(5)} = 2048$, and $N_{seg}^{(6)} = 1024$, for $it = 1, \cdots, 6$, i.e., the maximum level of hierarchy was 6. In the first experiment, the sensing ratio was fixed to 25%, whose value was obtained using Eq. 69, and the number of neighbors was varied from 4 to 32. The obtained overall accuracy curve of classification results, along with the mean and variance for each neighbor, is depicted in Fig. 54, where the presented results are the average of 20 trials. It can be observed that the largest variance occurs when using 4 neighbors as well as the best performance is obtained when using 24 neighbors for both datasets. Taking into account the previous results, in the second experiment, the number of neighbors is fixed as 24, and the number of scales was varied from 3 to 6. The obtained results are shown in Fig. 55, where it is observed that as the number of scales increases, the quality of classification improves as expected with the proposed method.

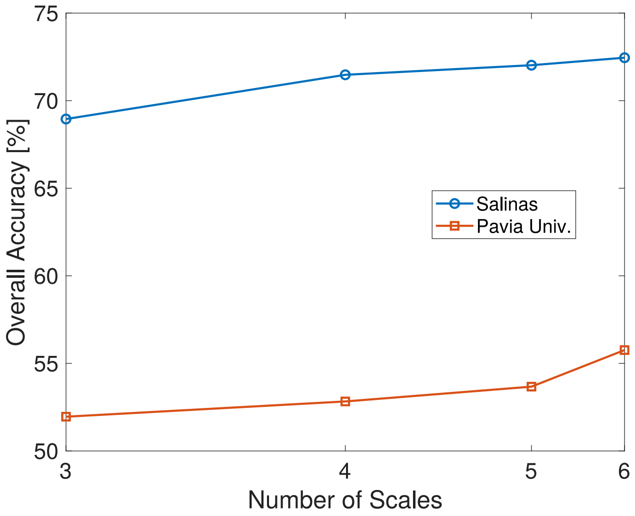


Figure 55. Overall clustering accuracy as a function of the number of scales in the proposed method.

In the last experiment, the results obtained with the proposed method is compared with the approach of using the same hierarchical scheme, but different designs of Δ. The first design, which I refer as “non-designed Δ”, corresponds to the first level of the proposed hierarchical approach, i.e., Δ is designed such as that all the pixels are grouped in square segments. The second design, which I refer as “FMR+Clustering”, corresponds to the approach described in (Garcia et al., 2018), where Δ is designed using super-pixels obtained from an RGB image which was acquired as side information. Note that the FMR method requires image reconstruction before performing spectral clustering. Figure 56, and Tables 28 and 29 show the visual and quantitative results obtained with the methods mentioned above. Specifically, tables show the numerical results for each of the land-cover classes (producer’s accuracy), overall accuracy (OA), average accuracy (AA), and Kappa (κ) coefficients (Lillesand et al., 2015), where the best value of each row is shown in bold font and the second best is underlined. All the results, except the Kappa coefficients, are given in percentage.

Table 28

Quantitative Results of different design approaches of Δ for the Salinas Valley Image.

Class	**Non-Designed Δ**	**FMR+Clustering**	**Proposed**
Broccoli-green-1	98.11	98.11	98.11
Broccoli-green-2	98.34	**98.77**	97.67
Fallow	7.91	**74.62**	8.70
Fallow-rough-plow	**96.48**	0.00	95.91
Fallow-smooth	92.91	**96.27**	93.09
Stubble	90.05	77.14	**97.07**
Celery	47.20	**99.17**	98.07
Grapes-untrained	78.91	**93.95**	55.15
Soil-vineyard-develop	93.42	0.48	**95.74**
Corn-senesced-green-weeds	**65.34**	59.43	50.27
Lettuce-romaine-4wk	19.19	2.15	**28.84**
Lettuce-romaine-5wk	0.00	**100.00**	99.53
Lettuce-romaine-6wk	0.00	**97.71**	96.51
Lettuce-romaine-7wk	0.00	83.27	**89.07**
Vineyard-untrained	0.00	3.10	**67.94**
Vineyard-vertical-trellis	**90.59**	82.79	77.37
AA	54.90	66.68	**78.07**
OA	62.55	63.12	**76.52**
Kappa	0.58	0.59	**0.74**

As observed, the proposed hierarchical method achieves the highest classification performance compared to the other approaches. Besides, note that the reconstruction time is avoided.

6.5. Conclusions

This chapter presented an approach to perform clustering directly on the compressed measurements. In particular, this chapter proposed an unsupervised and hierarchical method that takes advantage of Hadamard matrices structure and designs the sensing matrix of the SPC architecture such that the image reconstruction is avoided, enabling to extract clustering features. In general,

Table 29
Quantitative Results of different design approaches of Δ *for the Pavia University Image.*

Class	Non-Designed Δ	FMR+Clustering	Proposed
Asphalt	61.29	**91.75**	90.21
Meadows	26.30	21.98	**81.44**
Gravel	0.86	0.00	**4.05**
Trees	**77.84**	7.17	41.87
Metal sheets	**87.43**	71.23	86.77
Bare soil	30.11	**34.08**	32.63
Bitumen	**92.03**	0.00	0.00
Bricks	79.63	**93.97**	84.03
Shadows	89.14	**98.57**	92.12
AA	**60.51**	46.53	57.01
OA	49.19	43.86	**65.35**
Kappa	0.42	0.36	**0.57**

the presented results showed that overall accuracy of 78.94%, and 65.35% was obtained using the "Salinas", " Pavia University", and "Pavia Center" spectral image datasets, respectively.

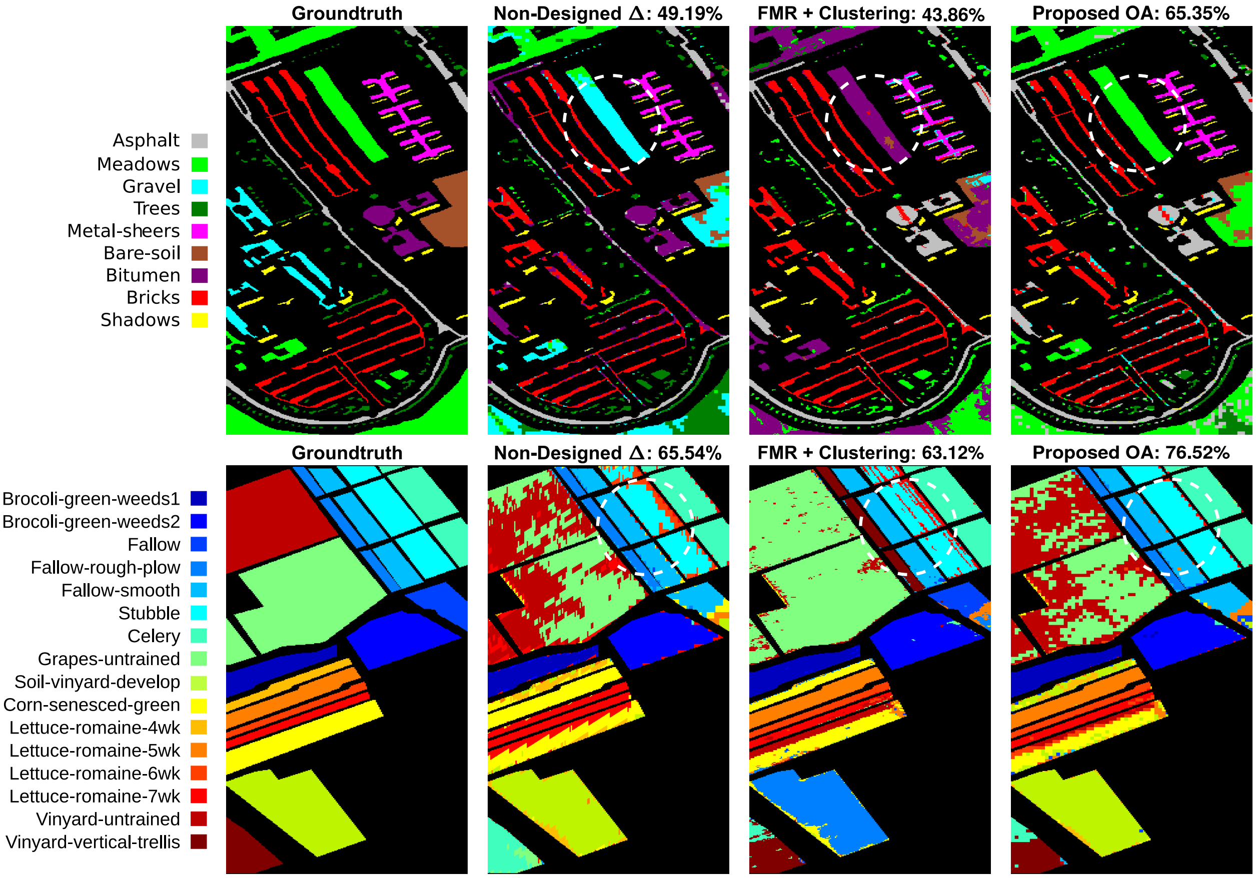


Figure 56. Visual clustering results on Pavia University (top) and Salinas data sets (bottom). The figure shows the ground truth and the results for the evaluated methods.

7. Conclusions and Future Work

This dissertation studied the problem of large-scale accurate unsupervised classification of HSIs. Among unsupervised methods, subspace clustering-based methods achieves high clustering accuracy and have become a popular tool. However, directly using subspace clustering methods on HSIs poses three main problems: (1) HSIs typically have noise due to their high variability of the spectral information and sensor noise (acquisition), which may decrease the subspace clustering performance if not handled appropriately; (2) due to the high-dimensional nature of HSIs and computational complexity of SC methods, most clustering algorithms can not be performed on the full HSIs but selected small regions of interest; (3) since SC methods ignore the spatial information in the HSIs, their discrimination capability is limited, hampering the clustering results' spatial homogeneity.

The hypothesis of this dissertation was that it is feasible to design a low-computational-complexity and robust algorithm that efficiently learns the low-dimensional structure of the data and provides an accurate representation coefficient matrix that incorporates the similarity information among the learned elements. To test this hypothesis, different algorithms were proposed and validated with experiments in the research works described through chapters 2-6.

The algorithm developed in chapter 2 used a 3D Gaussian filter to enhance the sparse coefficients matrix and reduce the negative impact of noise in the similarity graph before applying spectral clustering. This chapter shows that a smooth filtering in the representation matrix could improve the clustering accuracy by considering spatial information and reducing noise. Therefore, the use of this filtering procedure is adopted in chapters 4 and 5.

On the other hand, the algorithms developed in chapters 3 and 4 make use of superpixels extraction to group similar spectral pixels before subspace clustering. This technique improve the efficiency of subspace clustering since the representation matrix is built using fewer pixels and the clustering procedure can be splitted as shown in chapter 3.

Chapter 4 presents a fast and accurate subspace clustering algorithm for HSI by adopting the ideas from previous chapters (2 and 3) and looking for the most similar spectral pixels within each superpixel. Specifically, this algorithm selects the most representative spectral pixels within each superpixel and then constrains the remaining spectral pixels to be represented as a linear combination of them. This significantly reduces the clustering time and increases the accuracy as the representation matrix is built using only these most representative pixels and it encodes information about similarities between each subset's most representative spectral pixels and the whole HSI. Finally, the obtained coefficient matrix is enhanced using a similar strategy to the proposed in chapter 2, but using a 2D convolution instead of 3D to improve efficiency.

Finally, the algorithms developed in chapters 5 and 6 consider the spatial information to design the coded aperture of a CSI system, hence the spatial information of the spectral pixels is taken into account during sensing. Specifically, the algorithm in chapter 5 considers the similarity distance between two compressed spectral signatures and the amount of sensing information to design coding patterns and their location in a 3D-coded aperture used in the 3D-CASSI architecture, see chapter 1.5. On the other hand, chapter 6 uses k-means to group similar compressed spectral features and design a set of coded apertures in a hierarchical/adapting way.

Future work includes the implementation of the method for selecting the most representative spectral pixels to design the coded aperture, improving efficiency and accuracy in the clustering of HSI compressed measurements. Also, techniques developed in chapter 2 and 4 can be incorporated in a parallel algorithm to boost efficiency. Furthermore, considering the relevance of deep neural networks nowadays, it would also be interesting to design an unsupervised end-to-end deep network model to perform both code aperture design and clustering considering the spatial information in HSIs.

Bibliography

Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Abdolali, M., Gillis, N., and Rahmati, M. (2019). Scalable and robust sparse subspace clustering using randomized clustering and multilayer graphs. *Signal Processing*, 163:166–180.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2010). SLIC Superpixels. Technical Report June, EPFL Technical Report.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282.

Achlioptas, D. (2003). Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687.

Ahmad, M. (2020). A fast 3d cnn for hyperspectral image classification. *arXiv preprint arXiv:2004.14152*.

Akhtar, N., Shafait, F., and Mian, A. S. (2014). Sparse spatio-spectral representation for hyperspectral image super-resolution. In *ECCV*.

Annergren, M., Hansson, A., and Wahlberg, B. (2012). An admm algorithm for solving l1 regularized mpc. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 4486–4491. IEEE.

Arce, G. R., Brady, D. J., Carin, L., Arguello, H., and Kittle, D. S. (2014). Compressive coded aperture spectral imaging: An introduction. *IEEE Signal Processing Magazine*, 31(1):105–115.

Arguello, H. and Arce, G. R. (2011). Code aperture optimization for spectrally agile compressive imaging. *JOSA A*, 28(11):2400–2413.

Arguello, H. and Arce, G. R. (2013). Rank minimization code aperture design for spectrally selective compressive imaging. *IEEE Transactions on Image Processing*, 22(3):941–954.

Arguello, H. and Arce, G. R. (2014). Colored coded aperture design by concentration of measure in compressive spectral imaging. *IEEE Transactions on Image Processing*, 23(4):1896–1908.

Bacca, J., Correa, C. V., and Arguello, H. (2019a). A non-iterative reconstruction algorithm for single pixel spectral imaging with side information. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5.

Bacca, J., Correa, C. V., and Arguello, H. (2019b). Noniterative hyperspectral image reconstruction from compressive fused measurements. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Bacca, J., Hinojosa, C. A., and Arguello, H. (2017a). Kernel sparse subspace clustering with total variation denoising for hyperspectral remote sensing images. In *Mathematics in Imaging*, pages MTu4C–5. Optical Society of America.

Bacca, J., Vargas, H., and Arguello, H. (2017b). A constrained formulation for compressive spectral image reconstruction using linear mixture models. In *Proc. Conf. CAMSAP*, pages 1–5. IEEE.

Baraniuk, R. G. (2007). Compressive sensing. *IEEE signal processing magazine*, 24(4).

Baraniuk, R. G., Cevher, V., and Wakin, M. B. (2010). Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective. *Proceedings of the IEEE*, 98(6):959–971.

Bezdek, J. C. (2013). *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media.

Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250.

Bioucas-Dias, J. M. and Figueiredo, M. A. (2007). A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration. *Image Processing, IEEE Transactions on*, 16(12):2992–3004.

Bioucas-Dias, J. M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N., and Chanussot, J. (2013). Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and remote sensing magazine*, 1(2):6–36.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

Bovolo, F., Bruzzone, L., and Marconcini, M. (2006). A novel context-sensitive svm for classification of remote sensing images. In *2006 IEEE International Symposium on Geoscience and Remote Sensing*, pages 2498–2501.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine learning*, 3(1):1–122.

Briottet, X., Boucher, Y., Dimmeler, A., Malaplate, A., Cini, A., Diani, M., Bekman, H., Schwering, P., Skauli, T., Kasen, I., et al. (2006). Military applications of hyperspectral imagery. In *Targets and backgrounds XII: Characterization and representation*, volume 6239, page 62390B. International Society for Optics and Photonics.

Bruzzone, L., Chi, M., and Marconcini, M. (2006). A novel transductive svm for semisupervised classification of remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11):3363–3373.

Cai, Y., Zhang, Z., Cai, Z., Liu, X., Jiang, X., and Yan, Q. (2020). Graph convolutional subspace clustering: A robust subspace clustering framework for hyperspectral image. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):4191–4202.

Calderbank, R., Jafarpour, S., and Schapire, R. (2009). Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. *preprint*.

Camps-Valls, G., Gomez-Chova, L., Muñoz-Marí, J., Vila-Francés, J., and Calpe-Maravilla, J. (2006). Composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 3(1):93–97.

Camps-Valls, G., Tuia, D., Bruzzone, L., and Benediktsson, J. A. (2014). Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. *IEEE Signal Processing Magazine*, 31(1):45–54.

Candès, E. J. and Wakin, M. B. (2008). An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 8(6):679–698.

Cao, X., Yue, T., Lin, X., Lin, S., Yuan, X., Dai, Q., Carin, L., and Brady, D. J. (2016). Computational snapshot multispectral cameras: Toward dynamic capture of the spectral world. *IEEE Signal Processing Magazine*, 33(5):95–108.

Chartrand, R. (2007). Exact reconstruction of sparse signals via nonconvex minimization. *Signal Processing Letters, IEEE*, 14(10):707–710.

Chen, G. and Lerman, G. (2009). Spectral curvature clustering (scc). *International Journal of Computer Vision*, 81(3):317–330.

Chen, S. and Zhang, D. (2004). Robust image segmentation using fcm with spatial constraints based on new kernel-induced distance measure. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1907–1916.

Chen, X., Hong, W., Nie, F., He, D., Yang, M., and Huang, J. Z. (2018). Spectral clustering of large-scale data by directly solving normalized cut. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1206–1215. ACM.

Chen, Y., Li, G., and Gu, Y. (2017). Active orthogonal matching pursuit for sparse subspace clustering. *IEEE Signal Processing Letters*, 25(2):164–168.

Computational Intelligence Group (GIC) (2011). Hyperspectral remote sensing scenes. [Online]. Available: `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`.

Correa, C. V., Hinojosa, C. A., Arce, G. R., and Arguello, H. (2016). Multiple snapshot colored compressive spectral imager. *Optical Engineering*, 56(4):041309.

Das, S., Routray, A., and Deb, A. K. (2016). Noise robust estimation of number of endmembers in a hyperspectral image by eigenvalue based gap index. In *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5. IEEE.

Davenport, M. A., Boufounos, P. T., Wakin, M. B., and Baraniuk, R. G. (2010). Signal processing with compressive measurements. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):445–460.

Della Porta, C. J., Bekit, A. A., Lampe, B. H., and Chang, C.-I. (2019). Hyperspectral image classification via compressive sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 57(10):8290–8303.

Deng, C., Hu, X., Li, X., Suo, J., Zhang, Z., and Dai, Q. (2019). High fidelity single-pixel imaging. *IEEE Photonics Journal*, 11(2):1–9.

Diaz, N., Hinojosa, C., and Arguello, H. (2019). Adaptive grayscale compressive spectral imaging using optimal blue noise coding patterns. *Optics & Laser Technology*, 117:147–157.

Donoho, D., Tsaig, Y., Drori, I., and Starck, J. (2006). Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit 2006. *Preprint*.

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306.

Donoho, D. L. and Huo, X. (2001). Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862.

Du, Q. and Fowler, J. E. (2008). Low-complexity principal component analysis for hyperspectral image compression. *The International Journal of High Performance Computing Applications*, 22(4):438–448.

Duarte, M. F., Davenport, M. A., Takhar, D., Laska, J. N., Sun, T., Kelly, K. F., and Baraniuk, R. G. (2008). Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, 25(2):83–91.

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Dyer, E. L., Sankaranarayanan, A. C., and Baraniuk, R. G. (2013). Greedy feature selection for subspace clustering. *The Journal of Machine Learning Research*, 14(1):2487–2517.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

Eichenholz, J. M., Barnett, N., Juang, Y., Fish, D., Spano, S., Lindsley, E., and Farkas, D. L. (2010). Real-time megapixel multispectral bioimaging. In *BiOS*, pages 75681L–75681L. International Society for Optics and Photonics.

Eismann, M. T., Stocker, A. D., and Nasrabadi, N. M. (2009). Automated hyperspectral cueing for civilian search and rescue. *Proceedings of the IEEE*, 97(6):1031–1055.

Elhamifar, E. and Vidal, R. (2009). Sparse subspace clustering. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797. IEEE.

Elhamifar, E. and Vidal, R. (2011). Sparse manifold clustering and embedding. In *Advances in neural information processing systems*, pages 55–63.

Elhamifar, E. and Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781.

Fauvel, M., Tarabalka, Y., Benediktsson, J. A., Chanussot, J., and Tilton, J. C. (2013). Advances in spectral-spatial classification of hyperspectral images. *Proceedings of the IEEE*, 101(3):652–675.

Figueiredo, M. A., Nowak, R. D., and Wright, S. J. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):586–597.

Fox, S., Tridgell, S., Jin, C., and Leong, P. H. (2016). Random projections for scaling machine learning on fpgas. In *Field-Programmable Technology (FPT), 2016 International Conference on*, pages 85–92. IEEE.

Fu, H., Bian, L., and Zhang, J. (2020). Single-pixel sensing with optimal binarized modulation. *Opt. Lett.*, 45(11):3111–3114.

Galvis, L., Arguello, H., Lau, D., and Arce, G. R. (2017). Side information in coded aperture compressive spectral imaging. In *Emerging Digital Micromirror Device Based Systems and Applications IX*, volume 10117, page 101170H. International Society for Optics and Photonics.

Gao, L., Kester, R. T., Hagen, N., and Tkaczyk, T. S. (2010). Snapshot image mapping spectrometer (ims) with high sampling density for hyperspectral microscopy. *Optics express*, 18(14):14330–14344.

Garcia, H., Correa, C. V., and Arguello, H. (2011). Slic super-pixels for multi-resolution compressive spectral imaging reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell*, 33:2423–2435.

Garcia, H., Correa, C. V., and Arguello, H. (2018). Multi-resolution compressive spectral imaging reconstruction from single pixel measurements. *IEEE Transactions on Image Processing*, 27(12):6174–6184.

Garcia, H., Correa, C. V., and Arguello, H. (2018a). Multi-resolution compressive spectral imaging reconstruction from single pixel measurements. *IEEE Transactions on Image Processing*, 27(12):6174–6184.

Garcia, H., Correa, C. V., and Arguello, H. (2018b). Multi-Resolution Compressive Spectral Imaging Reconstruction from Single Pixel Measurements. *IEEE Transactions on Image Processing*, page 1.

Garcia, H., Correa, C. V., Villarreal, O., Pinilla, S., and Arguello, H. (2017). Multi-resolution reconstruction algorithm for compressive single pixel spectral imaging. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 468–472. IEEE.

Gauraha, N. (2018). Introduction to the lasso. *Resonance*, 23(4):439–464.

Geelen, B., Jayapala, M., Tack, N., and Lambrechts, A. (2013). Low-complexity image processing for a high-throughput low-latency snapshot multispectral imager with integrated tiled filters. In *SPIE Defense, Security, and Sensing*, pages 87431E–87431E. International Society for Optics and Photonics.

Gehm, M., John, R., Brady, D., Willett, R., and Schulz, T. (2007). Single-shot compressive spectral imaging with a dual-disperser architecture. *Optics express*, 15(21):14013–14027.

Gessesse, B., Bewket, W., and Bräuning, A. (2015). Model-based characterization and monitoring of runoff and soil erosion in response to land use/land cover changes in the modjo watershed, ethiopia. *Land degradation & development*, 26(7):711–724.

Ghamisi, P., Plaza, J., Chen, Y., Li, J., and Plaza, A. J. (2017a). Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geoscience and Remote Sensing Magazine*, 5(1):8–32.

Ghamisi, P., Yokoya, N., Li, J., Liao, W., Liu, S., Plaza, J., Rasti, B., and Plaza, A. (2017b). Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):37–78.

Goh, A. and Vidal, R. (2007). Segmenting motions of different types by unsupervised manifold clustering. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE.

Golub, G. H. and Reinsch, C. (1971). Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer.

Graña, M., Veganzons, M., and Ayerdi, B. (2020). Hyperspectral remote sensing scenes. `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`.

Green, R. O., Eastwood, M. L., Sarture, C. M., Chrien, T. G., Aronsson, M., Chippendale, B. J., Faust, J. A., Pavri, B. E., Chovit, C. J., Solis, M., et al. (1998). Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (aviris). *Remote Sensing of Environment*, 65(3):227–248.

Gu, B. and Sheng, V. S. (2017). A robust regularization path algorithm for ν-support vector classification. *IEEE Transactions on neural networks and learning systems*, 28(5):1241–1248.

Gu, D., Gillespie, A. R., Kahle, A. B., and Palluconi, F. D. (2000). Autonomous atmospheric compensation (aac) of high resolution hyperspectral thermal infrared remote-sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 38(6):2557–2570.

Gualtieri, J. A. and Cromp, R. F. (1999). Support vector machines for hyperspectral remote sensing classification. In *The 27th AIPR Workshop: Advances in Computer-Assisted Recognition*, pages 221–232. International Society for Optics and Photonics.

Guo, Y., Gao, J., and Li, F. (2014). Spatial subspace clustering for drill hole spectral data. *Journal of Applied Remote Sensing*, 8(1):083644.

Hahn, J., Rosenkranz, S., and Zoubir, A. M. (2014). Adaptive compressed classification for hyperspectral imagery. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 1020–1024. IEEE.

Hang, R., Liu, Q., Song, H., and Sun, Y. (2016). Matrix-based discriminant subspace ensemble for hyperspectral image spatial–spectral feature fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 54(2):783–794.

He, D., Shi, Q., Liu, X., Zhong, Y., and Zhang, X. (2021). Deep subpixel mapping based on semantic information modulated network for urban land use mapping. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–19.

He, L., Li, J., Liu, C., and Li, S. (2018). Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines. *IEEE Transactions on Geoscience and Remote Sensing*, 56(3):1579–1597.

He, M., Li, B., and Chen, H. (2017). Multi-scale 3d deep convolutional neural network for hyperspectral image classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3904–3908. IEEE.

He, W., Zhang, H., Zhang, L., Philips, W., and Liao, W. (2016a). Weighted sparse graph based dimensionality reduction for hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 13(5):686–690.

He, W., Zhang, H., Zhang, L., and Shen, H. (2015). Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):3050–3061.

He, Z., Liu, L., Deng, R., and Shen, Y. (2016b). Low-rank group inspired dictionary learning for hyperspectral image classification. *Signal Processing*, 120:209–221.

Heckel, R., Tschannen, M., and Bölcskei, H. (2015). Dimensionality-reduced subspace clustering. *arXiv preprint arXiv:1507.07105*.

Herrala, E., Okkonen, J. T., Hyvarinen, T. S., Aikio, M., and Lammasniemi, J. (1994). Imaging spectrometer for process industry applications. In *Optics for Productivity in Manufacturing*, pages 33–40. International Society for Optics and Photonics.

Hinojosa, C., Arguello, H., and Rueda, H. (2015). Analysis of matrix completion algorithms for spectral image estimation from compressive coded projections. In *2015 20th Symposium on Signal Processing, Images and Computer Vision (STSIVA)*, pages 1–7. IEEE.

Hinojosa, C., Bacca, J., and Arguello, H. (2018a). Coded aperture design for compressive spectral subspace clustering. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1589–1600.

Hinojosa, C., Bacca, J., and Arguello, H. (2018b). Coded aperture design for compressive spectral subspace clustering. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1589–1600.

Hinojosa, C., Bacca, J., and Arguello, H. (2018). Coded aperture design for compressive spectral subspace clustering. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1589–1600.

Hinojosa, C., Bacca, J., Vargas, E., Castillo, S., and Arguello, H. (2019a). Single-pixel camera sensing matrix design for hierarchical compressed spectral clustering. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

Hinojosa, C., Niebles, J. C., and Arguello, H. (2021a). Learning privacy-preserving optics for human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2573–2582.

Hinojosa, C., Ramirez, J. M., and Arguello, H. (2019b). Spectral-spatial classification from multi-sensor compressive measurements using superpixels. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3143–3147. IEEE.

Hinojosa, C., Ramirez, J. M., and Arguello, H. (2019c). Spectral-spatial classification from multi-sensor compressive measurements using superpixels. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3143–3147.

Hinojosa, C., Sanchez, K., Garcia, H., and Arguello, H. (2022). C-3spcd: coded aperture similarity constrained design forspatio-spectral classification of single-pixel measurements. *Appl. Opt.*, 61(8):E21–E32.

Hinojosa, C., Vera, E., and Arguello, H. (2021b). A fast and accurate similarity-constrained subspace clustering algorithm for hyperspectral image. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Hinojosa, C. A., Bacca, J., and Arguello, H. (2018a). Spectral imaging subspace clustering with 3-d spatial regularizer. In *Digital Holography and Three-Dimensional Imaging*, pages JW5E–7. Optical Society of America.

Hinojosa, C. A., Bacca, J., and Arguello, H. (2018b). Spectral imaging subspace clustering with 3-d spatial regularizer. In *Digital Holography and Three-Dimensional Imaging*, pages JW5E–7. Optical Society of America.

Hinojosa, C. A., Bacca, J., and Arguello, H. (2018c). Spectral imaging subspace clustering with 3-d spatial regularizer. In *Imaging and Applied Optics 2018 (3D, AO, AIO, COSI, DH, IS, LACSEA, LS&C, MATH, pcAOP)*, page JW5E.7. Optical Society of America.

Hinojosa, C. A., Rojas, F., Castillo, S., and Arguello, H. (2021c). Hyperspectral image segmentation using 3d regularized subspace clustering model. *Journal of Applied Remote Sensing*, 15(1):016508.

Hörig, B., Kühn, F., Oschütz, F., and Lehmann, F. (2001). Hymap hyperspectral remote sensing to detect hydrocarbons. *International Journal of Remote Sensing*, 22(8):1413–1422.

Horn, R. A., Horn, R. A., and Johnson, C. R. (1990). *Matrix analysis*. Cambridge university press.

Huang, S., Zhang, H., and Pižurica, A. (2019). Semisupervised sparse subspace clustering method with a joint sparsity constraint for hyperspectral remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(3):989–999.

James, R. C. (1961). Book review: Introduction to functional analysis. *Bulletin of the American Mathematical Society*, 67(4):344–346.

Ji, S., Xue, Y., and Carin, L. (2008). Bayesian compressive sensing. *Signal Processing, IEEE Transactions on*, 56(6):2346–2356.

Jiao, Y., Li, G., and Gu, Y. (2017). Principal angles preserving property of gaussian random projection for subspaces. In *Signal and Information Processing (GlobalSIP), 2017 IEEE Global Conference on*, pages 318–322. IEEE.

Kaski, S. (1998). Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proc. Conf. IJCNN*, volume 1, pages 413–418. IEEE.

Kerekes, J. P. and Schott, J. R. (2007). Hyperspectral imaging systems. *Hyperspectral data exploitation: Theory and applications*, pages 19–45.

Kittle, D., Choi, K., Wagadarikar, A., and Brady, D. J. (2010). Multiframe image estimation for coded aperture snapshot spectral imagers. *Applied Optics*, 49(36):6824–6833.

Kolesnikov, A., Zhai, X., and Beyer, L. (2019). Revisiting self-supervised visual representation learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lanthier, Y., Bannari, A., Haboudane, D., Miller, J. R., and Tremblay, N. (2008). Hyperspectral data segmentation and classification in precision agriculture: A multi-scale analysis. In *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*, volume 2, pages II–585. IEEE.

Li, G. and Gu, Y. (2017). Restricted isometry property of gaussian random projection for finite set of subspaces. *IEEE Transactions on Signal Processing*, 66(7):1705–1720.

Li, J., Bioucas-Dias, J. M., and Plaza, A. (2012). Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 50(3):809–823.

Li, J., Marpu, P. R., Plaza, A., Bioucas-Dias, J. M., and Benediktsson, J. A. (2013a). Generalized composite kernel framework for hyperspectral image classification. *IEEE transactions on geoscience and remote sensing*, 51(9):4816–4829.

Li, P., Hastie, T. J., and Church, K. W. (2006). Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 287–296. ACM.

Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., and Benediktsson, J. A. (2019). Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*.

Li, W., Prasad, S., and Fowler, J. E. (2013b). Classification and reconstruction from random projections for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):833–843.

Li, Z. and Chen, J. (2015). Superpixel segmentation using linear spectral clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1356–1363.

Li, Z., Suo, J., Hu, X., Deng, C., Fan, J., and Dai, Q. (2017). Efficient single-pixel multispectral imaging via non-mechanical spatio-spectral modulation. *Scientific reports*, 7:41435.

Liang, Y., Markopoulos, P. P., and Saber, E. (2019). Spatial–spectral segmentation of hyperspectral images for subpixel target detection. *Journal of Applied Remote Sensing*, 13(3):036502.

Lillesand, T., Kiefer, R. W., and Chipman, J. (2015). *Remote sensing and image interpretation*. John Wiley & Sons.

Lin, F., Jovanovic, M. R., and Georgiou, T. T. (2013). An admm algorithm for matrix completion of partially known state covariances. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 1684–1689. IEEE.

Lin, J. and Gunopulos, D. (2003). Dimensionality reduction by random projection and latent semantic indexing. In *proceedings of the Text Mining Workshop, at the 3rd SIAM International Conference on Data Mining*.

Lin, X., Liu, Y., Wu, J., and Dai, Q. (2014a). Spatial-spectral encoded compressive hyperspectral imaging. *ACM Transactions on Graphics (TOG)*, 33(6):233.

Lin, X., Wetzstein, G., Liu, Y., and Dai, Q. (2014b). Dual-coded compressive hyperspectral imaging. *Optics letters*, 39(7):2044–2047.

Liu, G., Zhang, Z., Liu, Q., and Xiong, H. (2019). Robust subspace clustering with compressed data. *IEEE Transactions on Image Processing*, 28(10):5161–5170.

Liu, J., Wu, Z., Wei, Z., Xiao, L., and Sun, L. (2013). Spatial-spectral kernel sparse representation for hyperspectral image classification. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens*, 6(6):2462–2471.

Liu, S., Shi, Q., and Zhang, L. (2021). Few-shot hyperspectral image classification with unknown classes using multitask deep learning. *IEEE Transactions on Geoscience and Remote Sensing*, 59(6):5085–5102.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Loncan, L., Almeida, L. B., Bioucas-Dias, J. M., Briottet, X., Chanussot, J., Dobigeon, N., Fabre, S., Liao, W., Licciardi, G. A., Simoes, M., et al. (2015). Hyperspectral pansharpening: A review. *arXiv preprint arXiv:1504.04531*.

Lopez, J., Hinojosa, C., and Arguello, H. (2021a). Fast subspace clustering algorithm with efficient similarity-constrained sampling for hyperspectral images. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

Lopez, J., Hinojosa, C. A., and Arguello, H. (2021b). Efficient subspace clustering of hyperspectral images using similarity-constrained sampling. *Journal of Applied Remote Sensing*, 15(3):036507.

Lu, G. and Fei, B. (2014). Medical hyperspectral imaging: a review. *Journal of biomedical optics*, 19(1):010901.

Magen, A. (2002). Dimensionality reductions that preserve volumes and distance to affine spaces, and their algorithmic applications. *Randomization and approximation techniques in computer science*, pages 953–953.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60.

Manolakis, D. and Shaw, G. (2002). Detection algorithms for hyperspectral imaging applications. *IEEE Signal Processing Magazine*, 19(1):29–43.

Mao, X. and Gu, Y. (2014). Compressed subspace clustering: A case study. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 453–457. IEEE.

Martín, G. and Bioucas-Dias, J. M. (2016). Hyperspectral blind reconstruction from random spectral projections. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(6):2390–2399.

Mehta, A. and Dikshit, O. (2017). Segmentation-based clustering of hyperspectral images using local band selection. *Journal of Applied Remote Sensing*, 11(1):015028.

Melgani, F. and Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on geoscience and remote sensing*, 42(8):1778–1790.

Meng, L., Li, G., Yan, J., and Gu, Y. (2018). A general framework for understanding compressed subspace clustering algorithms. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1504–1519.

Miller, A. (2002). *Subset selection in regression*. CRC Press.

Morris, H. R., Hoyt, C. C., and Treado, P. J. (1994). Imaging spectrometers for fluorescence and raman microscopy: acousto-optic and liquid crystal tunable filters. *Applied spectroscopy*, 48(7):857–866.

Mountrakis, G., Im, J., and Ogole, C. (2011). Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3):247–259.

Nalepa, J., Myller, M., Imai, Y., Honda, K.-I., Takeda, T., and Antoniak, M. (2020). Unsupervised segmentation of hyperspectral images using 3-d convolutional autoencoders. *IEEE Geoscience and Remote Sensing Letters*.

Needell, D. and Tropp, J. A. (2009). Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321.

Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2002). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856.

Paoletti, M., Haut, J., Plaza, J., and Plaza, A. (2019). Deep learning classifiers for hyperspectral imaging: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158:279–317.

Parikh, N., Boyd, S., et al. (2014). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239.

Parsons, L., Haque, E., and Liu, H. (2004). Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, 6(1):90–105.

Patel, N., Patnaik, C., Dutta, S., Shekh, A., and Dave, A. (2001). Study of crop growth parameters using airborne imaging spectrometer data. *International Journal of Remote Sensing*, 22(12):2401–2411.

Peng, X., Tang, H., Zhang, L., Yi, Z., and Xiao, S. (2015). A unified framework for representation-based subspace clustering of out-of-sample and large-scale data. *IEEE transactions on neural networks and learning systems*, 27(12):2499–2512.

Peng, X., Yu, Z., Yi, Z., and Tang, H. (2016). Constructing the l2-graph for robust subspace learning and subspace clustering. *IEEE transactions on cybernetics*, 47(4):1053–1066.

Peng, X., Zhang, L., and Yi, Z. (2013). Scalable sparse subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 430–437.

Qin, A., Shang, Z., Tian, J., Wang, Y., Zhang, T., and Tang, Y. Y. (2018). Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 16(2):241–245.

Rasti, B., Ulfarsson, M. O., and Sveinsson, J. R. (2015). Hyperspectral subspace identification using sure. *IEEE Geoscience and Remote Sensing Letters*, 12(12):2481–2485.

Rasti, B., Ulfarsson, M. O., and Sveinsson, J. R. (2016). Hyperspectral feature extraction using total variation component analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):6976–6985.

Rauhut, H. (2010). Compressive sensing and structured random matrices. *Theoretical foundations and numerical methods for sparse recovery*, 9:1–92.

Rizvi, S., Cao, J., and Hao, Q. (2020). High-speed image-free target detection and classification in single-pixel imaging. In Kimata, M., Shaw, J. A., and Valenta, C. R., editors, *SPIE Future Sensing Technologies*, volume 11525, pages 189 – 194. International Society for Optics and Photonics, SPIE.

Rueda, H., Arguello, H., and Arce, G. R. (2014). Compressive spectral imaging based on colored coded apertures. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 7799–7803. IEEE.

Rueda, H., Arguello, H., and Arce, G. R. (2015). Dmd-based implementation of patterned optical filter arrays for compressive spectral imaging. *JOSA A*, 32(1):80–89.

Sanchez, K., Hinojosa, C., and Arguello, H. (2018). Supervised classification of hyperspectral images using side information. In *Imaging and Applied Optics 2018 (3D, AO, AIO, COSI, DH, IS, LACSEA, LS&C, MATH, pcAOP)*, page JW5E.5. Optical Society of America.

Sanchez, K., Hinojosa, C., and Arguello, H. (2019a). Supervised spatio-spectral classification of fused images using superpixels. *Appl. Opt.*, 58(7):B9–B18.

Sanchez, K., Hinojosa, C., and Arguello, H. (2019b). Supervised spatio-spectral classification of fused images using superpixels. *Appl. Opt.*, 58(7):B9–B18.

Sanchez, K., Hinojosa, C., Arguello, H., Freiss, S., Sans, N., Kouamé, D., Meyrignac, O., and Basarab, A. (2021a). Subspace-based domain adaptation using similarity constraints for pneumonia diagnosis within a small chest x-ray image dataset. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1232–1235. IEEE.

Sanchez, K., Hinojosa, C., Garcia, H., Arguello, H., and Castillo, S. (2021b). Compressed-domain classification algorithm for spectral imaging based on designed single-pixel camera codification. In *OSA Imaging and Applied Optics Congress 2021 (3D, COSI, DH, ISA, pcAOP)*, page CTu2F.5. Optical Society of America.

Sankaranarayanan, A. C., Xu, L., Studer, C., Li, Y., Kelly, K. F., and Baraniuk, R. G. (2015). Video compressive sensing for spatial multiplexing cameras using motion-flow models. *SIAM Journal on Imaging Sciences*, 8(3):1489–1518.

Schlkopf, B., Smola, A. J., and Bach, F. (2018). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. The MIT Press.

Schütze, H., Manning, C. D., and Raghavan, P. (2007). *An introduction to information retrieval*. Cambridge University Press.

Sellami, A., Abbes, A. B., Barra, V., and Farah, I. R. (2020). Fused 3-d spectral-spatial deep neural networks and spectral clustering for hyperspectral image classification. *Pattern Recognition Letters*, 138:594–600.

Shaw, G. A. and Burke, H. K. (2003). Spectral imaging for remote sensing. *Lincoln laboratory journal*, 14(1):3–28.

Shi, Q., Liu, M., Li, S., Liu, X., Wang, F., and Zhang, L. (2021a). A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–16.

Shi, Q., Tang, X., Yang, T., Liu, R., and Zhang, L. (2021b). Hyperspectral image denoising using a

3-d attention denoising network. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–16.

Simões, M., Bioucas-Dias, J. M., Almeida, L. B., and Chanussot, J. (2014). A convex formulation for hyperspectral image superresolution via subspace-based regularization. *CoRR*, abs/1411.4005.

Soltanolkotabi, M., Candes, E. J., et al. (2012). A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617.

Stutz, D., Hermans, A., and Leibe, B. (2018). Superpixels: an evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27.

Thenkabail, P. S. and Lyon, J. G. (2016). *Hyperspectral remote sensing of vegetation*. CRC press.

Tian, X., Jiao, L., Yi, L., Guo, K., and Zhang, X. (2015). The image segmentation based on optimized spatial feature of superpixel. *Journal of Visual Communication and Image Representation*, 26:146–160.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tropp, J., Gilbert, A. C., et al. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666.

Tropp, J., Wright, S. J., et al. (2010). Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958.

Trucco, E. and Verri, A. (1998). *Introductory techniques for 3-D computer vision*, volume 201. Prentice Hall Englewood Cliffs.

Tschannen, M. and Bölcskei, H. (2016). Noisy subspace clustering via matching pursuits. *arXiv preprint arXiv:1612.03450*.

Tulczyjew, L., Kawulok, M., and Nalepa, J. (2020). Unsupervised feature learning using recurrent neural nets for segmenting hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*.

Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71.

Van Zee, F. G. and Van De Geijn, R. A. (2015). Blis: A framework for rapidly instantiating blas functionality. *ACM Transactions on Mathematical Software (TOMS)*, 41(3):14.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media.

Vargas, E., Arguello, H., and Tourneret, J. (2019a). Spectral image fusion from compressive measurements using spectral unmixing and a sparse representation of abundance maps. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–11.

Vargas, E., Espitia, Ǿ., Arguello, H., and Tourneret, J. (2019b). Spectral image fusion from compressive measurements. *IEEE Transactions on Image Processing*, 28(5):2271–2282.

Vargas, E., Pinilla, S., and Arguello, H. (2018). A fast endmember estimation algorithm from compressive measurements. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2210–2214. IEEE.

Vargas, H. and Arguello, H. (2019). A low-rank model for compressive spectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):9888–9899.

Vempala, S. S. (2005). *The random projection method*, volume 65. American Mathematical Soc.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

Vidal, R. (2011). Subspace clustering. *IEEE Signal Processing Magazine*, pages 52–68.

Vincent, O. R., Folorunso, O., et al. (2009). A descriptive algorithm for sobel image edge detection. In *Proceedings of Informing Science & IT Education Conference (InSITE)*, volume 40, pages 97–107. Informing Science Institute California.

Volpi, M. and Ferrari, V. (2015). Semantic segmentation of urban scenes by learning local class interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9.

Von Luxburg, U. (2007a). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

Von Luxburg, U. (2007b). A tutorial on spectral clustering, statistics and computing17 (4)(2007), 395-416. *Google Scholar Google Scholar Digital Library Digital Library*.

Wagadarikar, A., John, R., Willett, R., and Brady, D. (2008). Single disperser design for coded aperture snapshot spectral imaging. *Applied optics*, 47(10):B44–B51.

Wang, Y., Wang, Y.-X., and Singh, A. (2016). A theoretical analysis of noisy sparse subspace clustering on dimensionality-reduced data. *arXiv preprint arXiv:1610.07650*.

Wei, Q., Bioucas-Dias, J., Dobigeon, N., and Tourneret, J. (2015a). Hyperspectral and multispectral image fusion based on a sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 53(7):3658–3668.

Wei, Q., Bioucas-Dias, J., Dobigeon, N., Tourneret, J., Chen, M., and Godsill, S. (2016). Multiband image fusion based on spectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7236–7249.

Wei, Q., Dobigeon, N., and Tourneret, J. (2015b). Fast fusion of multi-band images based on solving a sylvester equation. *IEEE Transactions on Image Processing*, 24(11):4109–4121.

Williamson, D. P. and Shmoys, D. B. (2011). *The design of approximation algorithms*. Cambridge university press.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227.

Wright, S. J., Nowak, R. D., and Figueiredo, M. A. (2009). Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7):2479–2493.

Xing, Z., Zhou, M., Castrodad, A., Sapiro, G., and Carin, L. (2012). Dictionary learning for noisy and incomplete hyperspectral images. *SIAM Journal on Imaging Sciences*, 5(1):33–56.

Xu, X., Li, G., and Gu, Y. (2020). Unraveling the veil of subspace rip through near-isometry on subspaces. *IEEE Transactions on Signal Processing*, 68:3117–3131.

Xu, Y., Du, B., Zhang, F., and Zhang, L. (2018). Hyperspectral image classification via a random patches network. *ISPRS journal of photogrammetry and remote sensing*, 142:344–357.

Yan, J. and Pollefeys, M. (2006). A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European conference on computer vision*, pages 94–106. Springer.

Yang, L., Zhang, R., Yang, S., and Jiao, L. (2020). Hyperspectral image classification via slice sparse coding tensor based classifier with compressive dimensionality reduction. *IEEE Access*, 8:145207–145215.

Yang, X., Ye, Y., Li, X., Lau, R. Y., Zhang, X., and Huang, X. (2018). Hyperspectral image classification with deep learning models. *IEEE Transactions on Geoscience and Remote Sensing*, 56(9):5408–5423.

Yokoya, N., Grohnfeldt, C., and Chanussot, J. (2017). Hyperspectral and multispectral data fusion: A comparative review of the recent literature. *IEEE Geoscience and Remote Sensing Magazine*, 5(2):29–56.

Yokoya, N. and Iwasaki, A. (2013). Hyperspectral and multispectral data fusion mission on hyperspectral imager suite (hisui). In *Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International*, pages 4086–4089. IEEE.

Yokoya, N., Yairi, T., and Iwasaki, A. (2012a). Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 50(2):528–537.

Yokoya, N., Yairi, T., and Iwasaki, A. (2012b). Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 50(2):528–537.

You, C., Li, C., Robinson, D. P., and Vidal, R. (2018). Scalable exemplar-based subspace clustering on class-imbalanced data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–83.

You, C., Li, C.-G., Robinson, D. P., and Vidal, R. (2016a). Oracle based active set algorithm for scalable elastic net subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3928–3937.

You, C., Robinson, D., and Vidal, R. (2016b). Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3918–3927.

Yue, J., Fang, L., Rahmani, H., and Ghamisi, P. (2021). Self-supervised learning with adaptive distillation for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*.

Zelnik-Manor, L. and Irani, M. (2003). Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–287. IEEE.

Zhai, H., Zhang, H., Xu, X., Zhang, L., and Li, P. (2017). Kernel sparse subspace clustering with a spatial max pooling operation for hyperspectral remote sensing data interpretation. *Remote Sensing*, 9(4):335.

Zhai, H., Zhang, H., Zhang, L., Li, P., and Plaza, A. (2016). A new sparse subspace clustering algorithm for hyperspectral remote sensing imagery. *IEEE Geoscience and Remote Sensing Letters*, 14(1):43–47.

Zhang, H., Zhai, H., Zhang, L., and Li, P. (2016). Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6):3672–3684.

Zhang, L., Zhang, L., Du, B., You, J., and Tao, D. (2019). Hyperspectral image unsupervised classification by robust manifold matrix factorization. *Information Sciences*, 485:154–169.

Zhang, T., Szlam, A., Wang, Y., and Lerman, G. (2012). Hybrid linear modeling via local best-fit flats. *International journal of computer vision*, 100(3):217–240.

Zhang, X., Chew, S. E., Xu, Z., and Cahill, N. D. (2015). Slic superpixels for efficient graph-based dimensionality reduction of hyperspectral imagery. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI*, volume 9472, page 947209. International Society for Optics and Photonics.

Zhang, Y. (2014). Spatial resolution enhancement of hyperspectral image based on the combination of spectral mixing model and observation model. In *Image and Signal Processing for Remote Sensing XX*, volume 9244, page 924405. International Society for Optics and Photonics.

Zhang, Z., Li, X., Zheng, S., Yao, M., Zheng, G., and Zhong, J. (2020). Image-free classification of fast-moving objects using "learned"; structured illumination and single-pixel detection. *Opt. Express*, 28(9):13269–13278.

Appendices

Appendix A. Extension to CSI supervised classification

Part of this chapter has been adapted from the conference paper (Hinojosa et al., 2019c) (published), the conference paper (Sanchez et al., 2021b) (published), and the journal paper (Hinojosa et al., 2022) (published).

Spectral-Spatial Classification from Multi-Sensor Compressive Measurements Using Superpixels

Spectral imaging senses 2D spatial information along multiple wavelengths. Traditional sensing techniques require scanning all the scene across multiple spectral bands to construct the three-dimensional (3D) data cube (Shaw and Burke, 2003). Based on the acquired spectral/spatial resolution, spectral imaging sensors can be categorized in Hyperspectral (HS) and Multispectral (MS). Typically, HS devices capture hundreds of spectral bands of the scene, however, its spatial resolution is often lower compared to that obtained with an MS sensor, which has a low spectral resolution (Yokoya et al., 2017). In this sense, fusion techniques provide a methodological framework for finding a high spatial and spectral resolution image using the information from both HS and MS images (Ghamisi et al., 2017b; Yokoya et al., 2017). After applying a fusion technique, the spectral signatures (a.k.a spectral pixels) are commonly used as classification features for identifying and detecting different materials within the high-resolution spectral data cube (Ghamisi et al., 2017a,b; He et al., 2018; Hinojosa et al., 2018c; Sanchez et al., 2019b).

In general, spectral image classification is a very difficult task due to the inherent data complexity, high storage requirements and computational costs. Therefore, a preprocessing step to reduce the dimensions of the spectral imagery is often required (He et al., 2016a; Rasti et al., 2016). On the other hand, compressive spectral imaging (CSI) has recently emerged as a new spectral imaging acquisition approach which captures compressive 2D measurements of the entire data cube rather than directly acquiring all the voxels, hence reducing the data dimensionality (Arce et al., 2014; Hinojosa et al., 2015; Correa et al., 2016; Cao et al., 2016). Under the context of CSI, two data fusion algorithms have been recently proposed (Vargas et al., 2019b,a). In particular, these methods reconstruct a high spatial and spectral resolution image from multi-sensor compressive measurements. Then, an intuitive approach of spectral image classification from multi-sensor compressive measurements would involve two tasks: the reconstruction of the fused image from the HS and MS compressive measurements, and finally, the classification procedure itself. However, this approach implies expensive computational costs and further, the classification accuracy might be degraded due to multiple factors (Yokoya et al., 2017; Arce et al., 2014).

Although most of the CSI literature has focused on improving the speed and accuracy of the spectral image reconstruction (Arce et al., 2014; Arguello and Arce, 2014; Garcia et al., 2018a; Diaz et al., 2019), the image recovery stage, in general, is not actually necessary for performing many signal processing tasks (Davenport et al., 2010; Li et al., 2013b; Hinojosa et al., 2018). Indeed, the aim of this work is to extract features and perform the spectral image classification directly from multi-sensor compressive measurements without requiring to recover the whole data cube, thus, the CSI reconstruction phase is completely avoided. After acquiring the HS and MS compressive measurements, the proposed method perform a rearrangement and extrapolation procedure from CSI HS measures in order to extract the spectral features. Consecutively, the CSI MS measurements are rearranged before applying a superpixel algorithm (Achanta et al., 2012) in order to obtain the spatial features. Specifically, using this segmentation technique, the proposed method groups pixels with similar spectral characteristics within a nearby neighborhood, incorporating spatial information and hence boosting the classification performance. Finally, the HS and MS extracted features are concatenated before being evaluated in a SVM classifier with a polynomial kernel.

Proposed Methodology.

Compressive measurements acquisition. In this work, the 3D-CASSI sensing approach is adopted (Cao et al., 2016). In general, denote $\mathscr{F}$ as the spatio-spectral data cube, with $M \times N$ spatial dimensions, L spectral bands and entries denoted as $\mathscr{F}_{m,n,k}$, where m and n index the spatial coordinates, and k determines the k-th spectral band. As shown in Fig. 11, the 3D-CASSI sensing scheme first modulates the voxels of the spectral scene using a 3D coded aperture $\mathbf{C}^s$, whose entries are indexed as $C^s_{m,n,k}$. Then, the coded spectral scene is integrated in the focal plane array (FPA) detector, along the spectral axis. In CSI it is possible to acquire $S \ll L$ measurement shots, each one employing a different coded aperture, such that different measurements of the spectral data cube are acquired. Therefore, the output of the sensing process, at the (m,n)-th detector pixel and a specific snapshot s, can be expressed as

$$\hat{Y}^s_{m,n} = \sum_{k=0}^{L-1} \mathscr{F}_{m,n,k} C^s_{m,n,k}. \tag{70}$$

Note that for each spatial location of $\mathbf{C}^s$, it is assigned a coding pattern (optical filter) $\phi^s \in \mathbb{R}^L$, with entries $(\phi^s)_k \in \{0,1\}$, that modulates a spectral pixel in that particular position before being integrated at the FPA detector. Further, observe that there is a finite number of optical filters randomly distributed in $\mathbf{C}^s$. In this work, the 3D coded apertures are built with non-overlapping optical filters which cover all the spectrum.

The set of compressive measurements from (70) can be arranged in a matrix $\hat{\mathbf{Y}}$ with dimensions $S \times (MN)$, where each column contains the compressive measurements associated to a particular spectral pixel. Note that, each row of $\hat{\mathbf{Y}}$ contains the compressive measurements acquired by the s-th snapshot. However, column vectors of $\hat{\mathbf{Y}}$ contain the spectral pixel measurements in unordered form since, in a particular snapshot, the spectral pixels are encoded using distinct coding patterns. Then, the matrix $\hat{\mathbf{Y}}$ is not convenient for classification as its structure makes difficult to discriminate among compressive measurements. Denoting P as the number of different coding patterns, if the number of measurements shots S equals P, the entries of $\hat{\mathbf{Y}}$ can be rearranged to form a new matrix $\mathbf{Y}$, such that each row contains the compressed information acquired with a specific coding pattern ϕ^s. Note that this rearrangement is only possible when $S = P$ since in this case it can be guaranteed that, at a specific snapshot, one pixel is encoded only once by a different coding pattern and, at the end of the sensing procedure, all pixels are encoded by the whole set of S coding patterns. Indeed, there are few optical filters in practice hence the case $S = P$ will efficiently extract the information from the underlying data cube. Formally, the rearrangement can be expressed as

$$Y_{s,j} = \hat{Y}_{s',j} \quad \text{if } \hat{Y}_{s',j} = (\phi^s)^T \mathbf{f}_j \quad \forall s',$$

for $s, s' = 0, \cdots, S-1$, where $\mathbf{f}_j \in \mathbb{R}^L$ denotes the j-th spectral signature for $j = 0, \cdots, MN-1$. This rearrangement, depicted in Fig. 15, preserves the structure of the underlying data improving the classification results. Alternatively, defining the matrix of S coding patterns as $\Phi = \left[\phi^0, \phi^1, \cdots \phi^{S-1}\right]^T$, the problem of acquiring and rearranging the measurements $\hat{\mathbf{Y}}$ can be succinctly expressed as follows

$$\mathbf{Y} = \Phi\mathbf{F}, \tag{71}$$

where $\mathbf{F} \in \mathbb{R}^{L\times(MN)}$ is the spectral image in matrix form, and $\Phi \in \mathbb{R}^{S\times L}$ can be viewed as the projection matrix. In this work the MS ($\mathbf{F_m}$) and HS ($\mathbf{F_h}$) images can be modeled from $\mathbf{F}$ as follows:

$$\mathbf{F_m} = \mathbf{D_m}\mathbf{F} \tag{72}$$

$$\mathbf{F_h} = \mathbf{F}\mathbf{D_h}, \tag{73}$$

where $\mathbf{D_m} \in \mathbb{R}^{L_m\times L}$ and $\mathbf{D_h} \in \mathbb{R}^{MN\times M_hN_h}$ are the spectral and spatial downsampling matrices, with downsampling factor q and p, respectively.

Features Extraction. **HS feature extraction** Following the CSI acquisition model described in Section 1, the compressive measurements acquired by the CSI hyperspectral sensor can be succinctly expressed as

$$\mathbf{Y_h} = \Phi_\mathbf{h}\mathbf{F_h}, \tag{74}$$

where $\mathbf{\Phi_h} \in \mathbb{R}^{S_h \times L}$ is the coding pattern matrix, $\mathbf{Y_h} \in \mathbb{R}^{S_h \times (M_h N_h)}$ contains the compressive measurements in an ordered form, with S_h denoting the number of measurement shots and M_h, N_h the spatial dimensions of the image acquired by the HS CSI sensor.

Although the matrix $\mathbf{Y_h}$ can be used as a feature matrix to perform classification, the goal of this work is to label a high spectral and spatial resolution image taking into account the rich spectral information embedded in $\mathbf{Y_h}$. Then, a spatial extrapolation is performed by replicating the columns of $\mathbf{Y_h}$, where the replicated pixels are located to the corresponding high-spatial resolution positions. The extrapolation process can be formulated as follows

$$\omega_{\mathbf{h}}^{j} = \mathbf{Y_h}^{\left(\lfloor \frac{j'}{p} \rfloor + \frac{M}{p}\left[\lfloor \frac{j'}{M} \rfloor + \lfloor \frac{j'}{Mp} \rfloor\right]\right)}, \tag{75}$$

where $\omega_{\mathbf{h}}^{j}$ is the j-th column of the HS classification features $\mathbf{\Omega_h}$; $\mathbf{Y_h}^{(j')}$ is the j'-th column of the ordered compressive measurements $\mathbf{Y_h}$; and $\lfloor x \rfloor$ returns the greatest integer less or equal than x.

MS feature extraction

Similarly, the compressive MS measurements are acquired as

$$\mathbf{Y_m} = \mathbf{\Phi_m} \mathbf{F_m}, \tag{76}$$

where $\mathbf{\Phi_m} \in \mathbb{R}^{S_m \times L}$ is the coding pattern matrix, whose rows contain the coding patterns of the optical filters used for acquiring the MS compressive measurements. To incorporate the spatial neighborhood information, this work uses a superpixel technique. Superpixel algorithms group pixels into perceptually meaningful atomic regions or segments. This captures image redundancy, provide a convenient primitive from which to compute image features, and greatly reduce the complexity of subsequent image processing tasks. In this work, the segmented image is created by applying the efficient simple linear iterative clustering (SLIC) algorithm (Achanta et al., 2012) on the MS compressive measurements, which are first rearranged back to a $M \times N \times S_m$ image. To reduce the computational cost, before the segmentation, principal component analysis is applied on the MS compressive measurements and the three principal components are used as the base image for the segmentation. Note that the PCA analysis is only necessary if the number of acquired shots S_m is greater than 3.

Once the segmentation map has been created, it can be utilized with the MS compressive measurements to extract the classification features. Specifically, denote $\mathbf{p}^e$ as the vector containing the indexes of all pixels belonging to the superpixel e, the columns of MS feature matrix $\Omega_{\mathbf{m}}$ are created as follows

$$\omega_{\mathbf{m}}^{\mathbf{p}^e} = \frac{\sum_{l=0}^{n_e-1} \mathbf{Y_m}^{(\mathbf{p}^e)_l}}{n_e}, \quad \text{for } e = 0, \cdots, N_{seg}, \tag{77}$$

where N_{seg} is the number of segments generated by the superpixel algorithm, $(\mathbf{p}^e)_l$ denotes the l-th entry of the $\mathbf{p}^e$ vector and $\omega_{\mathbf{m}}^{\mathbf{p}^e}$ represents the columns in $\Omega_{\mathbf{m}}$ indexed by the vector $\mathbf{p}^e$. Note that Eq. (77) simply replace all pixels in a segment e by the mean pixel. This procedure incorporates the spatial neighboring information of the superpixel in the classification method.

Finally, the feature matrices obtained from both MS and HS images $(\Omega_{\mathbf{m}}, \Omega_{\mathbf{h}})$ are stacked in order to build the feature matrix of the high spatial and spectral resolution image whose columns

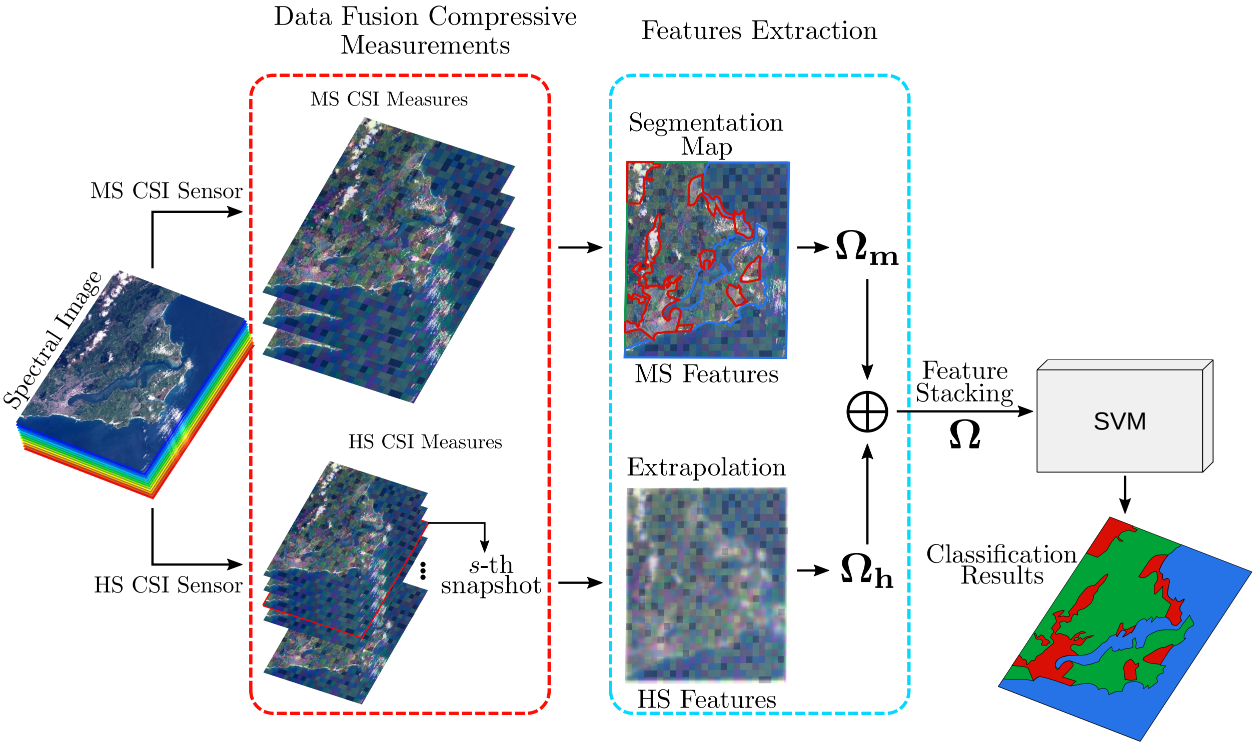


Figure 57. Workflow of the proposed CSI Spectral-Spatial classification approach.

shall be the input samples of a support vector machines (SVM) classifier. In other words, since the $\Omega_{\mathbf{m}}$ and $\Omega_{\mathbf{h}}$ have MN columns, the feature matrix is built as $\Omega = \left[\Omega_{\mathbf{h}}^T, \Omega_{\mathbf{m}}^T\right]^T$. In this work, the compression ratio is given by $\rho = \frac{S_h+S_m}{L}$. The flowchart of the proposed classification approach from multi-sensor compressive measurements is shown in Fig. 57.

Simulations and Results. The proposed classification method [4] was first evaluated on the Pavia University dataset, acquired by the Reflective Optics System Imaging Spectrometer (ROSIS-03) sensor (Computational Intelligence Group (GIC), 2011). This spectral image consists of 610 ×

---

[4] The MATLAB code along with some simulations can be found in `https://rebrand.ly/icip2019`.

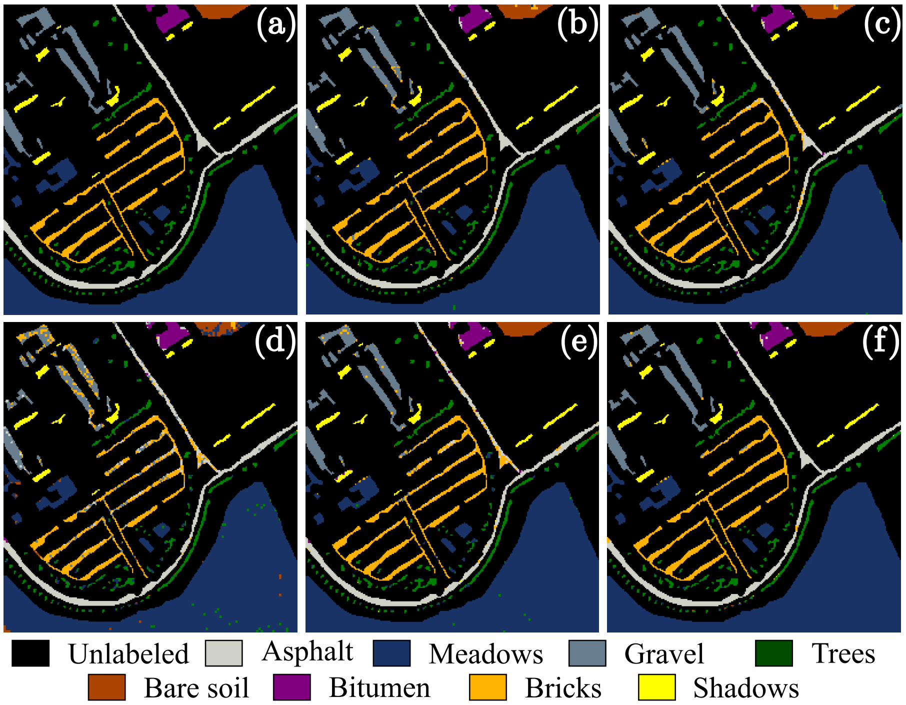
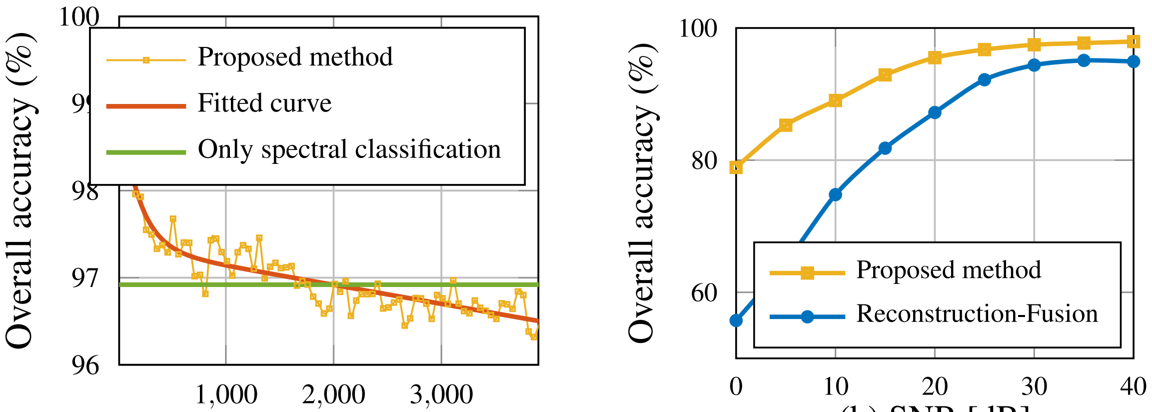


Figure 58. Classification maps on the Pavia University dataset. (a) Ground Truth. (b) Original Image, (c) Reconstruction-Fusion, (d) ACC, (e) Proposed-Noisy, (f) Proposed-Noiseless.

340 pixels and 103 spectral bands. In this work, a subset of this spectral image with dimensions $256 \times 256 \times 96$ was used in order to evaluate various classification approaches. Figure 58 (a) shows the ground-truth map of the Pavia University dataset, where each class corresponds to a distinct material in an urban cover. For all the experiments, a SVM classifier is trained using a polynomial kernel and 10% of the samples. In addition, the compression ratio of the acquired CSI measurements is set to $\rho = 25\%$.

In the first experiment, the number of superpixels is varied in order to analyze the impact of the clustering technique in the overall accuracy (OA). Specifically, Fig. 59 (A) shows the OA versus number of segments (N_{seg}) classification results on the University of Pavia image. As it can be observed, as the number of desired segments increases the overall accuracy decays exponentially,



(a) Number of Segments (b) SNR [dB]

Figure 59. (a) The overall accuracy of the proposed classification method versus the number of the superpixel segments and (b) the overall accuracy against the SNR of the compressive measurements.

as depicted with the fitted curve in red. In addition, note that, in general, the proposed method outperforms the spectral classification over the original image when using $N_{seg} \leq 1600$. Indeed, it can be seen from the figure that the incorporation of spatial neighboring information boost the classification accuracy up to 3% in OA, when the chosen number of superpixels is approximately the number of classes within the spectral scene. Therefore, the number of segments is fixed to $N_{seg} = 10$ for the subsequent experiments on the Pavia University dataset.

It is important to note that the measurements described by Eq. (71), in general, are noise free. However, in real CSI architecture, the compressive measurements are contaminated with noise due to the physical limitation of the sensor and imperfections of the acquisition system. Therefore, Eq. (71) should be rewritten as $\mathbf{Y} = \mathbf{\Phi}\mathbf{F} + \Sigma$, where $\Sigma \sim N(0, \sigma^2)$ represents the noise of the system. Figure 59 (B) shows the overall accuracy results of the proposed classification method versus the SNR of compressive measurements. For comparison purposes, the accuracy results of the Reconstruction-Fusion method is included in Fig. 59 (B). As can be seen in this figure, the proposed method outperforms the other method for the test interval.

Figures 58 (b), (c), (e) and (f) show the classification maps obtained when the SVM is applied on the original spectral image (Original Image), a reconstructed and fused spectral image from compressive measurements (Reconstruction-Fusion), the extracted features (Ω) from noisy compressive measurements (Proposed-Noisy), and the extracted features from noiseless compressive measurements (Proposed-Noiseless), respectively. In addition, the proposed method was compared with the Adaptive Compressed Classification (ACC) framework (Hahn et al., 2014), using SVM, in Fig. 58 (d). For the Reconstruction-Fusion classification approach, I implemented the spectral image reconstruction method developed in (Arguello and Arce, 2014) and the coupled nonnegative matrix factorization (CNMF) fusion technique (Yokoya et al., 2012a). For the Proposed-Noisy approach, the proposed method is applied to noisy compressive measurements with SNR = 25 dB.

In order to further validate the performance of the proposed approach, quantitative results are presented for the selected spectral scene in Table 30. All the presented results are the average of 10 experiments, each with different realizations of $\mathbf{\Phi_h}$ and $\mathbf{\Phi_m}$ coding patterns, and the best value of each row is shown in bold font. Furthermore, Table 30 shows the numerical results for each of the eight land-cover classes (producer's accuracy), overall accuracy (OA), average accuracy (AA), Kappa (κ) coefficients (Lillesand et al., 2015) and time. All the results, except the Kappa coefficients, are given in percentage.

From the Table 30, it can be clearly observed that the proposed classification approach pro-

Table 30
Performance of the various classification approaches on the Pavia University dataset.

Class	Original image	Recontruction-Fusion	ACC Framework	Proposed-Noisy	Proposed-Noiseless
Asphalt	86.80 ± 2.03	84.62 ± 1.05	91.20 ± 1.21	95.05 ± 4.62	**98.63** ± **0.70**
Meadows	99.07 ± 0.02	99.23 ± 0.22	95.78 ± 0.14	98.95 ± 0.17	**99.77** ± **0.02**
Gravel	82.39 ± 1.35	80.03 ± 6.67	79.62 ± 0.31	78.04 ± 4.58	**99.67** ± **0.16**
Trees	88.61 ± 2.41	91.55 ± 2.62	92.06 ± 0.27	86.86 ± 3.13	**93.35** ± **0.07**
Bare-Soil	61.96 ± 5.89	72.45 ± 5.89	85.57 ± 0.98	88.98 ± 6.46	**98.25** ± **2.47**
Bitumen	**93.29** ± **0.97**	90.82 ± 2.51	77.11 ± 0.16	93.70 ± 3.10	92.19 ± 0.97
Self-Block Bricks	90.40 ± 0.20	85.14 ± 3.19	83.16 ± 0.24	83.05 ± 1.19	**97.58** ± **1.03**
Shadows	**100.00** ± **0.00**	99.89 ± 0.15	98.47 ± 0.66	98.42 ± 0.74	98.74 ± 0.00
OA (%)	94.51 ± 0.35	94.05 ± 0.72	90.88 ± 0.43	94.55 ± 0.60	**98.90** ± **0.03**
AA (%)	87.81 ± 1.26	87.97 ± 0.01	87.87 ± 1.05	90.38 ± 0.86	**97.27** ± **0.40**
κ	0.91 ± 0.0062	0.90 ± 0.0119	0.88 ± 0.0147	0.91 ± 0.0105	**0.98** ± **0.0005**
Time (s)	1.17 ± 0.007	87.43 ± 1.77	24.97 ± 2.35	**0.66** ± **0.050**	0.74 ± 0.037

vide comparable results to applying the method directly on the original and reconstructed spectral data cube. In addition, performing the classification directly on the CSI measurements is significantly faster than performing all the processing in the complete and the reconstructed spectral data. As observed, the results show that when no noise is assumed, the achieved classification accuracy outperforms the results obtained with the full spectral data cube. As reported in some works, random projections are not sensitive to impulse noise thus can be used as a noise reduction method (Bingham and Mannila, 2001). In other words, the achieved results when using noise-free compressive measurements are due to the random projection, described in Eq. (71), removes noise from the acquired measurements. Finally, as clearly observed from Table 30, ACC is outperformed by the proposed method in both, noisy and noiseless case.

For the sake of completeness, I evaluate the proposed method on the Salinas dataset, captured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) (Computational Intelligence Group (GIC), 2011). This image has dimensions of 512×217 pixels and 204 spectral band. In this work, a subset of $217 \times 217 \times 192$ is used to evaluate the proposed classification method. Fig. 60(a) shows an RGB composite of the Salinas Valley spectral image and Fig. 60(b) shows the ground truth map of eight land-cover classes corresponding to different kinds of crops. Moreover, Fig. 60(c)-(f) show the classification maps obtained from the original image, the Reconstruction-Fusion approach, the proposed method using noisy compressive measurements (Proposed-Noisy), and the proposed method using noiseless compressive measurements (Proposed-Noiseless), respectively. As can be observed in these figures, the classification map generated by the proposed method over noiseless compressive measurements exhibits a superior performance compared with those yielded by the other approaches.

Figure 61 (a) shows the overall accuracy versus the compression ratio for the various classification approaches. Each point of these curves is obtaining by averaging 20 realizations of the respective experiment, and for each trial, a new set of colored coded apertures is built and a random set of training samples are selected. Furthermore, the rate of training samples is set to 10% and the number of super-pixels is fixed to $N_{seg} = 10$. As can be seen in this figure, the proposed classification approach using noiseless measurements has a remarkable performance in comparison with the other methods. Finally, Fig. 61 (b) shows the overall accuracy as the rate of training samples increases. As can be observed in this figure, the proposed method from noiseless measurements outperforms the other classification approaches for the entire evaluation interval, achieving an accuracy gain of at least 3%.

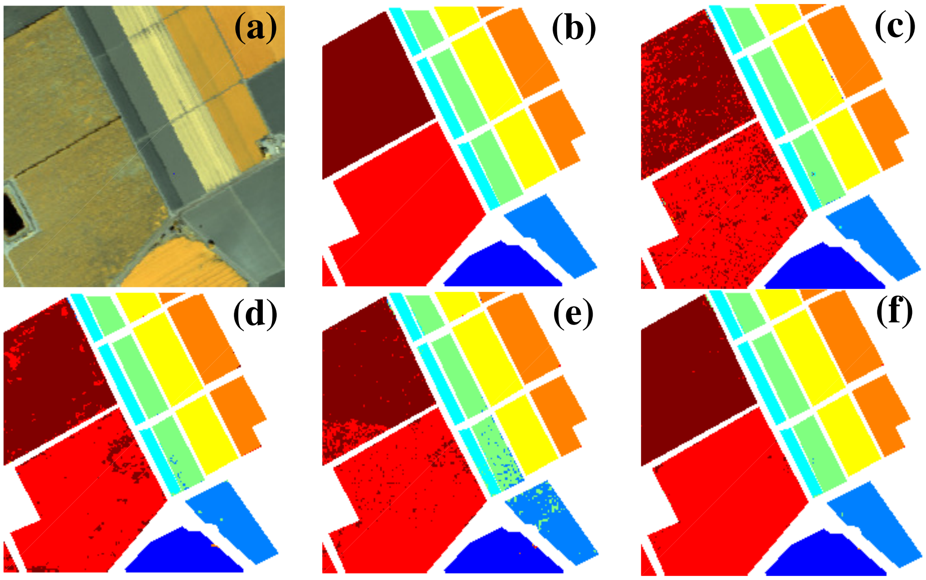


Figure 60. Salinas Valley dataset: (a) RGB composite of the spectral image, (b) Ground Truth. (c) Original Image, (d) Reconstruction-Fusion, (e) Proposed-Noisy, (f) Proposed-Noiseless.

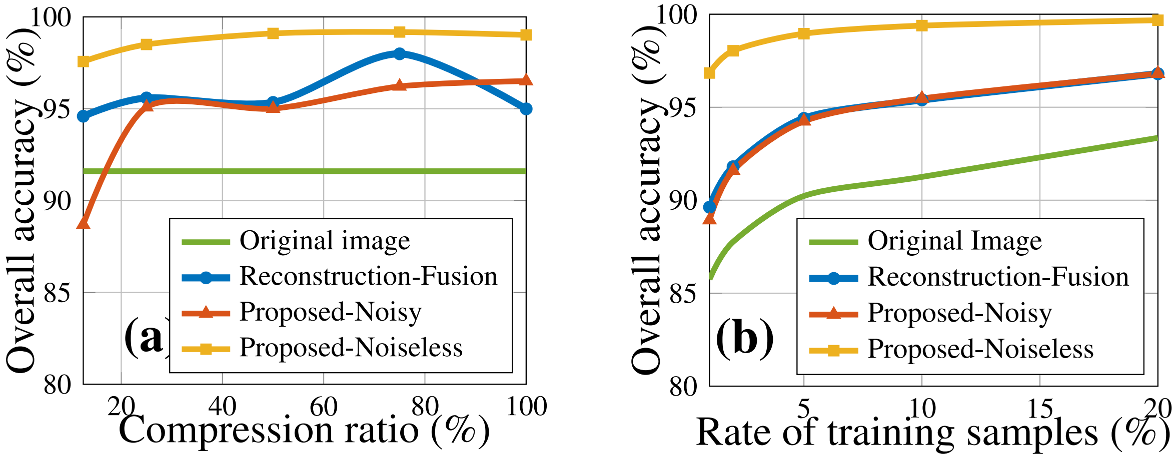


Figure 61. (a) The overall accuracy on the Salinas Valley dataset versus the compression ratio and (b) the overall accuracy versus the rate of training samples.

Conclusions. This work presented a spectral-spatial image classification approach, which perform all the processing tasks directly on multi-sensor compressive measurements. The proposed method performs a superpixel algorithm with the multispectral CSI measures in order to incorporate spatial neighboring information in the classification features. In addition, spectral and some spatial features are extracted from the hyperspectral CSI measures, using an extrapolation procedure. The proposed approach was validated trough some preliminary experiments. In general, the results show that performing the classification directly on the compressive measurements provides similar accuracy results, in a lesser time, compared with those provided by performing the classification on the original 3D spectral image and the reconstructed image pixels. Particularly, a maximum difference of approximately 4% in terms of OA was observed when comparing the classification results obtained using the original image with those achieved using the CSI multi-sensor measurements acquired with the 3D-CASSI sensing scheme.

C-3SPCD: Coded aperture similarity constrained design for Spatio-spectral

Spectral imaging (SI) is a technique used to acquire spatial information (2D) of a scene in many contiguous frequencies of the electromagnetic spectrum. The acquired data are typically regarded as a three-dimensional data cube, where two dimensions correspond to the spatial domain, and the third represents the spectral wavelengths.

Since points in a scene emit, absorb, scatter and reflect electromagnetic energy in different patterns according to their molecular composition, various materials and properties of the objects can be inferred from their detailed reflectance or spectral signatures of the pixels (a.k.a. spectral pixels). Therefore, acquisition systems for precise spectral measurements are useful in many applications such as agricultural assessment in remote sensing, mineral exploration, food quality control, biomedical imaging for noninvasive disease diagnosis, and surgical guidance (Manolakis and Shaw, 2002; Shaw and Burke, 2003; Lu and Fei, 2014). These applications usually require high-resolution images to better discriminate between specific details of the scene, which implies a massive amount of data, and therefore, high storage, transmission, and processing cost (Sanchez et al., 2019a).

Recently, compressive sampling (CS) theory has been applied to SI acquisition, leveraging compressive spectral imaging (CSI). CSI employs coding patterns to acquire 2D compressed projections of the 3D scene rather than direct measurements of all voxels (Lin et al., 2014a; Arce et al., 2014), which significantly reduces the sensing, storage, transmission and processing cost (Duarte et al., 2008; Arguello and Arce, 2014). Formally, given the vector form of a spectral image, $\mathbf{f} \in \mathbb{R}^{MNL}$ with $M \times N$ spatial pixels and L spectral bands, CSI assumes that such signal is S-sparse in a given basis Ψ, such that $\mathbf{f} = \Psi\theta$, with $S \ll MNL$. Then, the acquisition of K compressed projections, $\mathbf{g}$, of the 3D scene, can be expressed as $\mathbf{g} = \mathbf{H}\mathbf{f}$, where $\mathbf{H}$ is the sensing matrix of the system, and the coding patterns determine its entries. Using the compressed projections and taking advantage of the sparse representation of $\mathbf{f}$ on the transformation basis Ψ, an approximation of the spectral scene is obtained solving the optimization problem given by

$$\hat{\mathbf{f}} = \Psi \left\{ \underset{\theta}{\arg\min} \, ||\mathbf{H}\Psi\theta - \mathbf{g}||_2^2 + \lambda ||\theta||_1 \right\}, \tag{78}$$

where λ is a regularization parameter.

Related Works. In the last years, many CSI architectures have been developed. Some of the state-of-the-art architectures are the coded aperture snapshot spectral imager (CASSI) (Wagadarikar et al., 2008), the spatial-spectral encoded compressive hyperspectral imaging (SSCSI)

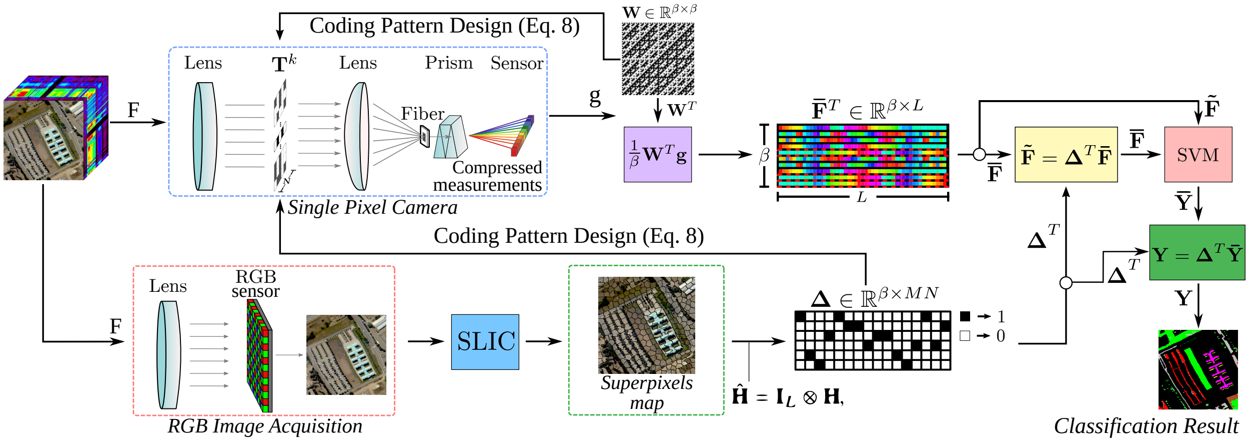


Figure 62. The workflow of the proposed SPC coding pattern design and compressed measurement classification. In general, the proposed method first acquires an RGB image and performs the SLIC (Achanta et al., 2012) algorithm to extract the superpixels map. Such a segmentation map is used altogether with a Hadamard matrix to design the coding patterns. Then, an SPC camera, with the designed patterns, is used to acquire the compressed measurements. Taking advantage of the coding pattern design approach, classification features can be extracted directly from the compressed measurements without the need to perform the scene reconstruction. Therefore, in this work, I use a SVM to classify the extracted features, and the full classification is obtained by upscaling the result using the superpixels information.

(Lin et al., 2014a), the dual-coded hyper-spectral imaging (DCSI) (Lin et al., 2014b). These architectures employ a 2D intensity detector that captures the projections of the spectral scene. However, in order to obtain high-resolution images, most applications require a high-resolution detector, which increases the implementation costs (Cao et al., 2016). On the other hand, it has been shown that the SPC architecture provides reliable spectral images with lower-cost hardware, when acquiring a large number of spectral bands, compared with 2D sensor systems (Li et al., 2017; Garcia et al., 2018a).

Typically, before performing any processing task, such as SI classification (Yang et al., 2020) and unmixing, CSI systems require to reconstruct the complete spectral scene, which is a non-required step in the proposed approach in this work. In general, the compressed measures acquisition process determines the quality of the reconstruction, which leads to high classification accuracy,an example is the work of Porta et al that use a random sampling matrix that satisfies the restricted isometry property (RIP) (Vargas and Arguello, 2019). Therefore, several works in CSI have focused on designing coding patterns as well as using additional information in order to improve the results of the reconstruction (Garcia et al., 2018a; Galvis et al., 2017; Hinojosa et al., 2018a). Nevertheless, solving the optimization problem described in (78) is computationally expensive, which also limits the applications of CSI.

Proposed Approach. Instead of designing a set of coding patterns to enhance the scene reconstruction, and then perform pixel classification, this works focuses on using an appropriate design of the SPC coding patterns to enable feature extraction directly from the compressed measurements and perform spectral pixel classification. In particular, the proposed method involves the acquisition of an RGB image from the scene as side information to design the SPC coding patterns. The design entails the use of the Simple Linear Iterative Clustering Algorithm (SLIC) (Achanta et al., 2012) to segment the image into superpixels. Then, this information is joined with a Hadamard matrix to construct the SPC sensing matrix (Sankaranarayanan et al., 2015). Our design approach allows incorporating the similarity information of neighboring pixels from the RGB image into the compressed measurements.

After the coding pattern design and the acquisition step, I use the fact that the transpose of a Hadamard matrix is closely related to its inverse to extract features from the compressed measurements. In this work, I employ a support vector machine classifier (SVM) to perform the classification on the extracted features (Della Porta et al., 2019). Finally, I use the information from the superpixel map to extrapolate the classification results obtaining the overall classification of the spectral pixels to allow a pixel-level classification, which is more powerful than a whole image label as presented in (Fu et al., 2020)-(Rizvi et al., 2020)-(Zhang et al., 2020). Figure 62 depicts the complete workflow of the proposed SPC coding pattern design and compressed classification. Through the manuscript, I will refer to the proposed coded aperture design and compressed-domain spatio-spectral classification approach, as C-3SPCD. This paper is organized as follows: Section 1 presents the mathematical description of the SPC employed in this work, along with the proposed coding patterns design. Then, Section 1 describes the complete algorithm for SPC compressed classification. In order to evaluate the performance of the proposed method, several simulation results are presented in Section 15, employing two remote sensing spectral datasets. Besides, an analysis of the impact of the number of superpixels in the overall classification results is presented in this section. To further verify the performance of the proposed classification method, a testbed implementation of the system was built in the laboratory, and experimental results are also included in Section 15.

SPC Sensing model and sensing matrix design. As observed from Fig. 62, the proposed C-3SPCD algorithm is mainly composed of two optical devices: an SPC architecture and an RGB sensor, using the same spatial resolution in the coded aperture in SPC and the RGB sensor, where an ideal response of the RGB sensor is assumed for all $L \geq 3$ spectral bands. Such systems employ different sampling strategies according to their optical configurations, allowing them to exploit distinct statistical properties of the spectral data, which leads to different sensing performance in terms of reconstruction quality, it is important to highlight that the performance depends on the reconstruction approach as the $l1$-based (Garcia et al., 2017), deep learning-based (Bacca et al., 2019a), and some other variants that assume smooth scenes (Deng et al., 2019). Then, assuming a corrected registration of the images acquired by each optical system, the linear model for the RGB sensing process is given by $\bar{\mathbf{g}} = \bar{\mathbf{H}}\mathbf{f}$, where $\bar{\mathbf{H}}$ is the sensing matrix of the RGB sensor. Notice that the acquired RGB image plays a crucial role in this work since it is used to design the coding patterns for the SPC. In this section, I first introduce the Single-pixel camera sensing model, and then I present the adopted design approach.

Single-Pixel Camera (SPC) Sensing Model. The single-pixel camera architecture is composed by a coding pattern $T^k_{(x,y)}$ that encodes the input light-rays in the same way for all spectral bands of the input scene $\mathbf{f}_{(x,y,\lambda)}$, where x,y are the spatial coordinates, λ accounts for the spectral wavelength, and k indexes each captured snapshot. From Fig. 62, note that in the proposed method, the coding patterns depend directly on the scene to be acquired, and it will be described in Section 1.

Mathematically, the spatial modulation in the SPC device for each band can be linearly represented as

$$f^k_{1(x,y,\lambda)} = f_{(x,y,\lambda)} T(x,y)^k, \tag{79}$$

where is evident that for this architecture the modulation is the same for all spectral wavelengths, later the modulated spectral image is condensed on a spatial point, which can be modeled as

$$g_{\lambda}^{k} = \int\int f_{1(x,y,\lambda)}^{k} dxdy, \tag{80}$$

later of the spatial condensation, the spectrum is acquired by a spectrometer, this discretization assume no overlapping bands as

$$g_{l}^{k} = \int_{\Lambda_l} g_(\lambda)^{k} rect\left(\frac{\lambda}{\Delta_e} - l\right) d\lambda, \tag{81}$$

where $\Lambda_l = \{\lambda | \Delta_e(l-1)/2 \leq \lambda \leq \Delta_e(l+1)/2\}$ represent the bandwidth of the l-th band, which can be rewrite in a discrete model as

$$g_{l}^{k} = (\mathbf{h}_k)^T \mathbf{f}_l, \tag{82}$$

for $k = 1,2,\ldots,K$, and $l = 1,2,\ldots,L$, where $(\mathbf{h}_k)^T$ is the vectorization of the k-th row of the coding pattern, and $\mathbf{f}_l$ is the vectorization of the l-th band of $\mathbf{F}$. The sensing model for all captured shots K for the l-th band can be expressed as

$$\mathbf{g}_l = \mathbf{H}\mathbf{f}_l, \tag{83}$$

where $\mathbf{g}_l = \left[g_l^1,\ldots,g_l^K\right]^T$, and $\mathbf{H} \in \mathbb{R}^{K\times MN}$ is given by $\mathbf{H} = [\mathbf{h}_1,\ldots,\mathbf{h}_K]^T$. On the other hand, the sensing matrix, representing the coding process for all bands, can be obtained as

$$\hat{\mathbf{H}} = \mathbf{I}_L \otimes \mathbf{H}, \tag{84}$$

where $\otimes$ represents the Kronecker product, and $\mathbf{I}_L$ is an $L \times L$ identity matrix, such that the resulting number of columns and rows of $\hat{\mathbf{H}}$ are MNL and KL, respectively. Then, the measurements for all bands are given by $\mathbf{g} = \left[(\mathbf{g}_1)^T, \ldots, (\mathbf{g}_L)^T\right]^T$, and the SPC sensing model for the full spectral image will be given by

$$\mathbf{g} = \hat{\mathbf{H}}\mathbf{f}. \tag{85}$$

Single-Pixel Sensing Matrix Design. In this work, the coding pattern design used depends on the scene to acquire and should satisfy

$$\mathbf{H}^T\mathbf{H}\mathbf{f}_l = \bar{\mathbf{f}}_l \approx \mathbf{f}_l, \tag{86}$$

where $\mathbf{H}^T\mathbf{H}$ cannot be an identity matrix because $\mathbf{H}$ is not full rank neither square (Sankaranarayanan et al., 2015). A solution to achieve the approximation in Eq. (86) is the use of superpixels in the sensing model, as previously shown in (Garcia et al., 2018a,b). Specifically, to design the data-dependent sensing matrix $\mathbf{H}$ in Eq. (86) a solution is that each snapshot captures a uniform region in the scene, i.e., take the same number of snapshots as spatial uniform regions, and on each snapshot, the coding pattern in the SPC only allows to pass the light coming from such regions. In general, the process to build the MR operator, denoted as Δ, starts by building a super-pixel map $\Gamma = \text{SLIC}(\bar{\mathbf{g}})$, whose entries are integer numbers that represent the label of the super-pixel to which each pixel is assigned. Note that the superpixels algorithm ensures that the pixels belonging to a super-pixel have a similar spectral signature. In this work, the SLIC algorithm (Achanta et al., 2010) is used to extract the superpixels. The map Γ is used to determine de MR operator Δ as in (Garcia et al., 2018b), where each row of Δ represents the formation of each super-pixel i.e. $\Delta \in \mathbb{R}^{\beta \times MN}$ for β superpixels. More specifically, Δ is obtained as

$$\Delta_{i,j} = \begin{cases} 1, & for \;\; vec(\mathbf{\Gamma})_j = i \\ \\ 0, & otherwise \end{cases}, \tag{87}$$

where $vec(\cdot)$ represents the vectorization operator, and $(\cdot)_j$ the j-th index of a vector.

It is important to note that the matrix Δ cannot be used directly because it implies very low transmittance in the coding patterns caused by few 1-valued entries per row of Δ, which state low signal-to-noise ratios in the sensing process. To solve this problem, an orthonormal matrix $\mathbf{W}$ is used to design $\mathbf{H}$ for each spectral band, which is scaled by $\sqrt{MN}$ to guarantee unit-norm columns as

$$\mathbf{H} = \left(\frac{1}{\sqrt{MN}}\right)\mathbf{W}\Delta. \tag{88}$$

The selection of $\mathbf{W}$ is critical in this work. In particular, I selected $\mathbf{W} \in \mathbb{R}^{\beta \times \beta}$ as a Hadamard matrix. Taking into account that the transpose of $\mathbf{W}$ is closely related to its inverse, i.e., $\mathbf{W}\mathbf{W}^T = \beta \mathbf{I}_\beta$, in the next section I propose a strategy to avoid CSI reconstruction and to acquire classification features directly from the compressed measurements.

Supervised Spatio-spectral classification model for SPC compressed measurements. Traditionally, after acquiring the SPC compressed measurements, it is required to perform the underlying high dimensional signal reconstruction using a computational expensive optimization algorithm (Baraniuk, 2007; Arce et al., 2014; Diaz et al., 2019). Once the estimation of the spectral scene is obtained, the labels of each spectral signature are determined by using a classifier such as SVM. However, taking advantage of the proposed coding patterns described in Section 1, it is possible to develop a strategy that focuses on classifying directly over the SPC measurements, not only boosting the computational performance but improving the classification accuracy. The complete workflow of the proposed C-3SPCD method is depicted in Fig. 62.

Recalling Eqs. (83) and (88), and considering that the transpose of $\mathbf{W}$ is closely related to its inverse, it is possible to obtain an estimation of the l-th band, $\bar{\mathbf{f}}_l$, of the spectral scene as

$$\bar{\mathbf{f}}_l = (1/\beta)\mathbf{W}^T\mathbf{g}_l = (1/\beta)\mathbf{W}^T\mathbf{W}\Delta\mathbf{f}_l = \Delta\mathbf{f}_l, \tag{89}$$

for each spectral band l. Note that, the product $\Delta\mathbf{f}_l$ decimates the vector $\mathbf{f}_l$ by integrating similar pixels into a single super-pixel. It can be seen as a feature extraction preprocessing step for classification, which takes into account the pixel neighboring information. In this sense, denoting

$$\bar{\mathbf{F}} = \left[\bar{\mathbf{f}}_1, \cdots, \bar{\mathbf{f}}_L\right] \in \mathbb{R}^{\beta \times L}, \tag{90}$$

as the feature matrix, which contains the spectral information of each super-pixel, the classification process reduces to determine the class label of the β spectral superpixels (rows of $\bar{\mathbf{F}}$). Then, the pixels in $\mathbf{F}$ are assigned the same label of the super-pixel to which they belong. Specifically, denoting as $\bar{\mathbf{Y}} \in \mathbb{Z}^{\beta}$ the result of classifying the rows of $\bar{\mathbf{F}}$, the final classification of $\mathbf{F}$ is given by

$$\mathbf{Y} = \Delta^T \bar{\mathbf{Y}}. \tag{91}$$

In this work, the classification is achieved using the support vector machine (SVM) classifier. Specifically, let denote $\zeta = \{\zeta_1, \cdots, \zeta_c\}$ as the set of c class labels, and $\mathscr{T} = \{\tau_1, \cdots, \tau_v\}$ as the set of v integers indexing the training samples. Then, let first consider the binary supervised classification problem, where each feature vector of the training set is associated with a binary output $\mathscr{Z}_{\mathscr{T}} \in \{-1, 1\}$. In this context, the SVM focuses on determining a hyperplane that optimally separates the samples belonging to different classes. This problem is equivalent to minimize the norm of a weight vector ω that is normal to the decision hyperplane (Mountrakis et al., 2011; Li et al., 2013a). Specifically, this classification method solves the following optimization problem

$$\begin{aligned} &\min_{\omega,\omega_0,\mathscr{X}_{\mathscr{T}}} \left\{ \frac{1}{2}\|\omega\|_2^2 + \eta \sum_{\mathscr{T}} \mathscr{X}_{\mathscr{T}} \right\} \\ &\text{s.t.} \quad \mathscr{Z}_{\mathscr{T}} \left(\varepsilon \left(\tilde{\mathbf{f}}_{\mathscr{T}} \right)^T \omega + \omega_0 \right) \geq 1 - \mathscr{X}_{\mathscr{T}} \\ &\qquad\qquad\qquad\qquad \mathscr{X}_{\mathscr{T}} \geq 0, \end{aligned} \tag{92}$$

for all $\mathscr{T} = \{\tau_1, \cdots, \tau_v\}$, where $\tilde{\mathbf{f}}_{\mathscr{T}}$ are the $\mathscr{T}$ row vectors of $\tilde{\mathbf{F}} = \Delta^T \bar{\mathbf{F}}$ used as training data; $\mathscr{X}_{\mathscr{T}}$ is the set of slack variables that considers the non separability between sets belonging to different classes; η is the regularization parameter that controls the influence of misclassified samples; ε is a function that maps the feature vector to the kernel space; ω_0 is the offset of the decision hyperplane with respect to the origin of the coordinate system (Schlkopf et al., 2018).

As mentioned before, SVMs are basically binary classifiers. However, in this work, the one-against-one multi-class strategy is adopted, where two classes are separately analyzed by implementing a binary classifier, ignoring, at the same time, the remaining classes (Gu and Sheng, 2017). Finally, using the training sample set $\{\tilde{\mathbf{f}}_{\mathscr{T}}\}_{\mathscr{T}=\tau_1}^{\tau_v}$, the problem of classifying each test pixel consists of finding the class ζ whose training pixel $\mathscr{T}$ is the nearest to the test pixel in the Euclidean distance

$$\bar{Y}_{j_s} = \min_{\zeta=\{\zeta_1,\cdots,\zeta_c\}} \|\bar{\mathbf{f}}_{j_s} - (\tilde{\mathbf{f}}_{\mathscr{T}})_\zeta\|_2^2, \tag{93}$$

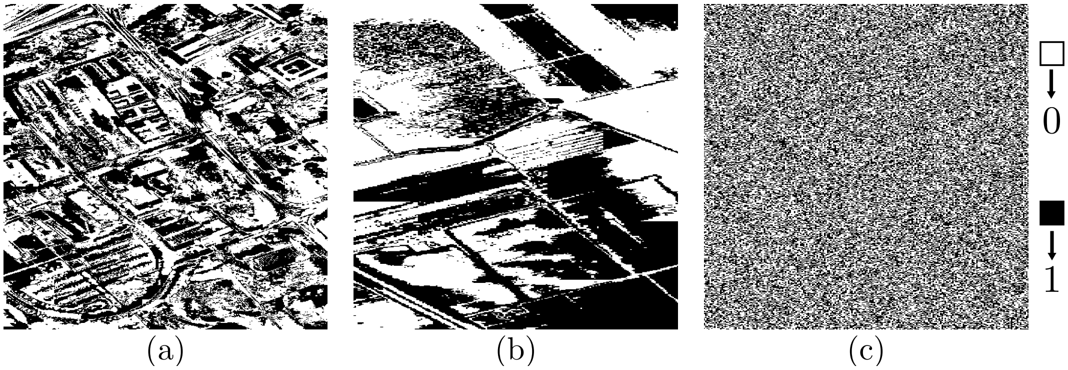
where j_s indexes the rows of $\bar{\mathbf{F}}$, and $(\tilde{\mathbf{f}}_{\mathscr{T}})_\zeta$ denotes the training pixels belonging to the class ζ. Notice that the SVM is trained using the rows of $\tilde{\mathbf{F}} \in \mathbb{R}^{MN\times L}$ (the spectral image estimation), and the classification is performed on the β feature vectors, i.e., rows of $\bar{\mathbf{F}}$.

Algorithm 10 summarizes the proposed method that takes into account the arrays Δ and $\bar{\mathbf{F}}$. The C-3SPCD method has mainly three advantages:

- The classification is performed directly on the SPC measurements $\mathbf{g}$, avoiding the expensive signal reconstruction which involves solving the problem in Eq. (78).
- As the number of superpixels $\beta \ll MN$, the proposed approach boosts the computational performance. Indeed, the complexity of the proposed compressed-domain classification algorithm is given by the complexity of the SLIC, which takes $O(MN)$ (Achanta et al., 2012),

Algorithm 10 Coded aperture similarity constrained design for Spatio-spectral Classification of Single-Pixel Measurements (C-3SPCD)

```
Input : Spectral image in vector form $\mathbf{f}$, RGB sensing matrix $\bar{\mathbf{H}}$, number of superpixels $\beta$, class labels $c$, number of
        training samples $v$.
Super-pixels Extraction:
$\bar{\mathbf{g}} = \bar{\mathbf{H}}\mathbf{f}$ ▷ Obtain the RGB image
$\Delta \leftarrow \text{SLIC}(\bar{\mathbf{g}}, \beta)$ ▷ $\Delta \in \mathbb{R}^{\beta\times MN}$ is given by Eq. (87)
Coding patterns Design:
$\mathbf{H} = \mathbf{W}\Delta$ ▷ Where $\mathbf{W}$ is a $\mathbb{R}^{\beta\times\beta}$ Hadamard matrix
SPC Measurement Acquisition:
$\mathbf{g}_l = \mathbf{H}\mathbf{f}_l$ ▷ $\mathbf{f}_l$ is the $l$-th spectral band in vector form.
Feature Extraction and Classification:
$\bar{\mathbf{F}} = (1/\beta)\mathbf{W}^{\mathbf{T}}\mathbf{g} = [\Delta\mathbf{f}_1,\cdots,\Delta\mathbf{f}_L] = [\bar{\mathbf{f}}_1,\cdots,\bar{\mathbf{f}}_L]$ ▷ $\bar{\mathbf{F}} \in \mathbb{R}^{\beta\times L}$
$\tilde{\mathbf{F}} = \Delta^T\bar{\mathbf{F}}$
$\zeta = \{\zeta_1,\cdots,\zeta_c\}$ ▷ Establish the Set of $c$ class labels
$\mathscr{T} = \{\tau_1,\cdots,\tau_v\}$ ▷ Index set of training samples (rows of $\tilde{\mathbf{F}}$)
Train a SVM classifier using Eq. (92)
Classify the features $\bar{\mathbf{f}}$ using Eq. (93)
Obtain the final classification result using Eq. (91)
```

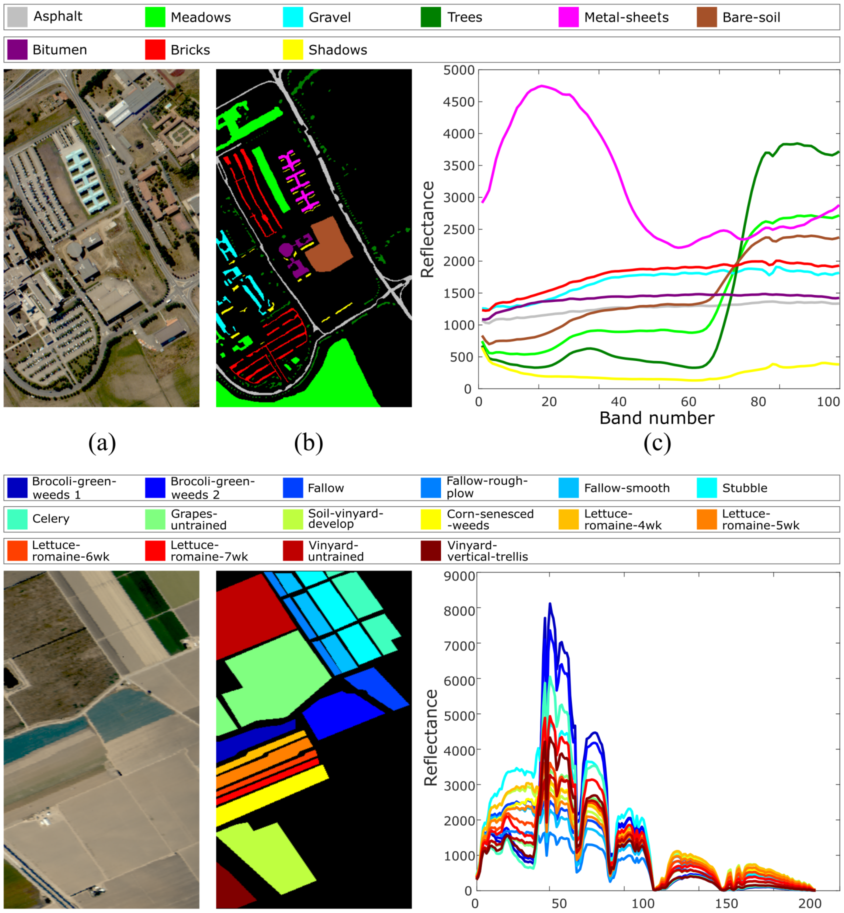


(a) (b) (c)

Figure 63. Examples of the designed similarity-constrained coded apertures for two different datasets (a) and (b) using Algorithm 10. An example of a random coded aperture is shown in (c) for reference.

and the complexity of the matrix multiplication $\Delta\bar{\mathbf{F}}$ which is $O(\beta MNL)$, and can be solved faster by state-of-the-art algorithms (Van Zee and Van De Geijn, 2015).

- Due to the use of superpixels, the neighboring spatial information is taken into account to extract the classification features; hence improving the accuracy.



(a) (b) (c)

(d) (e) (f)

Figure 64. RGB false-color images, ground truth maps, and spectral signatures of each land-cover class for (a)-(c) Pavia University and (d)-(f) Salinas Valley datasets, respectively.

Figure 63 show two examples of coded apertures obtained with the proposed design for the datasets shown in Fig. 64. We also show an example of a random coded aperture for reference in Fig. 63 (c). From the figures, I observed that the super-pixel map guides the design of the coded apertures.

Simulations and Results. This section presents the numerical results of the proposed method[5] for supervised classification of features extracted from SPC measurements using two remote sensing spectral datasets. The proposed workflow was implemented in MatLab (version 2018b), and all numerical experiments were performed on a computer with an Intel Xeon E5-1603 v3 at 2.80 GHz and 128-GB RAM. The used SVM implementation in this work adopts an error-correcting output codes (ECOC) model with the one-against-one strategy, which uses a default linear kernel function in each binary classification subproblem.

Data Description. The first dataset is the *Pavia University*, which was acquired over an urban area of northern Italy by the Reflective Optics System Imaging Spectrometer (ROSIS). The image consists of 103 spectral bands ranging from $0.43 - 0.84$ μm and exhibits 610×340 pixels with a spatial resolution of 1.3 meters per pixel. Figures 64 (a) - (c) show an RGB false-color composite of the dataset, along with the ground truth, and the characteristic spectral signatures of each class, respectively. The ground truth differentiates nine land-cover classes: asphalt, meadows, gravel, trees, metal sheets, bare soil, bitumen, bricks, and shadows.

The second spectral image used is the *Salinas Valley*; this dataset was collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over the Salinas Valley, California, USA. The size of the dataset is 512×217 pixels and 204 spectral bands across the spectral range from $0.24 - 2.40$ μm, with a spatial resolution of 3.7 meters per pixel. The Salinas ground truth con-

---

[5] A MatLab implementation of the proposed method can be found at `http://link.carloshinojosa.me/spatio-spectral-spc-classification`

tains 16 land-cover classes: broccoli green weeds 1, broccoli green weeds 2, fallow, fallow rough plow, fallow smooth, stubble, celery, grapes untrained, soil vineyard develop, corn senesced green weeds, lettuce romaine 4 weeks, lettuce romaine 5 weeks, lettuce romaine 6 weeks, lettuce romaine 7 weeks, vineyard untrained, and vineyard vertical trellis. An RGB false-color image of the dataset, the ground truth, and its spectral signatures are shown in Figs. 64 (d) - (f), respectively. The black pixels from the ground-truth maps are unlabeled regions, which are not considered in the classification results. For the experiments, I select the 5% of each class as training samples, and the remaining spectral pixels are used for testing.

Evaluation Metrics. To compare the classification performance of the proposed method, I use four standard metrics: users accuracy (UA), average accuracy (AA), overall accuracy (OA), and Kappa coefficient (Lillesand et al., 2015; Strehl and Ghosh, 2002; Schütze et al., 2007). In particular, UA, AA, OA, and Kappa coefficient can be obtained by means of an error matrix (a.k.a confusion matrix). Error matrices compare, on a category-by-category basis, the relationship between known reference data (ground truth) and the corresponding results of automated classification. Such matrices are square, with the number of rows and columns equal to the number of categories whose classification accuracy is being assessed.

- **Overall Accuracy (OA):** is computed by dividing the total number of correctly classified pixels (i.e., the sum of the elements along the major diagonal) by the total number of reference pixels.

- **User’s Accuracies (UA):** It is the accuracy of individual categories, which are computed by dividing the number of correctly classified pixels in each category by the total number of pixels that were classified in that category (the row total).

- **Average Accuracy (AA):** is the mean of User's accuracies.
- **Kappa Coefficient (κ):** is a measure of the difference between the actual agreement between reference data and an automated classifier and the chance agreement between the reference data and a random classifier. Specifically given an error matrix **Q** with r_q rows and N_q number of observations, the Kappa coefficient is given by

$$\text{Kappa} = \frac{N_q \sum_{i=1}^{r_q} Q_{ii} - \sum_{i=1}^{r_q} (Q_{i+} \cdot Q_{+i})}{N_q^2 - \sum_{i=1}^{r_q} (Q_{i+} \cdot Q_{+i})}, \tag{94}$$

where Q_{ii} is the number of observations in row i and column i (on the major diagonal); Q_{i+} is the total observations in row i (shown as marginal total to right of the matrix); and Q_{+i} is the total observations in column i (shown as marginal total at bottom of the matrix).

Approaches for SPC Measurements Classification. In general, traditional approaches to perform spectral pixel classification using compressed measurements requires first to reconstruct the underlying spectral image data cube using Eq. 78. Since the classification accuracy is directly affected by the quality of the reconstruction, different works have focused on designing strategies to enhance the reconstruction (Garcia et al., 2018a,b; Arce et al., 2014; Correa et al., 2016; Diaz et al., 2019). For instance, recent work in (Garcia et al., 2018a) proposes two approaches to improve the reconstruction of the underlying spectral image using SPC measurements and an RGB

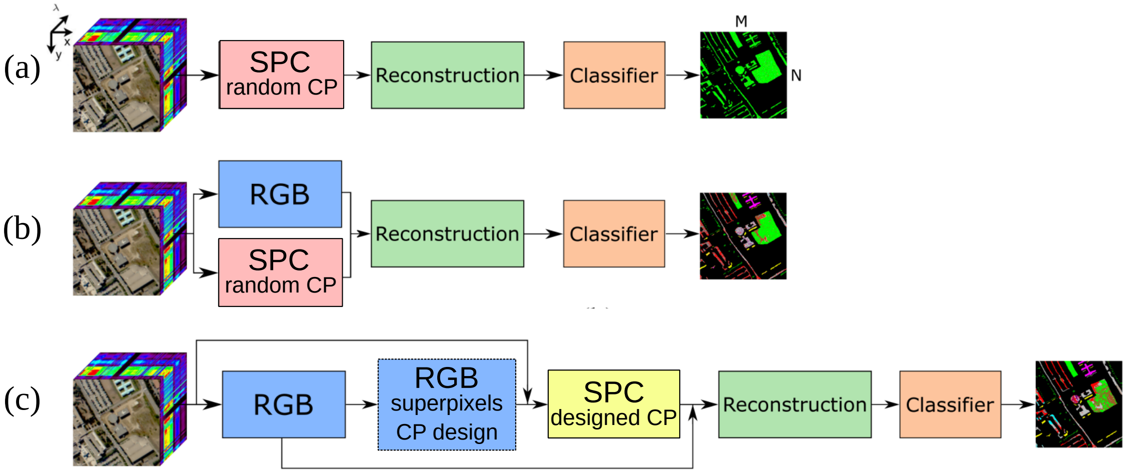


Figure 65. Flow diagram of classification approaches when reconstructing the SPC measurements using: (a) random CP (R-SPC), (b) random CP and stacking the RGB image with compressed measurements (R-SPCRGB1), and (c) the proposed designed CP and stacking the RGB image with the acquired measurements (R-SPCRGB2).

image that incorporates spatial information in the inverse problem. Specifically, one approach to performing reconstruction with side information is to stack the RGB measurements with the SPC measurements acquired with a random coding pattern and use the Eq. 78; this approach is shown in Fig. 65 (b), and I refer to it as R-SPCRGB1. Similarly, the second method also stacks the RGB and SPC measurements, but the acquisition is performed with a designed coding pattern instead of a random one, and the design strategy takes into account the RGB image information; this approach is shown in Fig. 65 (c), and I refer to it as R-SPCRGB2. It is important to notice that all of these methods require to reconstruct the underlying spectral image, solving an expensive inverse problem before performing the pixels classification. In the following subsections, I will compare the performance of the proposed C-3SPCD algorithm, which design the coded aperture and performs all the feature extraction and classification directly on the compressed domain, avoiding the scene reconstruction. For completeness, I will also compare the proposed method against the traditional approach of performing the reconstruction, followed by the classification, using only the SPC measurements acquired with a random coding pattern; I refer to it as R-SPC, and it is depicted in Fig. 65 (a). In these approaches, the reconstruction is performed using the Sparse Reconstruction by Separable Approximation (SpaRSA) algorithm (Wright et al., 2009). The basis representation used in this work is $\Phi = \Phi_1 \otimes \Phi_2$, where Φ_1 is a 2D wavelet Symmlet 8 basis, and Φ_2 is a 1D discrete cosine transform (DCT).

Impact of the Number of Superpixels in the Classification Accuracy. In the first experiment, the number of superpixels of the RGB image was varied to analyze its impact on the overall classification accuracy. Note that, since the superpixels extracted from the RGB image are used to design the coding pattern, which, at the same time, is used to acquire the SPC measurements, the number of segments β is directly related to the compression ratio. In the sensing model adopted in this paper, the compression ratio is given by $\gamma = \frac{\beta}{MN}$, where $\gamma \in [0,1]$. Also, note that due to the proposed design of $\mathbf{H}$ given in Eq. (88), $K = \beta$ in my experimental settings. The experiment was executed for values of $\beta = 2^n$ with n from 4 to 12 in steps of 1, i.e., for $\beta = 16, 32, 64, 128, 256, 512, 1024, 2048$, and 4096. The obtained results are plotted in Fig. 66, where the methods are compared in terms of OA. Note that, since $K = \beta$, the results of R-SPC and R-SPCRGB1 vary as β increases.

Figure 66 (a) presents the obtained results for the Pavia University. Note that, when the number of superpixels increases, the performance of the classification with the proposed method (blue line with circles) improves. However, in some cases the number of superpixels have a maximum performance point to go down and stabilize, it is usually because when the number of su-

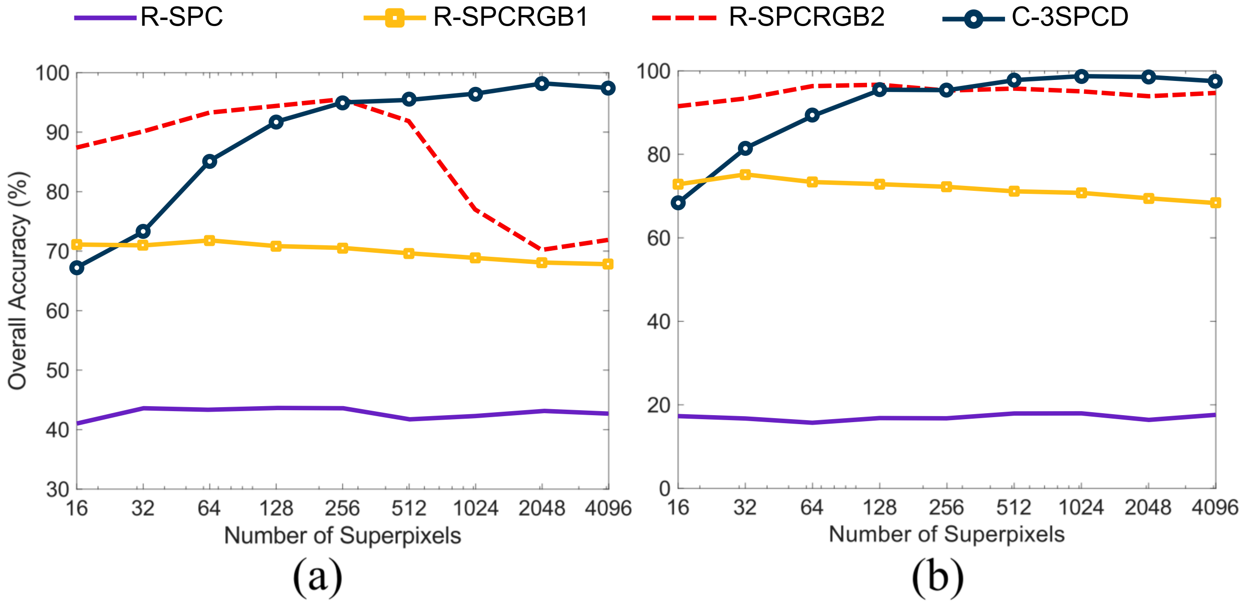


Figure 66. Overall accuracy of the classification by varying the number of superpixels (β) in (a) Pavia University and (b) Salinas Valley datasets.

perpixels increases the algorithm could have behavior similar to the traditional non-superpixels approaches (Garcia et al., 2011), until $\beta \approx 2048$ where the overall accuracy stabilizes at 98.61%. Besides, observe that R-SPCRGB2 Design (dotted red line) overcome the proposed method when β ranges between 16 and 256 superpixels. Then, in the Pavia University dataset, both techniques which implement designed coding patterns, instead of a random one, provide the highest accuracy results. Similarly, Fig. 66 (b) shows the results for the Salinas Valley dataset. The OA improves as the β increases until it reaches an OA of $\approx 99.6\%$, after 512 superpixels. Also, R-SPCRGB2 Design overcome the proposed method when $16 \leq \beta \leq 128$. Note that although R-SPCRGB2 Design provides the best results for some values of β in both datasets, the total time spent from the acquisition of the measurement to classification is significantly less by using the proposed method,

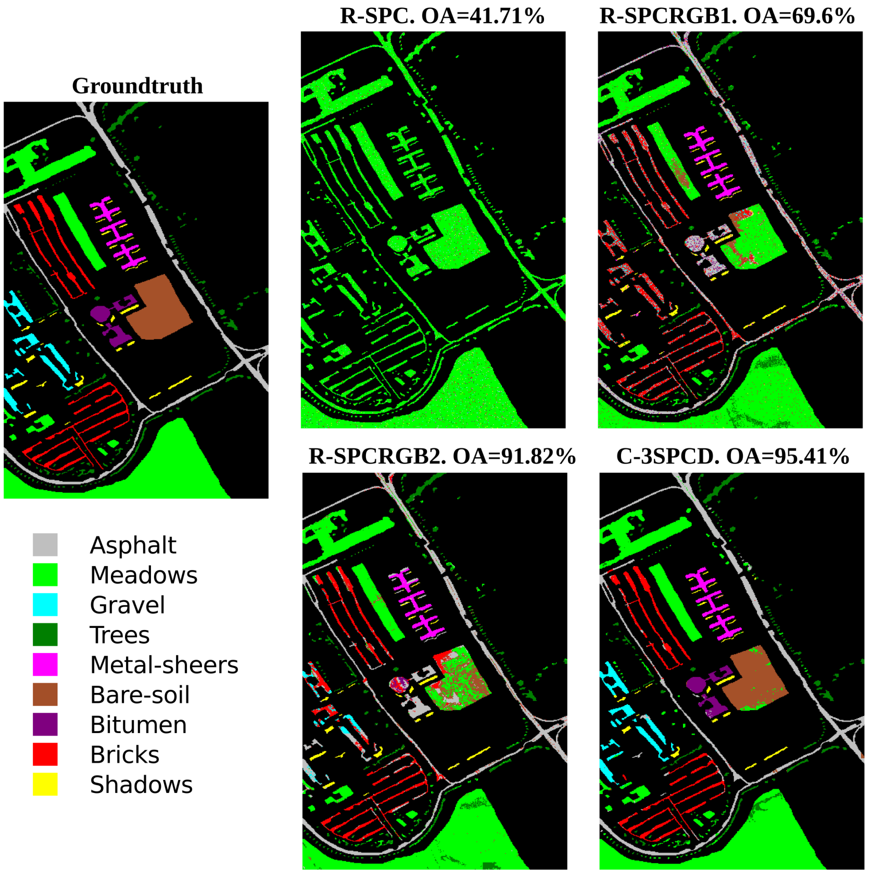


Figure 67. Visual classification results on Pavia Center dataset.

as shown in the next experiments. Besides, the low accuracy obtained by R-SPC and R-SPCRGB1, in both datasets, allows us to conclude that taking into account the spatial information provided by the RGB image directly in the SPC sensing procedure is crucial for the classification. It is important to notice that explicitly determining the β parameter for an image it is not a straightforward task as it can significantly vary depending on the different image characteristics. However, if the number of classes c within the image is known in advance, then an initial value for β can be set as $\beta = q_0 c$, with q_0 being a constant.

Table 31

Quantitative results on Pavia University dataset

Evaluation Metrics	**R-SPC**	**R-SPCRGB1**	**R-SPCRGB2**	**C-3SPCD**
Asphalt	2.35	74.56	90.79	**97.50**
Meadows	93.45	89.40	**97.81**	97.66
Gravel	1.14	17.63	78.35	**92.73**
Trees	1.60	70.46	**90.34**	83.67
Metal sheets	0.89	97.92	97.55	**99.22**
Bare soil	2.07	16.01	74.47	**96.52**
Bitumen	1.50	15.64	96.20	**97.59**
Bricks	0.92	67.76	90.96	**93.14**
Shadows	1.80	86.17	**96.88**	74.76
AA	11.75	59.51	90.37	**92.53**
OA	41.71	69.63	91.82	**95.41**
Kappa	0.87	58.48	89.06	**93.89**
SLIC Time	-	-	8.30	8.30
Reconstruction Time	2370.08	2644.08	3201.77	**0.00**
Classification Time	88.74	16.51	14.01	**1.83**
Total Time	2458.81	2660.58	3224.14	**10.13**

Visual Maps and Quantitative Results. To further validate the classification performance of the proposed method, visual and quantitative results are presented in Figs. 67 and 68, and Tables 31 and 32, respectively. In all the results shown below, the input parameter β is fixed to 512. In addition to the classification results measured using the evaluation metrics explained in section 15, the SLIC algorithm execution time, the SPC acquisition procedure, the reconstruction stage, and the classification algorithm times are also included in the tables. Furthermore, the optimal values are shown in bold font, and the second-best results are underlined. All the numerical results are given in percentage, except for the running time, which is given in seconds, and the Kappa coefficients. The presented results in the tables are the average of 15 realizations of each experiment, and presented visual results correspond to the best realization of each method.

From Fig. 67 and Table 31, it can be seen that the proposed classification approach (C-3SPCD) provides the best accuracy results for the Pavia University dataset. The results obtained with the R-SPCRGB2 Design also provide good OA results; however, the total time of this approach is much higher than the proposed method, due to the additional time of the reconstruction stage. Finally, Fig. 68 and Table 32 present the visual and numerical results on the Salinas Valley dataset. In the test executed on this scene, a high similarity between the classification accuracy achieved with the R-SPCRGB2 Design and the proposed method C-3SPCD is appreciated. Even though the best OA is attained by the C-3SPCD method, the best AA is obtained by the R-SPCRGB2 Design approach.

Note that the proposed method achieves classification accuracy results superior to those obtained with the other approaches, even though it does not stack the RGB image with the SPC measurements, as it happens in the second and third approach shown in Fig. 65. In contrast, the side information extracted as superpixels of the RGB image is used optimally to design the coding patterns to acquire the SPC measurements, which incorporate the spatial information, and allows performing the classification directly on the compressed domain. Besides, for both datasets, the proposed method takes significantly less total execution time than the others. The behavior of time is expected, mainly due to the absence of the reconstruction step (Reconstruction Time in Tables). Also, the classification time is less with the proposed approach. This is due to the proposed method C-3SPCD only need to classify β featured pixels instead of MN spectral pixels, and then Eq. (91) is used to obtain the final classification.

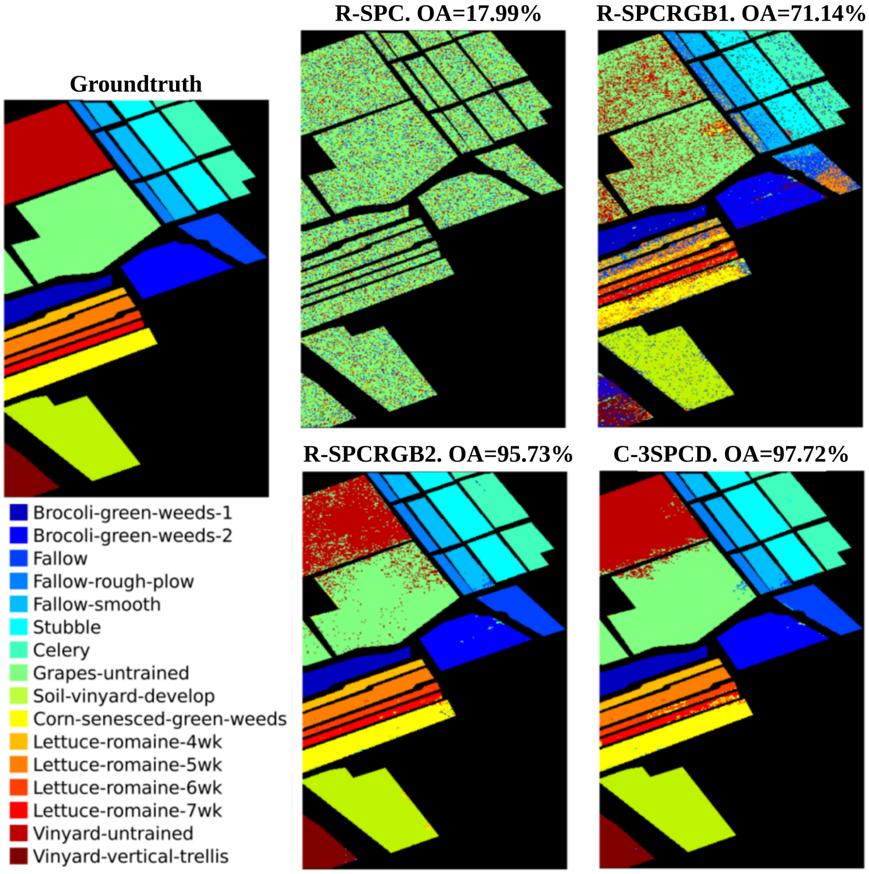


Figure 68. Visual classification results on the Salinas Valley dataset.

Experimental Testing.

Architecture Setup and Data Acquisition. A testbed that follows the workflow shown in Figure 62 was built in the laboratory to evaluate the proposed classification approach. This prototype, shown in Fig 69, uses a 100 mm objective lens to guide the input light-rays to be encoded by a Texas Instrument digital micromirror device (DMD). A relay lens, formed by two 100 mm lenses, is placed between the DMD and the non-polarizing beam splitter used to enable the acquisition of the SPC and the RGB measurements of the same scene. Precisely, the arm that acquires the SPC measurement consists of a 30 mm lens in conjunction with an F220SMA-A as

Table 32

Quantitative results on Salinas Valley dataset

Evaluation Metrics	**R-SPC**	**R-SPCRGB1**	**R-SPCRGB2**	**C-3SPCD**
Broccoli-green-1	2.79	92.68	**99.85**	99.30
Broccoli-green-2	3.17	89.29	99.09	**99.60**
Fallow	2.83	48.79	99.85	**98.94**
Fallow-rough-plow	2.44	60.55	98.78	**99.07**
Fallow-smooth	4.11	87.27	**97.31**	96.60
Stubble	4.45	92.47	**99.87**	99.37
Celery	3.13	90.44	**99.64**	99.50
Grapes-untrained	67.03	73.13	92.40	**97.03**
Soil-vineyard-develop	8.25	89.25	98.55	**99.56**
Corn-senesced-green	2.14	62.11	95.79	**97.86**
Lettuce-romaine-4wk	2.43	59.18	**98.88**	95.97
Lettuce-romaine-5wk	2.85	48.26	98.75	**98.91**
Lettuce-romaine-6wk	3.28	73.47	**94.21**	86.24
Lettuce-romaine-7wk	2.43	78.88	**94.39**	81.12
Vineyard-untrained	10.35	30.56	87.23	**97.08**
Vineyard-vertical	2.66	64.14	99.78	**99.89**
AA	7.77	71.28	**97.15**	96.63
OA	17.99	71.14	95.73	**97.72**
Kappa	1.84	67.81	95.24	**97.46**
SLIC Time	-	-	4.53	4.53
Reconstruction Time	2398.71	2686.37	2572.74	**0.00**
Classification Time	153.32	37.00	37.11	**2.39**
Total Time	2552.03	2723.36	2614.35	**6.88**

a condenser lens, which is connected through an optical fiber to an Ocean Optics Flame S-VIS-NIR-ES spectrometer, which is used as the single-pixel detector. The second arm that acquires the RGB images is compounded by a 30 mm lens and an AVT Stingray F-080B camera. It is important to remark that the RGB images are acquired one band each time, at the wavelengths 455 nm, 540 nm, and 656 nm. Note that, as shown in the implemented architecture (Fig. 69), the coding pattern (DMD) is located before the light bifurcation with the beam splitter, which

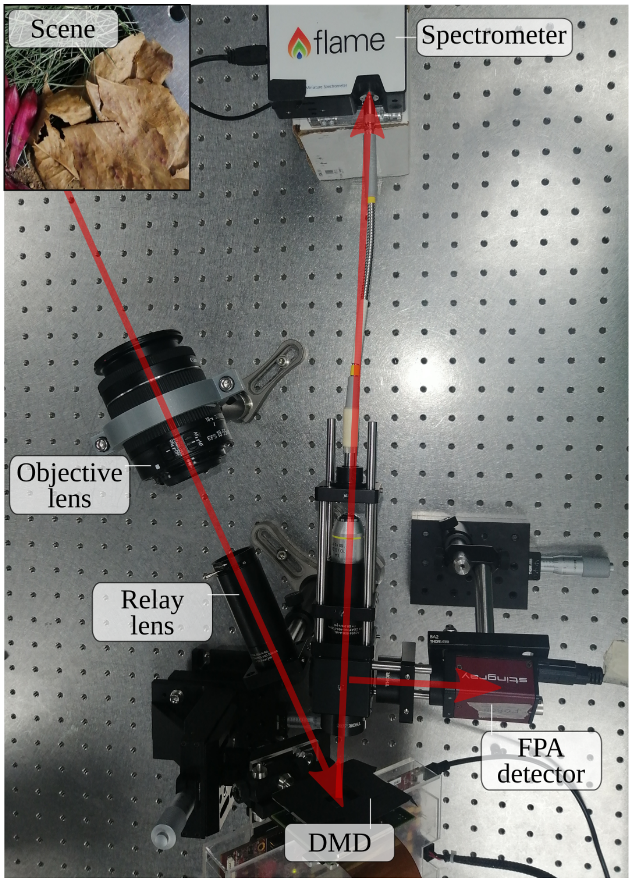
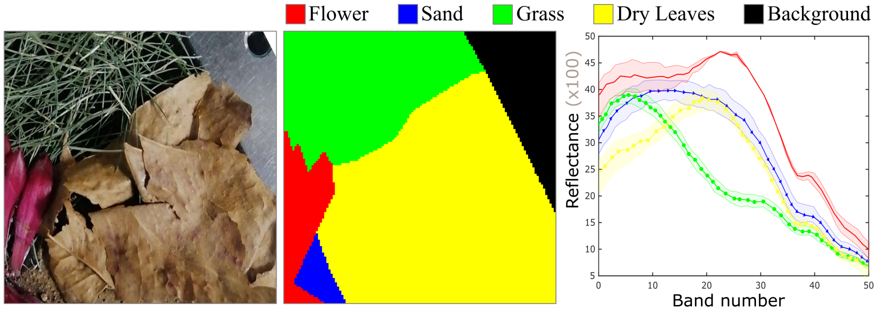


Figure 69. Laboratory implementation of the proposed SPC architecture, which acquires the compressed measurements and the RGB images, as side-information, to design the coding patterns.

guarantees the registration of the SPC and the RGB images. In this way, the dataset is suitable for the classification test without the need to include a stage of registration of the two measurements (SPC and RGB images).

Figure 70 (a) shows an RGB image of the scene under testing. Our target scene consists of a black card with four nature materials: flower, sand, grass, and dry leaves. We show the corresponding ground truth in Fig. 70 (b), and Fig. 70 (c) depicts the spectral signatures for each material. The spectral image to acquire has $M = N = 384$ pixels and $L = 50$ spectral bands. To



(a) (b) (c)

Figure 70. (a) RGB of the dataset acquired in the lab, (b) ground truth map, and (c) spectral signatures of each class.

perform the experiments, I select the 5 % of each class as training samples and the remaining data points as testing samples.

Impact of the Number of Superpixels in the Classification Accuracy. In this experiment, the number of superpixels was varied, as in section 15, to evaluate the classification accuracy. Figure 71 shows the results obtained with each approach of Fig. 65. As observed, the proposed method improves its performance when β increases until $\beta = 128$, where the classification accuracy achieves 94.66%. According to this, for application purposes, $\beta \geq 128$ superpixels would guarantee high classification accuracy. On the other hand, the R-SPCRGB2 approach achieves very similar results, but it takes a longer time. In contrast, the R-SPC and R-SPCRGB1 methods show low OA performance, which corresponds to similar behavior as in section 15.

Visual Maps and Quantitative Results. Figure 72 and Table 33 present the qualitative and quantitative results that I obtained after classify the spectral image acquired in the lab. Note that the proposed classification method achieves higher performance compared with the other methods. Specifically, an improvement of up to 37% in classification overall accuracy was obtained

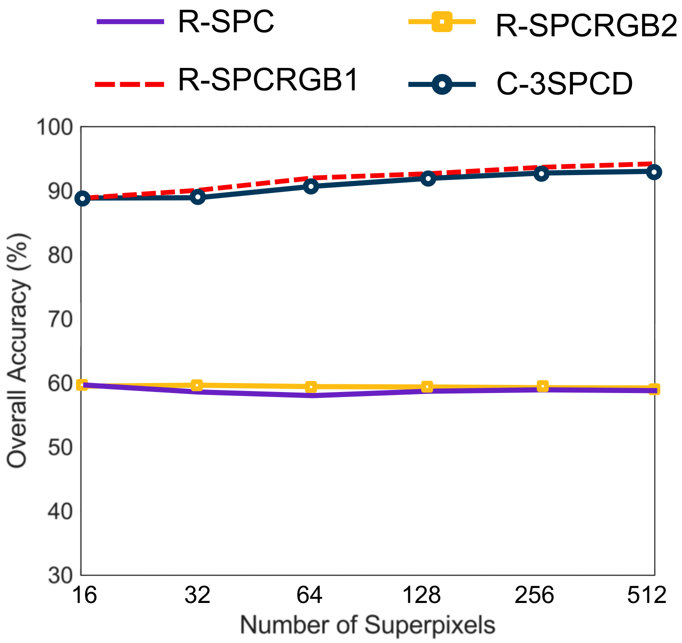


Figure 71. Overall accuracy of the classification by varying the number of superpixels (β) in the dataset acquired in the lab.

in comparison with the traditional approaches. Furthermore, in terms of total execution time, the proposed method is up to 70x faster.

Conclusion. This work presented an approach to classify single-pixel measurements directly on the compressed-domain called C-3SPCD, thus avoiding the traditional and computational expensive signal reconstruction step. Mainly, the proposed method acquires an RGB image and extracts superpixels information of the desired scene to carefully design the coding patterns for an SPC camera. Using the SPC compressed measurements and the acquired superpixels information, I proposed an algorithm to perform the spectral pixels classification directly. We developed extensive simulations using the well-known remote sensing images “Pavia University”, and “Salinas”, obtaining a global precision of 95.41%, and 97.72%. Besides, for the sake of completeness, I im-

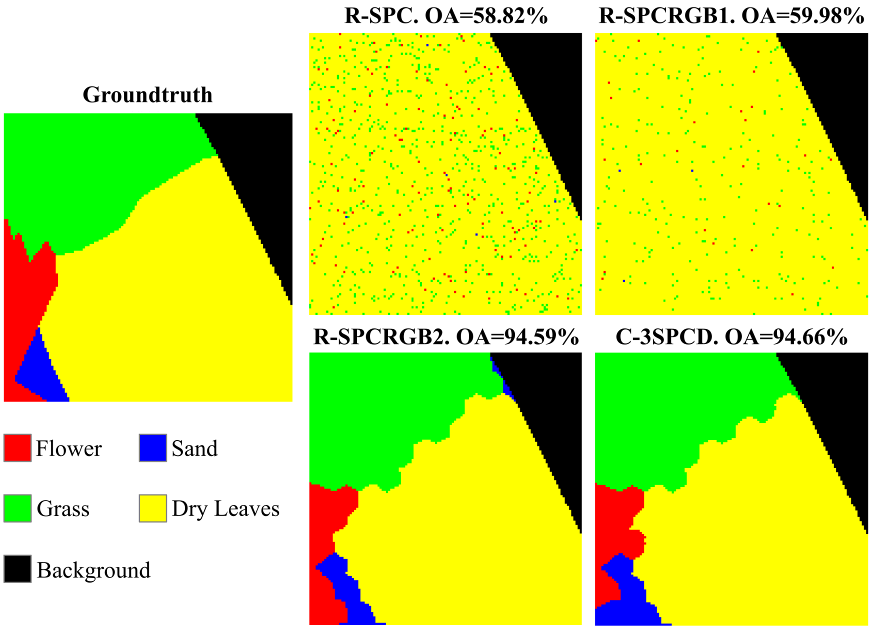


Figure 72. Visual classification results on the dataset acquired in the lab.

plemented in the high dimensional signal processing (HDSP) group's optics laboratory a testbed of the proposed methodology and acquired a spectral image with different materials. Then, the proposed method was tested on the acquired dataset and it achieves a classification accuracy of 94.66%.

Table 33

Quantitative results on the dataset acquired in the lab

	R-SPC	R-SPCRGB1	R-SPCRGB2	C-3SPCD
Dry Leaves	94.38	98.07	**98.09**	96.64
Grass	5.51	2.03	95.96	**98.16**
Flower	1.14	0.26	**71.28**	68.13
Sand	0.00	0.27	67.90	**89.66**
AA	25.26	25.16	83.31	**88.14**
OA	58.82	59.98	94.59	**94.66**
Kappa	0.71	0.29	89.95	**90.23**
SLIC Time	-	-	0.25	0.25
Reconstruction Time	26.91	32.55	32.73	**0.05**
Classification Time	3.75	2.21	0.18	**0.13**
Total Time	30.69	34.80	33.17	**0.43**

Appendix B. Extension to HSI supervised classification

Part of this chapter has been adapted from the journal paper (Sanchez et al., 2019a) (published) and the conference paper (Sanchez et al., 2018) (published).

Supervised spatio-spectral classification of fused images using superpixels

Spectral imaging (SI) is a technique used to acquire spatial information of a scene in many contiguous electromagnetic frequencies. The acquired image is commonly represented as a 3D datacube, where two dimensions correspond to the spatial information and the third one to the spectral domain. Traditional sensing techniques construct such datacube by scanning the scene, either spectrally or spatially in proportion to the desired spatial or spectral resolution. In general, SI is a valuable tool for remote sensing applications including precision agriculture, urban planning, military surveillance, etc (Bioucas-Dias et al., 2013; Martín and Bioucas-Dias, 2016; Thenkabail and Lyon, 2016).

Based on the spectral/spatial resolution, spectral range, and width and contiguousness of bands, spectral imaging sensors can be categorized as Hyperspectral (HS) or Multispectral (MS). In particular, HS devices capture hundreds of spectral bands of the scene of interest in a wavelength interval ranging from the visible region ($0.4-0.7\ \mu$m) to the infrared region ($\approx 2.4\ \mu$m). However, despite its high spectral resolution, HS imagery usually suffer from low spatial resolution compared to that obtained with an MS or a typical RGB sensor. Therefore, spectral image fusion has emerged as a challenging processing task in the remote sensing field that consists of the combination of a high-spectral but low-spatial resolution image with a low-spectral but high-spatial resolution image for obtaining a high spatial and spectral resolution image (Yokoya and Iwasaki, 2013). Based on the used theoretical approach, data fusion algorithms can be categorized as, component substitution, spectral unmixing, sparse signal representation, and Bayesian estimation (Yokoya et al., 2012b; Akhtar et al., 2014; Wei et al., 2015a,b; Simões et al., 2014; Wei et al., 2016). Traditionally, these algorithms fuse the information acquired by traditional spatio-spectral imaging sensors, resulting in high storage and processing costs (Yokoya et al., 2017). After applying a fusion technique, the spectral signatures of the pixels (a.k.a spectral pixels) are commonly processed to detect and classify different materials within the enhanced spatio-spectral datacube. Particularly, spectral image classification, which aims at assigning each spectral pixel to one class belonging to a set of categories, has drawn broad attention and has led to a variety of methods (Fauvel et al., 2013; Ghamisi et al., 2017b,a). In the literature, many of the methods have concentrated on exploring the role of the spectral signatures in classification, employing exclusively the spectral signatures for determining the classification maps. Since spectral images can be seen as a set of 2D images acquired at different spectral wavelengths, the spatial information is naturally another source of data that can be considered to improve the classification performance. Then, the introduction of spatial dependency offers the possibility to boost the pixel-wise classification methods (Li et al., 2012; Liu et al., 2013; Camps-Valls et al., 2014; Hang et al., 2016).

Among image processing methods, the superpixel technique is well known to efficiently embed the spatial neighboring information into the classification procedure. This technique is based on the over-segmentation strategy that partition an image into homogeneous subregions (Li and Chen, 2015; Stutz et al., 2018). Such subregions are usually irregular spatial units, but perceptually consistent, i.e., all pixel values in a superpixel area are most likely uniform, meaning that superpixel methods are adapted to real scenes (Tian et al., 2015). Therefore, combining superpixel partitioning with conventional spectral classification approaches is expected to boost the classification performance, considering not only the spectral behavior of the spectral signatures but the variations of spatial characteristics of the scene of interest.

In this paper, a supervised spatial-spectral classification approach of high spectral and spatial resolution fused images is developed, where the superpixel segmentation technique is used to incorporate the spatial information in the fused data. Specifically, the proposed algorithm first apply a superpixel algorithm to the rich spatial resolution RGB image in order to group spatial neighboring pixels, with similar characteristics, in segments or superpixels. Then, an optimization problem is formulated to fuse the high spectral information of an HS image with the spatial information given by the superpixels of the MS image. Finally, the pixels of the fused datacube are classified using a supervised classification algorithm. The main contribution of this paper is a methodology that fuses and, simultaneously, incorporates spatial information of the scene for boosting the classification accuracy. This approach, in turn, provides a framework for low computational complexity and accurate classification .

The paper is organized as follows: In section 2, the matrix model and some notation used through the paper are introduced. The mathematical description and flowchart of the proposed classification methodology is described in section 2. In order to evaluate the performance of the proposed method, several simulation results are presented in Section 5 with different spectral image datasets. Some concluding remarks are summarized in Section 5.

Matrix Model. In general, denote $\mathscr{F}$ as the spatio-spectral datacube, with $M \times N$ spatial dimensions, L spectral bands, and entries denoted as $\mathscr{F}_{n_1,n_2,k}$, where n_1 and n_2 index the spatial coordinates, and k determines the k-th spectral band. The datacube can be rearranged as a matrix $\mathbf{F} \in \mathbb{R}^{MN\times L}$ whose rows $\{\mathbf{f}_j^T\}_{j=0}^{MN}$ are the spectral signatures of the datacube. Specifically, the rearrangement can be expressed as

$$\begin{aligned}\mathbf{F} = \big[&[\mathscr{F}_{0,0,0}, \mathscr{F}_{1,0,0}, \cdots, \mathscr{F}_{0,1,0}, \cdots, \mathscr{F}_{(M-1),(N-1),0}]^T,\\ &\cdots, [\mathscr{F}_{0,0,(L-1)}, \cdots, \mathscr{F}_{(M-1),(N-1),(L-1)}]^T\big].\end{aligned} \tag{95}$$

In this work, the classification maps of a high spatial and spectral resolution image $\mathbf{F}$ are found from fused image obtained from an RGB and an HS image, denoted as $\mathbf{F_m} \in \mathbb{R}^{MN\times 3}$ and $\mathbf{F_h} \in \mathbb{R}^{M_hN_h\times L}$, respectively. Note that $\mathbf{F_m}$ can be composed by combining three spectral bands representing the red, green, and blue colors. Further, $\mathbf{F_h}$ can be seen as a blurred and spatially downsampled instance of $\mathbf{F}$ with $N_h = N/p$ and $M_h = M/p$, where p is the downsampling factor along the spatial coordinates. Mathematically, the matrices $\mathbf{F_m}$ and $\mathbf{F_h}$ can be expressed as

$$\mathbf{F_m} = \mathbf{F}\mathbf{D_m}, \tag{96}$$

$$\mathbf{F_h} = \mathbf{D_h}\mathbf{F}, \tag{97}$$

where $\mathbf{D_m} \in \mathbb{R}^{L\times 3}$ is a sparse selection matrix with only three nonzero values, i.e., a single one per column that indicates the selected spectral bands composing the RGB image and $\mathbf{D_h} \in \mathbb{R}^{M_hN_h \times MN}$ is a spatial downsampling that includes both the blur operation and the spatial decimation of $\mathbf{F}$ with downsampling factor p.

Proposed Spatio-Spectral Classification Approach. The aim of the proposed method is to incorporate the spatial information of the spectral scene in the classification features using an RGB image as side information. The flowchart of the algorithm is depicted in figure 73. In general, a rich spatial and spectral fused cube $\mathbf{F} \in \mathbb{R}^{M_mN_m \times L_h}$ is obtained using the spectral information of an HS image and the superpixels extracted from the RGB image. Then, the fused cube is later used as the input for a supervised pixel-based classifier.

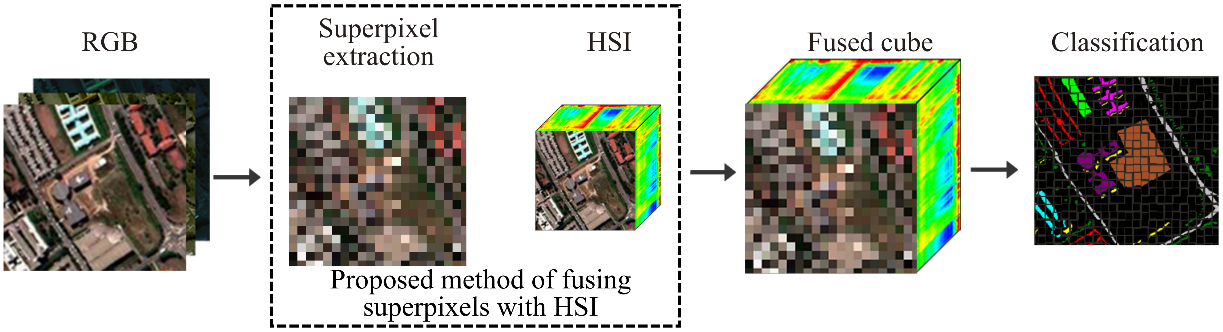


Figure 73. Flowchart of the proposed classification framework. An RGB image is segmented into a predefined number of superpixels. The superpixel image and the HS image are used to obtain a high spectral and spatial resolution which is used as classification features. Finally, a supervised classification algorithm is applied on the fused image.

Superpixels Segmentation. In figure 73, the first step in the proposed methodology is to segment an RGB image in superpixels. The superpixels technique is based on the over-segmentation strategy that gather a group of uniform pixels into homogeneous subregions. In this work, I use the simple linear iterative clustering (SLIC) superpixel algorithm (Zhang et al., 2015; Achanta et al., 2012). SLIC works in the 5-D space, where two coordinate components (x,y) depict the spatial location of the segment and the other three components corresponds to the RGB color channels. Given N_{seg} desired equally-sized superpixels, where the approximation size of each superpixel is N^2/N_{seg}, the first step of SLIC is to define a cluster center at every grid interval $S = \sqrt{N^2/N_{seg}} = N/\sqrt{N_{seg}}$. The algorithm assumes that the pixels associated with a cluster, lie in a $2S \times 2S$ area around the superpixel center on the (x,y) plane. Therefore, this becomes the search area for the pixels near to each cluster center. To avoid keeping the center on the edge of an object, it is transferred to the lowest gradient position in a 3×3 neighborhood. In the next step, for each cluster center, SLIC assigns the best matching pixels from the search area according to the distance measure

$$D_c = \sqrt{(R_j - R_{j'})^2 + (G_j - G_{j'})^2 + (B_j - B_{j'})^2}, \tag{98}$$

$$D_p = \sqrt{(x_j - x_{j'})^2 + (y_j - y_{j'})^2}, \tag{99}$$

$$D_t = D_c + \frac{m}{S} D_p, \tag{100}$$

where R_j, G_j, B_j corresponds to the color of the j-th pixel, and m controls the compactness of a superpixel, which is usually chosen as $m = 10$. The complexity of SLIC algorithm is $O(MN)$, since it needs to compute distances from any point to no more than eight cluster centers and the number of iterations is constant (Achanta et al., 2012). Notice that the region grows at most two times the cluster radius with the SLIC algorithm.

Once the segmentation map of the RGB image has been created using SLIC, the next step in the proposed method is to create the upsampling matrix $\mathbf{U} \in \mathbb{R}^{N_{seg} \times MN}$ which collects the superpixels information. First, denote $\mathbf{p}^e$ as the vector of size n_e containing the indexes of all pixels belonging to the superpixel e. Then, the nonzero values of the $e-th$ row of $\mathbf{U}$, denoted in vector form as $(\mathbf{u}_e)^T$, are determined by the entries of $\mathbf{p}^e$ and the value of n_e as follows

$$(\mathbf{u}_e)^T_{(\mathbf{p}^e)_l} = \frac{1}{n_e}, \quad \text{for } l = 0, \cdots, n_e, \tag{101}$$

where $(\mathbf{u}_e)^T_{(\mathbf{p}^e)_l}$ denotes the position in $\mathbf{u}_e$ indexed by the l-th entry of the $\mathbf{p}^e$ vector.

Problem Formulation. The following step in the proposed method is to perform the fusion operation $\mathbf{F_h}$ spectral information and the superpixels extracted from the RGB image $\mathbf{F_m}$. Using the notation given in section 2, the proposed fusion operation is formulated as the optimization problem

$$\min_{\mathbf{F}} \frac{1}{2} ||\mathbf{F_h} - \mathbf{D_h}\mathbf{F}||_F^2 + \lambda ||\mathbf{F}||_*, \tag{102}$$

where $||\cdot||_F$ stands for the Frobenius norm and λ is a regularization parameter. Since pixels in $\mathbf{F}$ that belongs to the same class share a common low-rank pattern, the nuclear norm minimization $||\mathbf{F}||_*$, in Eq. (102), is used to obtain the lowest-rank representation that captures the global structure of the data (He et al., 2016b) .

As the matrix $\mathbf{F}$ is obtained from the superpixels information, it is possible to relate the $\mathbf{U}$ matrix with $\mathbf{F}$ as

$$\mathbf{F} = \mathbf{U}^T \bar{\mathbf{F}}, \tag{103}$$

where $\bar{\mathbf{F}} \in \mathbb{R}^{N_{seg} \times L}$ contains the fused spectral information of each superpixel. Then, assuming that $\text{rank}(\bar{\mathbf{F}}) \approx \text{rank}(\mathbf{F})$, the optimization problem in Eq. (102) can be rewritten as

$$\min_{\bar{\mathbf{F}}} \frac{1}{2} \|\mathbf{F_h} - \mathbf{D_h}\mathbf{U}^T\bar{\mathbf{F}}\|_F^2 + \lambda \|\bar{\mathbf{F}}\|_*. \tag{104}$$

ADMM algorithm to solve the proposed optimization problem. The minimization in Eq. (104) can be efficiently solved by the well-known alternating direction method of multipliers (ADMM). First, an auxiliary matrix $\mathbf{Z}$ with the same size of $\bar{\mathbf{F}}$ is used to separate the problem. In this way, problem in Eq. (104) becomes

$$\begin{aligned} &\min_{\bar{\mathbf{F}},\mathbf{Z}} \frac{1}{2} \|\mathbf{F_h} - \mathbf{D_h}\mathbf{U}^T\bar{\mathbf{F}}\|_F^2 + \lambda \|\mathbf{Z}\|_* \\ &\text{s.t. } \mathbf{Z} = \bar{\mathbf{F}}. \end{aligned} \tag{105}$$

The augmented Lagrangian associated with the optimization in Eq. (105) can be written as

$$\mathscr{L}(\bar{\mathbf{F}},\mathbf{Z},\mathbf{G}) = \frac{1}{2} \|\mathbf{F_h} - \mathbf{D_h}\mathbf{U}^T\bar{\mathbf{F}}\|_F^2 + \lambda \|\mathbf{Z}\|_* + \frac{\rho}{2} \|\bar{\mathbf{F}} - \mathbf{Z} + \mathbf{G}\|_F^2, \tag{106}$$

where $\mathbf{G}$ is the scaled dual variable and $\rho > 0$ is the weighting of the augmented Lagrangian term (Parikh et al., 2014). The ADMM solution of Eq. (105) is summarized in Algorithm 11 that consists of minimizing $\bar{\mathbf{F}}, \mathbf{Z}$, and $\mathbf{G}$, alternatively. Specifically, the minimization of the three matrices consists of the following updates:

1) *Updating* $\bar{\mathbf{F}}$: $\bar{\mathbf{F}}^{(t)}$ is obtained by minimizing $\mathscr{L}$ with respect to $\bar{\mathbf{F}}$, while ($\mathbf{Z}$,$\mathbf{G}$) are fixed. We calculate the derivative of $\mathscr{L}$ with respect to $\bar{\mathbf{F}}$ and set it to zero to obtain $\bar{\mathbf{F}}$ as follows:

$$\bar{\mathbf{F}}^{(t)} = \frac{\mathbf{U}^T\mathbf{D_h}^T\mathbf{F_h} + \rho\mathbf{I}(\mathbf{Z}-\mathbf{G})}{\rho\mathbf{I} + \mathbf{U}^T\mathbf{D_h}^T\mathbf{D_h}\mathbf{U}} \tag{107}$$

2) *Updating* $\mathbf{Z}$: the update of $\mathbf{Z}$ is given by the singular value thresholding operator (Lin et al., 2013). Specifically, I compute the SVD of the symmetric matrix

$$\bar{\mathbf{F}} + \mathbf{G} = \hat{\mathbf{U}}\Sigma\hat{\mathbf{U}}^* \tag{108}$$

and apply the soft-thresholding operator to the resulting singular values

$$S_{1/\rho}(\Sigma) = \operatorname{diag}\left\{(\sigma_i - 1/\rho)_+\right\}, \tag{109}$$

where $(a)_+ := \max\{a,0\}$. Then, the update of $\mathbf{Z}$ is defined as

$$\mathbf{Z}^{(t)} = \hat{\mathbf{U}}S_{1/\rho}(\Sigma)\hat{\mathbf{U}}^*. \tag{110}$$

After obtain the estimation of $\bar{\mathbf{F}}$, the rich spatio-spectral image can be obtained using Eq. (103).

Supervised Classification. Given the spectral superpixels $\bar{\mathbf{F}}$ and the respective high spectral and spatial resolution image $\mathbf{F}$, obtained with Eq. (103), the multiclass supervised classifica-

Algorithm 11 ADMM sub-iterations to estimate $\bar{\mathbf{F}}$

```
Input : $\mathbf{F_h}, \lambda, \rho, \mathbf{D_h}, \mathbf{U}, MAXITER.$
Initialization :
$\mathbf{Z}^{(0)} = \mathbf{0}$, $\mathbf{G}^0 = \mathbf{0}$
Main iterations for $t = 1$ to $MAXITER$ do
    $\bar{\mathbf{F}}^{(t)} \leftarrow \arg\min_{\bar{\mathbf{F}}} \mathscr{L}(\bar{\mathbf{F}}, \mathbf{Z}^{(t-1)}, \mathbf{G}^{(t-1)})$  $\mathbf{Z}^{(t)} \leftarrow \arg\min_{\mathbf{Z}} \mathscr{L}(\bar{\mathbf{F}}^{(t)}, \mathbf{Z}, \mathbf{G}^{(t-1)})$
    $\mathbf{G}^{(t)} \leftarrow \mathbf{G}^{(t-1)} + \bar{\mathbf{F}} - \mathbf{Z}$
```

tion method aims at determining the class label that better fits the corresponding j_s-th pixel, for $j_s = 0, \cdots, N_{seg} - 1$. Notice that with the proposed method it is only necessary to determine the class label of the N_{seg} spectral superpixels (rows of $\bar{\mathbf{F}}$). Then, the pixels in $\mathbf{F}$ acquire the same label of the superpixel which their belongs to. As the number of superpixels $N_{seg} \ll MN$, the proposed approach boost the computational performance, as the next section shows. In this paper, the classification of the spectral scene is achieved using two well-known algorithms: the support vector machine (SVM) and the k-nearest neighbor (KNN). In addition, let's denote $\Omega = \{\omega_1, \cdots, \omega_c\}$ as the set of c class labels and $\Theta = \{\theta_1, \cdots, \theta_v\}$ as the set of v integers indexing the training samples.

SVM. Let first consider the binary supervised classification problem, where each feature vector of the training set is associated with a binary output $\beta_\theta \in \{-1, 1\}$ for $\Theta = \{\theta_1, \cdots, \theta_v\}$. In this context, the SVM focus on determining a hyperplane that optimally separates the samples belonging to different classes. This problem is equivalent to minimize the norm of a weight vector $\mathbf{h}$ that is normal to the decision hyperplane (Mountrakis et al., 2011; Li et al., 2013a). Specifically, this classification method solves the following optimization problem

$$\begin{aligned} \min_{\mathbf{h},h_0,\mathscr{X}_\Theta} & \left\{ \frac{1}{2}\|\mathbf{h}\|_2^2 + \eta \sum_{\Theta} \mathscr{X}_\Theta \right\} \\ \text{s.t.} \quad & \beta_\Theta \left(\gamma \left(\mathbf{f}_\Theta^T \right)^T \mathbf{h} + h_0 \right) \geq 1 - \mathscr{X}_\Theta \\ & \mathscr{X}_\Theta \geq 0, \end{aligned} \tag{111}$$

for all $\Theta = \{\theta_1, \cdots, \theta_v\}$, where $\mathbf{f}_\Theta^T$ are the Θ rows of $\mathbf{F}$ used as training data; $\mathscr{X}_\Theta$ is the set of slack variables that considers the non separability between sets belonging to different classes; η is the regularization parameter that controls the influence of the misclassified samples; γ is a nonlinear function that maps the feature vector to the kernel space; h_0 is the offset of the decision hyperplane with respect to the origin of the coordinate system (Schlkopf et al., 2018).

As mentioned before, SVM are basically binary classifiers. However, a multiclass strategy for the spectral image classification is required. To overcome this drawback, the one-against-one multiclass strategy is adopted, where two classes are separately analyzed by implementing a binary classifier, ignoring, at the same time, the remaining classes (Gu and Sheng, 2017). Finally, using the training sample set $\{\mathbf{f}_\Theta^T\}_{\Theta=\theta_1}^{\theta_v}$, the problem of classifying each test pixel consists of finding the class Ω whose training pixel Θ is the nearest to the test pixel in the Euclidean distance

$$\text{Class}(\bar{\mathbf{f}}_{j_s}^T) = \min_{\Omega=\Omega_1,\cdots,\Omega_c} \|\bar{\mathbf{f}}_{j_s}^T - (\mathbf{f}_\Theta^T)_{(\Omega)}\|_2^2, \tag{112}$$

where $\bar{\mathbf{f}}_{j_s}^T$ is the j_s-th row of $\bar{\mathbf{F}}$ and $(\mathbf{f}_\Theta^T)_{(\Omega)}$ denotes the training pixels (rows of $\mathbf{F}$) belonging to the class Ω. Notice that the SVM training is performed using the rows of $\mathbf{F}$ as training data and the classification is performed on the superpixels (rows of $\bar{\mathbf{F}}$).

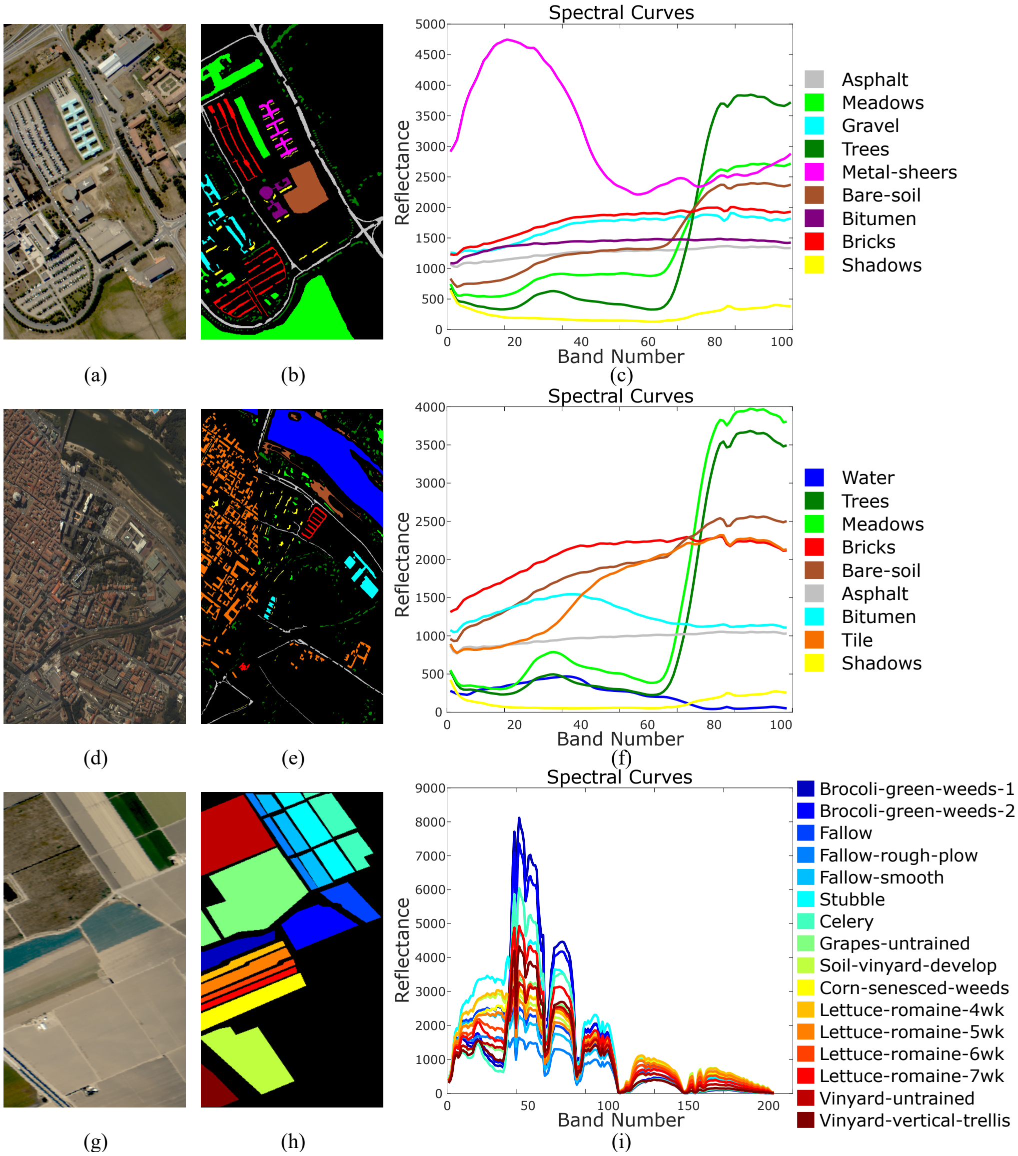
KNN. Following the same previous introduced notation, denote $\mathscr{N}_{j_s}$ as the set containing the k_n nearest neighbors of the j_s pixel $\bar{\mathbf{f}}_{j_s}^T$ with labels denoted as $\{\ell_1, \ell_2, \cdots, \ell_{k_n}\}$. Then, the KNN classifier finds the k_n nearest neighbors of the $\bar{\mathbf{f}}_{j_s}^T$ point in the training data and assigns such testing point to the most frequently occurring class of its k_n neighbors. The KNN perform the classification with the following majority voting rule:

$$\text{Class}(\bar{\mathbf{f}}_{j_s}^T) = \underset{\Omega=\Omega_1,\cdots,\Omega_c}{\arg\max} \sum_{i=1}^{k_n} \delta(\ell_i, \Omega), \tag{113}$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta function which is equal to 1 if all its arguments are equal, and 0 otherwise.

Simulations and Results. In this section, the performance of the proposed spectral image classification method is evaluated. In particular, I tested the classification framework on three datasets, the Pavia University, the Pavia Center, and the Salinas Valley. The results are compared against other approaches that fuse high spatial and spectral resolution images, followed by the classification. Pavia University is an image acquired over an urban area surrounding the University of Pavia, northern Italy, by the Reflective Optics System Imaging Spectrometer (ROSIS-03) airborne sensor. The size of dataset is 610×340 pixels and 103 spectral bands, with a high spatial resolution of 1.3 meters per pixel and a spectral coverage ranging from 0.43 to 0.84 μm. Figure 74 (a)-(c) shows the color composite of Pavia University, the corresponding ground truth image which differentiates nine land-cover classes: asphalt, meadows, gravel, trees, metal sheets, bare soil, bitumen, bricks and shadows; and the characteristic spectral signatures of each class, respectively. Pavia Center is an image also acquired with the ROSIS-03 sensor with size of 1096×715 pixels and 102 spectral bands, and it retains the same spatial and spectral resolution characteristics as the Pavia University dataset. The ground truth of Pavia Center contains nine classes: water, trees, asphalt, self-blocking bricks, bitumen, tiles, shadows, meadows and bare soil. A false-color image of the Pavia Center, the ground truth and its spectral signatures are shown in Fig.74 (d)-(f). The third spectral dataset with which the proposed method was tested is the Salinas image. This scene was collected in 1998 by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) on Salinas Valley, California, USA. The size of dataset is 512×217 pixels and 204 spectral bands in the range of 0.24 to 2.40 μm. The Salinas ground truth contains 16 land-cover classes: broccoli green weeds 1, broccoli green weeds2, fallow, fallow rough plow, fallow smooth, stubble, celery, grapes untrained, soil vinyard develop, corn senesced green weeds, lettuce romaine 4wk, lettuce romaine 5wk, lettuce romaine 6wk, lettuce romaine 7wk, vinyard untrained, and vinyard vertical trellis. Figure 74 (g)-(i) shows a false-color of Salinas Valley, its ground truth and the spectral signatures of each class, respectively.

Following the matrix model described in section 2, the HS images ($\mathbf{F_h}$) are obtained with a decimation ratio across the spatial coordinates of $p = 4$. Therefore, the Pavia University image was resized to a high-resolution-spectral image of $152 \times 85 \times 103$; the Pavia Center image was spatially reduced to 274×178 pixels and 102 spectral bands; and the Salinas Valley image to $128 \times 54 \times 204$. On the other hand, with a selection matrix applied to the original datasets, a high-



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

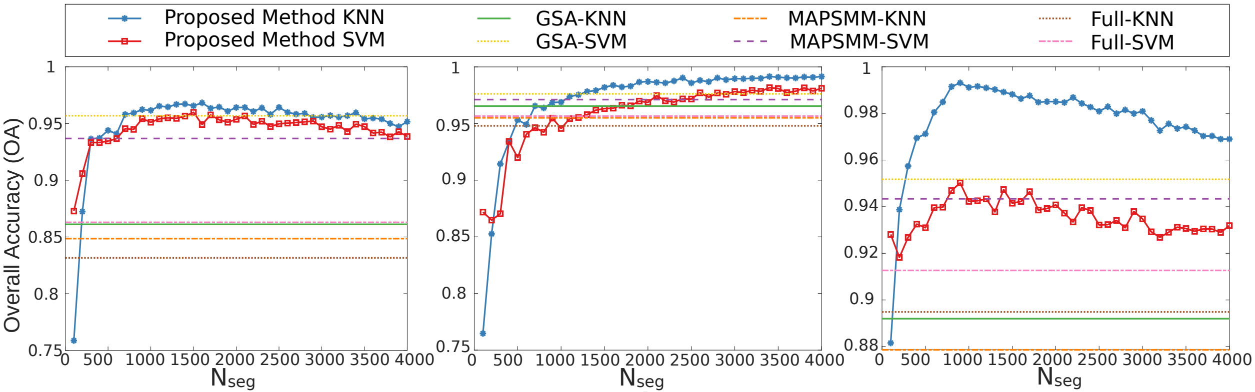
Figure 74. False-color images, ground truth maps and spectral signatures of each land-cover class for Pavia University (a)-(c), Pavia Center (d)-(f), and Salinas Valley (g)-(i), respectively.

resolution-spatial RGB ($\mathbf{F_m}$) image of each dataset was obtained. The size of the $\mathbf{F_m}$ images for Pavia University, Pavia Center and Salinas are $610 \times 340 \times 3$, $1096 \times 715 \times 3$, and $512 \times 217 \times 3$, respectively.

The experiments described below were performed by randomly selecting 5% of the labeled referenced data as training samples. Note that, in the presented results, I refer to "Full-data" when classifying the complete high spatial and spectral image, i.e., no fusion was performed. The aim of adding the Full-data results is to evidence the impact of incorporating spatial information in the classification performance. In addition, I compared the proposed approach with performing the fusion procedure with two state-of-the-art algorithms, before classifying the obtained high spatio-spectral pixels. Specifically, "GSA" refers to the results achieved classifying a high spectral and spatial resolution image obtained using the state-of-the-art fusion method GS Adaptive (GSA) algorithm (Loncan et al., 2015). Similarly, "MAP-SMM" corresponds to the results achieved classifying a high spectral and spatial resolution image reconstructed with the MAP-SMM algorithm (Zhang, 2014). In all the simulations, the datasets were classified with the SVM and KNN algorithms.

Impact of the number of superpixels and training samples in the classification accuracy. In the first experiment, the number of superpixels to cluster the image was varied to analyze the impact in the OA. In order to identify an adequate number of segments that provides the highest classification accuracy for each dataset, the value of N_{seg} was varied from 100 to 4000 in steps of 100. Figure 75 shows the overall accuracy results of the proposed method with both classifiers, SVM and KNN, by varying the number of segments N_{seg} in each dataset.

Figure 75 (a) presents the obtained results for the Pavia Unversity. Note that, as the number of desired segments (N_{seg}) increases, the overall accuracy increases exponentially until reaching $N_{seg} = 1600$ where the OA begins to decrease slowly, as depicted with the red and blue lines. Furthermore, note that the proposed method with KNN outperforms the results obtained by classifying the full-data and data obtained by the GSA and MAP-SMM fusion methods when $N_{seg} \geq 700$ and $N_{seg} \leq 2800$.



(a) (b) (c)

Figure 75. Overall accuracy of classification varying the number of segments (N_{seg}) in the proposed method when classifying (a) Pavia University, (b) Pavia Center, and (c) Salinas Valley. The obtained OA when using the full-data, and the GSA and the MAP-SSM fusion methods are provided for comparison purposes.

The obtained results for Pavia Center are shown in Fig. 75 (b). As observed, when the number of superpixels increases, the performance of the classification improves exponentially, until $N_{seg} \approx 3400$ where the overall accuracy begins to tend to $\approx 99.61\%$. As observed, in this particular image, all the methods perform well, however, with an adequate selected number of segments, the proposed method with KNN outperforms the others.

Figure 75 (c) shows the results obtained for the Salinas image. In this experiment, using the KNN classifier, the classification accuracy increases exponentially when the number of superpixels increases. However, when $N_{seg} = 900$, the OA begins to decrease. In this experiment, it is also important to note that the performance of the KNN classifier was significantly better than SVM. In addition the results obtained with the proposed method and the KNN classifier outperforms all the other methods when $N_{seg} \geq 300$.

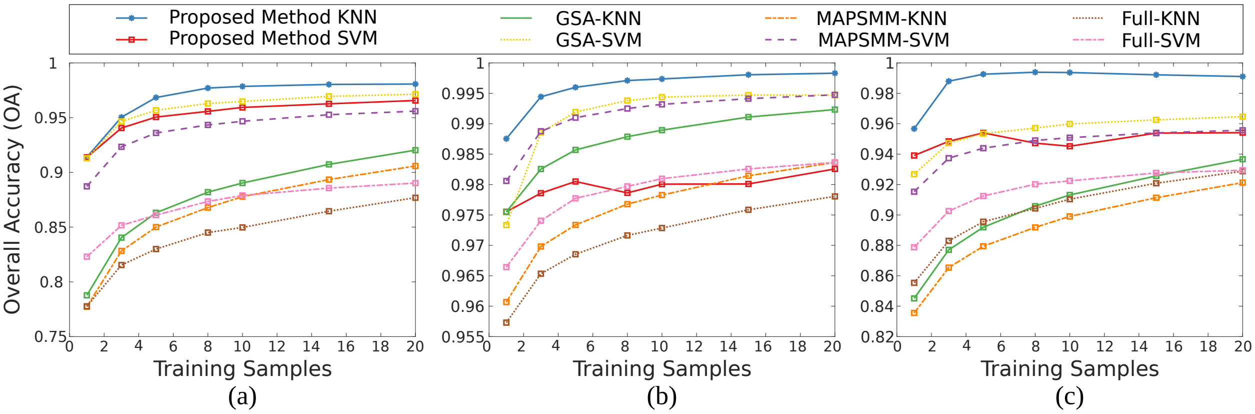


Figure 76. Overall accuracy of classification varying the number of training samples in the proposed method when classifying (a) Pavia University, (b) Pavia Center, and (c) Salinas Valley. The obtained OA when using the full-data, and the GSA and the MAP-SSM fusion methods are provided for comparison purposes.

In the second experiment, the number of samples used in the training stage was varied to analyze the impact in the OA. As observed in Fig. 76, all results obtained from the KNN classifier provides low-accuracy classification results in comparison with SVM, except for the proposed method which exhibits good performance for the tested number of training samples. This is expected since the KNN classifier takes more advantage of the spatial contextual information given by superpixels.

In general, for the proposed method with KNN, increasing the number of training samples will also improve the overall accuracy. However, when the number of training samples is greater than 8, the overall accuracy does not vary notably. In addition, from Fig. 76.(a), it can be noted that, for 1% of training samples, the proposed method with both KNN and SVM classifiers provides the same results as those given by GSA-SVM.

Classification Maps and Quantitative Results. To further validate the performance of the proposed framework, classification maps and quantitative results from Full data, GSA, MAP-SMM and the proposed approach, are presented in Figures (77)-(79) and Tables (34)-(36). In particular, table 34 shows the numerical results for each in the nine land-cover classes in Pavia University. The presented results of average accuracy (AA), OA, the Kappa coefficient, and the running time are the average of 25 realizations of each experiment. In the table, the optimal values are shown in bold font and the second-best results are underlined. All the numerical results, except the Kappa coefficients and the running time, are given in percentage.

From Fig. 77 and Table 34, it can be clearly seen that the proposed classification approach provides the best accuracy results in comparison with classifying the pixels of the full and the fused spectral datacubes. The OA values for the different classification approaches are shown in each subfigure of Fig. 77. For this dataset, the best accuracy classification results are provided by the proposed framework using the KNN classifier. In addition, using the proposed method with the KNN classifier provides the best classification time, as noticed in Table 34. All execution times shown in this paper includes the time of the fusion stage.

Figure 78 and Table 35 show the visual and numerical results of the Pavia Center dataset. Notice that when the KNN classifier is used with the proposed approach, the best overall accuracy and kappa coefficient are obtained. In addition, this method takes the shortest execution time. On the other hand, note that although using the SVM classifier with the proposed method the classification results are lower than using KNN, these are the second-best results in terms of overall accuracy, Kappa and time.

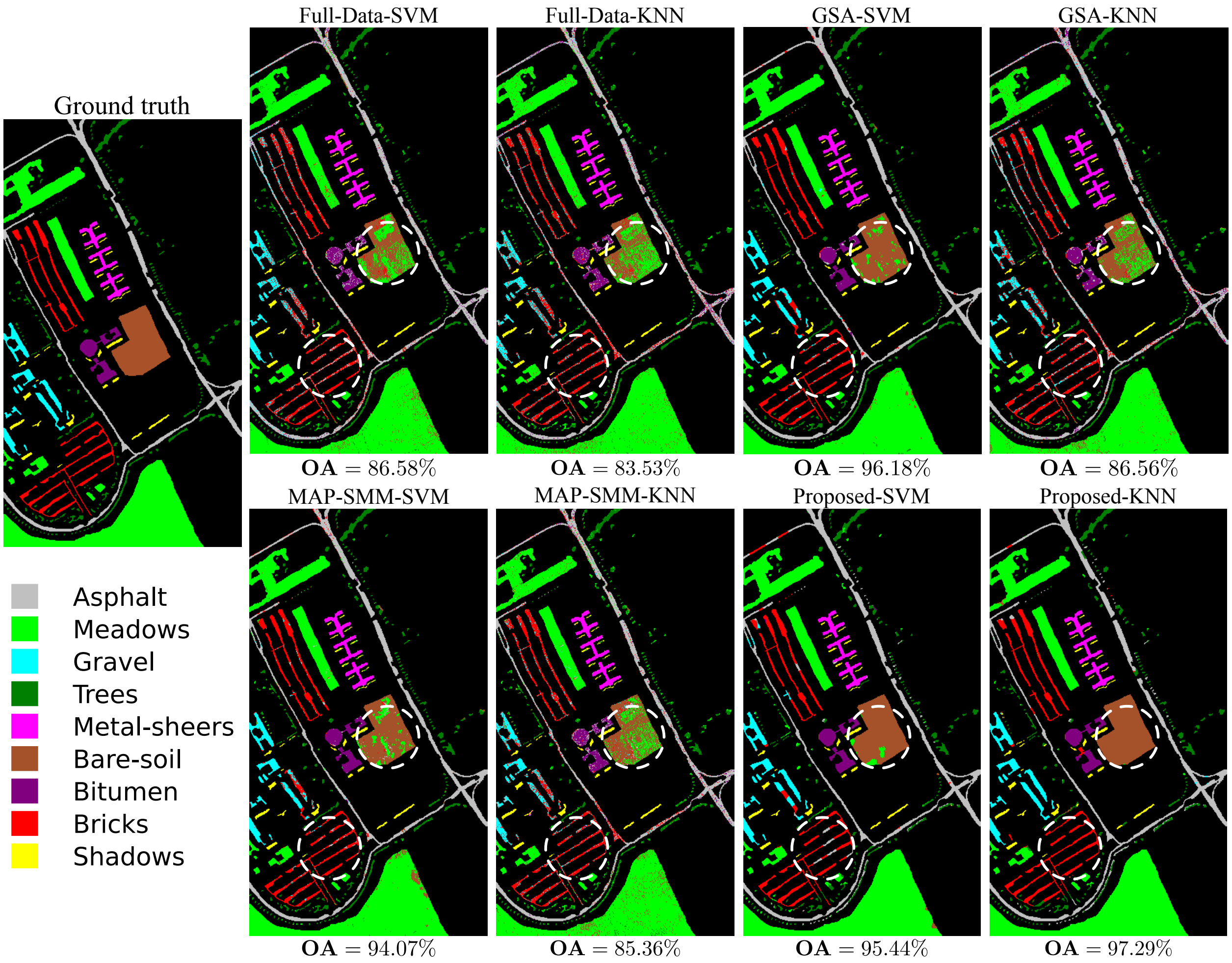


Figure 77. Visual classification results on Pavia University image. Figure shows the ground truth, and the results for Full-Data, GSA, MAP-SMM, and the Proposed approach using both SVM and KNN classifiers.

The performance of the classification of the Salinas Valley dataset is shown in Fig. 79 and the quantitative results are presented in table 36. This spectral image exhibits the greatest increment in classification performance when using the proposed method, compared to the other approaches. By applying the classifier directly on the original full data, OA classification results of 91.29% and 89.47% were obtained, in 33.424 and 17.903 seconds, respectively, whereas the proposed method using the KNN classifier achieves an OA of 99.21% in 1.414 seconds. In addition, note

Table 34
Quantitative results of the different classification methods for the Pavia University image.

	FULL		**GSA**		**MAP-SMM**		**PROPOSED**	
Class	**SVM**	**KNN**	**SVM**	**KNN**	**SVM**	**KNN**	**SVM**	**KNN**
Asphalt	86.41	76.48	**97.04**	81.00	94.94	82.40	96.10	96.28
Meadows	95.75	94.74	98.89	95.47	97.79	94.28	99.04	**99.60**
Gravel	69.18	62.34	87.34	66.72	82.93	70.83	90.40	**96.61**
Trees	91.51	83.74	**94.07**	85.96	91.91	74.66	74.39	77.81
Metal sheets	99.40	99.43	99.87	**99.92**	99.00	98.90	99.16	99.27
Bare soil	61.19	55.81	87.59	70.90	82.13	66.00	92.46	**99.44**
Bitumen	72.41	79.02	94.73	76.27	95.08	80.56	**98.70**	98.06
Bricks	76.53	77.89	93.03	76.58	91.60	79.32	93.34	**93.39**
Shadows	95.81	94.91	97.56	**98.89**	95.73	94.02	95.80	96.60
AA	83.13	80.48	94.46	83.52	92.34	82.33	93.27	**95.23**
OA	86.24	83.01	95.69	86.11	93.68	84.85	95.00	**96.66**
Kappa	0.8164	0.7731	0.9429	0.8158	0.9162	0.7985	0.9337	**0.9559**
Time	16.7052	23.9618	16.5859	25.1008	27.0535	38.1522	10.8399	**6.8039**

Table 35
Quantitative results of the different classification methods for the Pavia Center image.

	FULL		**GSA**		**MAP-SMM**		**PROPOSED**	
Class	**SVM**	**KNN**	**SVM**	**KNN**	**SVM**	**KNN**	**SVM**	**KNN**
Water	99.97	99.97	99.98	99.99	99.98	99.98	99.96	**100**
Trees	93.75	89.98	97.54	94.25	95.15	93.41	97.27	**98.56**
Meadows	86.27	87.58	95.95	92.02	87.16	87.27	93.07	**97.37**
Bricks	77.20	74.01	92.45	89.97	91.01	82.88	**99.48**	97.55
Bare soil	95.94	94.87	97.23	97.50	97.34	92.92	**99.38**	98.85
Asphalt	91.69	86.24	97.40	94.34	95.39	90.30	97.13	**99.29**
Bitumen	92.54	90.46	97.59	93.89	87.43	92.02	**99.07**	98.56
Tile	99.63	99.15	99.67	99.85	99.58	99.20	99.57	**99.93**
Shadows	96.70	94.32	98.32	**99.55**	90.97	98.29	90.68	99.10
AA	92.63	90.73	97.35	95.71	93.78	92.92	97.29	**98.80**
OA	97.74	96.84	99.11	98.54	98.00	97.51	99.13	**99.62**
Kappa	0.9680	0.9552	0.9874	0.9794	0.9716	0.9647	0.9877	**0.9946**
Time	35.5553	180.1488	54.4082	359.6640	58.2847	353.3343	10.9937	**7.3261**

that the second best classification results are obtained using the GSA fusion approach and the SVM classifier. However, the second best classification time was obtained by the proposed method and the SVM classifier. It is important to note that the good classification results obtained by the GSA

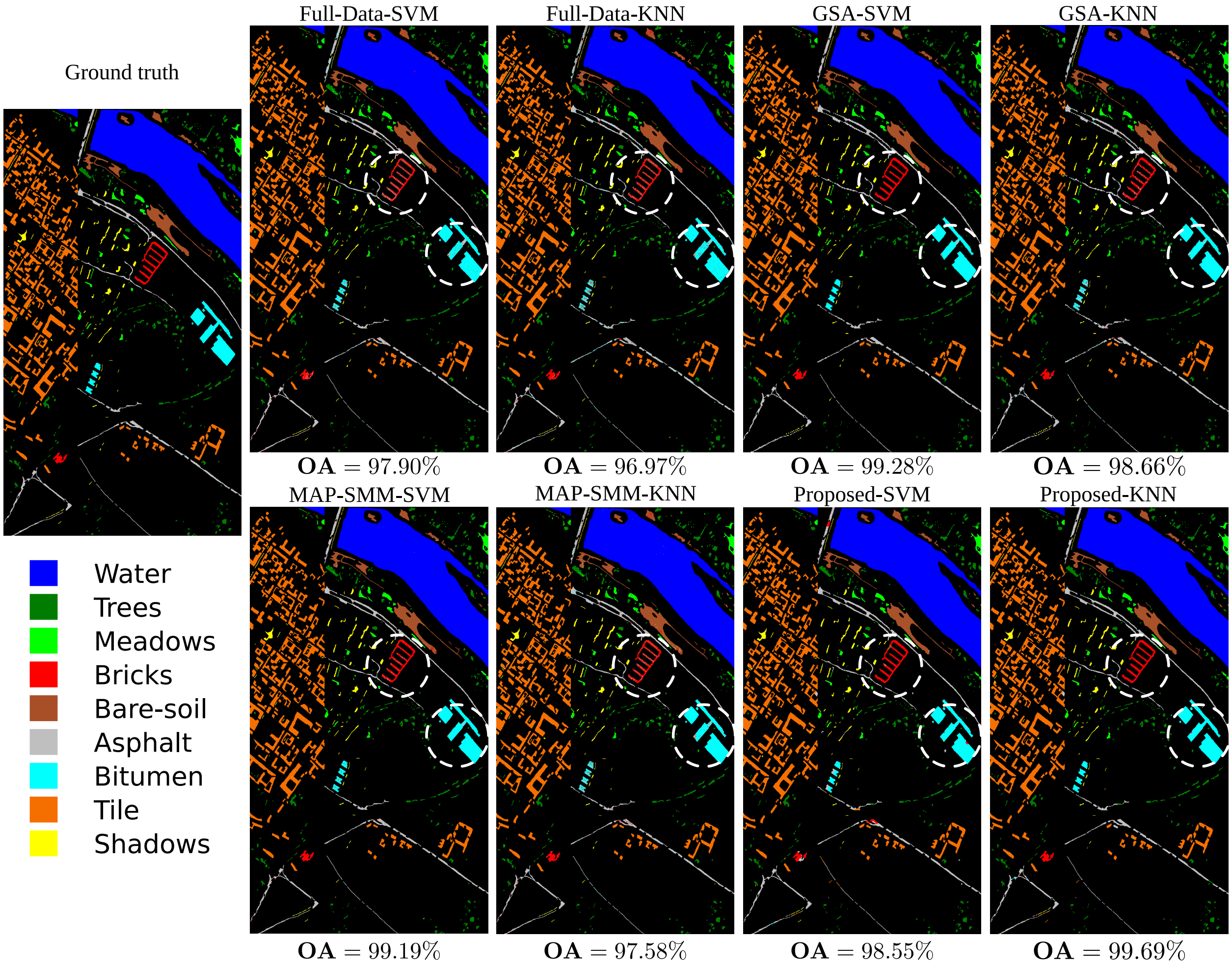


Figure 78. Visual classification results on Pavia Center image. Figure shows the ground truth, and the results for Full-Data, GSA, MAP-SMM, and the Proposed approach using both SVM and KNN classifiers.

and the MAP-SMM are greatly due to the denoising process performed in those methods.

Conclusions. A supervised classification approach from remote sensing fused images has been developed. This approach fuses the spatial information of an RGB image, extracted using superpixels, with the spectral information of an HS image. This fusion procedure is effectively formulated as an optimization problem which was solved using ADMM. The classification, using a supervised method, of the fused image boosts the classification performance, and therefore, it

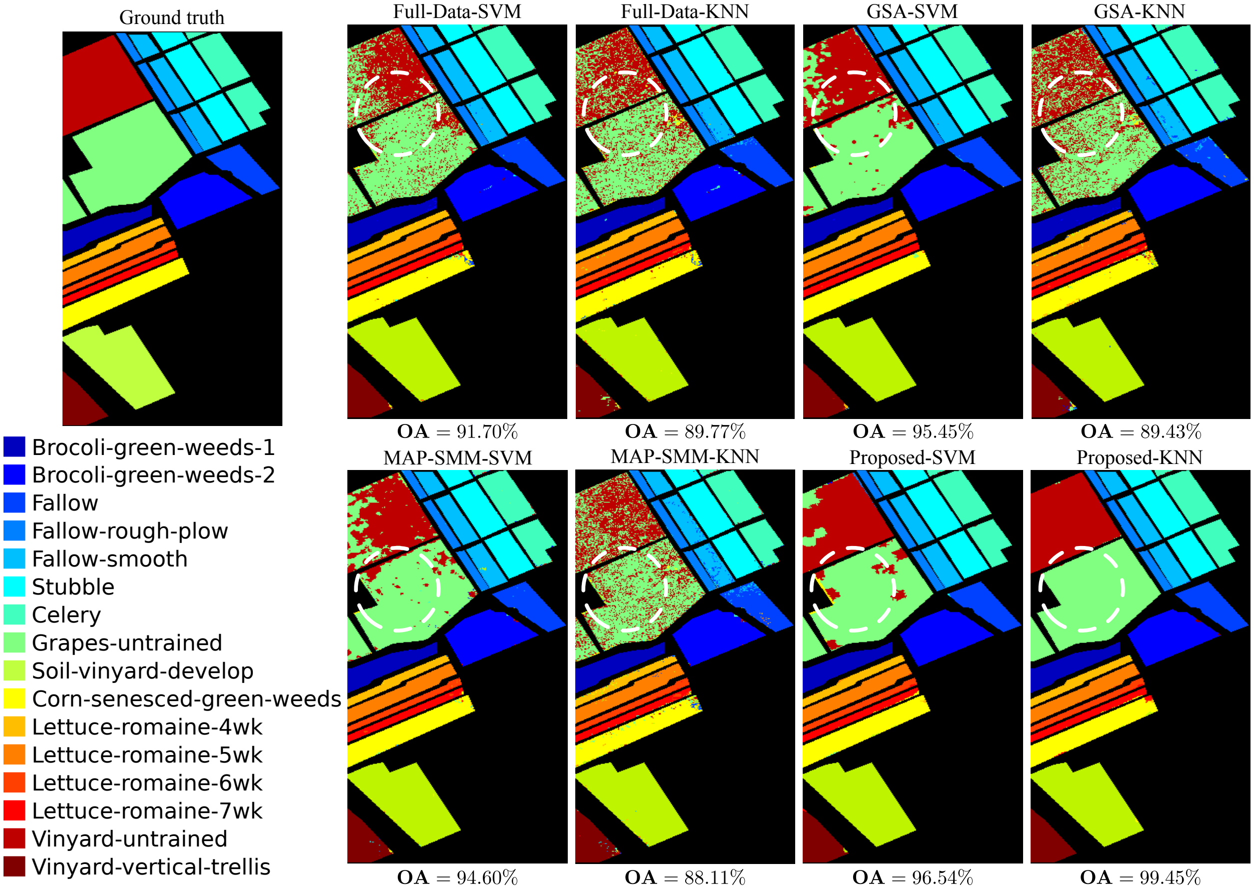


Figure 79. Visual classification results on Salinas Valley image. Figure shows the ground truth, and the results for Full-Data, GSA, MAP-SMM, and the Proposed approach using both SVM and KNN classifiers.

reduces the number of pixels that need to be classified, since the image is grouped in superpixels. Therefore, the overall classification time decreases in comparison with other tested approaches. In all performed experiments it can be noticed that adding spatial contextual information via superpixel segmentation significantly improves the classification results. Although the classification from the fused images, obtained with GSA and MAP-SMM methods, is good, the classification time is significantly greater than using the proposed method. In general, the framework developed in this paper provides a faster method to directly fuse and classify hyperspectral images instead of

Table 36

Quantitative results of the different classification methods for the Salinas Valley image.

	FULL		**GSA**		**MAP-SMM**		**PROPOSED**	
Class	**SVM**	**KNN**	**SVM**	**KNN**	**SVM**	**KNN**	**SVM**	**KNN**
Brocoli-green-1	98.83	99.00	99.48	98.13	99.47	99.74	**100**	99.86
Brocoli-green-2	99.23	99.10	**99.91**	99.78	99.33	99.12	99.80	99.60
Fallow	99.24	97.07	99.76	95.07	**99.91**	93.85	99.78	99.78
Fallow-rough-plow	98.86	99.52	98.09	98.39	98.58	98.87	**99.65**	99.57
Fallow-smooth	**99.35**	96.59	98.98	95.63	98.19	93.10	98.90	98.95
Stubble	99.83	99.70	**99.99**	99.86	99.86	99.50	99.92	99.79
Celery	99.47	99.41	99.48	98.96	99.08	98.61	**99.75**	99.60
Grapes-untrained	83.21	76.73	92.60	77.52	90.71	73.89	91.51	**99.72**
Soil-vinyard-develop	99.50	98.77	**99.99**	99.46	99.82	98.96	99.94	99.80
Corn-senesced-weeds	96.29	93.34	98.32	91.73	97.71	90.53	98.28	**98.67**
Lettuce-romaine-4wk	97.80	98.25	97.88	89.79	97.45	88.56	97.95	**98.93**
Lettuce-romaine-5wk	99.83	99.65	**99.97**	98.42	99.90	98.65	98.72	99.49
Lettuce-romaine-6wk	98.01	97.74	**98.67**	97.93	97.38	96.59	96.56	92.35
Lettuce-romaine-7wk	96.63	94.51	**98.90**	94.92	96.80	93.27	95.63	90.88
Vinyard-untrained	66.45	67.07	78.71	66.90	76.90	65.61	78.98	**99.28**
Vinyard-vertical-trellis	98.76	97.63	97.90	96.38	98.86	95.99	**99.02**	98.88
AA	95.71	94.63	97.42	93.68	96.87	92.80	97.15	**98.45**
OA	91.29	89.47	95.17	89.20	94.34	87.86	94.93	**99.21**
Kappa	0.9030	0.8828	0.9462	0.8797	0.9369	0.8649	0.9435	**0.9912**
Time	33.4255	17.9035	34.4387	18.8947	35.4682	18.4208	5.4616	**1.4140**

first reconstructing the high spatial and spectral resolution image and then perform classification.

Appendix C. Mathematical Proofs and Additional Algorithms

Proof of Theorem 1

Proof. According to the definition of $f(\Phi)$ in Eq. (55) we have that

$$f(\Phi) > 0, \tag{114}$$

because $\|\Phi^T\Phi - I\|_F^2 + \|\Phi\Phi^T - I\|_F^2 > 0$. Then, from Eq. (114), f is bounded below. Now, in order to prove that $\{\Phi^{i_t}\}$ generated by Algorithm 6 converges to a stationary point we need to ensure that $\{f(\Phi^{i_t})\}$ is a non-increasing sequence. Notice that from Line 1 of Algorithm 6 we have

$$f(\Phi^0) = 2SL(SL-2) + L + S, \tag{115}$$

which is the maximum possible value for f in Eq. 55, because $\Phi^{i_t} \in \{0,1\}^{S\times L}$. Then, following the iterations of Algorithm 6, specifically Lines 4, 11 and 24, we have that each row of Φ^{i_t} satisfies that

$$(\Phi^{i_t})_k = 0, \quad for\ \ k < \lambda_1^{i_t}\ \ or\ \ k > \lambda_2^{i_t}, \tag{116}$$

implying that the first and second constraints in Eq. 55 are satisfied. Further, according to Lines 5-11 in Algorithm 6, the spectral bandwidth is chosen, where fewer elements have been sensed for each row of Φ, which leads to minimize the term $\|\Phi^{i_t}(\Phi^{i_t})^T - \mathbf{I}\|_F^2$ in each iteration. Given the fact that the matrix Φ in Algorithm 6 is designed row by row, eliminating one-valued entries from a particular row, we have

$$\|\Phi^{i_t+1}(\Phi^{i_t+1})^T - \mathbf{I}\|_F^2 \leq \|\Phi^{i_t}(\Phi^{i_t})^T - \mathbf{I}\|_F^2. \tag{117}$$

Furthermore, in lines 13-17, the inner products between two adjacent columns of Φ are calculated in the chosen bandwidth. Then $\frac{1}{2}\Delta$ columns, with the smallest inner product, are uniformly selected. Therefore, we have that

$$\|(\Phi^{i_t+1})^T\Phi^{i_t+1} - \mathbf{I}\|_F^2 \leq \|(\Phi^{i_t})^T\Phi^{i_t} - \mathbf{I}\|_F^2. \tag{118}$$

Thus, taking into account Eq. (117) and Eq. (118) it can be concluded that

$$\begin{aligned} f(\Phi^{i_t+1}) &= \|\Phi^{i_t+1}(\Phi^{i_t+1})^T - \mathbf{I}\|_F^2 + \|(\Phi^{i_t+1})^T\Phi^{i_t+1} - \mathbf{I}\|_F^2 \\ &\leq f(\Phi^{i_t}), \end{aligned} \tag{119}$$

for all iterations. Also, combining Eq. (114) and Eq. (119) we have that the sequence $\{f(\Phi^{i_t})\} \to w^*$ for some $w^* > 0$ because of two facts. First from Eq. (119), $\{f(\Phi^{i_t})\}$ is a non-decreasing sequence and from Eq. (114) it is bounded below (James, 1961). Furthermore, given the fact that f is a continuous function we can ensure that there exists $\Phi^* \in \{0,1\}^{S\times L}$ such that $\{\Phi^{i_t}\}_{i_t\in\mathbb{N}} \to \Phi^*$ (James, 1961). Then, the sequences generated by Algorithm 6 converge to a stationary point. Also, considering the design procedure in Algorithm 6, Φ^* satisfies the design equality constraint in Eq. (55). Therefore, Eq. (55) can be efficiently solved via Algorithm 6. □

Proof of Theorem 2

Proof. Let $\mathbf{y}_j = \Phi\mathbf{f}_j$, where $\mathbf{y}_j \in \mathbb{R}^S$ and $\mathbf{f}_j \in \mathbb{R}^L$. Observe that, it is possible to decompose $\Phi = \hat{\Phi}\mathbf{J}$, where $\mathbf{J}$ can be viewed as a band-selection matrix which selects neighboring bands on the spectral pixel $\mathbf{f}_j$ and groups them in blocks, as $\mathbf{J}\mathbf{f}_j = \bar{\mathbf{f}}_j = \left[(\bar{\mathbf{f}}_j^0)^T, (\bar{\mathbf{f}}_j^1)^T, \cdots, (\bar{\mathbf{f}}_j^S)^T\right]$ with $\bar{\mathbf{f}}_j \in \mathbb{R}^{S\Delta}$ and $\bar{\mathbf{f}}_j^s \in \mathbb{R}^{\Delta}$. The structure of $\hat{\Phi}$ and $\mathbf{J}$ are depicted in Fig. 42 and mathematically described in Eq. (56) and Eq. (57), respectively.

Taking into account the structure of the designed matrix $\hat{\Phi}$, it is desired to show that the minimum and maximum singular values of $\hat{\Phi}$ are bounded. The following lemma establishes the relationship between the approximate isometry matrix and its singular values.

Lemma 1. Consider the decomposition of the matrix obtained from Algorithm 6 given by $\Phi = \hat{\Phi}\mathbf{J}$. Then, $\hat{\Phi}$ satisfies

$$\left\|\frac{2}{\Delta}\hat{\Phi}\hat{\Phi}^T - \mathbf{I}\right\|_2 \leq \rho$$

for some $\rho \in (0,1)$ with probability at least $1 - 2e^{-\rho^2/2}$. Furthermore,

$$1-\rho \leq \sigma_{min}\left(\frac{\sqrt{2}}{\sqrt{\Delta}}\hat{\Phi}\right) \leq \sigma_{max}\left(\frac{\sqrt{2}}{\sqrt{\Delta}}\hat{\Phi}\right) \leq 1+\rho$$

Proof. Taking into account the block rectangular structure of the matrix $\hat{\Phi}$ we have that $0 \leq (\hat{\Phi}\hat{\Phi}^T)_{i,j} \leq \Delta$. Notice that our designed matrix guarantees that $\mathbb{E}\left[(\hat{\Phi}\hat{\Phi}^T)_{i,i}\right] = \frac{\Delta}{2}$ and $\frac{2}{\Delta}\hat{\Phi}\hat{\Phi}^T - \mathbf{I} = \mathbf{0}$. For this reason, the Hoeffding's inequality for each diagonal entry is given by

$$P\left(\left|\frac{2}{\Delta}\left(\hat{\Phi}\hat{\Phi}^{T}\right)_{i,i}-1\right|\geq\rho\right)\leq 2\exp\left\{-\frac{\rho^{2}}{2}\right\}, \tag{120}$$

for $i = 0, \cdots S-1$ and some $\rho \in (0,1)$. Besides, the values outside the diagonal are 0. Therefore, from (120) we have

$$\left\|\frac{2}{\Delta}\hat{\Phi}\hat{\Phi}^{T}-\mathbf{I}\right\|_{2}\leq\rho, \tag{121}$$

with probability at least $1-2e^{-\rho^{2}/2}$. On the other hand, the following lemma establishes the bound of approximated isometry matrices.

Lemma 2. (Approximate isometry). Consider a matrix $\mathbf{B}$ that satisfies

$$||\mathbf{B}^{*}\mathbf{B}-\mathbf{I}||\leq\max(\rho,\rho^{2})$$

for some $\rho > 0$. Then,

$$1-\rho\leq\sigma_{min}(\mathbf{B})\leq\sigma_{max}(\mathbf{B})\leq 1+\rho.$$

Proof. The proof can be found in (Vershynin, 2010). □

Notice that, $\sigma_{min}(\mathbf{B})$ denotes the smallest non-zero singular value of $\mathbf{B}$. Using Lemma 2, for $\hat{\Phi}$, we can conclude that the following inequality holds

$$1-\rho \leq \sigma_{min}\left(\frac{\sqrt{2}}{\sqrt{\Delta}}\hat{\mathbf{\Phi}}\right) \leq \sigma_{max}\left(\frac{\sqrt{2}}{\sqrt{\Delta}}\hat{\mathbf{\Phi}}\right) \leq 1+\rho \tag{122}$$

with probability al lest $1-2e^{-\rho^2/2}$. □

Now, denote $\mathbb{W}_d$ as the set of all full rank submatrices $\tilde{\mathbf{F}}_d \in \mathbb{R}^{S\Delta \times Q_d}$ of $\bar{\mathbf{F}}_d = \mathbf{J}\mathbf{F}_d$. Therefore if $S \geq Q_d$, the rank of $\tilde{\mathbf{Y}}_d = \hat{\mathbf{\Phi}}\bar{\mathbf{F}}_d$ is preserved, i.e, $\tilde{\mathbf{Y}}_d$ has Q_d non-zero singular values. Notice that

$$\sigma_{min}\left(\hat{\mathbf{\Phi}}\tilde{\mathbf{F}}_d\right) = \sigma_{min}\left(\hat{\mathbf{\Phi}}\tilde{\mathbf{F}}_d\right)\frac{\sigma_{max}\left(\hat{\mathbf{\Phi}}\tilde{\mathbf{F}}_d\right)}{\sigma_{max}\left(\hat{\mathbf{\Phi}}\tilde{\mathbf{F}}_d\right)}. \tag{123}$$

Let $k = \sigma_{max}(\mathbf{B})/\sigma_{min}(\mathbf{B})$ be the condition number of $\mathbf{B}$, then from singular values inequalities (Horn et al., 1990), we have

$$\begin{aligned}\sigma_{min}\left(\hat{\mathbf{\Phi}}\tilde{\mathbf{F}}_d\right) &= \frac{1}{k}\sigma_{max}\left(\hat{\mathbf{\Phi}}\tilde{\mathbf{F}}_d\right)\\ &\geq \frac{1}{k}\sigma_{min}\left(\hat{\mathbf{\Phi}}\right)\sigma_{min}\left(\tilde{\mathbf{F}}_d\right).\end{aligned} \tag{124}$$

Given that $S \geq Q_d$, the matrix $\tilde{\mathbf{F}}_d^T\hat{\mathbf{\Phi}}^T\hat{\mathbf{\Phi}}\tilde{\mathbf{F}}_d$ is invertible thus, $\sigma_{min}\left(\hat{\mathbf{\Phi}}\tilde{\mathbf{F}}_d\right) = \tilde{\sigma}_{Q_d}\left(\hat{\mathbf{\Phi}}\tilde{\mathbf{F}}_d\right)$, and $k \geq 1$. Using the inequality of (122), we have

$$\tilde{\sigma}_{Q_d}\left(\hat{\mathbf{\Phi}}\tilde{\mathbf{F}}_d\right) \geq \frac{(1-\rho)}{k}\tilde{\sigma}_{Q_d}\left(\tilde{\mathbf{F}}_d\right). \tag{125}$$

Then, from (Elhamifar and Vidal, 2013, Theorem 3)

$$\max_{\tilde{\mathbf{F}}_d \in \mathbb{W}_d} \tilde{\sigma}_{Q_d}(\tilde{\mathbf{F}}_d) > \sqrt{Q_d} \max_{d \neq d'} \cos(\theta_{d,d'}), \tag{126}$$

we conclude that

$$\tilde{\sigma}_{Q_d}(\tilde{\mathbf{Y}}_d) \geq \frac{(1-\rho)}{k} \sqrt{Q_d} \max_{d \neq d'} \cos(\theta_{d,d'}), \tag{127}$$

holds with probability at least $1 - 2e^{-\rho^2/2}$. $\square$

ADMM algorithm to solve the optimization problem in Eq. 60.

Let's define an auxiliary matrix $\mathbf{U} \in \mathbb{R}^{MN \times MN}$ with the same size as the sparse coefficient matrix $\mathbf{Z}$ is defined to separate the variables. In this way, we only need to solve the following optimization problem

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{U},\bar{\mathbf{Z}}} \quad & \|\mathbf{Z}\|_1 + \frac{\lambda}{2}\|\mathbf{Y} - \mathbf{Y}\mathbf{U}\|_F^2 + \frac{\alpha}{2}\|\bar{\mathbf{Z}} - \mathbf{U}\|_F^2 \\ \text{s.t.} \quad & \mathbf{U}^T\mathbf{1}, \mathbf{U} = \mathbf{Z}-\operatorname{diag}(\mathbf{Z}) \end{aligned} \tag{128}$$

Two penalty terms corresponding to $\mathbf{U^T}\mathbf{1} = \mathbf{1}$ and $\mathbf{U} = \mathbf{Z} - \operatorname{diag}(\mathbf{Z})$ are then added to the penalty function of (128) to obtain the following optimization problem:

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{U},\bar{\mathbf{Z}}} \quad & \|\mathbf{Z}\|_1 + \frac{\lambda}{2}\|\mathbf{Y} - \mathbf{Y}\mathbf{U}\|_F^2 + \frac{\alpha}{2}\|\bar{\mathbf{Z}} - \mathbf{U}\|_F^2 \\ & + \frac{\rho}{2}\|\mathbf{U}^T\mathbf{1} - \mathbf{1}\|_2^2 + \frac{\rho}{2}\|\mathbf{U} - (\mathbf{Z}-\operatorname{diag}(\mathbf{Z}))\|_F^2 \\ \text{s.t.} \quad & \mathbf{U}^T\mathbf{1}, \mathbf{U} = \mathbf{Z}-\operatorname{diag}(\mathbf{Z}) \end{aligned} \tag{129}$$

It can be easily proved that the solutions to Eq. (60) and (128) coincide with that of (129). Next, we introduce a vector $\bar{\delta} \in \mathbb{R}^{MN}$ and a matrix $\bar{\Delta} \in \mathbb{R}^{MN \times MN}$ as Lagrange multipliers for the two equality constraints in (129) to obtain the Lagrange function

$$\begin{aligned}\mathscr{L}(\mathbf{Z},\mathbf{U},\bar{\mathbf{Z}},\bar{\delta},\bar{\Delta}) = \|\mathbf{Z}\|_1 + \frac{\lambda}{2}\|\mathbf{Y}-\mathbf{Y}\mathbf{U}\|_F^2 + \frac{\alpha}{2}\|\bar{\mathbf{Z}}-\mathbf{U}\|_F^2 \\ + \frac{\rho}{2}\|\mathbf{U}^T\mathbf{1}-\mathbf{1}\|_2^2 + \frac{\rho}{2}\|\mathbf{U}-(\mathbf{Z}-\mathrm{diag}(\mathbf{Z}))\|_F^2 \\ + \bar{\delta}^T(\mathbf{U}^T\mathbf{1}-\mathbf{1}) + \mathrm{tr}(\bar{\Delta}^T(\mathbf{U}-\mathbf{Z}+\mathrm{diag}(\mathbf{Z})))\end{aligned} \tag{130}$$

where tr(·) denotes the trace operator of a given matrix.

The optimization problem in Eq. (60) can then be divided intro three subproblems:

1. $\mathbf{U}^{(k+1)}$ is obtained by minimizing $\mathscr{L}$ with respect to $\mathbf{U}$, while $(\mathbf{Z}^{(k)},\bar{\mathbf{Z}}^{(k)},\bar{\delta}^{(k)},\bar{\Delta}^{(k)})$ are fixed. We calculate the derivative of $\mathscr{L}$ with respect to $\mathbf{U}$ and set it to zero to obtain the calculation formula of $\mathbf{U}$ as follows:

$$\begin{aligned}(\lambda\mathbf{Y}^T\mathbf{Y}+\alpha\mathbf{I}+\rho\mathbf{1}\mathbf{1}^T+\rho\mathbf{I})\mathbf{U}^{(k+1)} = \lambda\mathbf{Y}^T\mathbf{Y}+\alpha\bar{\mathbf{Z}}^{(k)} \\ +\rho(\mathbf{1}\mathbf{1}^T+\mathbf{Z}^{(k)}) - 1\delta^{(k)^T} - \bar{\Delta}^{(k)}.\end{aligned} \tag{131}$$

2. $\mathbf{Z}^{(k+1)}$ can be obtained by minimizing $\mathscr{L}$ with respect to $\mathbf{Z}$, while $(\mathbf{U}^{(k+1)},\bar{\mathbf{Z}}^{(k)},\bar{\delta}^{(k)},\bar{\Delta}^{(k)})$ are fixed

$$\begin{aligned}\mathbf{Z}^{(k+1)} &= \mathbf{J}-\mathrm{diag}(\mathbf{J}), \\ \mathbf{J} &= \Gamma_{\frac{1}{\rho}}\left(\mathbf{U}^{(k+1)}+\frac{\bar{\Delta}^{(k)}}{\rho}\right),\end{aligned} \tag{132}$$

where $\Gamma_{1/\rho}(\cdot)$ is a shrinkage-thresholding operator, $\Gamma_{1/\rho}(v) = (|u| - (1/\rho))_{+}\text{sgn}(v)$, and the operator $(\cdot)_{+}$ returns its arguments if it is nonnegative and returns zero otherwise. We then update $\bar{\mathbf{Z}}^{(k+1)}$ utilizing $\mathbf{Z}^{(k+1)}$ (see Fig. 43).

3. The Lagrange multipliers $\bar{\delta}^{(k+1)}$ and $\bar{\Delta}^{(k+1)}$ are obtained through a gradient ascent update with step size $\rho = 300$.

$$\bar{\delta}^{(k+1)} = \bar{\delta}^{(k)} + \rho\left(\mathbf{Z}^{(k+1)}\mathbf{1} - \mathbf{1}\right) \tag{133}$$

$$\bar{\Delta}^{(k+1)} = \bar{\Delta}^{(k)} + \rho\left(\mathbf{U}^{(k+1)} - \mathbf{Z}^{(k+1)}\right). \tag{134}$$

These three steps are repeated until convergence is achieved or the number of iterations exceeds the maximum iteration number. Specifically, the iteration is terminated when we have $\|\mathbf{U}^{(k)^T}\mathbf{1} - \mathbf{1}\|_\infty \leq \bar{\varepsilon}, \|\mathbf{U}^{(k)} - \mathbf{Z}^{(k)}\|_\infty \leq \bar{\varepsilon}, \|\mathbf{U}^{(k)} - \mathbf{U}^{k-1}\|_\infty \leq \bar{\varepsilon}$, where $\bar{\varepsilon}$ denotes the error tolerance for the primal and dual residuals.

Estimation of the number of subspaces

Typically, to test the performance of SC-based algorithms, it is assumed that the labels or ground truth are known. However, in practice, the number of clusters k_i could be unknown. In such a case, the value of k_i can be estimated using a rank selection technique that selects the rank and sparsity tuning parameters simultaneously for hyperspectral data (Rasti et al., 2015). Samiran et al. (Das et al., 2016) proposed estimated the number of clusters present in the HSI choosing of the optimal threshold value between the higher and smaller eigenvalues of the matrix covariance of the HSI. These approaches are interesting, but they estimate the number of clusters using direct information from the spectral image. Therefore, we decided to use the approach proposed by Von et al. (Von Luxburg, 2007a), which estimates the number of clusters using the Laplacian matrix **A**:

$$k_i \approx \underset{i\in[u_e-1]}{\arg\max}(\lambda_{i+1} - \lambda_i), \tag{135}$$

where $\lambda_1 \leq \lambda_2, \cdots, \leq \lambda_{u_e}$ are the eigenvalues of the normalized Laplacian of the graph given by **A** matrix.

Appendix D. Code Implementations and Repositories

The implementation of the algorithms developed in this thesis are publicly available in the following repositories.

Paper Title	Project Page / Code
A Fast and Accurate Similarity-constrained Subspace Clustering Algorithm for Hyperspectral Image Hinojosa et al. (2021b)	`https://tinyurl.com/SC-SSC`
Coded Aperture Design for Compressive Spectral Subspace Clustering Hinojosa et al. (2018b)	`https://tinyurl.com/HT-CS-CSC`
Hyperspectral image segmentation using 3D regularized subspace clustering model Hinojosa et al. (2021c)	`https://tinyurl.com/h-3DS-SSC`
C-3SPCD: Coded aperture similarity constrained design for Spatio-spectral Hinojosa et al. (2021c)	`https://tinyurl.com/HT-C-3SPCD`

Appendix E. Complementary works

This chapter describes additional research developed during my doctoral studies. Although such works are not directly related to the objectives of this thesis, they are inspired by different ideas derived from my studies in data similarity, unsupervised learning, classification, and coding patterns design.

Privacy-preserving vision systems (Hinojosa et al., 2021a). This work addressed the problem of user privacy preservation in computer vision applications. In particular, inspired by the trend of joint design of optics and algorithms, this work proposes an end-to-end deep neural network framework composed of an optical encoder, that provides hardware-level protection and a software decoder (deep neural network) that perform the visual task. The visual privacy protection provided by the optical layer is achieved by parameterizing the camera lens's point spread function (PSF) using the Zernike basis. This particular work ((Hinojosa et al., 2021a)) shows that the proposed privacy-preserving deep optics approach successfully degrades or inhibits private attributes while maintaining important features to perform human pose estimation. Results of this research include the publication of the work in the international conference on computer vision (ICCV) 2021 and an U.S. patent between Stanford University and the Universidad Industrial of Santander (U.S. Patent App. 63/212,528, 2021).

Subspace-based Domain Adaptation Using Similarity Constraints for Pneumonia Diagnosis within a Small Chest X-ray Image Dataset (Sanchez et al., 2021a). This work propose a subspace-based domain adaptation technique to increase pneumonia detection accuracy using a small training dataset. This dataset is augmented with automatically selected images from a large dataset acquired in a different medical center. This is performed by computing a subspace basis of the target domain dataset on which is projected the source dataset to find the most representative images. Augmenting the training set using the proposed method allows achieving an improvement from 90.03% to 96.18% in overall accuracy using the Xception neural network. This work was published in the IEEE 18th International Symposium on Biomedical Imaging (ISBI) 2021.

Adaptive grayscale compressive spectral imaging using optimal blue noise coding patterns (Diaz et al., 2019). This work proposes an adaptive grayscale coded aperture design for the coded aperture snapshot spectral imager (CASSI), which combines the advantages of blue noise and block-unblock coding patterns. Blue noise coding patterns are optimal and provide high-quality image reconstructions on regions of non-saturated compressed pixels. On the other hand, the block-unblock coding patterns provide redundancy in the sampling which helps to reduce the saturation in the detector. Further, the saturation is reduced between snapshots by using an adaptive filter that updates the entries of the grayscale coded aperture based on the previously acquired measurements. The proposed coded apertures are optimized such that the number of saturated measurements is minimized. Extensive simulations and an experimental setup were made using the CASSI sensing scheme, where the results show an improvement up to 2 dB of peak signal-to-noise ratio when the proposed coded aperture design is compared with traditional block-unblock coded apertures. This work was published in the journal “Optics & Laser Technology”, Volume

117, 2019. Also, this project was selected by the administrative department of science, technology, and innovation (MinCiencias) to grant me a Young Researcher scholarship that covered my expenses during part of my postgraduate studies.