



Feature Extraction and Data Reduction for Hyperspectral Remote Sensing Earth Observation

Jaime Zabalza

In the fulfilment of the requirement for the degree of
Doctor of Philosophy

Centre for Excellence in Signal and Image Processing
Department of Electronic and Electrical Engineering
University of Strathclyde

Supervised by
Doctor Jinchang Ren
Professor Stephen Marshall

© June 2015

Declaration

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Jaime Zabalza

© June 2015

Acknowledgements

I would like to thank a number of people for their guidance and help throughout the duration of my research. In particular, I express my most sincere gratitude to my supervisor, Doctor Jinchang Ren, for putting his faith in my skills and giving me this opportunity, and also cosupervisor, Professor Stephen Marshall, for his guidance and encouragement.

Thanks must also go to all coauthors in my publications, external partners in collaboration with my research group, the journal reviewers involved in the peer-review process of my publications, and also Doctor Paul Murray, Doctor Xiuping Jia, and Doctor Stephan Weiss, my VIVA convenor, external, and internal examiners, respectively. I am grateful as well to all my colleagues during this period for sharing their time and expertise, including special thanks to Professor Bruce Rafert from North Dakota State University for changing my personal view and thoughts about the remote sensing field.

This research would have been impossible without the funding from the University of Strathclyde and my supervisors. I really appreciate their economic support and I am confident that my research outcomes have returned this effort in the best possible way. I must thank as well the university staff for their guidance and treat.

Finally, on a more personal note, a big thank you to all my family and friends for their encouragement and understanding, with loving gratitude to Candela for her support in the distance.

Abstract

Earth observation and land-cover analysis became a reality in the last 2-3 decades thanks to NASA airborne and spacecrafts such as Landsat. Inclusion of Hyperspectral Imaging (HSI) technology in some of these platforms has made possible acquiring large data sets, with high potential in analytical tasks but at the cost of advanced signal processing. In this thesis, effective/efficient feature extraction methods are proposed.

Initially, contributions are introduced for efficient computation of the covariance matrix widely used in data reduction methods such as Principal Component Analysis (PCA). By taking advantage of the cube structure in HSI, onsite and real-time covariance computation is achieved, reducing memory requirements as well. Furthermore, following the PCA algorithm, a novel method called Folded-PCA (Fd-PCA) is proposed for efficiency while extracting both global and local features within the spectral pixels, achieved by folding the spectral samples from vector to matrix arrays.

Inspired by Empirical Mode Decomposition (EMD) methods, a recent and promising algorithm, Singular Spectrum Analysis (SSA), is introduced to hyperspectral remote sensing, performing extraction of features in the spectral (1D-SSA) and also the spatial (2D-SSA) domain. By successfully suppressing the noise and enhancing the useful signal, more effective feature extraction and data classification are achieved. Furthermore, a fast implementation of the SSA methods is also proposed, leading to reduction of computational complexity. In addition, combination of both spectral- and spatial-domain exploitation is also included, comprising data reduction.

Finally, promising Deep Learning (DL) approaches are evaluated by the analysis of Stacked AutoEncoders (SAEs) for feature extraction and data reduction, introducing a method called Segmented-SAE (S-SAE), working in local regions of the spectral domain. Preliminary results have validated its great potential in this context.

Contents

Contents	v
List of Figures	viii
List of Tables	x
Nomenclature	xvi
1 Introduction	1
1.1 Research Motivation	1
1.2 Original Contribution	3
1.3 Thesis Organisation	4
2 Background and Related Work	6
2.1 Introduction	6
2.2 Hyperspectral Remote Sensing Earth Observation	6
2.2.1 The Data Sources	7
2.2.2 Processing the Hyperspectral Cube	9
2.3 Related Data Processing Framework	11
2.3.1 Signal Processing Chain	11
2.3.2 Design of Experiments	15
2.3.3 Performance Assessment	18
2.4 A Review of Hyperspectral Feature Extraction	20
2.4.1 Spectral-Domain Methods	20
2.4.2 Joint Spatial-Spectral-Domain Methods	24
2.5 Summary	26
3 Structured Covariance (SC) for on-the-Fly Feature Extraction	28
3.1 Introduction	29
3.2 Data Acquisition and Covariance Computation	30
3.2.1 Data Acquisition in HSI	30

3.2.2	Conventional Computation of Covariance Matrix	31
3.3	Proposed Structured Covariance (SC)	33
3.3.1	Structured Covariance Schemes	34
3.3.2	Mathematical Equivalencies	37
3.3.3	Correction for Real-Time Covariance Computation	38
3.4	Analysis and Evaluation	40
3.4.1	Experiments	41
3.4.2	Results	42
3.5	Summary	46
4	Folded-PCA for Efficient and Effective Data Reduction	48
4.1	Introduction	49
4.2	The Conventional PCA Method	50
4.2.1	Algorithm Description	50
4.2.2	Variations of PCA	52
4.3	Proposed Folded-PCA	53
4.3.1	Algorithm Description	54
4.3.2	Implementation Considerations	56
4.3.3	Extension for Uneven Folding	58
4.4	Analysis and Evaluation	59
4.4.1	Experiments	59
4.4.2	Results	60
4.5	Summary	64
5	Spectral Extraction and Denoising by 1D-SSA	66
5.1	Introduction	67
5.2	The 1D-SSA Method in HSI	68
5.2.1	Algorithm Description	68
5.2.2	Application Example in HSI	71
5.3	Fast Implementation of 1D-SSA (F-1D-SSA)	72
5.3.1	Concept	73
5.3.2	Algorithm Description	73
5.4	Analysis and Evaluation	76
5.4.1	Experiments	76
5.4.2	Results	78
5.5	Summary	87
6	Spatial Extraction and Denoising by 2D-SSA	88
6.1	Introduction	89

6.2	The 2D-SSA Method in HSI	90
6.2.1	Algorithm Description	90
6.2.2	Application Example in HSI	94
6.3	Fast Implementation of 2D-SSA (F-2D-SSA)	96
6.3.1	Concept	96
6.3.2	Algorithm Description	97
6.4	Analysis and Evaluation	98
6.4.1	Experiments	99
6.4.2	Results	101
6.5	Summary	116
7	Segmented-SAE for Improved Data Reduction	117
7.1	Introduction	118
7.2	Stacked AutoEncoders (SAEs) in HSI	119
7.3	Proposed S-SAE	121
7.4	Analysis and Evaluation	123
7.4.1	Experiments	123
7.4.2	Results	124
7.5	Summary	131
8	Conclusions and Future Work	132
8.1	Introduction	132
8.2	Main Contributions	133
8.3	Future Work	135
	References	136
	Appendix A Remote Sensing Data Sets	150
A.1	92AV3C (Indian Pines) from AVIRIS (NASA)	150
A.2	Indian Pines B from AVIRIS (NASA)	152
A.3	Salinas C from AVIRIS (NASA)	154
A.4	Pavia University A (Pavia UA) from ROSIS (DLR)	155
A.5	Pavia Centre A (Pavia CA) from ROSIS (DLR)	156
A.6	Botswana A from HYPERION (NASA)	157
	Appendix B Publications by the Author	158
B.1	Journal Publications	158
B.2	Journal Publications under Consideration	160
B.3	Conference Publications	161

List of Figures

2.1	Examples of remote sensing by different satellites and aircrafts	7
2.2	From RGB to hyperspectral	8
2.3	Hyperspectral cube in remote sensing applications	8
2.4	Feature extraction and data mining for HSI	10
2.5	Spectral and spatial domains from hyperspectral cubes	10
2.6	Classical signal processing chain	11
2.7	AVIRIS acquisition instrument (image credit: JPL-NASA)	12
2.8	HYPERION onboard the EO-1 satellite (image credit: NASA)	13
3.1	3-D hyperspectral cube acquired by HSI devices	29
3.2	Basic scheme for pushbroom scanning and tuneable filtering	31
3.3	Subpartitions in sequential captures of a 3-D hyperspectral cube	32
3.4	Data matrix in conventional implementations	32
3.5	Conventional and SC computations in HSI	34
3.6	Mean OA using PCA ($F = 1$ to 10)	44
3.7	Approximated computation time after acquisition	46
4.1	Data matrix in conventional PCA	50
4.2	Folding procedure from vector to matrix array	54
4.3	Schematic representation of the covariance matrices	57
4.4	Folding procedure in the extended case	58
4.5	Mean OA for Indian Pines ($F = 40$, $N_h = 1$ to 40)	61
4.6	Mean OA for Indian Pines ($F = 10$ to 120)	62
4.7	Mean OA for Indian Pines B ($F = 10$ to 120)	62
4.8	Mean OA in the extended case for Indian Pines ($F = 10$ to 120)	64
5.1	Original and 1D-SSA-reconstructed pixels	72
5.2	HSI cube and its representative pixels	75
5.3	Workflow for (left) 1D-SSA and (right) F-1D-SSA	75
5.4	Mean OA for 92AV3C (9 classes) (10% training)	80
5.5	Difference between original and (F)-1D-SSA-reconstructed pixels	82

6.1	Decomposition of a HSI scene by 2D-SSA	95
6.2	Reconstruction of a HSI scene by 2D-SSA	96
6.3	HSI cube and its representative scenes	97
6.4	Workflow for (left) 2D-SSA and (right) F-2D-SSA	98
6.5	Original and (F)-2D-SSA-reconstructed scenes	111
7.1	Schematic representation of a basic AE	119
7.2	Schematic representation of a SAE (2 internal layers)	120
7.3	Difference between original and SAE-reconstructed pixels	121
7.4	Schematic representation of a S-SAE using two-layer SAEs	122
7.5	Correlation matrix with selected regions for 92AV3C	126
7.6	Correlation matrix with selected regions for Pavia CA	126
7.7	Mean OA for 92AV3C ($F = 5$ to 20)	128
7.8	Mean OA for Pavia CA ($F = 5$ to 20)	128
A.1	92AV3C data set with ground truth	150
A.2	Indian Pines B data set with ground truth	152
A.3	Salinas C data set with ground truth	154
A.4	Pavia UA data set with ground truth	155
A.5	Pavia CA data set with ground truth	156
A.6	Botswana A data set with ground truth (rotated 90° clockwise)	157

List of Tables

2.1	Workflow in the experiments	16
3.1	Main notation for conventional and SC computations	34
3.2	Mean OA using PCA ($F = 10$)	43
3.3	Matrices dimension and size (kB) for SC computations	44
3.4	Computational complexity in SC (offline)	45
3.5	Computational complexity in SC (during acquisition)	45
4.1	Matrices dimension for (Fd/S)-PCA computations	63
4.2	Computational complexity (MACs) in (Fd/S)-PCA	63
4.3	Number of MACs in (Fd/S)-PCA ($F = 30$)	63
5.1	Configuration for feature extraction methods in (F)-1D-SSA	78
5.2	Mean OA and McNemar's test [\mathcal{L}] (9 classes) (10% training)	79
5.3	Mean OA, CbC, and AA (10% training)	80
5.4	Mean OA and McNemar's test [\mathcal{L}] ($F = 15$) (10% training)	81
5.5	Mean cosine similarity scores in (F)-1D-SSA for 92AV3C	82
5.6	Mean OA and McNemar's test [\mathcal{L}] (best OAs)	83
5.7	Mean OA and McNemar's test [\mathcal{L}] in (F)-1D-SSA for 92AV3C	84
5.8	Mean OA and McNemar's test [\mathcal{L}] in (F)-1D-SSA for Pavia CA	84
5.9	Mean OA, CbC, and AA in (F)-1D-SSA for 92AV3C	85
5.10	Mean OA, CbC, and AA in (F)-1D-SSA for Pavia CA	85
5.11	Computational complexity (MACs) in (F)-1D-SSA	86
5.12	Number of MACs in (F)-1D-SSA	87
6.1	Configuration for feature extraction methods in (F)-2D-SSA	100
6.2	Mean OA and McNemar's test [\mathcal{L}] for 92AV3C	103
6.3	Mean OA and McNemar's test [\mathcal{L}] for Pavia UA	104
6.4	Mean OA and McNemar's test [\mathcal{L}] for Salinas C	105
6.5	Mean OA, CbC, and AA for 92AV3C (10% training)	106
6.6	Mean OA, CbC, and AA for Pavia UA (10% training)	106

6.7	Mean OA, CbC, and AA for Salinas C (10% training)	106
6.8	Computation time for 2D-EMD and 2D-SSA (*best OAs)	107
6.9	Mean OA and McNemar’s test [\mathcal{L}] for 92AV3C (best OAs)	108
6.10	Mean OA and McNemar’s test [\mathcal{L}] for Pavia UA (best OAs)	109
6.11	Mean OA and McNemar’s test [\mathcal{L}] for Salinas C (best OAs)	110
6.12	Mean cosine similarity scores in (F)-2D-SSA for 92AV3C	112
6.13	Mean OA and McNemar’s test [\mathcal{L}] in (F)-2D-SSA for 92AV3C	113
6.14	Mean OA, CbC, and AA in (F)-2D-SSA for 92AV3C	113
6.15	Computational complexity (MACS) in (F)-2D-SSA	114
6.16	Number of MACS in (F)-2D-SSA for 92AV3C	115
6.17	Computation time for (F)-2D-SSA for 92AV3C	115
7.1	Mean OA for SAE configuration ($F = 10$)	125
7.2	Configuration implemented for SAE	125
7.3	Configuration implemented for S-SAE	127
7.4	Mean OA and McNemar’s test [\mathcal{L}] (F) (best OAs)	127
7.5	Mean OA, CbC, and AA for 92AV3C	129
7.6	Mean OA, CbC, and AA for Pavia CA	129
7.7	Number of neural interconnections in (S)-SAE	130
A.1	92AV3C data set description	151
A.2	Indian Pines B data set description	153
A.3	Salinas C data set description	154
A.4	Pavia UA data set description	155
A.5	Pavia CA data set description	156
A.6	Botswana A data set description	157

Nomenclature

Roman Symbols

\mathbf{A}_{n_s}	Folded matrix in Fd-PCA
$\mathbf{a}_{n_s(n_h)}$	Row from the folded matrix in Fd-PCA
$\mathit{argmin}\{\}$	Argument of the minimum operator in computer science
\mathbf{B}_{n_s}	Pixel transformed by Fd-PCA
\mathbf{C}	Covariance matrix
$\mathbf{C}^{(p)}$	Covariance matrix from pixel partitions
$\mathbf{C}^{(r)}$	Covariance matrix from row partitions
$\mathbf{C}^{(c)}$	Covariance matrix from column partitions
$\mathbf{C}^{(b)}$	Covariance matrix from band partitions
$\mathbf{C}_{n_s}^{(p)}$	Partial covariance from pixel partitions
$\mathbf{C}_{n_r}^{(r)}$	Partial covariance from row partitions
$\mathbf{C}_{n_c}^{(c)}$	Partial covariance from column partitions
$\mathit{ceil}\{\}$	Ceiling operator in computer science
$\mathbf{CF}_{n_s}^{(p)}$	Correction factor for real-time SC pixel partition
$\mathbf{CF}_{n_r}^{(r)}$	Correction factor for real-time SC row partition
$\mathbf{CF}_{n_c}^{(c)}$	Correction factor for real-time SC column partition
\mathbf{CM}	Global correction matrix for real-time SC
\mathbf{cv}_k	Lagged vector from trajectory matrix in (F)-1D-SSA
$\mathit{diag}\{\}$	Diagonal operator in computer science
E	Size of the first layer in SAE
E_g	Size of the first layer for a segment in S-SAE
$\mathit{error}\{\}$	Error operator in computer science
$\mathit{expect}\{\}$	Expectation operator in computer science
F	Number of features in data reduction
F'	Number of row features in data reduction from Fd-PCA
F_g	Number of features in data reduction for a segment in S-SAE
$f\{\}$	Activation function operator in computer science
G	Number of segments in S-SAE
g	Index for segments in S-SAE

\mathbf{H}_{n_r}	Hankel-type matrix in (F)-2D-SSA
i	Generic 2-D position (i, j) of an element
j	Generic 2-D position (i, j) of an element
K	Complementary window in (F)-1D-SSA
K^{2D}	Complementary window in (F)-2D-SSA
k	Index for K (or K^{2D}) in (F)-2D/1D-SSA
L	Window size in (F)-1D-SSA
L_r	Window size (row terms) in (F)-2D-SSA
L_c	Window size (column terms) in (F)-2D-SSA
L^{2D}	Window size $(L_r \times L_c)$ in (F)-2D-SSA
L_{rank}	Rank of the trajectory matrix in (F)-2D/1D-SSA
L'	Complementary window in (F)-2D/1D-SSA (K or K^{2D})
l	Index related to the window size in (F)-2D/1D-SSA
M	Number of disjoint sets in (F)-2D/1D-SSA grouping
m	Index for disjoint sets in (F)-2D/1D-SSA grouping
$max\{\}$	Maximum operator in computer science
$mean\{\}$	Mean operator in computer science
$median\{\}$	Median operator in computer science
N_s	Spatial size (pixels) of the hyperspectral cube
n_s	Index for pixels from the hyperspectral cube
N_r	Spatial dimension (rows) of the hyperspectral cube
n_r	Index for rows from the hyperspectral cube
N_c	Spatial dimension (columns) of the hyperspectral cube
n_c	Index for columns from the hyperspectral cube
N_λ	Spectral dimension (bands) of the hyperspectral cube
n_λ	Index for bands from the hyperspectral cube
n	Generic index for the hyperspectral cube dimensions
N_h	Height (rows) of the folded matrix in Fd-PCA
n_h	Index for rows from the folded matrix in Fd-PCA
N_w	Width (columns) of the folded matrix in Fd-PCA
n_w	Index for columns from the folded matrix in Fd-PCA
N'_w	Width (columns) of the folded matrix (extended case) in Fd-PCA
\mathbf{P}	Data matrix from the hyperspectral cube
\mathbf{p}	Original pixel from the hyperspectral cube, indexed as \mathbf{p}_{n_s}
$\mathbf{P}_{n_r}^{(r)}$	Original row partition from the hyperspectral cube
$\mathbf{P}_{n_c}^{(c)}$	Original column partition from the hyperspectral cube
\mathbf{P}^{2D}	Band image from the hyperspectral cube, also $\mathbf{P}_{n_\lambda}^{(b)}$
$\bar{\mathbf{p}}$	Mean pixel from the hyperspectral cube
$\hat{\mathbf{p}}$	Reconstructed pixel in (S)AE

\mathbf{p}_g	Spectral segment of a pixel in S-SAE
$\hat{\mathbf{p}}_g$	Reconstructed spectral segment of a pixel in S-SAE
\mathbf{q}_{n_s}	Pixel transformed by PCA
$\mathbf{q}_{n_s}^t$	Pixel transformed and truncated by PCA
\mathbf{r}	Reduced internal signal in (S)AE
\mathbf{r}_g	Reduced internal signal for a segment in S-SAE
\mathbb{R}	Mathematical symbol for a real value
$round\{\}$	Rounding operator in computer science
$\mathbf{RW}_{(i,j)}$	Reallocated column vector from the window in (F)-2D-SSA
S_{test}	Number of samples or pixels in a test partition
$S_{correct}$	Number of samples correctly classified out of S_{test}
S_{ab}	Number of samples in confusion matrix by methods 'a' and 'b'
S_{ba}	Number of samples in confusion matrix by methods 'b' and 'a'
\top	Mathematical notation for transpose
T	Number of selected components in (F)-2D/1D-SSA
\mathbf{t}	Grouping set in (F)-2D/1D-SSA, indexed as \mathbf{t}_m
$trunc\{\}$	Truncation operator in computer science
\mathbf{U}	Eigenvectors matrix, where each column is an eigenvector
\mathbf{u}	Eigenvector in \mathbf{U} , indexed as \mathbf{u}_{n_λ} or \mathbf{u}_l
\mathbf{U}_t	Truncated eigenvectors matrix, where each column is an eigenvector
\mathbf{V}	Matrix of components, where each column is a component
\mathbf{v}	Component in \mathbf{V} from (F)-2D/1D-SSA, indexed as \mathbf{v}_l
$vec\{\}$	Vector operator in computer science
$\mathbf{W}_{(i,j)}$	Embedding window for (F)-2D-SSA, at position (i, j)
$\mathbf{w}_{l(i,j)}$	Row l^{th} in embedding window for (F)-2D-SSA, at position (i, j)
\mathbf{X}	Trajectory matrix in (F)-1D-SSA after embedding
\mathbf{X}_t	Grouping (or reconstructed) matrix from (F)-1D-SSA
\mathbf{X}_{rep}	Representative trajectory matrix in F-1D-SSA
\mathbf{X}^{2D}	Trajectory matrix in (F)-2D-SSA after embedding
\mathbf{X}_t^{2D}	Grouping (or reconstructed) matrix from (F)-2D-SSA
\mathbf{X}_{rep}^{2D}	Representative trajectory matrix in F-2D-SSA
\mathbf{X}_l	Component from SVD (or EVD) in (F)-2D/1D-SSA
$x_{(i,j)}$	Element in the matrix \mathbf{X}_t or \mathbf{X}_t^{2D} from (F)-2D/1D-SSA
\mathbf{Y}	Mean-adjusted data matrix from the hyperspectral cube
\mathbf{y}	Mean-adjusted pixel, indexed as \mathbf{y}_{n_s}
$\mathbf{Y}_{n_r}^{(r)}$	Mean-adjusted row partition from the hyperspectral cube
$\mathbf{Y}_{n_c}^{(c)}$	Mean-adjusted column partition from the hyperspectral cube
$\mathbf{Y}_{n_\lambda}^{(b)}$	Mean-adjusted band partition from the hyperspectral cube
\mathbf{Z}_m^{2D}	2-D signal obtained from (F)-2D-SSA

\mathbf{z}_m	1-D signal obtained from (F)-1D-SSA
\mathbb{Z}	Mathematical symbol for an integer value
\mathcal{Z}	McNemar's test of significance value

Greek Symbols

α	Threshold in the 1D-EMD configuration
β	Bias value in (S)AE, indexed as β_r or $\beta_{\hat{p}}$
η_t	Contribution percentage of a grouping t in SSA
γ	Configuration parameter in RBF kernel
Λ	Diagonal matrix from EVD
λ	Eigenvalue, indexed as λ_{n_λ} or λ_l
ω	Weight vector in (S)AE, indexed as ω_r or $\omega_{\hat{p}}$
τ	Threshold in the 2D-EMD configuration
θ_1	Threshold in the 1D-EMD configuration
θ_2	Threshold in the 1D-EMD configuration

Acronyms / Abbreviations

AA	Average Accuracy (%)
AE	AutoEncoder
AF	Adaptive Filtering
AFD	Adaptive Filtering with Derivative
AVIRIS	Airborne Visible/InfraRed Imaging Spectrometer
CbC	Class by Class accuracy (%)
DL	Deep Learning
DLR	German aerospace centre
DSP	Digital Signal Processing
EMD	Empirical Mode Decomposition
EMP	Extended Morphological Profile
ESA	European Space Agency
EVD	EigenValue Decomposition
EVG	EigenValue Grouping in (F)-2D/1D-SSA
Fd-PCA	Folded-PCA
F-SSA	Fast implementation of SSA (F-2D/1D-SSA)
HbH	Hankel-block-Hankel structure
HHT	Hilbert Huang Transform
HSI	HyperSpectral Imaging
ICA	Independent Component Analysis
IMFG	IMF Grouping in 2D/1D-EMD
IMF	Intrinsic Mode Function

JPL	Jet Propulsion Laboratory
MAC	Multiply ACcumulate
MNF	Maximum Noise Fraction
MP	Morphological Profile
NASA	National Aeronautics and Space Administration
NMF	Nonnegative Matrix Factorisation
NN	Neural Network
NoS	Number of Samples
OA	Overall Accuracy (%)
PCA	Principal Component Analysis
RBF	Radial Basis Function
RGB	Red, Green, and Blue image
ROSIS	Reflective Optics System Imaging Spectrometer
SAE	Stacked AutoEncoder
SC	Structured Covariance
SNR	Signal-to-Noise Ratio
S-PCA	Segmented-PCA
S-SAE	Segmented-SAE
SSA	Singular Spectrum Analysis
SVD	SingularValue Decomposition
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Research Motivation

In recent years, HyperSpectral Imaging (HSI) technology has become increasingly important, widely extending its applications to many different areas thanks to the spectral data acquired in hundreds of bands, which provides extremely useful information for material characterisation, recognition, classification, and many other related tasks. HSI provides spatial scenes or images in which every pixel is represented by a vector array of values acquired at different wavelengths over the spectrum. As a result, due to the information covered from a wide spectral range, HSI can detect minor difference of moisture, temperature, and chemical components changes. With such powerful capability, HSI technology is being introduced and employed in many different fields, leading to a vast number of applications areas such as food quality inspection [1, 2], medical diagnosis [3–5], pharmaceutical [6], and forensics [7, 8], among many others.

Nevertheless, remote sensing Earth observation is probably the most remarkable application that takes advantage of the spectral information provided by the HSI devices [9, 10]. In the last decades there is an increasing interest for remote sensing Earth observation, which has been translated into several missions and spacecraft launches from the National Aeronautics and Space Administration (NASA) and similar institutions such as the European Space Agency (ESA). To this end, HSI technology can provide inestimable help and solutions in the study of land images, where the high amount of spectral content available allows more appropriate characterisation of pix-

els in the image, making feasible the classification of pixels into different predefined labels according to the elements in nature they are representing.

Given the current context, several Earth observation missions from a few institutions have provided selected HSI data to be commonly used by the HSI research community worldwide. In order words, the data acquisition along general conditioning (usually associated with noise, water absorption effects, and different types of calibration) is mostly addressed in the missions, thus the research community focuses on dealing with the data in the best possible way with the available data sets provided by these remote sensing projects.

However, the high potentialities of hyperspectral remote sensing come at the cost of a proper way of Digital Signal Processing (DSP). In the first place, the spectral content available in HSI results in extremely large data, with related very expensive computational complexity. Moreover, the most challenging problem for investigation in hyperspectral remote sensing lies on how to extract appropriate features from the conditioned data and meet the need for following on data analysis and interpretation. In summary, with such HSI data, researchers are being challenged by numerous big-data projects requiring advance DSP, where several key problems can be highlighted as follows:

1. Most existing feature extraction methods involve considerable computational complexity, especially for embedded or portable devices and onboard Earth observation missions. Therefore, embedded systems along with online applications usually find remarkable restrictions in computational terms, making real-time operations infeasible, problems which seem aggravated in the case of HSI data.
2. Existing methods used for feature extraction and data reduction are found actually to provide a limited performance in terms of classification accuracy and efficiency, due to the failure in extracting the most representative information from the original data. How to improve the efficacy and efficiency needs to be addressed.
3. Most existing methods focus on performing the feature extraction in the spectral domain of the HSI data. Nevertheless, the spatial domain in HSI images

(with spatial correlation among the pixels) can also be employed alternatively to or to be combined with spectral domain extraction processing. Actually, such methodologies have been found to achieve better feature extraction and higher classification accuracy in the context of HSI.

4. Developments of Deep Learning (DL) algorithms in recent years have made this methodology a really promising tool in many fields. These methods are claimed to provide high level of abstraction in extracting features, which can lead to alternative perspectives and benefits much different to those from classical methods such as Principal Component Analysis (PCA). Nevertheless, it is found that currently few studies and research have been undertaken in DL hyperspectral remote sensing, where algorithms have not been properly assessed yet and further evaluation is desirable to fully explore its potential in HSI.

As a result, the present thesis is related to DSP and data mining for extracting features (generally referred to as feature extraction) in hyperspectral remote sensing, where the main objectives are to investigate and propose solutions in order to address the abovementioned challenging problems.

1.2 Original Contribution

In this thesis, several new methods for hyperspectral remote sensing feature extraction are proposed and evaluated. In general, these new methods aim to provide more effective features for higher classification accuracy and more efficiency in reducing the computational complexity, leading to potential advances in onboard and embedded devices. A more specific summary of these contributions can be highlighted in the following:

1. Methodology proposal for alternative computation of the covariance matrix of a hyperspectral cube, aiming to reduce the related complexity and allow onsite and even real-time covariance computation (on-the-fly feature extraction) [11] for some well-known feature extraction methods such as PCA.

2. Modification of the conventional PCA method [12], which is widely used for hyperspectral remote sensing data reduction, to improve its performance in terms of computational efficiency and data classification accuracy. With the proposed method [13] more effective feature extraction is achieved while, at the same time, the computational complexity of the algorithm is much reduced.
3. To propose using the Singular Spectrum Analysis (SSA) algorithm [14] for feature extraction in exploiting the spectral domain of HSI images. This method provides enhanced features with noise mitigated, increasing the classification accuracy, where a fast implementation of the approach is also proposed to further improve its efficiency [15, 16].
4. To propose an extended 2-D version of the SSA algorithm where both spatial- and spectral-domain information is combined for effective feature extraction, leading to much improved data classification accuracy [17]. A fast implementation of the approach is also proposed to further reduce the computational complexity of the algorithm.
5. Introductory evaluation on Stacked AutoEncoders (SAEs) inside the DL methodologies, where initial results seem promising and suggest continuing with this line of investigation.

Most contributions summarised above have been published in peer-reviewed international journals and conferences, with some of them currently under consideration, where a complete list is available in Appendix B, along with several coauthored publications.

1.3 Thesis Organisation

The remainder of the present thesis is divided into the following 7 chapters.

Chapter 2 provides an overview of the background introduction and related work in hyperspectral remote sensing, including a survey of feature extraction methods and the experimental conditions in the related research.

Chapter 3 proposes an alternative implementation for efficient computation of the covariance matrix from the hyperspectral cube. With this implementation, the whole covariance matrix can be determined by a group of partial covariance matrices, enabling fast and onsite (on-the-fly) feature extraction in accordance with the way the data is acquired.

Chapter 4 presents a variation of PCA, namely Folded-PCA (Fd-PCA). By folding the spectral vectors into matrices, the proposed approach can not only improve the efficiency but also the efficacy of data classification by extracting both global and local features from the HSI data.

Chapter 5 introduces the 1D-SSA method for feature extraction in HSI, including a complete description of the algorithm and examples of application. Then, a fast implementation is also presented, being both evaluated in comprehensive experiments. From the analysis, some recommendations are suggested in the 1D-SSA application to HSI data.

Chapter 6 evaluates the extended 2D-SSA method for feature extraction in the spatial domain of HSI, where the concept of 2D-SSA is described in detail along with an application example. In addition, a fast implementation is also proposed in a way similar to the 1-D case. The 2D-SSA method is benchmarked against several state-of-the-art techniques, especially with the 2-D Empirical Mode Decomposition (2D-EMD) method [18]. From the analysis, it is suggested that 2D-SSA provides similar efficacy as 2D-EMD but with much reduced complexity.

Chapter 7 briefly describes SAEs within a DL framework for feature extraction and data reduction. With a modified implementation based on spectral segmentation, both the efficiency and efficacy of feature extraction have been improved.

Finally, Chapter 8 summarises the contributions of the thesis and also discusses the future work as how to further improve the proposed approaches and beyond.

Chapter 2

Background and Related Work

2.1 Introduction

According to the motivation of the present thesis, background and related work in remote sensing Earth observation is introduced in this chapter. The hyperspectral remote sensing concept is described in Section 2.2, where key applications and the demand in feature extraction and data reduction are highlighted. A comprehensive description of the related experimental framework is provided in Section 2.3, including workflow of the signal processing, experiments design, and performance assessment. Section 2.4 surveys feature extraction methods in HyperSpectral Imaging (HSI), discussing the existing work. Finally, a brief summary is given in Section 2.5.

2.2 Hyperspectral Remote Sensing Earth Observation

The concept of remote sensing can be defined as the acquisition and study of information from an object of interest in the distance, without any physical contact or manipulation. More particularly, remote sensing refers to the instrument-based technology and application area for the detection, classification, and recognition of objects in Earth [19, 20], other planets, and celestial bodies including Mars [21] or the Moon [22]. This field has attracted increasing interest since Landsat-1 [23], the first satellite for Earth observation, was launched by the National Aeronautics and Space Administration (NASA) on 23rd July 1972. Nowadays more active exploration can be found, such as the Sentinel missions [24] from the European Space Agency (ESA).

2.2.1 The Data Sources

The remote sensing data source is the electromagnetic radiation reflected from the objects, captured in some conditions by an acquisition device or sensor. Since every object possesses unique characteristics, they can be determined by the reflected radiation, leading to discriminative information allowing accurate classification or recognition of different objects. This information can be provided in different data types including optical or radar images, while the sensors are carried by particular vehicles known as platforms, normally aircrafts or satellites in orbit for visual exploration (see Figure 2.1), providing land locations imagery. These images are subject to diverse analysis, being really useful for applications such as land use, urban distribution, or agriculture monitoring, among others.

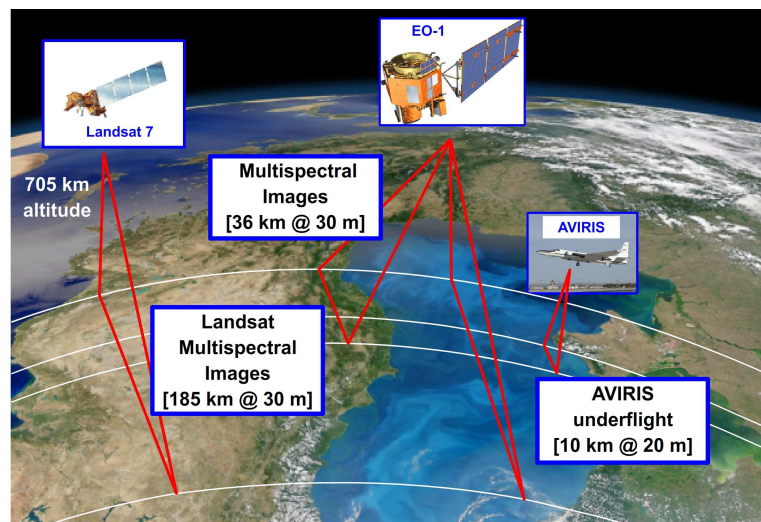


Fig. 2.1 Examples of remote sensing by different satellites and aircrafts

Advances in technology during recent decades have led to numerous devices and optical cameras ideal for remote sensing, taking into account technical aspects such as mechanical considerations, embedding structures for spacecrafts, engineering life, and atmospheric or geometric calibration. In fact, as long as more information is provided, higher potential is expected with relation to potential analysis, and that is the reason why HSI technology is introduced to remote sensing applications.

In contrast to usual Red, Green, and Blue (RGB) images, as an evolution from multispectral imaging, the HSI technology is able to capture high resolution data in the

spectral domain, with hundreds of wavelengths covering from visible to (near) infrared spectrum (see Figure 2.2). Unlike conventional RGB images, the wide spectral content in HSI is able to identify and characterise minute differences (unnoticeable to the naked eye) in terms of temperature, chemical composition, or moisture, among others.

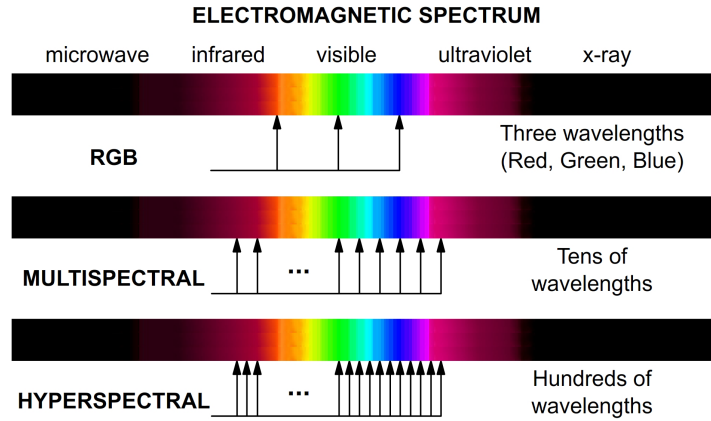


Fig. 2.2 From RGB to hyperspectral

From such amount of captured information, HSI technology leads to really powerful capabilities related to classification and recognition, being widely used in remote sensing and other conventional applications such as agriculture, geology, military surveillance, and astronomy [9, 10]. Thanks to the numerous and contiguous spectral bands, this technology additionally provides a unique and inestimable solution for an increasing number of application areas, becoming a research hotspot in recent years, as it has been successfully applied to emerging lab-based tasks [1–8, 25].

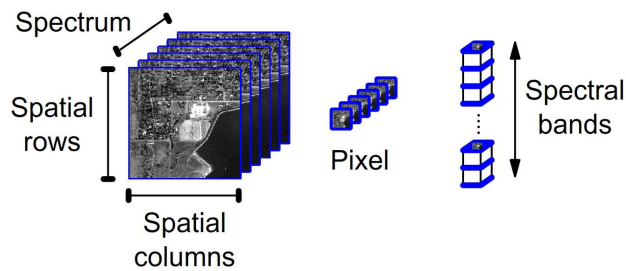


Fig. 2.3 Hyperspectral cube in remote sensing applications

HSI data is captured in a 3-D structure namely hyperspectral cube (Figure 2.3). With both spatial and spectral information being simultaneously acquired, the potential in data analysis and mining is greatly enhanced. In remote sensing Earth obser-

vation tasks, the pixels or samples from the cube are subject to study, where they are represented by a vector array containing hundreds of elements, i.e., the spectral bands or features forming each pixel. Therefore, with narrow bands in nm for the spectral sampling, hyperspectral remote sensing enables high discrimination ability among the pixels for data classification tasks such as land-cover analysis. However, this potential comes at the cost of extremely large data sets and expensive computation, leading to a main drawback of HSI, the high computational complexity.

2.2.2 Processing the Hyperspectral Cube

Data captured by HSI acquisition devices is usually subject to conditioning and other similar processes, including different types of calibration and noise removal. However, the conditioned data is still not ready for optimised analytical tasks, especially in the HSI case, where such large data makes difficult its comprehension and requires appropriate mining. Additionally, the original features or spectral values may not lead to a direct benefit from their use, because such large dimensionality of features usually masks valuable information contained in the hyperspectral cube.

Furthermore, when the number of pixels used in training classifier models is relatively reduced with relation to the hundreds of spectral features, the curse of dimensionality, also known as the Hughes effect [26], appears in HSI, especially in multi-class problems as the reduced amount of pixels in every class is not able to provide enough knowledge to the models for coping with so many features. Although this fact is somehow addressed by powerful classifiers such as Support Vector Machine (SVM) [27], adequate signal processing is required. Bearing in mind the considerable redundancy existing between neighbouring spectral bands in hyperspectral cubes, feature extraction and data reduction methods are feasible and usually crucial for achieving appropriate performance in the evaluation of HSI data.

The specific signal processing stage by which the main information from an original or conditioned data set can be extracted is generally known as feature extraction. This mining procedure picks up the potentialities contained in the data (see Figure 2.4), enhancing the significant knowledge while discarding the useless or even confusing misinformation. This processing and mining leads to resulting features representing

the data in a much more appropriate way for subsequent analysis. Therefore, better features allow the achievement of higher accuracy values in classification tasks.

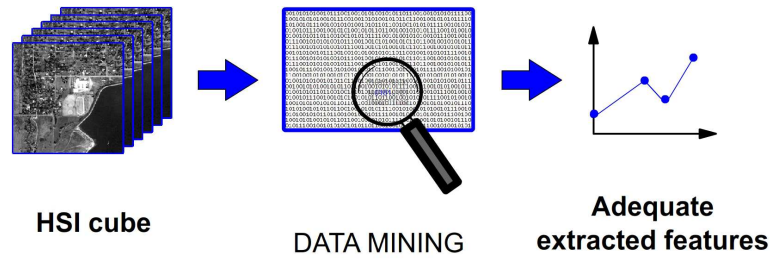


Fig. 2.4 Feature extraction and data mining for HSI

An adequate processing of the HSI cube in remote sensing applications has led to several feature extraction and data reduction methods available in the literature. These methods can be divided into groups according to diverse criteria [28]. A classification in terms of the exploited domain from the HSI cubes can be derived (see Figure 2.5).

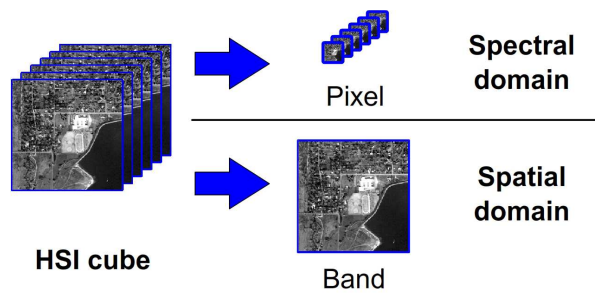


Fig. 2.5 Spectral and spatial domains from hyperspectral cubes

Bearing in mind that every pixel in HSI naturally forms a spectral profile or vector, the early literature has focused on the spectral content of pixels [29], especially because HSI presents high correlation between contiguous spectral bands, making feasible the reduction in dimensionality of features (data reduction). Hence, feature extraction in the spectral domain is emphasized regarding not only more effective features but also reduction in computational complexity and other considerations such as the curse of dimensionality problem [26]. In contrast, spatial-domain methods go further and take advantage of the spatial correlation in HSI images, reporting better results in recent years [18] whether combined or not with spectral extraction. Both cases are evaluated in the present thesis.

2.3 Related Data Processing Framework

For different HSI data sets, a common data processing framework is widely adopted by most researchers. This is discussed in detail in the following section to show how the experimental conditions are set for consistent comparisons.

2.3.1 Signal Processing Chain

In order to provide a better understanding of the hyperspectral remote sensing field, a description of the full processing chain can be summarised into 4 consecutive stages, i.e., data acquisition, data conditioning, feature extraction, and data classification, as shown in Figure 2.6. These are further discussed in detail below.

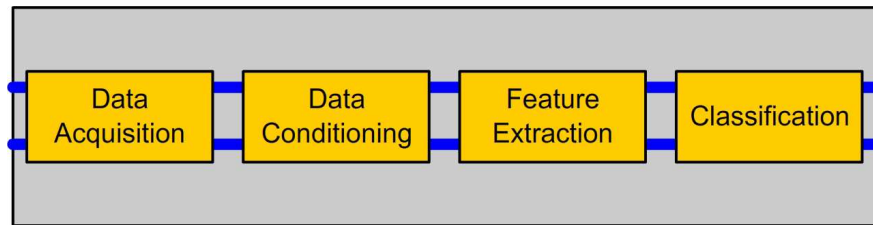


Fig. 2.6 Classical signal processing chain

A. Data Acquisition

This stage involves capturing the data to be processed in subsequent analytical tasks, which is not covered in this thesis as the existing data sets are used for analysis. As a result, the concept of data acquisition is replaced by data description, where comprehensive information is given for each data set employed in the experiments.

There are few data sets available for worldwide research, especially with corresponding ground truth. This is partly because labelling or assigning an adequate class to each pixel is extremely demanding, requiring manual and tedious procedures. Nevertheless, this fact ensures that common shared data is used in the vast majority of publications available in the literature, leading thus to appropriate comparisons.

A well-known acquisition instrument in hyperspectral remote sensing is the Airborne Visible/InfraRed Imaging Spectrometer (AVIRIS) [30] (see Figure 2.7), developed by the Jet Propulsion Laboratory (JPL) [31] at NASA, providing the first spectral

images around 1986 and becoming a hit in Earth remote sensing applications. It has provided the most widely used data sets in this field, with over 500 flights and more than 600 citations in peer-reviewed publications.



Fig. 2.7 AVIRIS acquisition instrument (image credit: JPL-NASA)

Onboard an aircraft at 20 km of altitude and about 700 km/h of velocity, AVIRIS is able to measure solar reflected spectrum from 400 nm to 2500 nm [30], capturing radiance in 224 contiguous spectral bands at approximately 10 nm intervals, with a spatial resolution of 20 m. Spectral, spatial, and radiometric calibrations are undertaken in laboratory and monitored inflight each year [30].

Another important acquisition instrument is the Reflective Optics System Imaging Spectrometer (ROSIS) [32]. ROSIS is a sensor developed in the German aerospace centre (DLR) being based on a design for a flight on an ESA platform [33]. The initial purpose of the ROSIS design was detecting spectral fine structures in coastal and inland waters [33], however, further application above land has been derived. This instrument was first tested in 1992 and operated in 1994 on the Falcon Jet from DLR, following redesign processes.

Based on a 2-D charge couple device, this acquisition sensor is able to provide spectral information in the range 430 nm - 860 nm, with a resolution of 4 nm, resulting in a total of 115 spectral channels [32], including related radiometric, spectral, and geometric corrections. Further information can be found in [32, 33].

Finally, being part of the NASA Earth observation EO-1 mission, and launched on 21st November 2000, the HYPERION sensor was the first spectrometer acquiring science-grade data from Earth orbit [34], providing spectral information in the range

400 nm - 2500 nm, with resolution of 10 nm, and more than 200 spectral bands (up to 242 bands available). Although onboard the EO-1 spacecraft (see Figure 2.8) at 700 km of altitude, a spatial resolution of 30 m is provided.

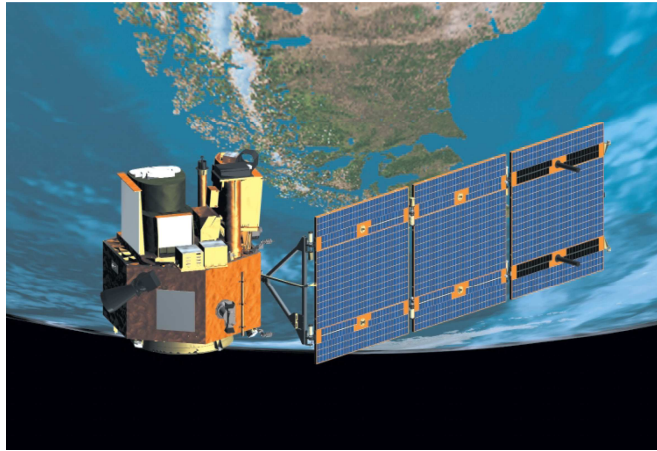


Fig. 2.8 HYPERION onboard the EO-1 satellite (image credit: NASA)

Similar to the other sensors, correction procedures are again involved in the acquisition stage, but including further considerations such as the prelaunch and in orbit radiometric calibration.

Several data sets from the abovementioned sensors are used in this thesis. The 92AV3C (Indian Pines), Indian Pines B, and Salinas C images are obtained from the AVIRIS instrument. On the other hand, the ROSIS sensor has provided the Pavia University A (Pavia UA) and Pavia Centre A (Pavia CA) scenes. Finally, the Botswana A data set from the HYPERION sensor is also used in the experiments. All the data sets used in this thesis are described in Appendix A.

B. Data Conditioning

The conditioning is conceived as a preprocessing stage applied after the data acquisition but prior to the feature extraction stage. With calibration and other procedures already undertaken, the conditioning at research level only involves few considerations, normally similar among most publications in the literature.

Data conditioning simply consists of removing some noisy spectral bands from the given hyperspectral cubes. The bands to be removed in each data set are selected according to state-of-the-art publications such as [18, 35] and general suggestions [36].

This proves the conditioning to be adequate in general terms and, moreover, consistency is achieved among the different publications, leading to fair comparisons to the current literature. Additionally, few particular labelled classes in the ground truth of some data sets are not included for classification in some experiments. The reason behind this fact is that some classes present a quite reduced Number of Samples (NoS) or pixels, so avoiding them is claimed to provide better statistical significance [37].

Information about the conditioning employed for each data set is available in Appendix A, where particular conditioning carried out for the experiments is described in the corresponding chapters.

C. Feature Extraction

Once the data is acquired and conditioned, the next stage in the signal processing workflow is the feature extraction. The concept of feature extraction, as already introduced in Section 2.2.2, is based on the idea of mining the main useful information from a given data set. Therefore, this stage is where the current literature focuses.

Initially, some discussion can be found with relation to the concept and difference between feature selection and feature extraction. Feature selection may be referred to discarding some of the original features, while the extraction concept seems related to some transformation procedure, such as projection or decomposition. Additionally, feature selection would imply a reduction in the dimensionality of features not always obtained from the extraction methods. The general concept of feature extraction is employed in the present dissertation.

On the other hand, methods can also be grouped into supervised, semisupervised, and unsupervised cases. Supervised (and semisupervised) methods require prior knowledge of the labelled class assigned to the different pixels. Yet this fact may lead to better features as these can be computed taking into account the class information from pixels, it requires prior knowledge not accessible or available most of the times. Additionally, even though the supervised case is expected to perform better, the general improvement is actually limited, and the unsupervised case results straight and easier to implement. That is the reason why unsupervised feature extraction methods are much more preferable and, accordingly, the present research is focused on them.

Different methods for feature extraction and data reduction are compared in the experiments under the same conditions, in order to evaluate their behaviour and performance on a consistent and comparable basis, according to some criteria (see Section 2.3.3). The contribution methods from the present research are evaluated and compared to some state-of-the-art methods from the literature, normally including the use of the original spectral features (Baseline case) as an initial reference for benchmarking.

D. Classification

Finally, in the last stage, testing pixels are allocated into several labelled classes by a supervised classification model, where the assigned labels are then compared to the real ones available in the ground truth of the data, resulting in a classification accuracy to be used for performance assessment. As the experiments are carried out under the same conditions and classifiers, the difference in classification accuracy is attributed to the features employed, i.e., the efficacy of the extracted features from the corresponding approaches.

The current literature suggests the use of machine-learning classifiers such as SVM [37]. Exploiting a margin-based criterion, SVM seems to be very robust to potential problems such as the Hughes effect [26]. With results proved to be better than other classifiers, SVM has been widely used in the HSI area [18, 29, 35, 37]. Furthermore, there are several accessible libraries supporting multiple functions of SVM, making its implementation easy even for embedded systems [38, 39]. Therefore, being considered the most powerful classifier in the HSI context, SVM is selected as the most appropriate classifier, where considerations for its implementation are available in the next section.

2.3.2 Design of Experiments

The design of experiments is straightforward (see Table 2.1). Some publicly available data sets with the mentioned conditioning are selected, covering the first 2 stages in the processing chain. Feature extraction is implemented by those methods to be evaluated, hence actually it is the classification stage that needs to be particularly designed, in considering training/testing partitions and configuration of the relevant classifiers, as described in the following.

Table 2.1 Workflow in the experiments

Data Description	Data sets employed
Data Conditioning	Spectral bands removed and labelled classes obviated if any
Feature Extraction	Methods to be evaluated
Classification	Training percentage for building the models

A. Training and Testing

Although all the pixels in a given hyperspectral cube can be included in the unsupervised feature extraction and data reduction, only the labelled ones are employed in training the classifier model and subsequent testing analysis, since the corresponding labels are required during the supervised classification process.

The labelled pixels are therefore divided into 2 groups, training and testing partitions, where no sample overlap is allowed, i.e., no pixel can be used in both groups at the same time. On one hand, the training partition is used to train the classifier, so a model is eventually achieved, appropriately constructed by dividing at the same time the training partition for validation purposes. This model is able to classify an inputted pixel into one of the predefined labelled classes. On the other hand, all pixels inside the testing partition are individually classified by the previously built model. Since the label of each testing pixel is known, it is possible to compare the real label to the predicted one by the classifier, resulting into an accuracy of classification (%) [20] by which the performance can be easily assessed.

The most common procedure to divide labelled pixels into the training and testing partitions is a stratified random sampling [20, 40] by some percentage (%). This procedure evidently produces a random selection of samples for a fair analysis, while at the same time it ensures an equal sample ratio among the labelled classes, i.e., the training and testing percentages are the same in each individual class. Obviously, the same training and testing partitions are employed to evaluate all the feature extraction methods for fair comparisons. Several different percentages can be employed, hence, the training percentages usually vary from 5% [41] to 35% [18] (with remaining 95% and 65% for testing, respectively).

The experiments are normally repeated several times (10 repetitions seems quite sensible) under the same conditions but employing a different training/testing partition. Therefore, several different stratified random samplings [20] provide different training partitions along with their corresponding testing pixels. This particular consideration makes the experiments time-consuming and resource-demanding, but it avoids systematic errors and provides adequate statistical significance [42], reporting the mean values from the repetitions.

B. SVM Implementation

The powerful SVM algorithm is implemented as the classifier by most researchers. Although initially being a binary classifier, the SVM can perform multiclass classification by means of approaches such as one-against-one or one-against-all [43]. To this end, there are several important libraries allowing an easy use of the SVM in multiclass tasks, where BSVM [44] and especially LIBSVM [45] are widely used in the current literature.

The SVM classifier has to be properly implemented in the experiments. In fact, different configurations can lead to variable performances [37], being a critical step to be taken into account. In the first place, the kernel type must be selected. Although there are a few kernel functions such as linear, polynomial, or sigmoid, the Radial Basis Function (RBF) Gaussian kernel is normally selected in many experiments based on previous experience and supported by main researchers and publications [18, 29, 35, 37], with some comparisons suggesting that the RBF Gaussian kernel generally performs better in hyperspectral remote sensing.

Following the RBF kernel selection, 2 key variables, the gamma (γ) and the penalty (cost) parameters need to be properly tuned. To this end, a grid search procedure [45, 46] is applied every time, where several values are assigned to these 2 parameters and the combination providing highest validation accuracy is selected to build the SVM model. Validation in most of the experiments is carried out by a 5-fold cross-validation on the training partition [46] (also 2-fold and 10-fold are common cases), with wide ranges comprising about 0.001-100 and 0.1-1000 for the γ and the cost parameters, respectively, to ensure an adequate (close to optimum) configuration.

Finally, other important considerations include the normalisation or scaling procedure applied to the training and testing partitions as required by a proper performance of SVM [46] [47], and the use of a MEX library [45] available for LIBSVM, allowing the SVM training and testing from well-known environments such as MATLAB [48].

2.3.3 Performance Assessment

The outcome obtained from the experiments has to be evaluated. This evaluation aims to differentiate among the feature extraction methods studied, where the assessment procedure is done on a comparative basis. Some commonly used metrics and criteria are discussed as follows.

A. Classification Accuracy

In land-cover analysis, with multiple classes defined in the ground truth, several classification accuracy values are of interest [20, 49]. Probably the most important measurement is the Overall Accuracy (OA), obtained by the accumulation of correctly classified pixels with relation to all those tested, i.e., $OA = S_{correct} / S_{test}$, where S_{test} is the total number of pixels in the testing partition, while $S_{correct}$ is the number of pixels correctly classified out of S_{test} . Additionally, it is also possible to obtain the particular accuracy for a given class in the data set, leading to the Class by Class (CbC) accuracy, where $CbC = (S_{correct} / S_{test})|_{class}$. These values are useful when some particular classes are of special importance in the analysis, or when comparing the performance of a given method among different classes. Finally, by simply averaging the CbC values from all the classes in the data, the Average Accuracy (AA) is achieved, thus $AA = mean\{CbC\}$. This is a good measurement when global improvements are expected regardless of the labelled classes or the NoS (pixels) available in each of them.

It is important to remark that there are other similar measurements such as the reliability or user's accuracy [20], where the accuracy or efficacy is measured from a slightly different point of view. Yet the reliability measurement can be interesting, OA, CbC, and AA values are much preferred and employed in the current literature, probably because both indicators are correlated, i.e., higher accuracy generally leads to higher reliability and vice versa.

B. Statistical Significance

Apart from the classification accuracy values, complementary indicators providing statistical significance are commonly employed in land-cover analysis, where the most used is the McNemar's test of significance [50]. This test provides statistical significance at a confidence level of 95% when $|\mathcal{Z}| > 1.96$, being \mathcal{Z} positive when the technique evaluated beats a given reference method (usually the Baseline case), and negative in the opposite case. This value is simply defined as $\mathcal{Z} = (S_{ab} - S_{ba}) / \sqrt{S_{ab} + S_{ba}}$, where S_{ab} represents the number of pixels that have been correctly classified by method 'a' but misclassified by method 'b' (reference), being S_{ba} the opposite.

There are some different indicators such as the kappa coefficient [20, 49, 50], which has also achieved considerable diffusion in many publications, yet some recent discussions [51] have suggested that the kappa coefficient may be inadequate in the remote sensing context, even leading to flawed results, while some other measurements such as the quantity disagreement and the allocation disagreement components [51] are recently attracting attention of researchers [52, 53].

C. Computational Efficiency

The performance assessment on feature extraction methods consists of not only efficacy but also complementary considerations related to efficiency. In other words, a given method can be better not because it leads to higher classification accuracy but because, achieving similar accuracy values, the computational cost required in the implementation is somehow reduced.

The computational cost derived from the algorithms, normally expressed in Multiply ACcumulates (MACs), gives a straight reference of the methods complexity, especially with regard to its implementation in digital devices. Therefore, the number of MACs (alternatively number of multiplications and additions) is normally employed as efficiency measurement [54].

The computation time is also a good parameter describing the performance of the feature extraction methods [18]. It is normally associated with the computational complexity, as those methods based in simpler algorithms are usually easier to implement and faster to compute.

Finally, the memory requirement, normally expressed in kB, is an additional parameter to be taken into account for efficiency purposes, especially given the limited memory capabilities in portable and onboard devices. Additionally, the number of features (dimensionality of features) can also be an indirect measurement of computational complexity, where a larger number of features obviously involve more memory requirement and complexity.

2.4 A Review of Hyperspectral Feature Extraction

Advanced data analysis and mining is crucial for appropriate classification and related tasks in numerous scientific and engineering applications. After data acquisition and conditioning, feature extraction becomes vital to extract the useful information from the hyperspectral cube.

Due to the hyperspectral cube nature in HSI, the methods exploiting information only from the spectral domain have been much widely used. Spatial-domain information is still useful in feature extraction, however, it has been seldom addressed and few methods have considered it, where a joint combination of spectral-spatial-domain information is commonly preferred. A survey of feature extraction in hyperspectral remote sensing is carried out in the following, considering both spectral- and spatial-domain methods as detailed below.

2.4.1 Spectral-Domain Methods

Initially, the introduction of HSI data has been associated with the use of original features or spectral profiles from the cubes as a direct input to the classifier [37]. This high dimensionality of features is considered powerful as it provides comprehensive information along the spectral domain, leading to potentially good characterisation. However, the large number of features in combination with the usually limited number of samples available in supervised classification training can result in the curse of dimensionality or Hughes effect [26], where a high number of features requires a vast number of samples to perform properly, which normally is not the case, and can lead to insufficient training of the classification models.

This important drawback can be solved by a simply reduction of features, where the covariance matrix is widely employed in performing data reduction and projecting samples into a lower dimensional feature space. To this end, several methods such as Principal Component Analysis (PCA) [12, 55] consider the covariance matrix computation in their algorithms. Nevertheless, in the hyperspectral remote sensing context, HSI data presents high size not only in spectral but also in spatial terms. Therefore, while the number of features can be in hundreds, the number of samples (pixels) can be above thousands, leading to complexity in computing the covariance matrix from HSI images, introducing difficulties in the case of portable devices and embedded systems, with considerable restrictions in terms of power and/or memory.

Some algorithms and mathematical tricks have been proposed in order to relieve the covariance computation in HSI [56, 57], based on the employment of smaller partitions of data to construct the covariance matrix, even in parallel threads. However, these algorithms have not been associated with the actual acquisition procedure given in HSI instruments, where the sensors capture smaller partitions of data in a sequential manner [58, 59]. Therefore, the time between capture steps available in the sensors can be potentially used for efficiency. Yet the computation of the covariance matrix is well-known and the related concepts are not new, there is still a gap in applying these theories to HSI applications, where the tricks can be better applied to the different acquisition procedures for onboard and real-time feature extraction.

Despite the abovementioned difficulties, PCA has probably been the most widely used method for feature extraction and data reduction in hyperspectral remote sensing [12]. When applying PCA, the original data is orthogonally projected by eigenvectors obtained through EigenValue Decomposition (EVD) applied to the covariance matrix. The resulting features are sorted by the represented amount of variance, therefore, the dimensionality of features can be truncated to the first few features, since they already contain most of the variance from the original data, usually around 90-99%.

Based on PCA, several modified methods have been proposed in the literature, where the conventional PCA implementation is changed in some degree to achieve particular benefits dependent on each case. It is possible to find diverse variants, from simple modifications of the algorithm to more complex alterations, including

thus the modified-PCA [60], incremental-PCA [61], or the interesting Segmented-PCA (S-PCA) [62]. In S-PCA, the conventional algorithm is independently applied into different segments of the spectrum, leading to the extraction of local information and becoming a reference method. Some more variants are found in recent years, such as the 2-D PCA (2D-PCA) introduced to the field of face recognition [63], which achieved great popularity and several variations from this particular method, although it is found to be related to the block-based-PCA [64]. Other remarkable works are related to the kernel-PCA [65] and the nonlinear-PCA [66] methods.

However, conventional PCA presents a series of drawbacks not only related to computational complexity, which can be expensive for HSI data, but also in terms of inefficacy in extracting the adequate information from the large spectral pixels. Most of the variants present the same problem inside the HSI context, and yet more alternative implementations have been proposed again for efficiency [67, 68], the tradeoff problem of extracting adequate features with reduced complexity remains unsolved.

Leaving aside the PCA method, there are several more feature extraction and data reduction techniques in the literature that have also become current state of the art. These include some well-known approaches such as the Independent Component Analysis (ICA) [69, 70], the Maximum Noise Fraction (MNF) [71, 72], and the Nonnegative Matrix Factorisation (NMF) [73] methods, yet no considerable improvement has been reported with relation to the classical PCA.

The ICA method [69] considers that several signals, denoted as independent components, are mixed in the related samples or pixels. Therefore, ICA separates sources assuming that the components are non-Gaussian signals statistically independent from each other, which is not necessarily the case in HSI. A different situation is found for MNF [71, 72]. This method, also known as Minimum Noise Fraction or noise-adjusted-PCA, transforms and sorts the features in terms of noise content, so the noisiest variables are truncated for noise removal. Finally, the NMF [73] technique imposes restrictions on positive values during a factorisation procedure, which is claimed to improve the final features. The performance of these methods seems limited [41] and inconsistent due to some causes, including the prior noise estimation required in MNF, and the initial values for iterations in both ICA and NMF, reducing their reliability.

Some other feature extraction (also feature selection) and data reduction methods with relative recognition in the research community include projection pursuit [74], band selection [75], steepest ascent [37], and methods based on machine learning [35] and sparse representation of data [76, 77], highlighting nonlinear methods such as kernel-ICA [78, 79], ISOMAP approaches in [80, 81], and locally linear embedding [82, 83]. Even though the present thesis is focused on unsupervised methodologies, it is also possible to remark some important and well-known supervised methods including the Bhattacharyya and Jeffries-Matusita distances, the well-known spectral angle mapper, mutual information, the vegetation index or NVDI, linear discriminant analysis [84], decision boundary feature extraction [85], and nonparametric weighted feature extraction [86].

However, these methods present diverse and several drawbacks, from complexity in the implementation or unclear methodology, to small improvements in the performance. In some cases, the algorithms are quite simple but the efficacy of the features is not exactly brilliant. In other cases, the methods are remarkably effective but at the cost of a complex implementation or particular restrictions and limitations. That is probably the reason why methods such as PCA remain being widely used nowadays. More information about the main or classical methods can be found in the comparisons undertaken in [41, 42, 87]. Additionally, a great compilation work on data mining, feature extraction, and data reduction is available in [28], where the methods are divided into several groups for easy understanding.

From the wide methodology found in the literature, the decomposition-based approaches are of high interest and inspiration to the present research. As HSI data is prone to noise, it is interesting the idea of decomposing the spectral pixels in a way that the noise can be removed. Therefore, decomposition methods such the discrete wavelet transform [88], the discrete cosine transform [89], and especially the Empirical Mode Decomposition (EMD) [90] are found inspiring, presenting ideas of interest. The basic EMD, being part of the Hilbert Huang Transform (HHT), is used in nonlinear and nonstationary time series analysis [90, 91], where it decomposes a signal into several few components or subseries namely Intrinsic Mode Functions (IMFs), and has some reported applications including speech recognition [92].

The reported results using 1D-EMD in hyperspectral remote sensing [18] are not good. However, a similar implementation can be applied by the Singular Spectrum Analysis (SSA) algorithm, a recent technique for time series analysis and forecasting [14, 93]. Being based on the well-known Singular Value Decomposition (SVD), the 1D-SSA method is expected to provide appropriate decompositions with enhanced features and reduced complexity, as those methods such as 1D-EMD are really expensive, requiring empirical iterations. The SSA algorithm has been already introduced to hyperspectral remote sensing, yet the evaluation in [94] is insufficient and, moreover, no considerations are derived with relation to the computational complexity.

Finally, Deep Learning (DL) methodology has been recently proposed in hyperspectral remote sensing, not only for classification including artificial Neural Networks (NNs) [95, 96], but also for data reduction, highlighting the introduction of Stacked AutoEncoders (SAEs), where high levels of abstraction are proved to achieve appropriate features [97], with similar works in [98, 99]. Other DL approaches being recently evaluated are the deep belief networks [100] and the restricted Boltzmann machine [101].

In this case, yet the ideas are inspiring, the computational cost usually involved in DL methods is prohibitive, and severe criticism [102] complains about the lack of theoretical justifications, since most DL methods are usually seen as a black box. However, there is still a promising potential in the DL context, where further research and studies are needed to properly assess this brand new topic.

2.4.2 Joint Spatial-Spectral-Domain Methods

After the comprehensive spectral feature extraction review, and bearing in mind the 3-D cube structure, it is clear that although the high resolution is only found in the 1-D spectral domain, there is still a 2-D spatial domain that can be employed for an effective feature extraction. Indeed, the limited contribution of spectral-domain feature extraction has been remarked in recent years, especially with relation to the potential achieved by including spatial information, where the spatial domain can also provide useful clues. Yet there are some approaches working only in the spatial domain, most spatial-domain methods actually involve both spectral- and spatial-domain extraction.

Initially, the spatial postprocessing after undertaking spectral-domain classification is found of high interest. This option is explored in [42], where morphological features are combined with the MNF method, the SVM classifier, and a fuzzy-NN-based spatial postprocessing. Nevertheless, this evaluation shows limited contributions, suggesting that the exploitation of the spatial domain should be more favorable when introduced before the classification stage, simply because the models can be more effectively trained.

More comprehensive analysis is found with relation to mathematical morphology processing, with opening and closing operators, and the well-known median filtering. Therefore, current state-of-the-art methods include [103], where a morphological processing is applied to a particular PCA component from the HSI data. This method, called Morphological Profile (MP), has attracted attention in hyperspectral remote sensing, where in [104], the MP method is extended to include several few PCA components, resulting in the Extended MP (EMP) method. Furthermore, this EMP approach is later evaluated with SVM as a classifier [105], with excellent results. However, this methodology has the morphology operators as a main actors, which means that the EMP performance is highly dependent on the shape and size of the structuring element associated with the operators. In addition, there are some unresolved questions, such as the number of PCA components to use and the optimum number of opening and closing operations to apply.

Another state-of-the-art approach, the Adaptive Filter (AF) method in [106], implements a spatial median filter in combination with the MNF method applied in the spectral domain. The kernel size in the median filter is variable, implemented according to the Signal-to-Noise Ratios (SNRs) obtained from the MNF method. Therefore, those transformed bands in which the SNR is adequate are treated by a small kernel size, while the noisy bands are subject to a higher size in order to mitigate the presence of noise. This work includes a couple of variants for the AF implementation, where the AF with Derivative (AFD) seems to achieve slightly higher classification accuracy. Although the performance reported is considerable, again the effect of the kernel size in the median filtering is not clear and probably inconsistent, highly dependent on neighbourhood pixels.

There are also some methods considering exclusively the spatial-domain extraction. For instance, the unsharp filtering [107] has been implemented to enhance high-frequency content in a 2-D image, boosting the edges of the various elements in the scene. In [108, 109], wavelet decomposition of 2-D images is employed for a spatial adaptive denoising, resulting in higher HSI classification accuracy. However, the improvement from these methods seems quite limited.

More recently, in [18], the 2-D extension of the previously cited EMD, the 2D-EMD method, is introduced for spatial-domain feature extraction in HSI cubes. This extended 2-D version can be applied to images for extracting edges and local structures [18, 110, 111]. Therefore, in the HSI context, the application of 2D-EMD to every spectral scene leads to a potential decomposition and subsequent reconstruction using the low-order IMFs and discarding the rest, as the spatial structure is captured by these low-order components while higher-order ones lack this content [18].

The 2D-EMD method has been proved to beat the current state-of-the-art techniques [18], with extremely high classification accuracy reported and international recognition. On the other hand, however, the 2D-EMD implementation is based on empirical iterations, involving extremely high computational complexity and endless computation times, which makes it unfeasible in some situations. The same authors in [18] suggest a simpler implementation in order to overcome this problem, nevertheless, the cheaper implementation results on degraded accuracy values.

For that reason, similarly to the 1-D case in the spectral-domain methodology, the extended 2D-SSA algorithm [112, 113], already introduced to some other fields [114, 115], can be evaluated in hyperspectral remote sensing for effective and efficient feature extraction in the spatial domain.

2.5 Summary

This chapter comprises the background and related work on hyperspectral remote sensing Earth observation, providing a survey of the topic. Firstly, the concepts of remote sensing and hyperspectral technology are described, where 3-D hyperspectral cubes are the key elements to study.

Secondly, the experimental framework in hyperspectral remote sensing is introduced, presenting a classical signal processing chain with 4 stages, i.e., data acquisition, data conditioning, feature extraction, and data classification. The data acquisition and conditioning is normally addressed by the remote sensing institutions and laboratories (NASA, JPL, DLR, or ESA). Therefore, the research community focuses on the feature extraction stage, where a posterior classification by SVM is usually employed for land-cover analysis. To this end, conditioned data is split into the training and testing partitions, where the performance can be assessed in terms of classification accuracy, statistical metrics, and computational complexity evaluations.

Finally, a review of feature extraction methods is carried out, where they are divided into 2 main groups, i.e., spectral-domain and joint spatial-spectral-domain methods. These have motivated the work as presented in this thesis, with contributions reported in detail in the next 5 chapters.

Chapter 3

Structured Covariance (SC) for on-the-Fly Feature Extraction

Numerous feature extraction methods such as Principal Component Analysis (PCA) [12] usually require the computation of a covariance matrix that contains a valuable amount of information from a given set of samples. However, obtaining the covariance matrix from data such as HyperSpectral Imaging (HSI) can lead to computational problems due to the large size of hyperspectral cubes.

Taking advantage of how the spectral data acquisition is carried out in the HSI devices, various Structured Covariance (SC) schemes are proposed, where the covariance matrix is obtained by computing smaller partitions of the cube. As such schemes have a low memory requirement, they can be potentially applied for onsite feature extraction while the data is acquired in a real-time manner, benefiting subsequent tasks in coding, compression, or transmission of hyperspectral data.

This chapter is organised as follows. Section 3.1 and Section 3.2 introduce the HSI data acquisition procedures and the conventional way of covariance matrix computation. In Section 3.3, the proposed SC is explained, Section 3.4 discusses detailed experiments and analysis, followed by the summary of the contributions in Section 3.5. The major contents of this chapter have been published in the OSA Applied Optics journal, being this paper referenced as [11].

3.1 Introduction

In HSI, the acquired data forms a 3-D structure known as hyperspectral cube, with both spatial and spectral domains of information. This data structure, shown in Figure 3.1, can be indexed as $N_r \times N_c \times N_\lambda$, where the 3 symbols represent dimensions of row, column, and spectral content of the cube, respectively.

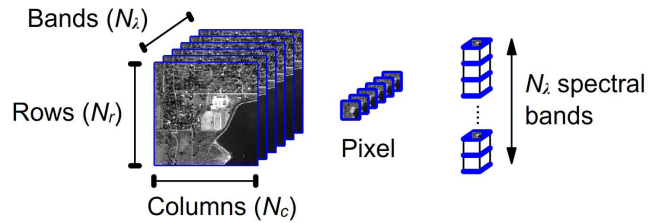


Fig. 3.1 3-D hyperspectral cube acquired by HSI devices

Nevertheless, processing this high amount of data contained in hyperspectral cubes is actually a demanding task. As the spectral dimension can easily exceed hundreds of bands, the data processing is similar to video analytics, which are generally time-consuming and memory-intensive tasks. That is the reason why feature extraction and data reduction methods are usually applied to hyperspectral cubes.

PCA [12, 55, 116] is probably one of the most widely used algorithms for HSI data reduction, being a state-of-the-art method in this context. The PCA technique, as well as other different methods, has to deal with the original cube in order to proceed to its implementation. As an initial step in these implementations, the spectral covariance matrix from the overall cube needs to be obtained. Conventional implementations simply convert the hyperspectral cube into a $N_\lambda \times N_r N_c$ matrix to undertake subsequent mathematical operations. However, since the $N_r N_c$ dimension of this matrix can achieve extremely large values (over 100 kB), its processing needs complex and powerful requirement, usually resulting in memory and computing problems.

On the other hand, regarding the acquisition procedures in most HSI devices, the hyperspectral cube acquisition is performed in a sequential manner, where smaller 2-D data slices are captured as a subspace of the final cube. Moreover, the size of these subspace partitions is determined by the type of sequential acquisition undertaken in the device, where the time gap between sequential captures can be employed more

efficiently. Therefore, considering the nature of the hyperspectral cube, and taking advantage of the current HSI acquisition technology, the covariance matrix can be computed more efficiently, sequentially dealing with smaller data partitions and, hence, partial covariance matrices.

In fact, 4 different implementation schemes are proposed as an alternative to the conventional covariance computation, with corresponding theoretical explanation and complete experimental results validating the effectiveness of the SC methodology.

3.2 Data Acquisition and Covariance Computation

Acquisition of the HSI cube is carried out using some procedures, normally in a sequential manner. Then, once the HSI data is captured in the hyperspectral cube, the covariance matrix is usually computed for data reduction and feature extraction methods by the conventional implementation. In this section, these mentioned aspects are briefly described.

3.2.1 Data Acquisition in HSI

Devices in HSI have developed diverse technologies for data acquisition. With relation to these, the devices can be classified into 3 main types, i.e., sequential, simultaneous, and pseudo-simultaneous [58, 59]. Indeed, the sequential type is the most common acquisition procedure, from which the present proposal is derived. At the same time, sequential procedures comprise 2 main groups, scanning and filter-based approaches. A generic optical scheme from these 2 groups is shown in Figure 3.2.

The scanning procedure consists of acquiring partial and discrete scenes in the spatial domain but including their whole spectral range, i.e., partial data is sequentially captured along the spatial domain, where the corresponding spectra are acquired in one. Simply by repeating the spatial acquisition along the covered spatial size, the whole cube is finally generated. Hence, depending on the actual spatial size covered, the scanning methods can be divided in the pixel scanning, where a single pixel is captured at every step, and the line scanning, usually known as pushbroom, in which a portion of pixels forming a row or column in the image is captured every time.

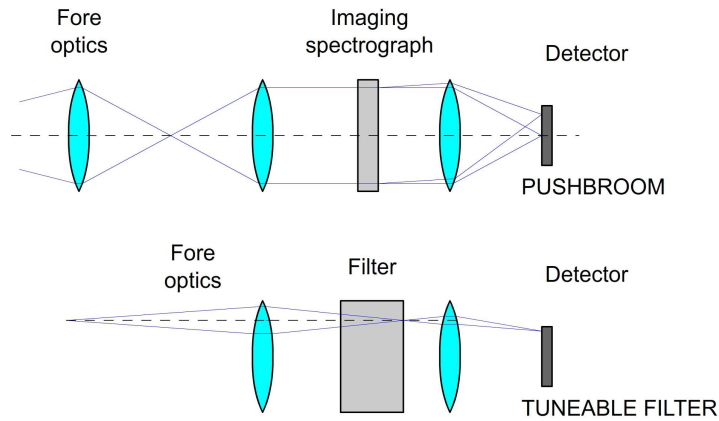


Fig. 3.2 Basic scheme for pushbroom scanning and tuneable filtering

Filter-based procedures, unlike the scanning ones, simply capture the whole spatial size at once, but just for a unique wavelength in the spectrum. Therefore, a singular wavelength is added at every step until completing the cube. In addition, 2 types of filter can be used in these techniques, a passive filter, which captures specific wavelengths, and a tuneable filter, prepared for capturing a range and usually preferred.

All in all, the different sequential acquisition procedures in HSI are schematically represented in Figure 3.3, where it is easy to discern the diverse partitions or subspaces (pixel, row, column, and band) employed in the sequential capture. Taking advantage of the time gap between sequential steps in the device, the SC proposal can be implemented for constructing the covariance matrix efficiently, as an alternative to the conventional case.

3.2.2 Conventional Computation of Covariance Matrix

Diverse methods for feature extraction and data reduction require the computation of a covariance matrix at a given point. For example, the widely used PCA method [12, 55, 116] is able to transform a large set of correlated features into a smaller set of uncorrelated ones by orthogonal projection. This projection is carried out thanks to the information contained in the spectral covariance matrix, to which an EigenValue Decomposition (EVD) is applied. Therefore, the covariance matrix is the key element of the method and its implementation.

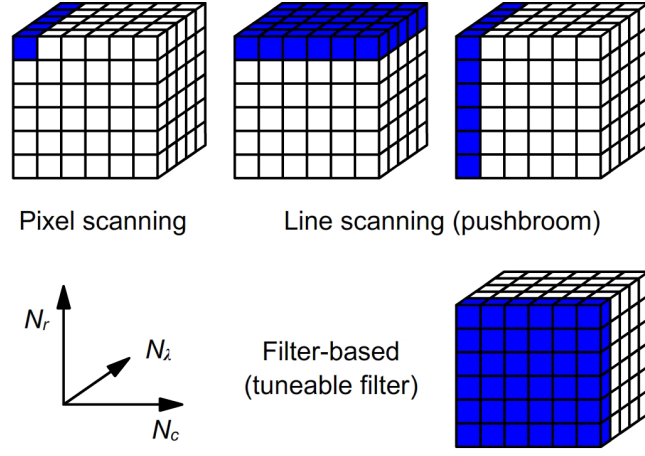


Fig. 3.3 Subpartitions in sequential captures of a 3-D hyperspectral cube

In HSI, the initial 3-D cube is reallocated into a 2-D matrix, namely data matrix, where each row represents the spectral bands or features (N_λ), while the columns contain the pixels or samples ($N_s = N_r \times N_c$). This data matrix, shown in Figure 3.4, is employed to compute the covariance matrix in the conventional way.

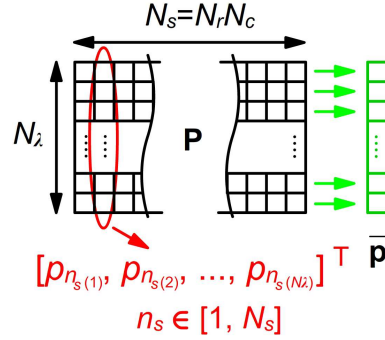


Fig. 3.4 Data matrix in conventional implementations

Given a sample or spectral vector $\mathbf{p}_{n_s} = [p_{n_s(1)}, p_{n_s(2)}, \dots, p_{n_s(N_\lambda)}]^\top$ in the cube, having $n_s \in [1, N_r N_c]$ as the number of samples and N_λ as the number of features, the mean sample from the hyperspectral cube is defined as

$$\bar{\mathbf{p}} = \frac{1}{N_r N_c} \sum_{n_s=1}^{N_s} \mathbf{p}_{n_s}. \quad (3.1)$$

The mean-adjustment of all samples in the data matrix is a requirement to compute the covariance matrix, and this is easily done by subtracting the mean sample as

$$\mathbf{y}_{n_s} = \mathbf{p}_{n_s} - \bar{\mathbf{p}} \quad n_s \in [1, N_s], \quad (3.2)$$

where \mathbf{y}_{n_s} is the resulting mean-adjusted sample used for the covariance matrix calculation. Consequently, given the matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{N_s}]$, where all samples are mean-adjusted, the covariance matrix of \mathbf{P} , called \mathbf{C} , is formulated as

$$\begin{aligned} \mathbf{C} &= \text{expect} \left\{ (\mathbf{p}_{n_s} - \text{expect}\{\mathbf{p}_{n_s}\}) (\mathbf{p}_{n_s} - \text{expect}\{\mathbf{p}_{n_s}\})^\top \right\} \\ &\approx \text{expect}\{\mathbf{y}_{n_s} \mathbf{y}_{n_s}^\top\} \\ &\approx \mathbf{Y} \mathbf{Y}^\top \end{aligned} \quad , \quad (3.3)$$

being $\text{expect}\{\}$ the expectation over the pixel dimension, omitting the dividing term and considering $\mathbf{C} = \mathbf{Y} \mathbf{Y}^\top$ for simplicity, where $\mathbf{Y} \in \mathbb{R}^{N_\lambda \times N_s}$ and $\mathbf{C} \in \mathbb{R}^{N_\lambda \times N_\lambda}$.

As can be derived from these equations, conventional computation of the covariance matrix in HSI presents 2 main disadvantages. On one hand, it needs the whole cube (or data matrix) to achieve the mean sample and mean-adjusted data matrix \mathbf{Y} , which removes any possibility of computing covariance in real time simultaneously to the acquisition. On the other hand, and more importantly, as the mean-adjusted data matrix $\mathbf{Y} \in \mathbb{R}^{N_\lambda \times N_s}$ presents extremely large dimensions, calculation of the covariance matrix by Equation 3.3 can cause contiguous memory problems. In order to address these disadvantages, the SC proposal is introduced in Section 3.3.

3.3 Proposed Structured Covariance (SC)

Since the covariance matrix computation in HSI presents such problems and, at the same time, common HSI acquisition procedures are known to be sequential, it is possible to take advantage of the 3-D hyperspectral cube, where subspace partitions of data are sequentially captured and accessed, in order to efficiently compute the covariance matrix for feature extraction and data reduction methods.

Working with subspace partitions, the covariance matrix computation becomes a sequential implementation in which, at every step, the matrices and multiplications involved are smaller in size, significantly reducing the memory requirements. Moreover, this sequential implementation also allows real-time computation, although since subpartitions are initially processed without being mean-adjusted, they require a correction. This correction step, to be applied for real-time cases, is also derived in this

section. Additionally, it is worth mentioning that the SC schemes could also be implemented in multiple parallel threads for further efficiency.

The contribution in this chapter provides a total of 4 SC schemes, including pixel-, row-, column-, and band-based approaches for efficient covariance computation. These are compared to the conventional case in Figure 3.5 for easy understanding to readers. The main mathematical notations and symbols employed in the SC methodology and conventional case are included in Table 3.1. In the following, the different SC schemes are discussed.

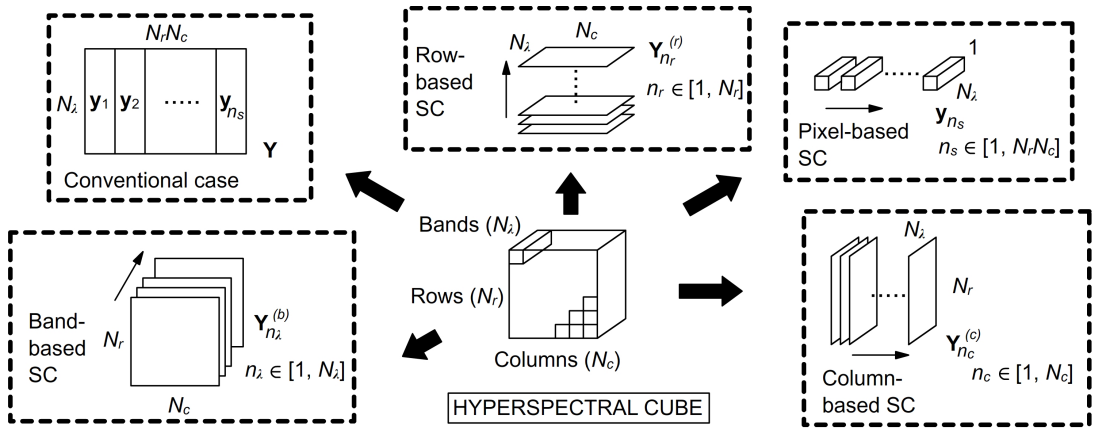


Fig. 3.5 Conventional and SC computations in HSI

Table 3.1 Main notation for conventional and SC computations

Partition	Covariance (partial)	Covariance (full)
Conventional	$\mathbf{Y} \in \mathbb{R}^{N_\lambda \times N_r N_c}$	$\mathbf{C} = \mathbf{Y}\mathbf{Y}^\top \in \mathbb{R}^{N_\lambda \times N_\lambda}$
Pixel	$\mathbf{y}_{n_s} \in \mathbb{R}^{N_\lambda \times 1}$	$\mathbf{C}_{n_s}^{(p)} = \mathbf{y}_{n_s} \mathbf{y}_{n_s}^\top$
Row	$\mathbf{Y}_{n_r}^{(r)} \in \mathbb{R}^{N_\lambda \times N_c}$	$\mathbf{C}_{n_r}^{(r)} = \mathbf{Y}_{n_r}^{(r)} (\mathbf{Y}_{n_r}^{(r)})^\top$
Column	$\mathbf{Y}_{n_c}^{(c)} \in \mathbb{R}^{N_\lambda \times N_r}$	$\mathbf{C}_{n_c}^{(c)} = \mathbf{Y}_{n_c}^{(c)} (\mathbf{Y}_{n_c}^{(c)})^\top$
Band	$\mathbf{Y}_{n_\lambda}^{(b)} \in \mathbb{R}^{N_r \times N_c}$	$\mathbf{C}^{(b)}(i, j) = \text{vec}\{\mathbf{Y}_{n_\lambda=i}^{(b)}\} (\text{vec}\{\mathbf{Y}_{n_\lambda=j}^{(b)}\})^\top$

3.3.1 Structured Covariance Schemes

Based on the sequential subpartitions of the HSI cube, it is possible to derive several different schemes by which the SC proposal is implemented. These schemes are based on pixel, row, column, and band partitions, as described in the following.

A. Pixel-based SC Scheme

In this scheme, only one sample or pixel \mathbf{y}_{n_s} obtained from the pixel scanning procedure is used in order to partially compute the covariance matrix, where $n_s \in [1, N_s]$ and $\mathbf{y}_{n_s} \in \mathbb{R}^{N_\lambda \times 1}$. Final covariance matrix $\mathbf{C}^{(p)}$ is then obtained by accumulation of partial covariances $\mathbf{C}_{n_s}^{(p)}$ from all the pixels in the hyperspectral cube, as

$$\mathbf{C}^{(p)} = \sum_{n_s=1}^{N_s} \mathbf{C}_{n_s}^{(p)}, \quad \mathbf{C}_{n_s}^{(p)} = \mathbf{y}_{n_s} \mathbf{y}_{n_s}^\top. \quad (3.4)$$

Therefore, the partial covariances are computed simply by the vector \mathbf{y}_{n_s} , with dimensions $N_\lambda \times 1$, much smaller than dimensions $N_\lambda \times N_r N_c$ from the conventional case. On the other hand, the number of partial covariances has now increased to $N_r N_c$.

B. Row-based SC Scheme

In this scheme, 2-D partitions of data along the row direction of the cube obtained from the pushbroom or line scanning procedure are employed. Defining these row partitions as $\mathbf{Y}_{n_r}^{(r)} \in \mathbb{R}^{N_\lambda \times N_c}$, where $n_r \in [1, N_r]$, each partition $\mathbf{Y}_{n_r}^{(r)}$ is expressed as

$$\mathbf{Y}_{n_r}^{(r)} = [\mathbf{y}_{n_r}, \mathbf{y}_{n_r+N_r}, \dots, \mathbf{y}_{n_r+(N_c-1)N_r}]. \quad (3.5)$$

Consequently, the covariance matrix $\mathbf{C}^{(r)} \in \mathbb{R}^{N_\lambda \times N_\lambda}$ can be derived as the summation of N_r partial covariance matrices $\mathbf{C}_{n_r}^{(r)}$ obtained from $\mathbf{Y}_{n_r}^{(r)}$ as follows

$$\mathbf{C}^{(r)} = \sum_{n_r=1}^{N_r} \mathbf{C}_{n_r}^{(r)}, \quad \mathbf{C}_{n_r}^{(r)} = \mathbf{Y}_{n_r}^{(r)} (\mathbf{Y}_{n_r}^{(r)})^\top, \quad (3.6)$$

where the partial covariances are computed simply by $\mathbf{Y}_{n_r}^{(r)}$, with dimensions $N_\lambda \times N_c$ much reduced in comparison to $N_\lambda \times N_r N_c$ from the conventional case.

C. Column-based SC Scheme

In this scheme, 2-D partitions of data along the column direction of the hyperspectral cube obtained as an alternative to the row-based approach in the pushbroom scanning procedure are used. Defining these column partitions as $\mathbf{Y}_{n_c}^{(c)} \in \mathbb{R}^{N_\lambda \times N_r}$, where $n_c \in [1, N_c]$, each partition $\mathbf{Y}_{n_c}^{(c)}$ is expressed as

$$\mathbf{Y}_{n_c}^{(c)} = [\mathbf{y}_{1+N_r(n_c-1)}, \mathbf{y}_{2+N_r(n_c-1)}, \dots, \mathbf{y}_{N_r+N_r(n_c-1)}]. \quad (3.7)$$

Consequently, the covariance matrix $\mathbf{C}^{(c)} \in \mathbb{R}^{N_\lambda \times N_\lambda}$ can be derived as the summation of N_c partial covariance matrices $\mathbf{C}_{n_c}^{(c)}$ obtained from $\mathbf{Y}_{n_c}^{(c)}$ as follows

$$\mathbf{C}^{(c)} = \sum_{n_c=1}^{N_c} \mathbf{C}_{n_c}^{(c)}, \quad \mathbf{C}_{n_c}^{(c)} = \mathbf{Y}_{n_c}^{(c)} (\mathbf{Y}_{n_c}^{(c)})^\top, \quad (3.8)$$

where the partial covariances are again computed simply by $\mathbf{Y}_{n_c}^{(c)}$, with dimensions $N_\lambda \times N_r$ reduced in comparison to $N_\lambda \times N_r N_c$ from the conventional case.

D. Band-based SC Scheme

Unlike the 3 previous schemes, the band-based case is derived not from scanning procedures but from the tuneable filtering, capturing all the spatial size for a selected wavelength at once. In this scheme, 2-D partitions of data along the band direction of the hyperspectral cube are employed.

Defining these band partitions as $\mathbf{Y}_{n_\lambda}^{(b)} \in \mathbb{R}^{N_r \times N_c}$, where $n_\lambda \in [1, N_\lambda]$, each partition $\mathbf{Y}_{n_\lambda}^{(b)}$ can be represented as

$$\mathbf{Y}_{n_\lambda}^{(b)} = \begin{bmatrix} y_{1(n_\lambda)} & \cdots & y_{N_r(N_c-1)+1(n_\lambda)} \\ \vdots & \ddots & \vdots \\ y_{N_r(n_\lambda)} & \cdots & y_{N_r N_c(n_\lambda)} \end{bmatrix}. \quad (3.9)$$

It is important to remark that now the selected partition $\mathbf{Y}_{n_\lambda}^{(b)}$ possesses all spatial pixels in a unique spectral band, missing the rest of bands. This fundamental difference makes it impossible to compute and accumulate partial covariances like in the rest of schemes. For that reason, the sequential covariance computation is based on matrix elements. Given an element in position (i, j) , it can be expressed as

$$\mathbf{C}_{(i,j)}^{(b)} = \text{vec}\{\mathbf{Y}_{n_\lambda=i}^{(b)}\} (\text{vec}\{\mathbf{Y}_{n_\lambda=j}^{(b)}\})^\top, \quad (3.10)$$

where operator $\text{vec}\{\}$ simply reallocates the 2-D partition $\mathbf{Y}_{n_\lambda}^{(b)}$ into a vector with dimensions $1 \times N_r N_c$, which multiplication in Equation 3.10 leads to a scalar value.

Therefore, the final covariance matrix is obtained through progressive incorporation of elements (i, j) computed from bands i and j , respectively, where vector dimensions of $1 \times N_r N_c$ are remarkably reduced in comparison to conventional $N_\lambda \times N_r N_c$.

3.3.2 Mathematical Equivalencies

The SC proposal aims to efficiently compute the covariance matrix, proposing alternative implementations that lead to exactly the same numerical results, i.e. the SC schemes are equivalent to the conventional case. In this section, it is proved that the covariance matrix from a SC scheme is just the same as the conventional one.

Following the conventional implementation, covariance matrix \mathbf{C} is expressed as $\mathbf{C} = \mathbf{Y}\mathbf{Y}^\top \in \mathbb{R}^{N_\lambda \times N_\lambda}$. Hence, every element (i, j) can be simply defined as

$$\mathbf{C}_{(i,j)} = \sum_{n_s=1}^{N_r N_c} y_{n_s(i)} y_{n_s(j)}. \quad (3.11)$$

On the other hand, following the pixel-based SC scheme, from a single pixel partition, partial covariance is achieved as

$$\mathbf{C}_{n_s}^{(p)} = \mathbf{y}_{n_s} \mathbf{y}_{n_s}^\top = \begin{bmatrix} y_{n_s(1)} y_{n_s(1)} & \cdots & y_{n_s(1)} y_{n_s(N_\lambda)} \\ \vdots & \ddots & \vdots \\ y_{n_s(N_\lambda)} y_{n_s(1)} & \cdots & y_{n_s(N_\lambda)} y_{n_s(N_\lambda)} \end{bmatrix}_{N_\lambda \times N_\lambda}. \quad (3.12)$$

In accordance with Equation 3.4, the covariance matrix is obtained through the accumulation of these partial covariances as

$$\begin{aligned} \mathbf{C}^{(p)} &= \sum_{n_s=1}^{N_r N_c} \mathbf{C}_{n_s}^{(p)} \\ &= \begin{bmatrix} \sum_{n_s=1}^{N_r N_c} y_{n_s(1)} y_{n_s(1)} & \cdots & \sum_{n_s=1}^{N_r N_c} y_{n_s(1)} y_{n_s(N_\lambda)} \\ \vdots & \ddots & \vdots \\ \sum_{n_s=1}^{N_r N_c} y_{n_s(N_\lambda)} y_{n_s(1)} & \cdots & \sum_{n_s=1}^{N_r N_c} y_{n_s(N_\lambda)} y_{n_s(N_\lambda)} \end{bmatrix}_{N_\lambda \times N_\lambda}. \end{aligned} \quad (3.13)$$

Now, directly comparing Equation 3.11 and Equation 3.13, it is evident that both matrices are just the same, proving the 2 implementations to be equivalent to each other. Equivalency to the rest of SC schemes is also proved by similar mechanisms.

3.3.3 Correction for Real-Time Covariance Computation

It is important to remark that the SC schemes can be simply implemented on the whole hyperspectral cube once the acquisition process is finished, computing in the first place the mean-adjustment to the data matrix. By this implementation, the memory requirements are highly reduced, with subsequent advantages.

However, the potential of the SC schemes goes beyond relieving memory requirements, as the sequential procedures in HSI acquisition allow fast implementations in real time, without requiring parallel implementation as suggested in [56, 57]. In that sense, the mean-adjustment step has to be considered for real-time implementations.

During the acquisition process, the sequentially captured partitions of data are obviously not mean-adjusted. Therefore, in place of employing the mentioned partitions \mathbf{y}_{n_s} , $\mathbf{Y}_{n_r}^{(r)}$, and $\mathbf{Y}_{n_c}^{(c)}$ for pixel-, row-, and column-based schemes respectively, now only the non-mean-adjusted equivalent partitions \mathbf{p}_{n_s} , $\mathbf{P}_{n_r}^{(r)}$, and $\mathbf{P}_{n_c}^{(c)}$ can be used. On the other hand, this is not the case of the band-based scheme, as all pixels from a single wavelength are captured at the same time, making the mean-adjustment from Equation 3.2 feasible in real time. Therefore, no correction is needed in the band-based scheme.

In order to illustrate the real-time correction solution, the pixel partition is selected again. As already mentioned, the sequentially obtained pixels \mathbf{p}_{n_s} are not mean-adjusted, unlike required by the conventional implementation. For that reason, a correction factor \mathbf{CF} based on the mean pixel in Equation 3.1 must be applied as

$$\begin{aligned} \mathbf{y}_{n_s} \mathbf{y}_{n_s}^\top &= \mathbf{p}_{n_s} \mathbf{p}_{n_s}^\top + \mathbf{CF}_{n_s}^{(p)} \\ \mathbf{CF}_{n_s}^{(p)} &= \bar{\mathbf{p}} \bar{\mathbf{p}}^\top - \mathbf{p}_{n_s} \bar{\mathbf{p}}^\top - \bar{\mathbf{p}} \mathbf{p}_{n_s}^\top \end{aligned} \quad (3.14)$$

where it is easy to realise that the correction factor $\mathbf{CF}_{n_s}^{(p)} \in \mathbb{R}^{N_\lambda \times N_\lambda}$ is derived from the corresponding pixel \mathbf{p}_{n_s} and $\bar{\mathbf{p}}$, with relation to the product of subtracted values

$$\begin{aligned} y_{n_s(i)} y_{n_s(j)} &= (p_{n_s(i)} - \bar{p}_{(i)}) (p_{n_s(j)} - \bar{p}_{(j)}) \\ &= p_{n_s(i)} p_{n_s(j)} + \bar{p}_{(i)} \bar{p}_{(j)} - p_{n_s(i)} \bar{p}_{(j)} - \bar{p}_{(i)} p_{n_s(j)} \end{aligned} \quad (3.15)$$

Hence, in real-time cases, the pixel-based SC scheme provides the covariance matrix as in Equation 3.4 adding the correction factor at the end of the acquisition

$$\mathbf{C}^{(p)} = \sum_{n_s=1}^{N_r N_c} \mathbf{p}_{n_s} \mathbf{p}_{n_s}^\top + \sum_{n_s=1}^{N_r N_c} \mathbf{CF}_{n_s}^{(p)}. \quad (3.16)$$

Regarding the row-based scheme, similarly to the pixel case, a correction factor needs to be introduced

$$\begin{aligned} \mathbf{Y}_{n_r}^{(r)} (\mathbf{Y}_{n_r}^{(r)})^\top &= \mathbf{P}_{n_r}^{(r)} (\mathbf{P}_{n_r}^{(r)})^\top + \mathbf{CF}_{n_r}^{(r)} \\ \mathbf{CF}_{n_r}^{(r)} &= \bar{\mathbf{p}} \bar{\mathbf{p}}^\top - \mathbf{P}_{n_r}^{(r)} [\bar{\mathbf{p}}, \dots, \bar{\mathbf{p}}]_{N_\lambda \times N_c}^\top - [\bar{\mathbf{p}}, \dots, \bar{\mathbf{p}}]_{N_\lambda \times N_c} (\mathbf{P}_{n_r}^{(r)})^\top, \end{aligned} \quad (3.17)$$

being the correction factor $\mathbf{CF}_{n_r}^{(r)} \in \mathbb{R}^{N_\lambda \times N_\lambda}$ derived from the corresponding partition $\mathbf{P}_{n_r}^{(r)}$ and the mean pixel $\bar{\mathbf{p}}$, where the covariance matrix is now expressed as

$$\mathbf{C}^{(r)} = \sum_{n_r=1}^{N_r} \mathbf{P}_{n_r}^{(r)} (\mathbf{P}_{n_r}^{(r)})^\top + \sum_{n_r=1}^{N_r} \mathbf{CF}_{n_r}^{(r)}. \quad (3.18)$$

Analogously to the row case, a correction factor is added in the column-based scheme in the same terms

$$\begin{aligned} \mathbf{Y}_{n_c}^{(c)} (\mathbf{Y}_{n_c}^{(c)})^\top &= \mathbf{P}_{n_c}^{(c)} (\mathbf{P}_{n_c}^{(c)})^\top + \mathbf{CF}_{n_c}^{(c)} \\ \mathbf{CF}_{n_c}^{(c)} &= \bar{\mathbf{p}} \bar{\mathbf{p}}^\top - \mathbf{P}_{n_c}^{(c)} [\bar{\mathbf{p}}, \dots, \bar{\mathbf{p}}]_{N_\lambda \times N_r}^\top - [\bar{\mathbf{p}}, \dots, \bar{\mathbf{p}}]_{N_\lambda \times N_r} (\mathbf{P}_{n_c}^{(c)})^\top, \end{aligned} \quad (3.19)$$

being the correction factor $\mathbf{CF}_{n_c}^{(c)} \in \mathbb{R}^{N_\lambda \times N_\lambda}$ derived from the corresponding partition $\mathbf{P}_{n_c}^{(c)}$ and the mean pixel $\bar{\mathbf{p}}$. Then, a similar expression for covariance matrix is achieved

$$\mathbf{C}^{(c)} = \sum_{n_c=1}^{N_c} \mathbf{P}_{n_c}^{(c)} (\mathbf{P}_{n_c}^{(c)})^\top + \sum_{n_c=1}^{N_c} \mathbf{CF}_{n_c}^{(c)}. \quad (3.20)$$

Once developed the mathematical expressions for covariance computation in real time, it is not difficult to see that the 3 correction factors in Equation 3.16, 3.18, and 3.20 are actually equivalent to each other, so a global correction matrix is defined as

$$\begin{aligned} \mathbf{CM} &= \sum_{n_s=1}^{N_r N_c} \mathbf{CF}_{n_s}^{(p)} = \sum_{n_r=1}^{N_r} \mathbf{CF}_{n_r}^{(r)} = \sum_{n_c=1}^{N_c} \mathbf{CF}_{n_c}^{(c)} \\ &= \sum_{n_s=1}^{N_r N_c} (\bar{\mathbf{p}} \bar{\mathbf{p}}^\top - \mathbf{p}_{n_s} \bar{\mathbf{p}}^\top - \bar{\mathbf{p}} \mathbf{p}_{n_s}^\top). \end{aligned} \quad (3.21)$$

Moreover, it is not necessary to achieve the end of the acquisition process for the correction matrix to be calculated, as this matrix can be updated sequentially during the

acquisition. Developing the correction matrix for the pixel-based scheme, an element in position (i, j) can be expressed as

$$\begin{aligned} \mathbf{CM}_{(i,j)} &= \sum_{n_s=1}^{N_r N_c} \bar{p}_{(i)} \bar{p}_{(j)} - \sum_{n_s=1}^{N_r N_c} p_{n_s(i)} \bar{p}_{(j)} - \sum_{n_s=1}^{N_r N_c} \bar{p}_{(i)} p_{n_s(j)} \\ &= N_r N_c (\bar{p}_{(i)} \bar{p}_{(j)}) - \bar{p}_{(j)} \sum_{n_s=1}^{N_r N_c} p_{n_s(i)} - \bar{p}_{(i)} \sum_{n_s=1}^{N_r N_c} p_{n_s(j)} \end{aligned} \quad (3.22)$$

then, multiplying and dividing the 2nd and 3rd elements in Equation 3.22 by the same factor $N_r N_c$, the following is obtained

$$\begin{aligned} \bar{p}_{(j)} N_r N_c \frac{1}{N_r N_c} \sum_{n_s=1}^{N_r N_c} p_{n_s(i)} &= \bar{p}_{(j)} N_r N_c \bar{p}_{(i)} \\ \bar{p}_{(i)} N_r N_c \frac{1}{N_r N_c} \sum_{n_s=1}^{N_r N_c} p_{n_s(j)} &= \bar{p}_{(i)} N_r N_c \bar{p}_{(j)} \end{aligned} \quad (3.23)$$

Therefore, it is possible to further express the correction matrix \mathbf{CM} as

$$\mathbf{CM}_{(i,j)} = -N_r N_c \bar{p}_{(i)} \bar{p}_{(j)}. \quad (3.24)$$

In the end, the matrix \mathbf{CM} can be obtained by

$$\mathbf{CM}_{(i,j)} = -\frac{1}{N_r N_c} \sum_{n_s=1}^{N_r N_c} p_{n_s(i)} \sum_{n_s=1}^{N_r N_c} p_{n_s(j)}, \quad (3.25)$$

hence, the different elements (i, j) from the correction matrix can be achieved by the accumulation of partial values inline with the acquisition process. Finally, the covariance matrices from the SC implementations in real time are defined as

$$\begin{aligned} \mathbf{C}^{(p)} &= \sum_{n_s=1}^{N_r N_c} \mathbf{p}_{n_s} \mathbf{p}_{n_s}^\top + \mathbf{CM} \\ \mathbf{C}^{(r)} &= \sum_{n_r=1}^{N_r} \mathbf{P}_{n_r}^{(r)} (\mathbf{P}_{n_r}^{(r)})^\top + \mathbf{CM} \\ \mathbf{C}^{(c)} &= \sum_{n_c=1}^{N_c} \mathbf{P}_{n_c}^{(c)} (\mathbf{P}_{n_c}^{(c)})^\top + \mathbf{CM} \end{aligned} \quad (3.26)$$

3.4 Analysis and Evaluation

The SC concept proposed for efficient computation of the covariance matrix in HSI, comprehensively described in the previous section, is now subject to an appropriate

analysis by which to prove the actual benefits from this contribution, using MATLAB [48] environment and with related content also available in [11].

The experimental setup for this particular evaluation is introduced in the following, where additional information can be found in Chapter 2. The key point of the analysis is to show the equivalency of the proposed SC implementation in terms of classification accuracy, while efficiency measurements indicate a reduction in memory requirements along with the possibility of onsite and even real-time computation.

3.4.1 Experiments

Following the generic setup commonly applied to the present research, it is possible to divide experiments into the 4 main stages described in Section 2.3.1. Some considerations about these experimental settings are given in the following.

A. Data Description

In this particular chapter, a total of 3 HSI remote sensing data sets are used, where each of them comes from a different sensor. This proves that the SC implementations work efficiently regardless of the acquisition source. These data sets are:

1. Indian Pines [36] (Appendix A.1).
2. Pavia UA [36] (Appendix A.4).
3. Botswana A [36] (Appendix A.6).

B. Data Conditioning

Some noisy spectral bands are discarded according to the description available in Appendix A for each data set, respectively. On the other hand, all labelled classes in the available ground truth are included in the land-cover analysis.

C. Feature Extraction

In this particular case, attention is focused not only on the resulting features, but also on the intermediate covariance matrix, which is needed for implementing feature extraction and data reduction methods such as PCA [12, 55, 116].

Hence, in this stage, the obtained covariance matrix from the conventional implementation is compared to the ones from the SC proposals. The covariance matrix is computed from the whole hyperspectral cube in each case, in an unsupervised manner with relation to the labelled pixels. The different methods evaluated for computing the covariance matrix in PCA are:

1. Conventional (Section 3.2.2).
2. SC-Pixel (Section 3.3.1 A).
3. SC-Row (Section 3.3.1 B).
4. SC-Column (Section 3.3.1 C).
5. SC-Band (Section 3.3.1 D).

The main comparisons are made in terms of memory requirements, computational complexity (number of multiplications and additions), and approximated computation time for real-time implementation onsite acquisition devices.

D. Classification

The classification analysis here is undertaken in order to prove that resulting features from the PCA method using the covariance matrix from conventional and SC implementations are just the same. In other words, alternative implementations lead to the same covariance matrix, which translates into the same PCA features and, therefore, the same classification results.

Having the LIBSVM [45] as a classifier, Support Vector Machine (SVM) models are trained with 30% of the labelled pixels in each class. With random training and testing partitions, classification accuracy is provided in terms of Overall Accuracy (OA), presenting the mean values out of 10 repetitions.

3.4.2 Results

Several evaluations are carried out to assess the contributions in the present chapter [11]. In the first place, (A) accuracy in land-cover classification with PCA is compared

for the different covariance implementations. Once it is proved that the SC schemes provide an equivalent covariance matrix leading to the same features, some efficiency aspects are compared, including (B) memory requirements, (C) computational complexity, and (D) computation time from the different implementations. These comparisons are available in the following.

A. Classification Accuracy

Given that the resulting covariance matrix from all the SC schemes is equivalent to the matrix obtained by the conventional implementation, any of them can be used inside the PCA method implementation, so the dimensionality of features is equally reduced to, in this case, $F = 10$ elements. Then, the new samples with reduced dimensionality are classified by SVM, achieving some classification OA (%) presented in Table 3.2. As expected, all the covariance matrices from the SC schemes lead to exactly the same features and, therefore, same results.

Table 3.2 Mean OA using PCA ($F = 10$)

Method	Indian Pines	Pavia UA	Botswana A
Conventional	80.24 ± 0.31	96.93 ± 0.20	94.34 ± 1.67
SC-Pixel	80.24 ± 0.31	96.93 ± 0.20	94.34 ± 1.67
SC-Row	80.24 ± 0.31	96.93 ± 0.20	94.34 ± 1.67
SC-Column	80.24 ± 0.31	96.93 ± 0.20	94.34 ± 1.67
SC-Band	80.24 ± 0.31	96.93 ± 0.20	94.34 ± 1.67

Moreover, the PCA method with the different covariance matrices is implemented for several different number of features, from 1 to 10. By doing this, the classification accuracy trend with relation to the number of features F is represented in Figure 3.6 for the 3 data sets, where results from conventional and SC schemes are again exactly the same.

The classification OA tends to increment with more features employed, although this increment is faster for the Pavia UA and the Botswana A data sets (see $F = 1, 2$). In general, when $F > 5$ the increment is reduced and it stabilises near $F = 10$ for the 3 data sets. Obviously, these results are the same for any approach employed in the covariance matrix computation.

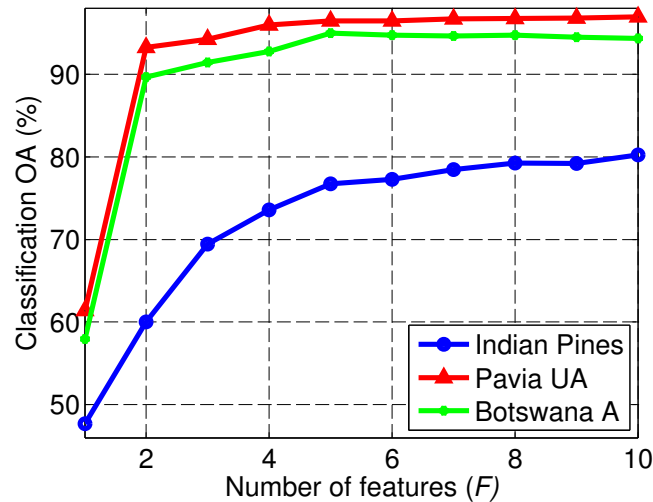


Fig. 3.6 Mean OA using PCA ($F = 1$ to 10)

B. Memory Requirements

Memory needs in terms of matrices size are compared among the different cases and data sets in Table 3.3, where MATLAB [48] data format is double (float), which means 8 B for each data value.

Table 3.3 Matrices dimension and size (kB) for SC computations

Method	Matrices dimension	Indian Pines	Pavia UA	Botswana A
Conventional	$N_\lambda \times N_r N_c$	33640	18540	26100
SC-Pixel	$N_\lambda \times 1$	1.6	0.83	1.2
SC-Row	$N_\lambda \times N_c$	232	124	87
SC-Column	$N_\lambda \times N_r$	232	124	348
SC-Band	$1 \times N_r N_c$	168	180	180

As derived from this table, the conventional case demands massive contiguous memory when computing the covariance matrix, however, the SC implementations significantly relieve this requirement, as the memory size is reduced by $N_r N_c$, N_r , N_c , and N_λ factors for pixel, row, column, and band partitions, respectively.

C. Computational Complexity

Complexity in terms of number of multiplications and additions is also evaluated in Table 3.4, which presents the values for both conventional and SC implementations. The

number of multiplications comprises 2 parts, the mean-adjustment and the data matrix product. Then, the number of additions contains 3 parts, mean-adjustment, partial covariance calculation, and partial covariance accumulation. As expected, regardless of the implementation case, the total number of multiplications and additions is again just the same, where the only difference resides in the distribution of these operations.

Table 3.4 Computational complexity in SC (offline)

Method	Multiplications	Additions
Conventional	$N_\lambda + N_\lambda^2 N_r N_c$	$N_\lambda (2N_r N_c - 1) + N_\lambda^2 (N_r N_c - 1) + 0$
SC-Pixel	$N_\lambda + N_\lambda^2 N_r N_c$	$N_\lambda (2N_r N_c - 1) + 0 + N_\lambda^2 (N_r N_c - 1)$
SC-Row	$N_\lambda + N_\lambda^2 N_r N_c$	$N_\lambda (2N_r N_c - 1) + N_\lambda^2 N_r (N_c - 1) + N_\lambda^2 (N_r - 1)$
SC-Column	$N_\lambda + N_\lambda^2 N_r N_c$	$N_\lambda (2N_r N_c - 1) + N_\lambda^2 (N_r - 1) N_c + N_\lambda^2 (N_c - 1)$
SC-Band	$N_\lambda + N_\lambda^2 N_r N_c$	$N_\lambda (2N_r N_c - 1) + N_\lambda^2 (N_r N_c - 1) + 0$

The computational complexity is evaluated again but for the real-time case. Therefore, multiplications and additions are now distributed along the sequential steps given during the acquisition process, as expressed in Table 3.5. According to the SC description, the pixel scheme provides a minimum computation in each step, at the cost of taking a total of $N_r N_c$ sequential scans to complete acquisition. Moving through the different SC schemes, a logical tradeoff between step complexity and number of steps is found. The global computational complexity is practically the same for all the cases.

Table 3.5 Computational complexity in SC (during acquisition)

Method	Step (n) complexity		Step (n)	After	
	Multiplications	Additions		Mult.	Add.
SC-Pixel	N_λ^2	$3N_\lambda^2$	$N_r N_c$	$2N_\lambda^2$	N_λ^2
SC-Row	$N_\lambda^2 N_c$	$3N_\lambda^2 N_c$	N_r	$2N_\lambda^2$	N_λ^2
SC-Column	$N_\lambda^2 N_r$	$3N_\lambda^2 N_r$	N_c	$2N_\lambda^2$	N_λ^2
SC-Band	$N_r N_c (2n - 1) + 1$	$N_r N_c (2n + 1) - 2n$	N_λ	0	0

Lastly, it is important to remark that the computational cost of all the implementations can, in turn, be approximately reduced from N_λ^2 to $N_\lambda (N_\lambda + 1)/2$ thanks to the symmetry of the covariance matrix, as suggested in [56]. Nevertheless, this fact as well as the potential for parallel implementations and/or calibration corrections is not considered for simplicity.

D. Computation Time

Finally, a comparison of the computation time is performed for the real-time case. Figure 3.7 shows the simulated time required for both the conventional case and SC implementations.

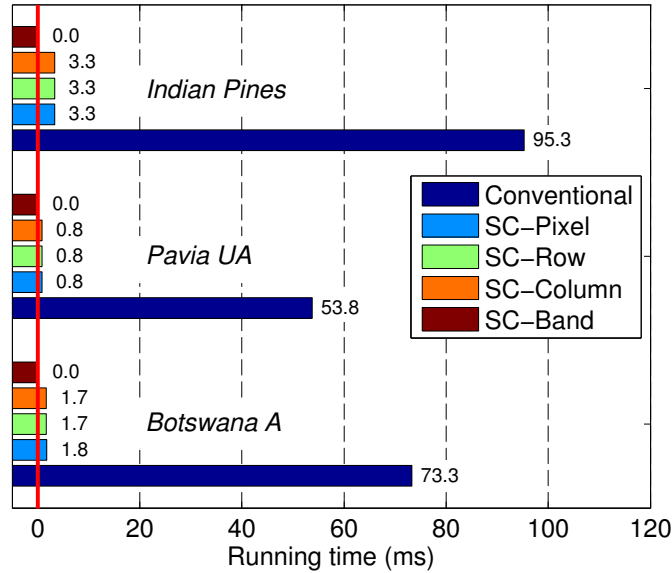


Fig. 3.7 Approximated computation time after acquisition

Even though the band scheme requires no extra time after acquisition (red line), the rest of schemes only need a few extra ms to complete the computation of the covariance matrix. Therefore, the computation time is clearly reduced, leading to faster implementations feasible for onboard or embedded devices.

3.5 Summary

Computation of the covariance matrix is a fundamental aspect for feature extraction and data analysis. However, the conventional computation of the covariance matrix presents some drawbacks when dealing with HSI remote sensing data, such as large matrices leading to challenging memory requirements, especially for portable or embedded systems such as those in satellite or spacecraft platforms. In addition, the acquisition procedures carried out by the HSI devices present a sequential process in which smaller subspaces or partitions of the hyperspectral cube are progressively cap-

tured. Therefore, taking into account these considerations, it is actually possible to benefit from the HSI acquisition procedures in efficient computation of the covariance matrix onsite and satisfy the requirements of real-time data analysis.

Inline with the hyperspectral cube acquisition process, the proposed SC schemes can reduce memory and time requirements as an alternative to the offline conventional implementation. A total of 4 SC schemes are proposed, according to the different subspaces available in a hyperspectral cube, i.e., pixel-, row-, column-, and band-based partitions. Each partition has different characteristics and can thus be used for diverse purposes, leading to interesting potential in more efficient cameras and acquisition devices.

Chapter 4

Folded-PCA for Efficient and Effective Data Reduction

Due to the large dimensionality of features in hyperspectral remote sensing, feature extraction and data reduction methods are usually implemented in subsequent analytical tasks. Probably the most widely used and well-known method for HyperSpectral Imaging (HSI) data reduction is Principal Component Analysis (PCA) [12], which performs an orthogonal projection of the original data according to variance distribution.

However, the PCA method applied to HSI data presents some drawbacks, negatively affecting its performance. Because of the hyperspectral cube nature, the application of PCA requires high computational complexity, with large memory requirements and relatively low efficacy in extracting appropriate features from the HSI data. In order to improve the PCA performance in HSI applications, a PCA variant is proposed in this chapter, where the spectral vector of pixels is folded into a matrix for a reduction in computational cost. Additionally, this proposal namely Folded-PCA (Fd-PCA) leads to an alternative covariance matrix with different properties thanks to the folding procedure in the PCA workflow, resulting in classification benefits.

With folded samples, the new covariance matrix seems to extract both global and local structure in the spectral domain, providing enhanced information in the features, which translates into higher classification accuracy. Moreover, working with folded samples reduces the computational cost and also the memory requirements in the implementation, as shown from comparisons to PCA [12]. These contributions have been published [13] in the ISPRS Journal of Photogrammetry and Remote Sensing.

With relation to the layout of this chapter, the context of research and description of conventional PCA can be found in Section 4.1 and Section 4.2, respectively. Then, Fd-PCA is explained in Section 4.3, including theoretical comparisons. Subsequent analysis and results are in Section 4.4, with summarised remarks in Section 4.5.

4.1 Introduction

The high dimensionality of features present in HSI data leads to great potential in many data analysis problems, but also requires effective feature extraction and data reduction. Thanks to the correlation among contiguous spectral bands in HSI, some data reduction methods are usually applied to the spectral domain of hyperspectral cubes, such as PCA [12, 55, 116], Independent Component Analysis (ICA) [69, 70] or Maximum Noise Fraction (MNF) [71, 72], forming part of the current state-of-the-art methods. In fact, PCA is probably the most widely used method for data reduction in HSI, which can reduce the original number of features from hundreds to only a few, maintaining a sensible performance in terms of classification accuracy.

Conventional implementation of PCA, as already described in Chapter 3, reallocates the initial 3-D hyperspectral cube in a 2-D data matrix from which the covariance matrix is computed. This point essentially comprises 2 challenges. On one hand, this computation can lead to problems when the dimensions in the hyperspectral cube are extremely large, causing even software such as MATLAB [48] to crash because of contiguous memory management. On the other hand, treating all spectral bands in the same way to compute the covariance matrix fails to pick up the disparate contributions arisen from the whole spectrum domain.

In order to solve these disadvantages, a PCA variant called Fd-PCA is proposed. With this alternative algorithm, the new covariance matrix is achieved by accumulation of partial covariances, computed through much smaller matrices thanks to the folding procedure carried out on the spectral pixels or samples. Therefore, using samples not in a vector form but in a matrix representation makes feasible the achievement of a new covariance matrix, where the correlation among spectral bands is more effectively addressed, while at the same time computational burden is much reduced.

4.2 The Conventional PCA Method

PCA is a well-known unsupervised method for feature extraction and data reduction [12, 55, 116]. This method can reduce the dimensionality of features removing the correlation among the original data, by means of an orthogonal projection and posterior truncation of the transformed features.

In general, the PCA method has a clear mathematical background and its implementation is usually straightforward. Not surprisingly, PCA has been widely employed in a vast number of applications, where initial data samples are transformed into a lower dimensional feature space for effective, efficient, or fast analysis such as those in coding [117] or classification [41, 118]. In the following, the PCA algorithm is described and formulated with relevant detail.

4.2.1 Algorithm Description

The conventional implementation of PCA consists of 3 clearly differentiated steps, i.e., (A) covariance matrix computation, (B) EigenValue Decomposition (EVD), and (C) data projection. These steps are explained for particular application of PCA to HSI data reduction.

A. Covariance Matrix Computation

Given a spectral pixel $\mathbf{p}_{n_s} = [p_{n_s(1)}, p_{n_s(2)}, \dots, p_{n_s(N_\lambda)}]^\top$ in the hyperspectral cube, where $n_s \in [1, N_s]$ and N_λ represents the number of spectral bands, the 3-D cube is reallocated into a 2-D matrix with size $N_\lambda \times N_s$, called data matrix, as shown in Figure 4.1.

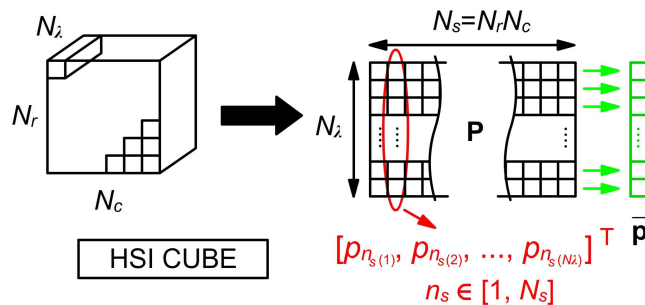


Fig. 4.1 Data matrix in conventional PCA

Covariance matrix computation requires a mean-adjustment procedure by which the mean spectral vector or sample is subtracted to all spectral vectors in the data matrix. In other words, $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{N_s}]$ is the matrix containing the mean-adjusted spectral vectors \mathbf{y}_{n_s} achieved by subtracting the mean spectral vector $\bar{\mathbf{p}}$ which is computed from the whole cube as expressed in the following

$$\begin{aligned} \mathbf{y}_{n_s} &= \mathbf{p}_{n_s} - \bar{\mathbf{p}} \\ \bar{\mathbf{p}} &= \frac{1}{N_s} \sum_{n_s=1}^{N_s} \mathbf{p}_{n_s} \end{aligned} \quad (4.1)$$

Then, covariance matrix \mathbf{C} is obtained by the matrix \mathbf{Y} as

$$\begin{aligned} \mathbf{C} &= \mathit{expect} \left\{ (\mathbf{p}_{n_s} - \mathit{expect}\{\mathbf{p}_{n_s}\}) (\mathbf{p}_{n_s} - \mathit{expect}\{\mathbf{p}_{n_s}\})^\top \right\} \\ &\approx \mathit{expect} \left\{ \mathbf{y}_{n_s} \mathbf{y}_{n_s}^\top \right\} \\ &\approx \mathbf{Y} \mathbf{Y}^\top \end{aligned} \quad (4.2)$$

where $\mathbf{Y} \in \mathbb{R}^{N_\lambda \times N_s}$, $\mathbf{C} \in \mathbb{R}^{N_\lambda \times N_\lambda}$, and $\mathit{expect}\{\}$ is used for denoting mathematical expectation, applied over the pixel dimension. For simplicity, the dividing term is omitted and $\mathbf{C} = \mathbf{Y} \mathbf{Y}^\top$ is considered. In HSI, the size of matrix \mathbf{Y} can be extremely large, and the multiplication performed in Equation 4.2 can be problematic when the resources are limited, such as in the case of portable or embedded devices.

B. EVD

The information contained in the covariance matrix is effectively exploited by EVD, where the spectral covariance \mathbf{C} is decomposed into a multiplication of diagonal and orthonormal matrices as in Equation 4.3

$$\mathbf{C} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top. \quad (4.3)$$

The matrix $\mathbf{\Lambda} = \mathit{diag}\{\lambda_1, \lambda_2, \dots, \lambda_{N_\lambda}\}$ is diagonal containing the eigenvalues of \mathbf{C} , while \mathbf{U} is an orthonormal matrix composed by the related eigenvectors $[\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{N_\lambda}]$. The eigenvalues are sorted in a way that $\lambda_1 > \lambda_2 > \dots > \lambda_{N_\lambda}$, and the smallest ones (along with their corresponding eigenvectors) are usually discarded for data reduction, only selecting the first few eigenvalues.

C. Data Projection

Finally, through the matrix \mathbf{U} of eigenvectors, the spectral vector \mathbf{y}_{n_s} can be transformed into an uncorrelated vector \mathbf{q}_{n_s} as

$$\mathbf{q}_{n_s} = \mathbf{U}^\top \mathbf{y}_{n_s} = \begin{bmatrix} \mathbf{u}_{(1,1)} & \mathbf{u}_{(1,2)} & \cdots & \mathbf{u}_{(1,N_\lambda)} \\ \mathbf{u}_{(2,1)} & \mathbf{u}_{(2,2)} & \cdots & \mathbf{u}_{(2,N_\lambda)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{(N_\lambda,1)} & \mathbf{u}_{(N_\lambda,2)} & \cdots & \mathbf{u}_{(N_\lambda,N_\lambda)} \end{bmatrix}_{N_\lambda \times N_\lambda} \begin{bmatrix} y_{n_s(1)} \\ y_{n_s(2)} \\ \vdots \\ y_{n_s(F)} \\ \vdots \\ y_{n_s(N_\lambda)} \end{bmatrix}_{N_\lambda \times 1}. \quad (4.4)$$

The projected vector \mathbf{q}_{n_s} can be truncated for data reduction, where the dimensionality of features is then reduced to $F \ll N_\lambda$. Since the eigenvalues are sorted in descent order, the smallest ones (or less representative eigenvalues) can be omitted. This implies the use of only the first F eigenvectors, expressed as

$$\begin{aligned} \mathbf{q}_{n_s}^t &= \text{trunc}\{\mathbf{q}_{n_s}\} = [q_{n_s(1)}, q_{n_s(2)}, \dots, q_{n_s(F)}]^\top \\ &= \begin{bmatrix} \mathbf{u}_{(1,1)} & \mathbf{u}_{(1,2)} & \cdots & \mathbf{u}_{(1,N_\lambda)} \\ \mathbf{u}_{(2,1)} & \mathbf{u}_{(2,2)} & \cdots & \mathbf{u}_{(2,N_\lambda)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{(F,1)} & \mathbf{u}_{(F,2)} & \cdots & \mathbf{u}_{(F,N_\lambda)} \end{bmatrix}_{F \times N_\lambda} \begin{bmatrix} y_{n_s(1)} \\ y_{n_s(2)} \\ \vdots \\ y_{n_s(F)} \\ \vdots \\ y_{n_s(N_\lambda)} \end{bmatrix}_{N_\lambda \times 1}. \end{aligned} \quad (4.5)$$

4.2.2 Variations of PCA

Due to the wide applications of PCA, several extensions and variations have been proposed. These variants simply aim to improve the efficiency or efficacy of the conventional PCA method. For instance, in [60] a modified-PCA is proposed, by which using only a selected amount of training data instead of all available samples to compute the covariance matrix provides some speedup in the implementations. Another example can be found in [61] with the incremental-PCA technique. This interesting variant

enables updating of the eigenspace according to previous and new data samples. However, a different variant proposing segmentation, in this case namely Segmented-PCA (S-PCA) [62], has attracted most of the attention.

The S-PCA variant basically consists of implementing conventional PCA separately into different segments of data along the spectrum. The spectral bands are segmented into several groups in accordance with the correlation between band pairs, which is claimed to extract local information from the spectral domain achieving enhanced analysis in classification and data compression tasks [62]. This variant is of particular interest for this research, especially with relation to the extraction of local structures in the spectral domain.

Finally, apart from hyperspectral applications, some other research areas have also proposed interesting variants, where the 2-D version of PCA (2D-PCA) [63] for face recognition tasks in computer vision stands out. The 2D-PCA method is a really motivating approach that reduces the memory requirements along with the computation time while, at the same time, the reduced samples achieve higher classification accuracy. Further discussion about 2D-PCA can be found in [119–121], where in [64] it is proved that the 2D-PCA variant is a particular case of the already introduced block-based-PCA.

4.3 Proposed Folded-PCA

Regarding some PCA variants already introduced in the literature, 2 key points have brought inspiration to propose the Fd-PCA method. On one hand, the spectral segmentation considered by S-PCA highlights the fact that not all the spectral bands contribute the same to extract useful information from the spectral domain. On the other hand, the 2D-PCA variant in face recognition applied to 2-D images remarks that spatial correlation of pixels in this particular case can be addressed by computing samples as 2-D matrices. Merging these 2 concepts together, Fd-PCA is achieved.

While conventional PCA has spectral vectors as inputs, Fd-PCA folds each sample into a 2-D matrix, enabling a 2D-PCA-style analysis [122, 123]. However, the aim of this folding would not be to solve the spatial disruption as in 2D-PCA [63], but to

reduce the computational complexity, while extracting local structures from the spectral domain in a similar way to the S-PCA method [62]. In the following, Fd-PCA is introduced in more detail.

4.3.1 Algorithm Description

Fd-PCA reallocates each sample or spectral vector into a matrix (Figure 4.2), from which a partial covariance matrix can be obtained and then accumulated until the final covariance matrix from all samples in the hyperspectral cube is achieved, to undertake the subsequent EVD and data projection steps.

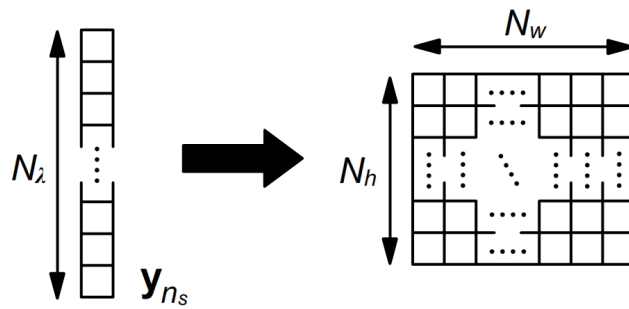


Fig. 4.2 Folding procedure from vector to matrix array

Therefore, in the 2-D matrix, each row contains a group of N_w bands, being the complete set of spectral bands folded into a total of N_h groups or segments. According to this, when $N_h = 1$, Fd-PCA simply degrades to conventional PCA. In the other extreme case, when $N_h = N_\lambda$, it results in no band grouping being implemented. Obviously, Fd-PCA performance depends on the N_h and therefore N_w parameters, where the folding scheme determines the additional information that can be extracted with relation to conventional PCA. Additionally, even though each group or segment can contain a different number of bands, the main algorithm is derived for an even number of bands, simply for an easier understanding. Later, Section 4.3.3 explains how this basic implementation can be extended to the case where uneven bands are used in the folding procedure.

Taking into account the high correlation and redundancy between spectral bands in HSI, as well as the grouping or clustering of bands already implemented for band selection and feature characterisation [124, 125], it is clear that the folding procedure

with appropriate parameters leads to extract not only global but also local structure, proving to be better than conventional PCA [41, 116], with higher discrimination ability and lower computational complexity.

Given a spectral sample $\mathbf{y}_{n_s} = [y_{n_s(1)}, y_{n_s(2)}, \dots, y_{n_s(N_\lambda)}]^\top$, Fd-PCA folds it into a matrix with dimensions $N_h \times N_w$, where $N_\lambda = N_h N_w$. This matrix looks like

$$\mathbf{A}_{n_s} = \begin{bmatrix} \mathbf{a}_{n_s(1)} \\ \mathbf{a}_{n_s(2)} \\ \vdots \\ \mathbf{a}_{n_s(N_h)} \end{bmatrix}_{N_h \times N_w}, \quad (4.6)$$

$$\mathbf{a}_{n_s(n_h)} = [y_{n_s(1+N_w(n_h-1))}, y_{n_s(2+N_w(n_h-1))}, \dots, y_{n_s(N_w+N_w(n_h-1))}]$$

where $n_h \in [1, N_h]$ represents the number of band groups or segments, all of them with N_w bands. Given a matrix \mathbf{A}_{n_s} , its corresponding partial covariance is obtained as

$$\mathbf{C}_{n_s} = \mathbf{A}_{n_s}^\top \mathbf{A}_{n_s}, \quad \mathbf{C}_{n_s} \in \mathbb{R}^{N_w \times N_w}, \quad (4.7)$$

and, therefore, the final covariance matrix from all N_s samples in the hyperspectral cube is achieved by accumulation

$$\mathbf{C}_{FdPCA} = \frac{1}{N_s} \sum_{n_s=1}^{N_s} \mathbf{C}_{n_s} = \frac{1}{N_s} \sum_{n_s=1}^{N_s} \mathbf{A}_{n_s}^\top \mathbf{A}_{n_s}. \quad (4.8)$$

Once the Fd-PCA covariance matrix is obtained, the same methodology in Equation 4.3 is applied in order to compute the EVD. However, the main difference here is that \mathbf{C}_{FdPCA} has dimensions $N_w \times N_w$, much smaller in relation to the conventional case where covariance was sized $N_\lambda \times N_\lambda$, or what is the same, $N_h N_w \times N_h N_w$. This fact directly translates into a lower memory requirement and, even better, reduced computational complexity during the EVD step.

After the EVD, final data projection is carried out. In the Fd-PCA case, Equation 4.4 and Equation 4.5 used in conventional PCA need some adjustment in order to cope with the folded 2-D sample, i.e., the matrix $\mathbf{A}_{n_s} \in \mathbb{R}^{N_h \times N_w}$ instead of the sample vector \mathbf{y}_{n_s} . Denoting F as the reduced number of features to be obtained from Fd-PCA, the matrix of eigenvectors is truncated to F' where $F = N_h F'$ and $\mathbf{U}_t \in \mathbb{R}^{N_w \times F'}$.

Therefore, F' features are extracted from each row in \mathbf{A}_{n_s} by multiplying it with the truncated \mathbf{U}_t , expressed as

$$\mathbf{B}_{n_s} = \mathbf{A}_{n_s} \mathbf{U}_t, \quad \mathbf{A}_{n_s} \in \mathbb{R}^{N_h \times N_w}, \quad \mathbf{U}_t \in \mathbb{R}^{N_w \times F'}. \quad (4.9)$$

Obviously, this results in a total of $F = N_h F'$ extracted features, where the product of smaller matrices with relation to the conventional PCA method leads to lower computational cost. The reduced sample is now defined as $\mathbf{B}_{n_s} \in \mathbb{R}^{N_h \times F'}$, still in matrix form. Simply by reallocating the resulting matrix into a vector array, the reduced sample is ready for subsequent analysis.

4.3.2 Implementation Considerations

In order to further understand the effects of folding vector samples into matrix form, it is essential to find out the mathematical relationship present between the covariance matrix obtained by conventional PCA [12], the proposed Fd-PCA approach, and also the interesting S-PCA [62] method.

Deriving the expression for conventional covariance matrix used in PCA, in terms of band groups or segments $\mathbf{a}_{n_s(n_h)}$ introduced in Equation 4.6, then

$$\begin{aligned} \mathbf{C}_{PCA} &= \frac{1}{N_s} \sum_{n_s=1}^{N_s} \mathbf{y}_{n_s} \mathbf{y}_{n_s}^\top \\ &= \frac{1}{N_s} \sum_{n_s=1}^{N_s} \begin{bmatrix} \mathbf{a}_{n_s(1)}^\top \mathbf{a}_{n_s(1)} & \mathbf{a}_{n_s(1)}^\top \mathbf{a}_{n_s(2)} & \cdots & \mathbf{a}_{n_s(1)}^\top \mathbf{a}_{n_s(N_h)} \\ \mathbf{a}_{n_s(2)}^\top \mathbf{a}_{n_s(1)} & \mathbf{a}_{n_s(2)}^\top \mathbf{a}_{n_s(2)} & \cdots & \mathbf{a}_{n_s(2)}^\top \mathbf{a}_{n_s(N_h)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{n_s(N_h)}^\top \mathbf{a}_{n_s(1)} & \mathbf{a}_{n_s(N_h)}^\top \mathbf{a}_{n_s(2)} & \cdots & \mathbf{a}_{n_s(N_h)}^\top \mathbf{a}_{n_s(N_h)} \end{bmatrix}. \end{aligned} \quad (4.10)$$

On the other hand, the covariance matrix achieved by Fd-PCA thanks to the folded samples can be expressed as

$$\begin{aligned} \mathbf{C}_{FdPCA} &= \frac{1}{N_s} \sum_{n_s=1}^{N_s} \mathbf{A}_{n_s}^\top \mathbf{A}_{n_s} \\ &= \frac{1}{N_s} \sum_{n_s=1}^{N_s} (\mathbf{a}_{n_s(1)}^\top \mathbf{a}_{n_s(1)} + \mathbf{a}_{n_s(2)}^\top \mathbf{a}_{n_s(2)} + \cdots + \mathbf{a}_{n_s(N_h)}^\top \mathbf{a}_{n_s(N_h)}). \end{aligned} \quad (4.11)$$

Furthermore, given a particular segment n_h and assuming for simplicity a segmentation by equally sized partitions, covariance from S-PCA can also be expressed as

$$\mathbf{C}_{S\text{PCA}-n_h} = \frac{1}{N_s} \sum_{n_s=1}^{N_s} \mathbf{a}_{n_s(n_h)}^\top \mathbf{a}_{n_s(n_h)}. \quad (4.12)$$

By paying attention to both expressions in Equation 4.10 and Equation 4.11, it is easy to realise that the new covariance matrix from the folded samples is constructed by the accumulation of the $N_w \times N_w$ portions from the main diagonal of the conventional covariance matrix, the same portions that are achieved, yet computed separately, by S-PCA. This important fact can be schematically represented by Figure 4.3.

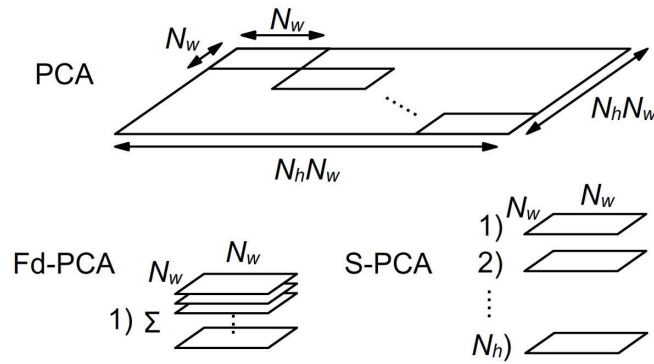


Fig. 4.3 Schematic representation of the covariance matrices

While in the S-PCA method there are as many covariance matrices as band groups or segments in the partition, with subsequent PCA independent implementations for each of them, Fd-PCA combines all the locally extracted covariance portions and, therefore, the following EVD needs to be solved just once, leading to a reduction in computational complexity.

How to select an appropriate configuration, i.e., values for the parameters N_h (or N_w) and F (or F') is important to achieve a good performance from Fd-PCA. In a similar way to S-PCA [62], the parameters selection has to be related to the common distribution of correlation between the spectral bands.

In the HSI cube, contiguous spectral bands present high correlation, which makes it convenient to group these bands into smaller subsets. From [62], it is derived that spectral bands are grouped according to the correlation matrix between each pair of bands. Therefore, a similar criterion must be used when grouping (or folding) the

spectral samples. Even though the number of bands grouped in each segment can vary, Fd-PCA is initially proposed with an even number of bands in all segments, which significantly simplifies the problem but still giving good results. Hence, the number of bands N_w in the N_h segments are selected based on the common or average size from the clusters observed in the main diagonal of the correlation matrix. With relation to the values for F' , they should be a small portion of N_w , i.e., about 10-25%. This is because a large number for F' may lead to noise extraction.

4.3.3 Extension for Uneven Folding

The Fd-PCA proposal has been introduced considering an even number of bands in each segment, so the folding process from vector to matrix is straightforward and sensible for an easier understanding of the method. This can be regarded as a basic configuration of Fd-PCA, already able to provide interesting benefits.

However, Fd-PCA is not limited to the basic case. Actually, the extraction of local structures from the spectral domain may be compromised, as the widths of the different segments are normally different to each other. Therefore, a more general implementation of Fd-PCA, with uneven bands, is possible for higher efficacy in extracting features, which is simply achieved by enlarging every group or segment to match the width of the largest group, N'_w (see Figure 4.4). This enlargement is simply carried out by filling in zeros at the end of the corresponding segment. In that sense, the extended Fd-PCA may lead to a tradeoff between efficiency and efficacy in some data sets, since a larger value for N'_w may translate into a higher potential in extracting local structures but, at the same time, less reduction in computational complexity is achieved.

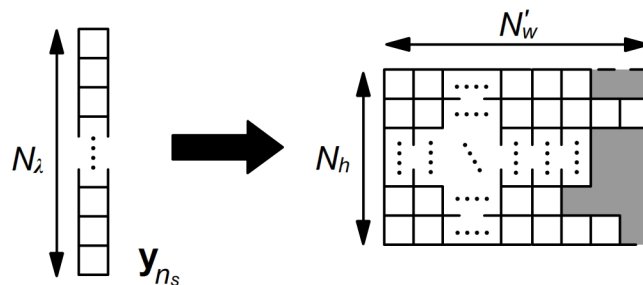


Fig. 4.4 Folding procedure in the extended case

4.4 Analysis and Evaluation

The main objective of the evaluations in the present chapter [13] is to compare the performance of the conventional PCA method to the proposed Fd-PCA one. Their global performance is assessed by several measurements, including the classification Overall Accuracy (OA) in land-cover analysis, the computational cost (number of required Multiply ACcumulates (MACs)), and memory requirements (matrices size).

4.4.1 Experiments

The general workflow of the experiments is described here, according to the stages defined in Section 2.3.1 and including some of the data described in Appendix A. The MATLAB [48] environment is employed for all the experiments.

A. Data Description

The Indian Pines data set is selected for evaluating the Fd-PCA method, along with the Indian Pines B, both from the same sensor [30], as they are considered appropriate hyperspectral images. Further analysis can be found in the publication derived from this work [13], where micro-doppler radar data [126] is also included for military remote sensing applications. The data sets selected for evaluations in this chapter are:

1. Indian Pines [36] (Appendix A.1).
2. Indian Pines B [127] (Appendix A.2).

B. Data Conditioning

Both Indian Pines and Indian Pines B are employed with the resulting 200 spectral bands after the removal process indicated in Appendix A, where all labelled classes in the ground truth (16 and 20, respectively) are included for classification.

C. Feature Extraction

A total of 3 different feature extraction methods are considered in the evaluations. In the first place, the Baseline case (original N_λ spectral features) is considered to be a useful reference to which compare the rest of methods. Then, the proposed Fd-PCA

method is compared to conventional PCA and also to S-PCA, a state-of-the-art method in this context, leading to unavoidable interest in comparisons. The main configuration parameters implemented are $N_h = 10$ and $N_w = 20$, determined by the criterion mentioned in Section 4.3.2 and also empirical evaluations [13]. These parameters are employed in both Fd-PCA and S-PCA for fair comparisons. The number of reduced features F is selected from 5 to 30, in steps of 5 [41].

D. Classification

The powerful Support Vector Machine (SVM) is employed as a classifier, where the multiclass library BSVM [44] is selected in this case (percentage of 30% for training) with general design of experiments as described in Section 2.3.2.

4.4.2 Results

Comprehensive analysis and results are obtained to evaluate the Fd-PCA method. The results are divided into different parts, related to (A) configuration of the Fd-PCA method, (B) classification accuracy, (C) memory requirements, (D) computational complexity, and (E) classification accuracy for uneven folding. These are described in the following.

A. Configuration for Fd-PCA

As described in Section 4.3.2, the final performance obtained from the Fd-PCA method depends on the number of segments to define, i.e., the configuration of parameters N_h and N_w . To illustrate this fact, Figure 4.5 presents the classification accuracy (OA) for different values of N_h reducing the data to $F = 40$ features.

These configuration results evaluate the Fd-PCA performance for different values of N_h (1, 2, 5, 10, 20, and 40) that lead to an equal number of bands in each segment, given that $N_\lambda = N_h N_w$. From the trend in Figure 4.5 is clear that the classification accuracy has its maximum peak when $N_h = 10$, with the rest of values degrading the accuracy when they approach 1 or become larger than 40. Actually, the case with $N_h = 10$ seems to fulfil the recommendation introduced in Section 4.3.2 about general distribution of spectral correlation and, therefore, this is the configuration selected in

the following experiments. In addition, some other considerations must be highlighted, such as the accuracy degrading to exactly the same value of conventional PCA when $N_h = 1$, and the general improvement in all the values implemented for N_h , even for few (2) or many (40) segments in the folded matrix.

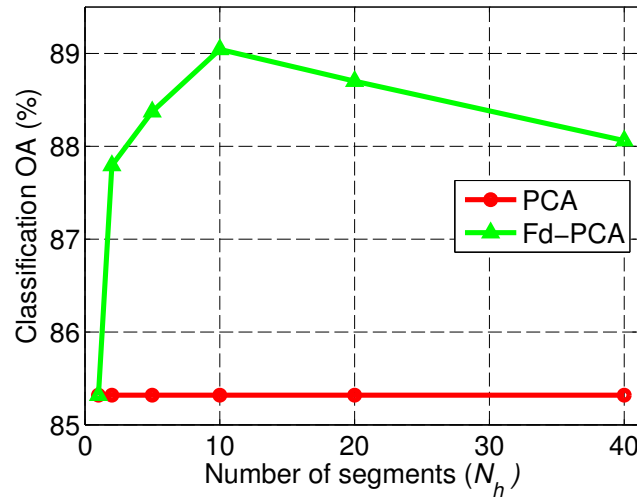


Fig. 4.5 Mean OA for Indian Pines ($F = 40$, $N_h = 1$ to 40)

B. Classification Accuracy

Fd-PCA is compared to the Baseline case, the conventional PCA, and also the S-PCA method in terms of classification OA, as shown in Figure 4.6 and Figure 4.7, depicting the global trend and behaviour of accuracy with different number of features F . In general, with more features extracted in the reduction, more information and better accuracy in the classification tasks is expected, where the Baseline case is not dependent on F , as it employs the original spectral profiles.

Fd-PCA outperforms PCA when $F < 100$ approximately, which is 50% of the original dimensionality of features. In addition, for $F = 20$ to 80, Fd-PCA provides higher accuracy than both Baseline and S-PCA. With relation to the general behaviour, for conventional PCA, as long as more features are included, better classification accuracy is achieved, where the trend is monotonic approaching the Baseline reference. However, the Fd-PCA can generate improved accuracy even for a reduced number of features, due to the local spectral information extracted. With more features included, the accuracy improves until a given limit, degrading then in a nonmonotonic behaviour, due to the noise contained in the less significant features added.

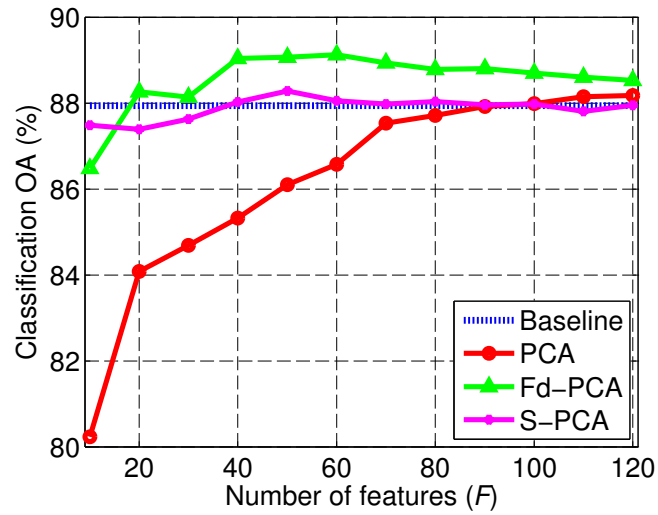


Fig. 4.6 Mean OA for Indian Pines ($F = 10$ to 120)

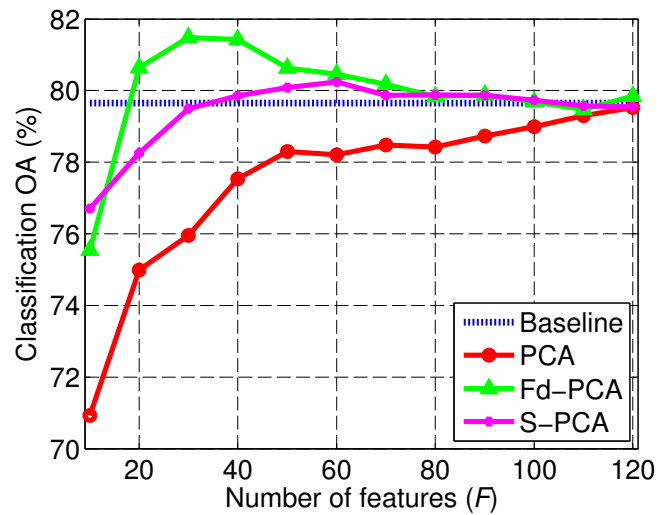


Fig. 4.7 Mean OA for Indian Pines B ($F = 10$ to 120)

C. Memory Requirements

With relation to contiguous memory needs, Table 4.1 shows the dimension of the main matrices involved in the 3 PCA steps, i.e., data, covariance, and projection matrices. Minimum savings achieved by Fd-PCA are about N_h^2 , yet the covariance matrix dimension presents a saving factor of N_s , which makes Fd-PCA better than the S-PCA method, given that $N_s \gg N_h$.

Indeed, a particular configuration with $N_h = 10$ requires less than 1% of memory in comparison to the conventional PCA method, and different configurations can be implemented depending on the case and objective.

Table 4.1 Matrices dimension for (Fd/S)-PCA computations

Method \ Step	Data matrix	Covariance matrix	Projection matrix
PCA	$N_s \times N_h N_w$	$N_h N_w \times N_h N_w$	$N_h N_w \times F$
Fd-PCA	$N_h \times N_w$	$N_w \times N_w$	$N_w \times F/N_h$
Saving factor	N_s	N_h^2	N_h^2
S-PCA	$N_s \times N_w$	$N_w \times N_w$	$N_w \times F/N_h$
Saving factor	N_h	N_h^2	N_h^2

D. Computational Complexity

The complexity associated with the PCA approaches, including conventional, Fd-PCA, and S-PCA, is stated in Table 4.2, being expressed in terms of MACs for the 3 PCA steps, i.e., covariance matrix computation, EVD, and data projection. The global saving factor achieved by Fd-PCA is approximately N_h , i.e., the height of the folded matrix, being slightly higher than the saving from S-PCA. Expressing the MACs in numbers for the particular case with $N_h = 10$ and $F = 30$ (Table 4.3), the computational complexity is reduced to approximately 10% for both data sets, which means an order of magnitude less.

Table 4.2 Computational complexity (MACs) in (Fd/S)-PCA

Method \ Step	Covariance computation	EVD	Data projection
PCA	$N_s N_h^2 N_w^2$	$N_h^3 N_w^3$	$N_s N_h N_w F$
Fd-PCA	$N_s N_h N_w^2$	N_w^3	$N_s N_w F$
Saving factor	N_h	N_h^3	N_h
S-PCA	$N_s N_h N_w^2$	$N_h N_w^3$	$N_s N_w F$
Saving factor	N_h	N_h^2	N_h

Table 4.3 Number of MACs in (Fd/S)-PCA ($F = 30$)

Method \ Data set	Indian Pines	Indian Pines B
PCA	9.752e8	1.043e9
Fd-PCA	9.672e7	1.035e8
Saving factor	10.0827	10.0773
S-PCA	9.679e7	1.036e8
Saving factor	10.0754	10.0676

E. Classification Accuracy for Uneven Folding

As a final evaluation in the experiments, the Fd-PCA method is now implemented for the general case in which the N_h segmented regions in the folding procedure are not the same width (see Section 4.3.3). The new configuration still presents $N_h = 10$ segments but instead of using a common $N_w = 20$, it employs an uneven distribution $N_{w(n_h)} = (15, 21, 24, 16, 13, 13, 21, 21, 28, 28)$, following correlation criteria, with $N'_w = \max\{N_{w(n_h)}\} = 28$.

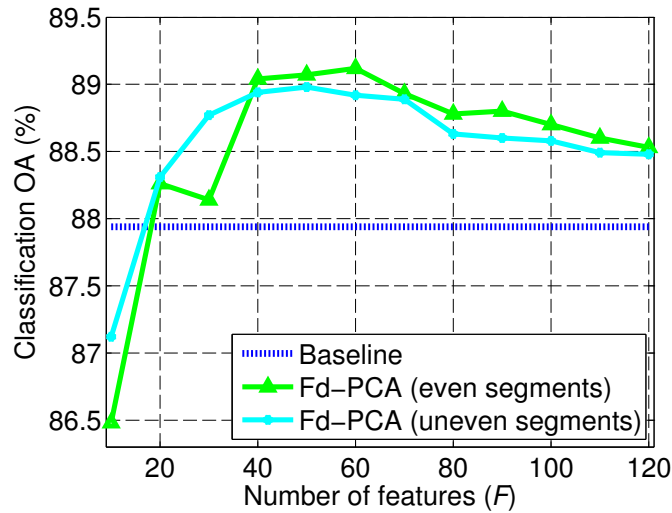


Fig. 4.8 Mean OA in the extended case for Indian Pines ($F = 10$ to 120)

As shown in Figure 4.8, the new configuration is able to provide good classification accuracy, similar to the basic configuration with some cases ($F = 10$ and $F = 30$) leading to slightly better results.

4.5 Summary

Large data such as HSI images in remote sensing require the application of feature extraction and data reduction methods, where PCA is probably the most widely used method for that purpose. However, the conventional PCA method comprises 3 main drawbacks, i.e., high computational cost, excessive memory requirements, and poor efficacy when dealing with HSI data.

In order to solve these disadvantages, a folding procedure is introduced in the implementation of PCA. With appropriate folding in the samples or pixels, the new co-

variance matrix achieved makes Fd-PCA able to extract local information from the spectral domain, providing a real added value to the classification tasks. The improvement in classification accuracy seems related to the extraction of local information from the spectral covariance matrix, which provides features with higher discrimination ability to the classifier models. Additionally, the computational complexity is reduced by an order of magnitude, and the memory requirements are relieved as well.

Fd-PCA proves to beat conventional PCA and S-PCA, a similar variant, being also able to surpass the accuracy obtained from the original spectral profiles, yet Fd-PCA employs a much reduced dimensionality of features F . In summary, Fd-PCA is clearly a potential method for efficient and effective data reduction in hyperspectral remote sensing.

Chapter 5

Spectral Extraction and Denoising by 1D-SSA

In this chapter, a recent technique for time series analysis and forecasting, Singular Spectrum Analysis (SSA), is introduced to hyperspectral remote sensing for effective feature extraction in the spectral domain (1D-SSA). This general algorithm has been previously applied in several fields including climatic, meteorological, and geophysical applications as well as generic data mining in social science [128], among others, reporting increasing interest.

The key point in 1D-SSA is that an original signal can be decomposed into several components such as varying trends, oscillations, and noise [14]. Therefore, the application of 1D-SSA to the spectral profiles from the hyperspectral cubes can decompose original pixels for a later reconstruction in which, removing noisy components, the discrimination capability of the resulting features (reconstructed pixels) is much improved, leading to increased classification accuracy in land-cover analysis.

Although 1D-SSA improves the efficacy of feature extraction, its computational complexity is considerable as in the pixel-based implementation, an individual SingularValue Decomposition (SVD) analysis is needed for each pixel. To solve this drawback, a fast implementation is proposed. This implementation, namely Fast-1D-SSA (F-1D-SSA), only requires a unique SVD applied to a representative pixel, selected as the mean or the median pixel from the whole hyperspectral cube. The output from this SVD is a single transform matrix for all samples, leading to similar accuracy values in classification tasks but with reduced complexity in extracting the features.

In the present chapter, after a background introduction in Section 5.1, 1D-SSA and F-1D-SSA methods are discussed in Section 5.2 and Section 5.3, respectively. Experimental evaluations and analysis are presented in Section 5.4, with a summary given in Section 5.5. This research on the proposed 1D-SSA method and its fast implementation has been published in 2 IEEE journal papers, referenced as [15, 16].

5.1 Introduction

The powerful capabilities of HyperSpectral Imaging (HSI) are based on the hundreds of spectral bands or features available in each pixel or sample. However, HSI data is also prone to noise, which on the contrary, can reduce the discrimination ability and the classification accuracy. To this end, it is desirable to decompose pixels in the spectral domain such that noise can be removed or mitigated. In this decomposition context, an inspiring piece of research is found in [18], where the use of the Empirical Mode Decomposition (EMD) method on the spectral pixels is briefly evaluated.

As the main part of the Hilbert Huang Transform (HHT), an algorithm employed for nonlinear and nonstationary data analysis [90, 91, 129], the EMD method is able to decompose 1-D signals into a few components namely Intrinsic Mode Functions (IMFs) for a posterior reconstruction by only specific IMFs, being used in signal processing tasks such as speech recognition [92]. A proper reconstruction of the original 1-D signal by using a few IMFs provides new features from which enhanced properties are expected. However, a brief application of 1D-EMD in [18] showed not very much improvement at all, but actually a deterioration of classification accuracy.

In this chapter, the application of 1D-SSA, unlike 1D-EMD, is proved to lead to increased classification accuracy, as the spectral pixels and corresponding features are enhanced, becoming a promising methodology for feature extraction in HSI. Additionally, as 1D-SSA requires a pixel-based implementation with considerable computational complexity, a fast implementation of SSA in HSI (F-1D-SSA) is proposed, only requiring a single SVD computation applied to a representative pixel from the whole cube. This fast implementation provides similar good results as 1D-SSA but with reduced complexity, as reported by the comprehensive evaluations in Section 5.4.

5.2 The 1D-SSA Method in HSI

The background of the SSA algorithm is associated with time series analysis and forecasting, including diverse applications from social science or market research to generic data mining. Its origins are related to the former Soviet Union, apparently during the 80s. Based on earlier works [130, 131], and with subsequent publications in the last 20 years, SSA has provided important research attracting attention in recent years [14, 93, 128], including preliminary but insufficient analysis in HSI [94].

In the present section, the SSA concept including mathematical description and related capabilities is briefly introduced. Furthermore, an application example in HSI is detailed for a better understanding.

5.2.1 Algorithm Description

Being based on SVD, the main purpose of SSA is to decompose an original series into several independent components or subseries, where each of them is related to the eigenvalues from the SVD analysis. These individual components can be grouped together to produce others, basically interpretable as varying trends, oscillations, or noise. According to this, the main capabilities of SSA can be summarised as [14]:

1. Extraction of trends and smoothing.
2. Extraction of periodic components.
3. Complex trends and periodicities with varying amplitude.
4. Finding structures in short time series.
5. Envelopes of oscillating signals.

These potential abilities make SSA a promising method, however, how to handle SSA components, including those related to noise, has not yet been properly assessed for hyperspectral remote sensing. Mathematical formulation of the algorithm for application in HSI is described in the following.

Given a pixel $\mathbf{p} = [p_1, p_2, \dots, p_{N_\lambda}]^T \in \mathbb{R}^{N_\lambda}$ from the HSI cube, which is actually a 1-D signal, it can be transformed by 1D-SSA according to the next steps.

A. Embedding

Selecting a window with size $L \in \mathbb{Z}$ given $L \in [1, N_\lambda]$, a trajectory matrix \mathbf{X} from the pixel \mathbf{p} is constructed as in Equation 5.1, having in each column a lagged vector \mathbf{cv}_k expressed as $\mathbf{cv}_k = [p_k, p_{k+1}, \dots, p_{k+L-1}]^\top \in \mathbb{R}^L$, where $k \in [1, K]$ and $K = N_\lambda - L + 1$.

$$\mathbf{X} = \begin{bmatrix} p_1 & p_2 & \cdots & p_K \\ p_2 & p_3 & \cdots & p_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ p_L & p_{L+1} & \cdots & p_{N_\lambda} \end{bmatrix} = [\mathbf{cv}_1, \mathbf{cv}_2, \dots, \mathbf{cv}_K]. \quad (5.1)$$

Trajectory matrix \mathbf{X} is Hankel type, as its antidiagonals have identical values. Moreover, the SSA algorithm can be implemented symmetrically in 2 intervals, thanks to the properties of \mathbf{X} [14]. These intervals are defined by $L \in [1, \text{round}\{N_\lambda/2\}]$ and $L \in [\text{ceil}\{(N_\lambda + 1)/2\}, N_\lambda]$, where $\text{round}\{\}$ and $\text{ceil}\{\}$ are operators representing the rounding and ceiling functions in computer science. Therefore, for a given L , an equivalent implementation can be achieved for another $L' = K$, with exactly the same output. Finally, an important remark is that selection of the extremes from the interval $L \in [1, N_\lambda]$, i.e., $L = 1$ or $L = N_\lambda$, results in the same original pixel.

B. SVD

The SVD of trajectory matrix \mathbf{X} can be expressed as in Equation 5.2, where even though $L_{rank} \leq L$ equals to the rank of \mathbf{X} and, therefore, the number of available components, $L_{rank} = L$ is considered for simplicity

$$\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2 + \cdots + \mathbf{X}_L. \quad (5.2)$$

The SVD of \mathbf{X} is actually equivalent to the EigenValue Decomposition (EVD) of the resulting matrix from $\mathbf{X}\mathbf{X}^\top$, with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L$ and corresponding eigenvectors $[\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_L]$. Following the SVD (or equivalent EVD), matrix \mathbf{X} (and in last terms the pixel \mathbf{p}) is decomposed into several matrices $\mathbf{X}_l | l \in [1, L]$, each of them known as elementary matrix, related to its corresponding eigenvalue as

$$\mathbf{X}_l = \sqrt{\lambda_l} \mathbf{u}_l \mathbf{v}_l^\top, \quad \mathbf{v}_l = \frac{\mathbf{X}^\top \mathbf{u}_l}{\sqrt{\lambda_l}}. \quad (5.3)$$

The collection $(\sqrt{\lambda_l}, \mathbf{u}_l, \mathbf{v}_l)$ is usually called the l^{th} eigentriple, where matrices \mathbf{U} and \mathbf{V} in Equation 5.4 are known as matrix of empirical orthogonal functions and matrix of principal components, respectively,

$$\begin{aligned} \mathbf{U} &= [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_L] \in \mathbb{R}^{L \times L} \\ \mathbf{V} &= [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L] \in \mathbb{R}^{K \times L} \end{aligned} \quad (5.4)$$

C. Grouping

This particular step refers to the selection and grouping of all or some of the L components obtained from the SVD. In general terms, the set of L components is divided into M disjointed sets $\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_M$, where $\sum |\mathbf{t}_m| = L$ and $m \in [1, M]$. Then, defining one of the divided sets as $\mathbf{t} = (l_1, l_2, \dots, l_T)$, the corresponding matrix $\mathbf{X}_{\mathbf{t}}$ from the group \mathbf{t} can be expressed as $\mathbf{X}_{\mathbf{t}} = \mathbf{X}_{l_1} + \mathbf{X}_{l_2} + \dots + \mathbf{X}_{l_T}$. Therefore, the original trajectory matrix \mathbf{X} can also be expressed in grouping terms as

$$\mathbf{X} = \mathbf{X}_{\mathbf{t}_1} + \mathbf{X}_{\mathbf{t}_2} + \dots + \mathbf{X}_{\mathbf{t}_M}. \quad (5.5)$$

For easy understanding, expression in Equation 5.2 corresponds to a basic grouping where $M = L$ and $T = 1$, each set being made by an individual component. Finally, the contribution of a grouping matrix $\mathbf{X}_{\mathbf{t}}$ with relation to the original \mathbf{X} is derived from its related eigenvalues as

$$\eta_{\mathbf{t}} = \sum_{l \in \mathbf{t}} \lambda_l \bigg/ \sum_{l=1}^L \lambda_l. \quad (5.6)$$

D. Diagonal Averaging

Resulting matrices $\mathbf{X}_{\mathbf{t}_m} | m \in [1, M]$ from the grouping step are not necessarily Hankel type as the original trajectory matrix. Nevertheless, they have to be so in order to project them into 1-D signals.

This hankelisation procedure is carried out by the average in the antidiagonals of $\mathbf{X}_{\mathbf{t}_m}$, as these values contribute to the same element in the derived 1-D vector. Denoting $\mathbf{z}_m = [z_{m(1)}, z_{m(2)}, \dots, z_{m(N_\lambda)}]^\top \in \mathbb{R}^{N_\lambda}$ as the 1-D signal from $\mathbf{X}_{\mathbf{t}_m}$, it can be obtained by the expression in Equation 5.7, where $x_{(i,j)}$ are the elements in $\mathbf{X}_{\mathbf{t}_m}$

$$z_{m(n_\lambda)} = \begin{cases} \frac{1}{n_\lambda} \sum_{i=1}^{n_\lambda} x(i, n_\lambda - i + 1) & 1 \leq n_\lambda \leq L \\ \frac{1}{L} \sum_{i=1}^L x(i, n_\lambda - i + 1) & L < n_\lambda < K \\ \frac{1}{N_\lambda - n_\lambda + 1} \sum_{i=n_\lambda - K + 1}^L x(i, n_\lambda - i + 1) & K \leq n_\lambda \leq N_\lambda \end{cases}. \quad (5.7)$$

Repeating this in every matrix \mathbf{X}_{t_m} , the original pixel \mathbf{p} is expressed as

$$\mathbf{p} = \mathbf{z}_1 + \mathbf{z}_2 + \cdots + \mathbf{z}_M = \sum_{m=1}^M \mathbf{z}_m. \quad (5.8)$$

In conclusion, original \mathbf{p} can also be reconstructed using only few specific components depending on the reconstruction purpose, where discarding components from small eigenvalues in HSI can lead to the avoidance of noise.

5.2.2 Application Example in HSI

An original pixel or spectral profile in HSI can be decomposed and later be reconstructed by 1D-SSA. A specific reconstruction based on the main eigenvalue components from the SVD, while discarding less representative or noisy ones, leads to an improved profile or features. In other words, it is feasible to replace the original spectral signature from a pixel by an enhanced profile providing improved discrimination ability for classification tasks.

How to determine an appropriate 1D-SSA-reconstructed profile is dependent on 2 key parameters. The first parameter is the size L of the embedding window, which states how many components are extracted during the SVD decomposition. For example, given a window size $L = 10$, a total of 10 components are obtained, one per each eigenvalue from the SVD. According to the SVD, components related to largest eigenvalues are much more significant than those related to the rest. Therefore, the grouping or addition of main components provides a significant reconstruction.

The second parameter, called EigenValue Grouping (EVG), refers to the combination of extracted components that is selected for the reconstruction. For example, when the EVG includes all components, the reconstruction is exactly the same as the original profile. On the other hand, if the components from the smallest eigenvalues

are discarded by the EVG, then the resulting profile is probably less noisy, since noise is usually located in the small eigenvalue components. Finally, when the EVG avoids the component from the largest eigenvalue, the reconstruction is not adequate since it lacks the main information from the original signal.

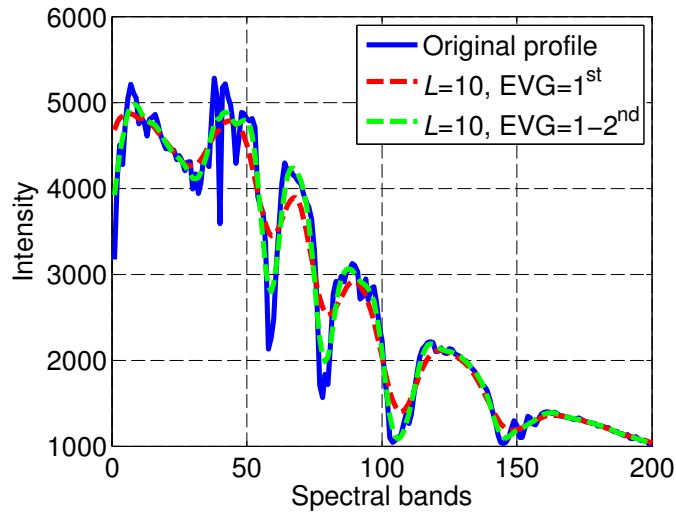


Fig. 5.1 Original and 1D-SSA-reconstructed pixels

An example of 1D-SSA application to a HSI pixel is provided in Figure 5.1, where an original spectral pixel from the 92AV3C data set [132] and reconstructed ones using parameter $L = 10$ with $\text{EVG}=1^{\text{st}}$ and $\text{EVG}=1-2^{\text{nd}}$ are plotted. The new profiles preserve the basic trend of the original signal, as the largest components are included in the reconstruction. Additionally, by avoiding the less representative components, noise content is potentially reduced, with subsequent better feature extraction and classification accuracy, as shown in Section 5.4.2.

5.3 Fast Implementation of 1D-SSA (F-1D-SSA)

The implementation of 1D-SSA in HSI is applied in pixel-based terms, in other words, 1D-SSA is individually applied to all the pixels involved in the classification tasks. Since the spatial size of HSI cubes in remote sensing can easily achieve tens of thousands, this implies extremely large number of SVD computations, the main step involved in the SSA algorithm.

Even though enhanced features and better classification accuracy values are expected from the 1D-SSA method, thus introducing added value to the data analysis, the high computational complexity derived from the pixel-based SVD requirement is a considerable drawback to be addressed. To this end, a fast implementation based on a unique SVD implementation is proposed to further prove the potential of SSA in HSI.

5.3.1 Concept

The proposed application of 1D-SSA to HSI data works individually in every pixel. Nevertheless, it still employs the same configuration parameters for all transformed pixels. Therefore, not only the selected components in EVG but also the window size L is common to all the cases. This consideration allows the F-1D-SSA method.

As the pixels are acquired by the same sensor, presenting same dimensionality of features (N_λ), and the embedding process is equal to all of them, the assembly of lagged vectors in the trajectory matrix (see Section 5.2.1) structure is as well the same. This common embedding process, applied before the SVD step, leads to similar transformation matrices (eigenvectors) for every pixel, so eventually a unique matrix can be commonly applied to all of them. Additionally, the distribution of general, system, and environmental noise tends to be consistent (even other aspects, such as water absorption regions) in pixels from the same hyperspectral cube, acquired under the same conditions. In consequence, a single set of eigenvectors can perfectly project the spectral samples into reconstructed ones where noise is mitigated.

Therefore, due to the common embedding procedure applied to all pixels, the orthonormal basis derived from a single SVD is able to transform all the spectral profiles in the same terms, where the SVD is applied to a signal representative of the whole data to transform. A mathematical description of this fast implementation is provided in the remaining part of the present section.

5.3.2 Algorithm Description

Taking a look at Equation 5.2, the SVD step on the trajectory matrix decomposes it into different elements according to the eigenvalues. The elements \mathbf{X}_l are dependent

on \mathbf{v}_l , and are both outputted from the SVD. Then, according to Equation 5.3, it is possible to substitute \mathbf{v}_l in the expression defining \mathbf{X}_l , so actually the elements from the SVD can be defined in terms of the trajectory matrix and eigenvectors \mathbf{u}_l as

$$\mathbf{X}_l = \sqrt{\lambda_l} \mathbf{u}_l \left(\frac{\mathbf{X}^\top \mathbf{u}_l}{\sqrt{\lambda_l}} \right)^\top. \quad (5.9)$$

Just by rearranging some basic SSA formulation in Equation 5.9, on one hand it is proved that the F-1D-SSA is mathematically feasible and, on the other hand, it can be implemented simply by that equation. Therefore, any pixel embedded with the appropriate window size L in a trajectory matrix \mathbf{X} can be subject to an SSA decomposition by means of some predefined eigenvectors \mathbf{u}_l .

This is the central issue allowing the use of a unique set of eigenvectors in order to transform all the pixels from a given hyperspectral cube. In the following, a further description of algorithm formulation is included for clarity.

A. Single SVD Analysis

Once it is clear that using a single set of eigenvectors is possible to apply the 1D-SSA method to all pixels in a given hyperspectral cube, the next question is what signal the unique SVD has now to be applied to.

With relation to this issue, from the literature it is known that the mean and the median computations over sets of pixels have been actively employed in HSI related tasks for feature extraction or data classification [133, 134]. Consequently, it is expected that both mean and median computations can be used here for achieving a representative spectral pixel from the whole data set.

Hence, the representative signal to which the unique SVD is applied is computed as the average (or median) pixel from the total N_s pixels in the hyperspectral cube (see Figure 5.2). Whether the mean or the median pixel from the whole cube, a unique signal is introduced, being considered an appropriate input to the SVD. As already explained, the representative pixel has to be embedded by the same window size L employed in the conventional analysis, leading thus to the representative trajectory matrix \mathbf{X}_{rep} to which apply the single SVD analysis.

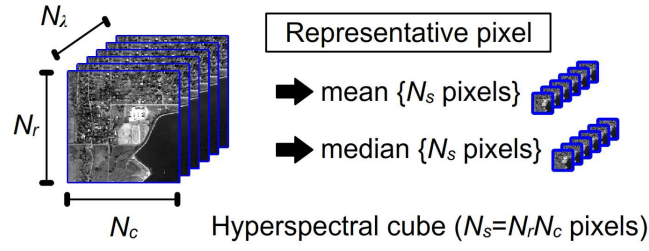


Fig. 5.2 HSI cube and its representative pixels

B. Grouping Considerations

The grouping of components is carried out not necessarily by the strict addition of components individually transformed, but can be implemented by a joint transformation, where denoting \mathbf{t} as the group of selected components, the grouping is achieved through a single multiplication as

$$\mathbf{X}_{\mathbf{t}} = \sqrt{\lambda_{\mathbf{t}}} \mathbf{U}_{\mathbf{t}} \left(\frac{\mathbf{X}^{\top} \mathbf{U}_{\mathbf{t}}}{\sqrt{\lambda_{\mathbf{t}}}} \right)^{\top} = \mathbf{U}_{\mathbf{t}} (\mathbf{U}_{\mathbf{t}}^{\top} \mathbf{X}) \quad (5.10)$$

$$\mathbf{U}_{\mathbf{t}} = [\mathbf{u}_{l_1}, \mathbf{u}_{l_2}, \dots, \mathbf{u}_{l_T}] \in \mathbb{R}^{L \times T}$$

C. Workflow of F-1D-SSA

The schematic workflows of both 1D-SSA and F-1D-SSA are clearly shown in Figure 5.3, for easy comparison highlighting their differences. In F-1D-SSA only the embedding, grouping, and diagonal averaging steps are implemented for all the N_s pixels, as the transformation matrix from the single SVD is commonly employed.

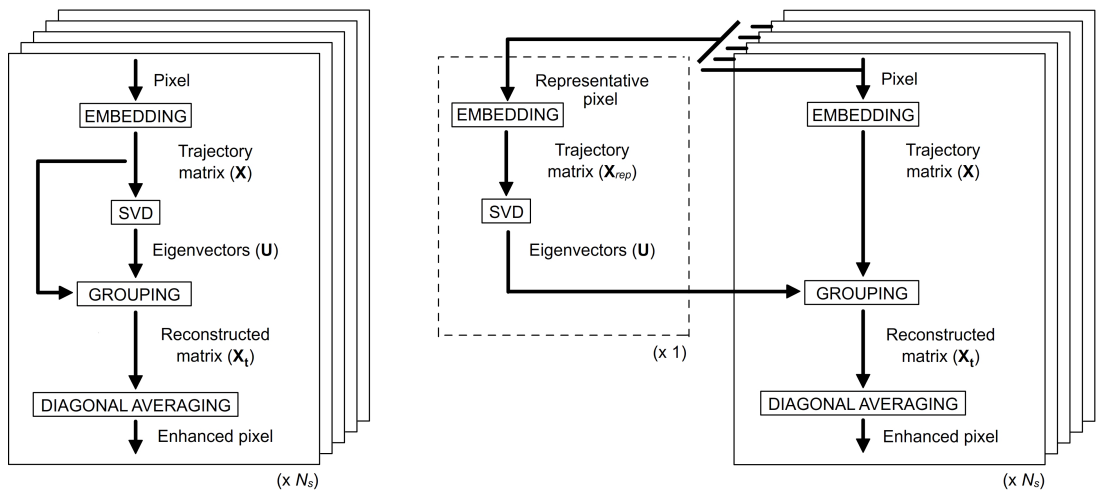


Fig. 5.3 Workflow for (left) 1D-SSA and (right) F-1D-SSA

By a unique SVD applied to a representative pixel, whether it is the mean or the median spectral profile from the whole cube, a remarkable reduction in complexity is achieved in applying F-1D-SSA to HSI data. The corresponding evaluation of computational cost can be found in Section 5.4.2.

5.4 Analysis and Evaluation

Once both 1D-SSA and its fast implementation, F-1D-SSA, are introduced and described, proper analysis is undertaken to prove the enhancement in feature extraction and data classification, including a comparison to other state-of-the-art methods and evaluations on the differences between 1D-SSA and F-1D-SSA [15, 16]. According to the general setup from Section 2.3 and implemented in MATLAB [48], particular description of the experiments in the present research is available in the following.

5.4.1 Experiments

Considerations in the 4 classical stages (see Section 2.3.1) can be found here, where comprehensive description is offered with relation to the feature extraction stage.

A. Data Description

A total of 3 data sets are employed to evaluate the 1D-SSA and F-1D-SSA methods, including scenes from Airborne Visible/InfraRed Imaging Spectrometer (AVIRIS) [30] and Reflective Optics System Imaging Spectrometer (ROSIS) [32]. These are:

1. 92AV3C [132] (Appendix A.1).
2. Salinas C [36] (Appendix A.3).
3. Pavia CA [36] (Appendix A.5).

B. Data Conditioning

Noise and water absorption bands are removed according to the literature as indicated in Appendix A for each data set. Regarding the number of labelled classes, the 2 options in the 92AV3C data set are evaluated, i.e., first the case with the most significant

9 classes, and later the original case with 16 classes. For the other 2 data sets, all available classes are considered in the classification analysis.

C. Feature Extraction

The feature extraction stage for the evaluations on both methods (F)-1D-SSA can be divided into 2 parts for easy understanding and progressive comparisons. Initially, the conventional 1D-SSA is compared to the Baseline case using original features, and the inspiring 1D-EMD method. Later, once the performance and behaviour of 1D-SSA are clear, a further analysis is carried out comparing the fast implementation and other state-of-the-art methods such as Principal Component Analysis (PCA) [116], Independent Component Analysis (ICA) [69], Maximum Noise Fraction (MNF) [71], and Nonnegative Matrix Factorisation (NMF) [73], under different data and conditions.

In the first place, 1D-SSA is compared to the Baseline and the 1D-EMD methods. For this purpose, the 92AV3C and Salinas C data sets are employed both including 9 labelled classes, with 10% of samples for training the classifier models. The 1D-EMD implementation is done by the available code in [135], with the stopping criterion discussed in [129]. Stopping thresholds θ_1 , θ_2 , and α are experimentally determined, satisfying $\theta_2 = 10\theta_1$ as suggested. The IMF Grouping (IMFG) is configured as in [18], where combinations of 1st, the 1-2nd, and the 1-3rd IMFs are selected. On the other hand, the 1D-SSA implementation includes several combinations of window size L (5, 10, 20, and 40) and EVG (1st, 1-2nd, 1-5th, and 1-10th).

In the second place, F-1D-SSA is evaluated by comparisons to classical methods such as PCA, ICA, MNF, and NMF in a similar context, where some experimental setups are changed to provide further added value to the analysis. Therefore, the Salinas C data set is replaced by the urban Pavia CA image, 92AV3C now includes all 16 classes, and the percentage of samples to train the classifier is reduced to 5%, in a more challenging context. The F-1D-SSA method is configured in exactly the same terms as 1D-SSA, for fair comparisons. With relation to the classical methods, the main parameter is simply the number of reduced features (F), where MATLAB provides adequate libraries for PCA, ICA, and NMF, while in the MNF case an implementation based on the Green's method [71] is employed.

The configuration implemented for each method is summarised in Table 5.1. Additionally, the 1D-SSA technique is also combined with PCA in some experiments to prove their compatibility for not only feature extraction but also data reduction.

Table 5.1 Configuration for feature extraction methods in (F)-1D-SSA

Method	Parameters	Values adopted
Baseline	N/A	N/A
PCA	Dimensionality of features (F)	From 5 to original dimensionality in steps of 5 features (best one)
ICA		
MNF		
NMF		
1D-EMD	Thresholds θ_1 , θ_2 , and α IMFG	0.8, 8, and 0.05 1 st , 1-2 nd , 1-3 rd
(F)-1D-SSA	Window size L EVG	5, 10, 20, 40 1 st , 1-2 nd , 1-5 th , 1-10 th

D. Classification

Several classification accuracy values in land-cover analysis, i.e., Overall Accuracy (OA), Class by Class (CbC), and Average Accuracy (AA) values [20, 49] (see Section 2.3.3) are the main performance measurements when comparing the different feature extraction methods, including McNemar's test of significance [50] having Baseline as a reference. Enhanced features are expected to provide better interclass discrimination ability, so the accuracy achieved by the classifier is higher.

The LIBSVM library [45] for Support Vector Machine (SVM) classifier is selected with the Radial Basis Function (RBF) Gaussian kernel, reporting the mean values from 10 independent repetitions in each case. As already mentioned above, 2 different contexts are implemented in the experiments. Initially, the 92AV3C and Salinas C data sets with 10% of samples for training and 9 classes each are employed. Later, 92AV3C includes all the 16 labelled classes, and Salinas C is replaced by the Pavia CA scene (7 classes), both with a reduced 5% of samples for training.

5.4.2 Results

The evaluations provided comprise (A) 1D-SSA comparison to 1D-EMD and the Baseline case, (B) 1D-SSA behaviour analysis, (C) combination of 1D-SSA with posterior

PCA, (D) features comparison between 1D-SSA and F-1D-SSA, (E) comparison to classical state-of-the-art methods, and (F) computational complexity in (F)-1D-SSA. These are explained in the following.

A. Classification Accuracy

The 1D-SSA method is initially compared to the Baseline case for benchmarking reference, and also to the inspiring decomposition method 1D-EMD. In Table 5.2 several results from different configurations prove that the Baseline case can be easily surpassed by 1D-SSA, while the analysis supports that the 1D-EMD method decreases the classification accuracy, as stated in [18].

Table 5.2 Mean OA and McNemar’s test [\mathcal{L}] (9 classes) (10% training)

Method	Parameters	92AV3C	Salinas C
Baseline	N/A	85.59 [-0.00]	98.61 [-0.00]
1D-EMD	IMFG=1 st	56.44 [-44.1]	96.21 [-16.4]
	IMFG=1-2 nd	65.52 [-33.8]	96.25 [-16.2]
	IMFG=1-3 rd	75.49 [-20.1]	97.31 [-11.0]
1D-SSA	$L=5$, EVG=1 st	88.78 [+9.24]	98.76 [+2.03]
	$L=5$, EVG=1-2 nd	88.02 [+7.47]	98.69 [+1.21]
	$L=10$, EVG=1 st	88.68 [+8.87]	98.68 [+0.78]
	$L=10$, EVG=1-2 nd	88.49 [+8.57]	98.76 [+2.00]

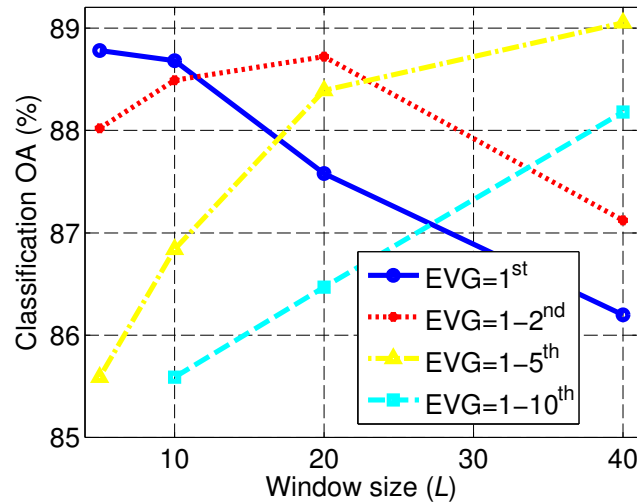
Complementary accuracy measurements are reported in Table 5.3, in this case for the Baseline and a 1D-SSA case ($L = 5$, EVG=1st). From this table, the general improvement in all classes it is clearly seen, regardless of the Number of Samples (NoS).

B. 1D-SSA Behavior

Results for 1D-SSA along all possible configurations described in Table 5.1 are plotted into Figure 5.4 for the 92AV3C data set. From the general behaviour observed (similar to the Salinas C case) a reconstruction based on large EVG for small L results in noise inclusion, which can be understood as a noisy region. On the contrary, having a small EVG with relation to a large L leads to loss of useful information, stated as lossy region. Therefore, an optimum performance is achieved by an intermediate or stable region between the 2 extreme cases.

Table 5.3 Mean OA, CbC, and AA (10% training)

92AV3C				Salinas C			
Class	NoS	Baseline	1D-SSA	Class	NoS	Baseline	1D-SSA
■ (2)	1434	80.71	84.81	■ (1)	240	95.32	95.56
■ (3)	834	72.03	80.99	■ (2)	3400	99.93	99.92
■ (5)	497	89.98	92.73	■ (3)	1957	99.71	99.80
■ (6)	747	97.23	97.77	■ (4)	599	99.13	98.42
■ (8)	489	99.00	99.11	■ (5)	1155	97.77	98.40
■ (10)	968	76.83	83.54	■ (6)	1414	99.99	99.99
■ (11)	2468	83.99	86.11	■ (7)	848	99.62	99.65
■ (12)	614	80.45	84.91	■ (8)	5890	99.23	99.35
■ (14)	1294	98.27	98.41	■ (15)	159	25.52	32.59
AA (%)		86.50	89.82	AA (%)		90.69	91.52
OA (%)		85.59	88.78	OA (%)		98.61	98.76

**Fig. 5.4** Mean OA for 92AV3C (9 classes) (10% training)

These 3 regions described above are indeed clearly identified in Figure 5.4, with noisy ($L = 10$, $\text{EVG}=1\text{-}5^{\text{th}}$), lossy ($L = 40$, $\text{EVG}=1^{\text{st}}$), and stable ($L = 5$, $\text{EVG}=1^{\text{st}}$) regions. In that sense, the tendency for $\text{EVG}=1^{\text{st}}$ is going from the stable to the lossy region when L increases, while the tendency for $\text{EVG}=1\text{-}10^{\text{th}}$ is the opposite, going from noisy to stable, as the selection of 10 components is large and requires big windows for optimum performance. As a concluding remark, it is important to highlight that some configurations ($L = 5$, $\text{EVG}=1\text{-}5^{\text{th}}$) and ($L = 10$, $\text{EVG}=1\text{-}10^{\text{th}}$) achieve the same accuracy as the Baseline case, simply because the selection of all available components leads to the same original features.

C. Combination of 1D-SSA with Posterior PCA

The introduced 1D-SSA method keeps the original dimensionality of features, i.e., the resulting samples contain the same number of features (N_λ), although they are different from the original ones, leading to enhanced profiles and better classification accuracy in land-cover analysis.

In order to prove the potential of 1D-SSA, further data reduction can be achieved from the 1D-SSA method resulting in reduced profiles by using a classical method such as PCA [116]. In Table 5.4, a basic PCA is applied to both Baseline and the 1D-SSA method, reducing the number of features to $F = 15$. Even though the new accuracy values appear slightly worse than those in Table 5.2, this is simply due to the lower feature dimension used, where these results actually support the 1D-SSA method, as significant improvement is achieved again with relation to the Baseline-PCA case, in the data reduction context.

Table 5.4 Mean OA and McNemar’s test [\mathcal{L}] ($F = 15$) (10% training)

Method	Parameters	92AV3C	Salinas C
Baseline-PCA	N/A	84.15 [-0.00]	98.68 [-0.00]
1D-SSA-PCA	$L=5$, EVG=1 st	87.93 [+9.64]	98.92 [+3.06]
	$L=5$, EVG=1-2 nd	86.25 [+5.75]	98.92 [+3.22]
	$L=10$, EVG=1 st	87.89 [+9.50]	98.86 [+2.49]
	$L=10$, EVG=1-2 nd	88.06 [+9.96]	98.96 [+3.73]

D. Comparison of Extracted Features in (F)-1D-SSA

The fast implementation of 1D-SSA is expected to produce similar features and, therefore, similar performance in classification accuracy terms, yet with much more reduced computational complexity in comparison to the conventional implementation.

In the first place, enhanced spectral profiles from conventional 1D-SSA and F-1D-SSA (using the mean computation for obtaining the representative pixel) are visualised in Figure 5.5 for a given configuration $L = 10$, EVG=1-5th, and a randomly selected pixel from the 92AV3C data set [132]. As expected from the explanations in Section 5.3.1, the resulting profiles are almost identical, being difficult to visually differentiate them for both conventional and fast implementation, whether the mean or median scheme is employed in F-1D-SSA.

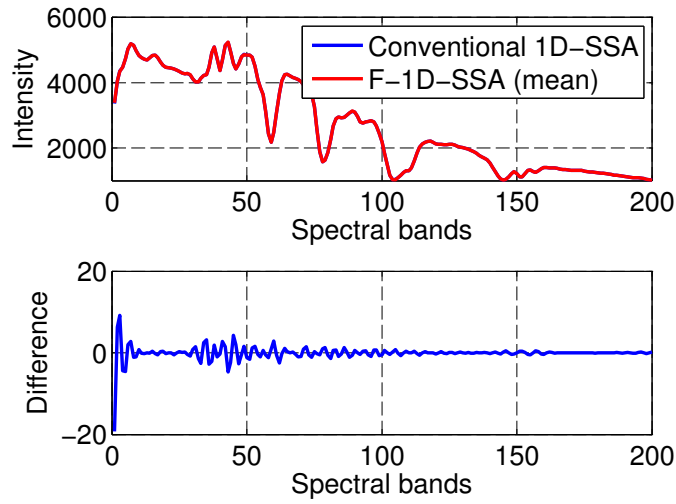


Fig. 5.5 Difference between original and (F)-1D-SSA-reconstructed pixels

Table 5.5 Mean cosine similarity scores in (F)-1D-SSA for 92AV3C

Conventional 1D-SSA				
$L \setminus \text{EVG}$	1st	1-2nd	1-5th	1-10th
5	99.7514	99.9092	100	N/A
10	99.3228	99.7716	99.9367	100
20	98.4455	99.3706	99.8411	99.9408
40	97.5394	98.4087	99.5434	99.8438
F-1D-SSA (mean)				
$L \setminus \text{EVG}$	1st	1-2nd	1-5th	1-10th
5	99.7513	99.9091	100	N/A
10	99.3228	99.7716	99.9365	100
20	98.4472	99.3732	99.8411	99.9410
40	97.5342	98.3275	99.5489	99.8433
F-1D-SSA (median)				
$L \setminus \text{EVG}$	1st	1-2nd	1-5th	1-10th
5	99.7513	99.9091	100	N/A
10	99.3226	99.7716	99.9362	100
20	98.4469	99.3730	99.8411	99.9412
40	97.5356	98.3392	99.5506	99.8434

To further prove the similarity between 1D-SSA and F-1D-SSA enhanced profiles, the well-known cosine distance [136] is introduced as a measurement quantifying the resemblance of (F)-1D-SSA-reconstructed profiles to the original ones. This is done for every individual pixel in the 92AV3C data set [132], reporting the average value from all of them, and computed for all possible configurations, as reconstruction is dependent on the window L and EVG parameters. The cosine similarity scores prove

that no significant difference is found between conventional and fast 1D-SSA implementation (see Table 5.5), as the dissimilarity increases with the window size while it decreases for larger EVG in the same terms for all implementations.

E. Classification Accuracy for (F)-1D-SSA and other Methods

This group of experiments now aims to compare both 1D-SSA [15] and F-1D-SSA [16] including Baseline, 1D-EMD [18], and also classical methods [28, 41] such as PCA, ICA, MNF, and NMF. This time, the 92AV3C data set with all 16 classes and the urban Pavia CA are employed, both with a 5% percentage for training the SVMs.

Firstly, Table 5.6 reports the best OA values obtained by all methods except F-1D-SSA. As can be seen, the 1D-SSA method outperforms the rest simply with the configuration $L = 10$ and $\text{EVG}=1^{\text{st}}$, showing again the efficacy of this method. Indeed, 1D-SSA beats the Baseline accuracy, from 78% to over 82% for the 92AV3C image, and from an already high 97.1% to over 97.35% for the Pavia CA scene. The rest of the methods evaluated are surpassed as well, where the classical ones provide limited accuracy with relation to the SSA methodology. This is also supported by the McNemar’s tests of significance.

Table 5.6 Mean OA and McNemar’s test [\mathcal{Z}] (best OAs)

Method	Parameters	92AV3C	Salinas C
Baseline	N/A	78.07 [-0.00]	97.10 [-0.00]
PCA	$F=15$ and $F=5$	77.01 [-2.36]	97.06 [-0.15]
ICA	$F=20$ and $F=5$	76.90 [-2.61]	96.93 [-0.74]
MNF	$F=10$ and $F=5$	78.03 [-0.13]	97.16 [+0.12]
NMF	$F=70$ and $F=10$	78.58 [+1.28]	97.15 [+0.25]
1D-EMD	IMFG= 1^{st}	48.33 [-47.2]	68.23 [-40.7]
	IMFG= $1\text{-}2^{\text{nd}}$	52.28 [-41.8]	79.55 [-30.7]
	IMFG= $1\text{-}3^{\text{rd}}$	65.40 [-24.2]	90.71 [-16.6]
1D-SSA	$L=10$, $\text{EVG}=1^{\text{st}}$	82.13 [+10.9]	97.35 [+1.19]

Moreover, the classical methods comprise some other disadvantages, for example, ICA and NMF are affected by the initial values in iterations, and MNF is highly dependent on the algorithm used to estimate the noise. On the contrary, the 1D-SSA method is generally reliable, consistent, and leads to better classification accuracy.

Table 5.7 Mean OA and McNemar's test [\mathcal{L}] in (F)-1D-SSA for 92AV3C

Parameters		1D-SSA	F-1D-SSA (mean)	F-1D-SSA (median)
L	EVG			
5	1 st	82.15 [+11.1]	82.12 [+11.2]	82.13 [+11.2]
5	1-2 nd	80.68 [+7.58]	80.54 [+7.17]	80.54 [+7.16]
10	1 st	82.13 [+10.9]	82.19 [+10.9]	82.17 [+11.0]
10	1-2 nd	81.67 [+9.78]	81.94 [+10.8]	82.06 [+11.0]
10	1-5 th	79.68 [+4.85]	79.85 [+5.50]	79.73 [+5.06]
20	1 st	80.82 [+7.40]	80.86 [+7.44]	80.87 [+7.44]
20	1-2 nd	82.15 [+10.9]	82.06 [+10.7]	82.05 [+10.5]
20	1-5 th	81.67 [+9.86]	81.63 [+9.85]	81.49 [+9.51]
20	1-10 th	79.13 [+3.29]	79.47 [+4.39]	79.47 [+4.38]
40	1 st	79.46 [+3.74]	78.61 [+1.48]	78.61 [+1.47]
40	1-2 nd	80.29 [+5.67]	80.64 [+6.90]	80.65 [+6.82]
40	1-5 th	82.56 [+12.0]	82.19 [+11.1]	82.39 [+11.6]
40	1-10 th	81.52 [+9.65]	81.14 [+8.58]	81.15 [+8.59]
Global mean		81.07 [+8.21]	81.02 [+8.15]	81.02 [+8.13]

Table 5.8 Mean OA and McNemar's test [\mathcal{L}] in (F)-1D-SSA for Pavia CA

Parameters		1D-SSA	F-1D-SSA (mean)	F-1D-SSA (median)
L	EVG			
5	1 st	97.16 [+0.41]	97.16 [+0.38]	97.16 [+0.36]
5	1-2 nd	97.00 [-0.23]	97.01 [-0.47]	97.01 [-0.47]
10	1 st	97.35 [+1.19]	97.36 [+1.25]	97.36 [+1.23]
10	1-2 nd	97.30 [+0.99]	97.12 [+0.16]	97.12 [+0.15]
10	1-5 th	97.22 [+0.61]	97.05 [-0.20]	97.05 [-0.25]
20	1 st	97.05 [-0.15]	97.07 [-0.10]	97.06 [-0.12]
20	1-2 nd	97.23 [+0.81]	97.38 [+1.60]	97.33 [+1.24]
20	1-5 th	97.06 [+0.07]	97.01 [-0.41]	96.92 [-0.81]
20	1-10 th	97.10 [+0.04]	97.03 [-0.37]	97.02 [-0.38]
40	1 st	96.79 [-1.85]	96.84 [-1.84]	96.85 [-1.82]
40	1-2 nd	97.19 [+0.51]	97.09 [-0.49]	97.28 [+0.71]
40	1-5 th	97.31 [+1.10]	97.42 [+1.72]	97.40 [+1.64]
40	1-10 th	97.05 [-0.09]	97.24 [+0.89]	97.12 [+0.22]
Global mean		97.14 [+0.26]	97.14 [+0.16]	97.13 [+0.13]

On the other hand, the results from all configurations in (F)-1D-SSA are now reported in Table 5.7 and Table 5.8. From these results, it is clearly proved the similarity of (F)-1D-SSA to each other, whether the mean or the median spectral profile is employed as a representative sample from the hyperspectral cube. Occasionally, some (F)-1D-SSA configurations can degrade the classification accuracy, which indicates

the importance of an appropriate configuration in accordance with the SSA behaviour (see Figure 5.4). Finally, the CbC and AA values are reported in Table 5.9 and Table 5.10 for 92AV3C (16 classes) and Pavia CA, respectively, where similar improvements are achieved in most labelled classes.

Table 5.9 Mean OA, CbC, and AA in (F)-1D-SSA for 92AV3C

Class	NoS	Baseline	1D-SSA	F-1D-SSA (mean)	F-1D-SSA (median)
■ (1)	54	37.84	75.29	75.29	74.71
■ (2)	1434	74.71	81.57	81.67	81.28
■ (3)	834	60.71	69.04	70.03	69.57
■ (4)	234	54.01	65.09	64.59	64.37
■ (5)	497	87.25	89.66	89.56	89.34
■ (6)	747	93.06	93.23	93.30	93.26
■ (7)	26	57.08	82.08	82.08	82.08
■ (8)	489	96.88	96.29	96.42	96.42
■ (9)	20	22.11	44.74	43.68	44.21
■ (10)	968	66.55	72.71	72.76	72.72
■ (11)	2468	81.19	82.92	82.94	83.19
■ (12)	614	68.70	81.87	82.35	82.18
■ (13)	212	95.27	96.22	96.07	96.12
■ (14)	1294	94.71	94.84	94.31	94.62
■ (15)	380	44.68	44.02	43.99	44.27
■ (16)	95	82.89	84.89	85.22	84.89
AA (%)		69.85	78.40	78.39	78.33
OA (%)		78.07	82.13	82.19	82.17

Table 5.10 Mean OA, CbC, and AA in (F)-1D-SSA for Pavia CA

Class	NoS	Baseline	1D-SSA	F-1D-SSA (mean)	F-1D-SSA (median)
■ (1)	447	100	100	100	100
■ (2)	28	23.08	20.77	23.46	23.46
■ (3)	347	87.42	89.18	87.93	87.93
■ (4)	1213	95.54	96.04	96.20	96.19
■ (5)	3512	98.77	98.90	98.93	98.93
■ (6)	893	97.10	97.11	97.23	97.23
■ (9)	43	99.75	100	100	100
AA (%)		85.85	86.00	86.25	86.25
OA (%)		97.10	97.35	97.36	97.36

F. Computational Complexity

From the experimental outcome previously described, it is clear that F-1D-SSA is able to provide similar features to the ones from 1D-SSA, leading to the same improvement in discrimination ability and, therefore, classification accuracy. However, the fast implementation aims to reduce the significant complexity involved in 1D-SSA.

In the F-1D-SSA implementation, the unique SVD computation directly translates into a saving factor of N_s (number of pixels) in the SVD step. Therefore, even though the rest of steps present the same cost, the global complexity of the algorithm is reduced given the considerable complexity associated with SVD. In order to show the benefits from F-1D-SSA, a brief analysis on the 1D-SSA method complexity in accordance with [54] is described as follows.

Firstly, the embedding step simply reallocates a vector array into a matrix form, hence, no Multiply ACcumulates (MACs) are associated with the process. Then, the SVD complexity can vary depending on the implementation proposed. As already mentioned in Section 5.2.1, even though SVD is normally formulated as in [14], it can be easily computed through an equivalent EVD applied to $\mathbf{X}\mathbf{X}^\top$ ($L^2K + L^3$), faster than the SVD complexity ($L^2K + LK^2 + K^3$) suggested by [54, 137]. Thirdly, the grouping step can be implemented by 2 multiplications in Equation 5.10 ($2KLT$), and finally, the diagonal averaging requires N_λ multiplications and LK additions for every pixel, yet it can be more appropriately approximated by a total of N_λ MACs per pixel.

Table 5.11 Computational complexity (MACs) in (F)-1D-SSA

Step	1D-SSA	F-1D-SSA	Saving factor
Embedding	N/A	N/A	1
SVD	$(L^2K + L^3)N_s$	$(L^2K + L^3)1$	N_s
Grouping	$(2KLT)N_s$	$(2KLT)N_s$	1
Diagonal Averaging	$N_\lambda N_s$	$N_\lambda N_s$	1

Regarding Table 5.11, the different SSA implementations share the complexity from 3 steps, while the SVD is reduced by a saving factor equal to the number of pixels transformed. Putting some values to these expressions, the number of MACs required under different configurations is compared for 92AV3C and Pavia CA in Table 5.12. As expected, saving factors of 3-4 validate the fast implementation. Additionally,

it is worth remarking that the same analysis with the SVD complexity suggested by [54, 137] results in much higher saving factors, achieving reductions of even hundreds.

Table 5.12 Number of MACs in (F)-1D-SSA

<i>L</i>	92AV3C		Pavia CA	
	5	40	5	40
EVG	1st	1-10th	1st	1-10th
1D-SSA	1.51e8	9.47e9	8.23e7	4.84e9
F-1D-SSA	4.50e7	2.71e9	2.43e7	1.14e9
Saving factor	3.33	3.49	3.38	4.24

5.5 Summary

The SSA algorithm, being recently evaluated in several and diverse applications, is able to decompose a 1-D signal into trends, oscillation components, or noise, among others, which involves great potential in signal processing. Based on the well-known SVD, 1D-SSA allows the decomposition of HSI pixels, where extracted components can be employed for improved reconstructions with noise content suppressed. An evaluation of the 1D-SSA behaviour derived from different configurations is carried out, by which it is feasible to suggest recommendations, as 1D-SSA seems to perform according to 3 different regions, i.e., noisy, stable, and lossy.

Nevertheless, the 1D-SSA method requires an individual SVD for each pixel, leading to considerable complexity. In order to solve this drawback, a fast implementation F-1D-SSA is proposed, in which only a single SVD computation is required. Comprehensive evaluations conclude that the fast implementation provides almost identical features and classification accuracy, while the computational complexity has been dramatically reduced.

Therefore, the (F)-1D-SSA methods are found to be effective in decomposing HSI pixels in the spectral domain, avoiding noisy components and achieving higher classification accuracy in land-cover analysis. (F)-1D-SSA beats several current state-of-the-art methods such as PCA, ICA, MNF, and NMF, proving to be more effective and reliable, with further possibilities still to be explored.

Chapter 6

Spatial Extraction and Denoising by 2D-SSA

Although the 1-D Singular Spectrum Analysis (1D-SSA) method presented in Chapter 5 is able to remove the noise from spectral profiles and increment the classification accuracy in hyperspectral remote sensing, it only exploits the spectral-domain information from hyperspectral cubes. As suggested in the literature that the use of 2-D spatial information can also increase the performance of features, the introduction of 2D-SSA in HyperSpectral Imaging (HSI) becomes a natural step of extension.

The 2D-SSA method is able to decompose 2-D scenes into several components for a later reconstruction with local structure extracted and noise highly mitigated, leading to outstanding classification accuracy in hyperspectral remote sensing. Being benchmarked against several state-of-the-art methods, and especially 2-D Empirical Mode Decomposition (2D-EMD) [18], 2D-SSA is proved to achieve the best results in most cases, only comparable to the 2D-EMD method. However, while 2D-EMD requires an iterative procedure in its implementation, 2D-SSA is based on the well-known Singular Value Decomposition (SVD), demanding less complexity and resulting in faster computation time for extracting the features.

Moreover, similarly to the 1-D case [16], a fast implementation Fast-2D-SSA (F-2D-SSA) is introduced, where a unique SVD is required, being applied to a representative spectral scene from the whole cube. F-2D-SSA is able to dramatically reduce the computational complexity of 2D-SSA, including important reduction in computation time, leading to potential applications in portable and embedded systems.

The organisation of this chapter is described as follows. The background of the 2D-SSA context is given in Section 6.1, with description in Section 6.2. Then, the F-2D-SSA method is introduced in Section 6.3, while Section 6.4 and Section 6.5 offer further discussions and a summary of the contributions, respectively. It is worth noting that the work on 2D-SSA has been published in IEEE Transactions on Geoscience and Remote Sensing [17], where its fast implementation is also under consideration by another journal at the moment this thesis is submitted.

6.1 Introduction

As every pixel in HSI forms a spectral vector, pixel-based analysis [29, 35, 37, 41] is broadly employed for remote sensing Earth observation. The pixels or samples can be certainly characterised by their spectral profile, endowing the classifier with hundreds of features for enhanced discrimination analysis.

However, the limited contribution from the spectral-domain exploitation has been remarked in recent years, especially when it is compared to new methodologies exploiting the spatial domain, normally achieving improved accuracy. Feature extraction methods considering the spatial domain have been introduced with significant results reported, including the Morphological Profile (MP) [103–105] or the Adaptive Filter with Derivative (AFD) [106] techniques, among others. In this context, an inspiring research is derived from the 2D-EMD method [18], with excellent classification accuracy but extremely high complexity, as it is based on empirical iterations.

Following the success of 1D-SSA [15, 16] and inspired by the 2D-EMD [18] research, the 2D-SSA method is introduced to hyperspectral remote sensing, covering the spatial-domain exploitation suggested in the literature. The 2D-SSA method is expected to provide high classification accuracy similar to the 2D-EMD case, but requiring much lower computational complexity and faster computation time for extracting the features, as 2D-SSA is based on the well-known SVD while 2D-EMD works with endless and expensive iterations. In addition, this computational difference is especially highlighted when a fast implementation (F-2D-SSA) is employed, similarly to the 1-D case (see Section 5.3).

6.2 The 2D-SSA Method in HSI

The 2D-SSA algorithm [112, 113] is actually an extension from the original SSA [14], with few reported evaluations in different fields [114, 115]. In this chapter, 2D-SSA for effective feature extraction in HSI is introduced, where the 2D-SSA method is detailed with an application example in HSI given to explain how it works.

6.2.1 Algorithm Description

With the same capabilities as described in Section 5.2.1 in discussing 1D-SSA, the extended 2D-SSA method aims to extract spatial information through the main eigenvalues from the SVD computation, while noise, normally found in the small eigenvalues, is avoided or mitigated. Indeed, the 2D-SSA application in HSI is similar to the one for 2D-EMD, as they are individually applied to every spectral band in the hyperspectral cube. One of the main differences is probably that while 2D-SSA works in eigenvalue terms [112], the 2D-EMD components are related to frequency [18], so the noise mitigation in 2D-EMD seems not really feasible, as noise is also contained in the main Intrinsic Mode Function (IMF) components and related IMF Grouping (IMFG).

Similar to the 1-D case, the 2D-SSA algorithm also has 4 different steps, which include (A) embedding a 2-D signal, (B) computing SVD on the embedded signal, (C) grouping selected SVD components for reconstruction, and (D) diagonal averaging to achieve the reconstructed signal. They are described in detail as follows.

A. Embedding a 2-D Image

Given a 2-D image \mathbf{P}^{2D} with dimensions $N_r \times N_c$ expressed as

$$\mathbf{P}^{2D} = \begin{bmatrix} P_{(1,1)} & P_{(1,2)} & \cdots & P_{(1,N_c)} \\ P_{(2,1)} & P_{(2,2)} & \cdots & P_{(2,N_c)} \\ \vdots & \vdots & \ddots & \vdots \\ P_{(N_r,1)} & P_{(N_r,2)} & \cdots & P_{(N_r,N_c)} \end{bmatrix} \in \mathbb{R}^{N_r \times N_c}, \quad (6.1)$$

it is embedded into a 2-D trajectory matrix \mathbf{X}^{2D} by means of a 2-D window with size $L^{2D} = L_r \times L_c$, where $L_r \in [1, N_r]$ and $L_c \in [1, N_c]$ are integer values. This window,

required to construct the 2-D trajectory matrix, is represented as in Equation 6.2, where (i, j) refers to the position of its top-left corner

$$\mathbf{W}_{(i,j)} = \begin{bmatrix} P(i,j) & P(i,j+1) & \cdots & P(i,j+L_c-1) \\ P(i+1,j) & P(i+1,j+1) & \cdots & P(i+1,j+L_c-1) \\ \vdots & \vdots & \ddots & \vdots \\ P(i+L_r-1,j) & P(i+L_r-1,j+1) & \cdots & P(i+L_r-1,j+L_c-1) \end{bmatrix}. \quad (6.2)$$

The window $\mathbf{W}_{(i,j)}$ can also be expressed as in Equation 6.3, for incoming formulation convenience

$$\mathbf{W}_{(i,j)} = \begin{bmatrix} \mathbf{w}_{1(i,j)} \\ \mathbf{w}_{2(i,j)} \\ \vdots \\ \mathbf{w}_{L_r(i,j)} \end{bmatrix} \quad \mathbf{w}_{l(i,j)} = \begin{bmatrix} P(i+l-1,j) \\ P(i+l-1,j+1) \\ \vdots \\ P(i+l-1,j+L_c-1) \end{bmatrix}^\top. \quad (6.3)$$

In order to construct the 2-D trajectory matrix, this window has to be placed all over the image \mathbf{P}^{2D} . Therefore, a raw scanning starting from the top-left to the bottom-right corner of the image, taking as a reference the element in position (i, j) , yields to a total number of $K^{2D} = (N_r - L_r + 1)(N_c - L_c + 1)$ actual positions of the window, where $i \in [1, N_r - L_r + 1]$ and $j \in [1, N_c - L_c + 1]$. All these positioned windows are reallocated into a column vector \mathbf{RW} as

$$\mathbf{RW}_{(i,j)} = \begin{bmatrix} \mathbf{w}_{1(i,j)}^\top \\ \mathbf{w}_{2(i,j)}^\top \\ \vdots \\ \mathbf{w}_{L_r(i,j)}^\top \end{bmatrix} = \begin{bmatrix} P(i,j) \\ P(i,j+1) \\ \vdots \\ P(i,j+L_c-1) \\ P(i+1,j) \\ \vdots \\ P(i+L_r-1,j+L_c-1) \end{bmatrix} \in \mathbb{R}^{L_r L_c \times 1}, \quad (6.4)$$

so the 2-D trajectory matrix $\mathbf{X}^{2D} \in \mathbb{R}^{L^{2D} \times K^{2D}}$ from embedding the image \mathbf{P}^{2D} is finally expressed by

$$\mathbf{X}^{2D} = \begin{bmatrix} \mathbf{R}\mathbf{W}_{(1,1)}^\top \\ \mathbf{R}\mathbf{W}_{(1,2)}^\top \\ \vdots \\ \mathbf{R}\mathbf{W}_{(1,N_c-L_c+1)}^\top \\ \mathbf{R}\mathbf{W}_{(2,1)}^\top \\ \vdots \\ \mathbf{R}\mathbf{W}_{(N_r-L_r+1,N_c-L_c+1)}^\top \end{bmatrix}^\top. \quad (6.5)$$

This 2-D trajectory matrix \mathbf{X}^{2D} , analogously to the 1-D case, has a Hankel-block-Hankel (HbH) structure, which can be seen as

$$\mathbf{X}^{2D} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \cdots & \mathbf{H}_{N_r-L_r+1} \\ \mathbf{H}_2 & \mathbf{H}_3 & \cdots & \mathbf{H}_{N_r-L_r+2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{L_r} & \mathbf{H}_{L_r+1} & \cdots & \mathbf{H}_{N_r} \end{bmatrix}_{L_r \times (N_r-L_r+1)}, \quad (6.6)$$

being every of the submatrices \mathbf{H}_{n_r} Hankel type as expressed in Equation 6.7, thus making the matrix \mathbf{X}^{2D} Hankel type in block terms

$$\mathbf{H}_{n_r} = \begin{bmatrix} P(n_r,1) & P(n_r,2) & \cdots & P(n_r,N_c-L_c+1) \\ P(n_r,2) & P(n_r,3) & \cdots & P(n_r,N_c-L_c+2) \\ \vdots & \vdots & \ddots & \vdots \\ P(n_r,L_c) & P(n_r,L_c+1) & \cdots & P(n_r,N_c) \end{bmatrix}_{L_c \times (N_c-L_c+1)}. \quad (6.7)$$

B. SVD

In the second step, a SVD computation is performed on the matrix \mathbf{X}^{2D} . Indeed, the SVD of \mathbf{X}^{2D} is equivalent to the EigenValue Decomposition (EVD) of the matrix obtained from $\mathbf{X}^{2D}(\mathbf{X}^{2D})^\top$, resulting in $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{L^{2D}}$ eigenvalues and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{L^{2D}}] \in \mathbb{R}^{L^{2D} \times L^{2D}}$ corresponding eigenvectors from \mathbf{X}^{2D} .

According to the SVD (or equivalent EVD), the 2-D trajectory matrix is decomposed into the addition of several components $\mathbf{X}^{2D} = \mathbf{X}_1 + \mathbf{X}_2 + \cdots + \mathbf{X}_{L^{2D}}$, where the matrices $\mathbf{X}_l | l \in [1, L^{2D}]$ are related to the corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{L^{2D}}$, being formulated as

$$\mathbf{X}_l = \sqrt{\lambda_l} \mathbf{u}_l \mathbf{v}_l^T, \quad \mathbf{v}_l = \frac{(\mathbf{X}^{2D})^\top \mathbf{u}_l}{\sqrt{\lambda_l}}. \quad (6.8)$$

C. Grouping

Once the SVD has been applied to the 2-D trajectory matrix, resulting in several components, in the grouping step particular components (\mathbf{X}_l) are selected for reconstruction of the image \mathbf{P}^{2D} being the remaining ones discarded.

Grouping all L^{2D} components into M disjoint sets denoted as $\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_M$, having $\sum |\mathbf{t}_m| = L$ and $m \in [1, M]$, let $\mathbf{t} = (l_1, l_2, \dots, l_T)$ be one of the groups. In that case, the matrix $\mathbf{X}_{\mathbf{t}}^{2D}$ related to the \mathbf{t} grouping is therefore defined as $\mathbf{X}_{\mathbf{t}}^{2D} = \mathbf{X}_{l_1} + \mathbf{X}_{l_2} + \dots + \mathbf{X}_{l_T}$. On the other hand, original matrix \mathbf{X}^{2D} can also be represented as in Equation 6.9 by the addition of all groups if none component is discarded,

$$\mathbf{X}^{2D} = \mathbf{X}_{\mathbf{t}_1} + \mathbf{X}_{\mathbf{t}_2} + \dots + \mathbf{X}_{\mathbf{t}_M}. \quad (6.9)$$

Accordingly, the particular contribution of each grouping matrix $\mathbf{X}_{\mathbf{t}}^{2D}$ related to the matrix \mathbf{X}^{2D} depends on the corresponding eigenvalues, derived as

$$\eta_{\mathbf{t}} = \sum_{l \in \mathbf{t}} \lambda_l / \sum_{l=1}^{L^{2D}} \lambda_l. \quad (6.10)$$

From an implementation point of view, this grouping and selection of components can be carried out by a simple product of 3 matrices. Taking expressions in Equation 6.8, a single group \mathbf{t} containing the selected components can be obtained by

$$\mathbf{X}_{\mathbf{t}}^{2D} = \mathbf{U}_{\mathbf{t}} ((\mathbf{X}^{2D})^\top \mathbf{U}_{\mathbf{t}})^\top = \mathbf{U}_{\mathbf{t}} (\mathbf{U}_{\mathbf{t}}^\top \mathbf{X}^{2D}), \quad (6.11)$$

where $\mathbf{U}_{\mathbf{t}} = [\mathbf{u}_{l_1}, \mathbf{u}_{l_2}, \dots, \mathbf{u}_{l_T}]$ is a matrix whose columns are the eigenvectors related to each of the selected components for reconstruction. As already explained in Section 5.2.2, the parameter used for the selection of components is the EigenValue Grouping (EVG), i.e., $\text{EVG} = l_1 - l_T$. Therefore, the resulting matrix $\mathbf{X}_{\mathbf{t}}^{2D}$ leads to a reconstruction of the original signal \mathbf{P}^{2D} , in which some components can be avoided. On the other hand, a selection including the total number of components simply results in the same original 2-D signal.

D. Diagonal Averaging

Once the grouping matrix \mathbf{X}_t^{2D} is obtained, the inverse procedure to that undertaken in the embedding step needs to be performed. However, in order to go from grouping matrix \mathbf{X}_t^{2D} to final 2-D image, \mathbf{X}_t^{2D} needs to be HbH type. For that reason, a diagonal averaging similar to the one in Section 5.2.1 is applied to every block in Equation 6.7, and then applied among the blocks in Equation 6.6 [112]. This is a simple average procedure along the values contributing to the same element (i, j) in the 2-D image.

Denoting \mathbf{Z}_m^{2D} as the resulting image from a particular eigenvalue component, and M as the total number of components from the SVD, i.e., $M = L^{2D}$, the original image \mathbf{P}^{2D} can be obtained by the addition of all components

$$\mathbf{P}^{2D} = \mathbf{Z}_1^{2D} + \mathbf{Z}_2^{2D} + \dots + \mathbf{Z}_M^{2D} = \sum_{m=1}^M \mathbf{Z}_m^{2D}. \quad (6.12)$$

The application of 2D-SSA in HSI looks for a reconstruction of \mathbf{P}^{2D} in which the main spatial trend and structure is extracted by selecting the main components.

6.2.2 Application Example in HSI

The data acquired in HSI is presented in a 3-D structure known as hyperspectral cube (see Section 2.2), where the same spatial content is provided for hundreds of different spectral bands. Therefore, regarding the application of the 2D-SSA method to HSI cubes, the implementation is carried out in a spectral-based procedure, where every band presents a 2-D image to which 2D-SSA is applied individually.

Given a randomly selected band (at 667 nm) from the 92AV3C data set [132], the spectral 2-D scene can be treated by the 2D-SSA method. The original image and extracted components with $L_r = 5$, $L_c = 5$ ($L^{2D} = 25$) can be found in Figure 6.1, where detailed spatial structure can be perceived.

The reconstruction by few main components leads therefore to extraction of main trend and noise avoidance for a particular scene. Hence, if this reconstruction is applied to every spectral band in the cube, the preserved main trend, local structure, and mitigated noise provide great potential in the resulting features. Similarly to the 1-D case, the efficacy of 2D-SSA is dependent on 2 key parameters, the size of the embed-

ding window (L^{2D}) and the selected components in the reconstruction (EVG), where again the total number of components available from the SVD is equal to the size of the window ($L^{2D} = L_r \times L_c$).

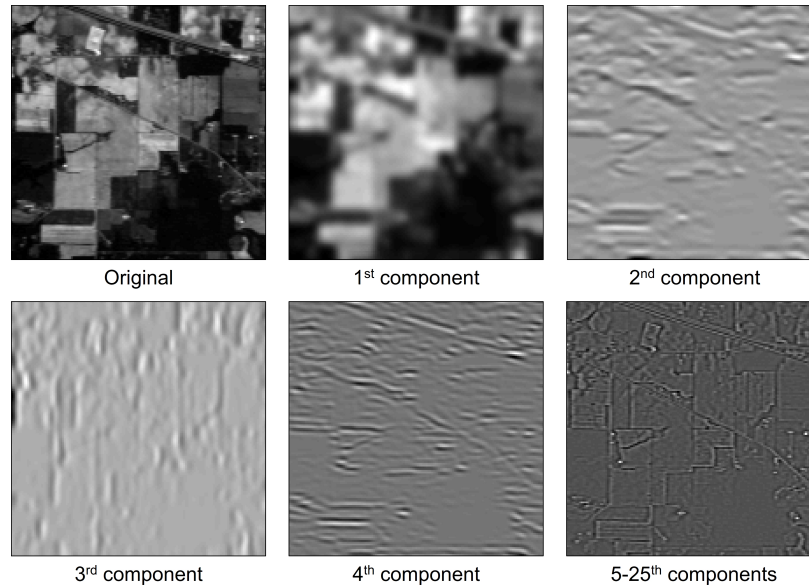


Fig. 6.1 Decomposition of a HSI scene by 2D-SSA

Several reconstruction examples from the same image (at 667 nm) are shown in Figure 6.2, where $L_r = 5$ and $L_c = 5$ ($L^{2D} = 25$). As long as more components are selected in the reconstruction, the new image becomes more similar to the original one, where selection by $\text{EVG} = 1 - L^{2D\text{th}}$ components leads to exactly the same image.

When comparing to the 1D-SSA method (see Section 5.2.2), it is clear that the extended 2-D version provides the extraction of main trends and local structure in the spatial domain, something unfeasible in the 1D-SSA case, homogenising spatial elements in the image and increasing the intraclass similarities, which results in higher classification accuracy. On the other hand, the avoidance of less significant components results in noise mitigation, which is a common advantage for both 1-D and 2D-SSA with relation to the EMD methods [18].

Finally, it is important to remark that the configuration parameters used in the 2D-SSA method are exactly the same for all the spectral bands or scenes in the hyperspectral cube. Therefore, all the spectral scenes are computed in the same terms, similarly to the 1-D case, in which all pixels are transformed by the same configuration, allowing the fast implementation proposed in Section 5.3.

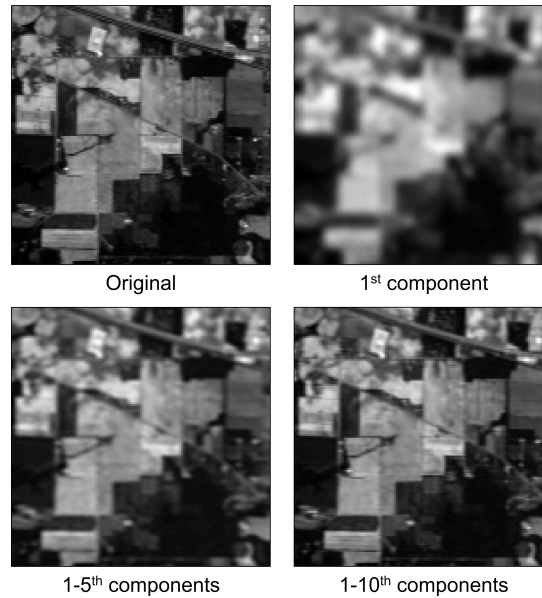


Fig. 6.2 Reconstruction of a HSI scene by 2D-SSA

6.3 Fast Implementation of 2D-SSA (F-2D-SSA)

Even though 2D-SSA is expected to be tremendously effective in feature extraction and subsequent data classification, it requires a band-level implementation, needing a complete SVD computation in each spectral band from the hyperspectral cube. In consequence, the complexity of the method when applied to HSI data, with hundreds of bands, is considerable.

Bearing in mind the good performance obtained from the 1D-SSA fast implementation [16], a fast version now for the 2-D case is introduced in this section, where a unique SVD is implemented for all the spectral bands, leading to reduced costs and high potential in onboard remote sensing and Earth observation.

6.3.1 Concept

Analogously to the previous 1-D case, the 2D-SSA configuration parameters are common for all the individual band scenes. As a result, the embedding procedure presents the same conditions and structure regardless of the spectral band being computed. This fact allows the proposed fast implementation F-2D-SSA.

As already described in Section 5.3.1, the use of the same configuration parameters along the whole cube, especially for the embedding procedure, leads to some benefits

such as the similarity of eigenvectors (and therefore transformation matrix) regardless of the information inputted to the SVD. Therefore, by embedding a spectral scene in a particular window with size L^{2D} , the 2D-SSA transformation can be obtained from a common set of eigenvectors, which are achieved by a unique SVD computation applied to a representative spectral scene.

6.3.2 Algorithm Description

From Equation 6.11, it is clear that the 2D-SSA extraction is based on a product of matrices, where the transformation (or eigenvectors) matrix \mathbf{U}_t is the key element by which resulting scenes are obtained. In other words, given an input image, simply by embedding with an appropriate window L^{2D} , it can be subject to the 2D-SSA algorithm using a predefined \mathbf{U}_t .

In the following, details are given about the single SVD computation that is still required to obtain the predefined eigenvectors matrix, and about the global implementation workflow of F-2D-SSA.

A. Representative Scene for a Single SVD

A single SVD computation, leading to the unique set of eigenvectors to be used in the whole cube reconstruction, demands again a careful determination of the representative signal, in this case a 2-D image to which the SVD is applied.

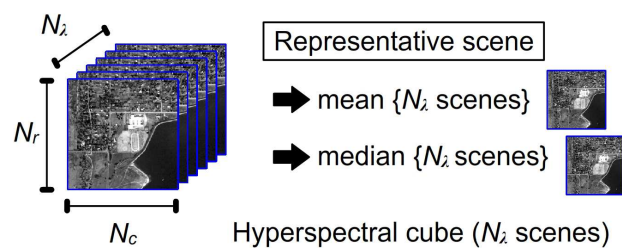


Fig. 6.3 HSI cube and its representative scenes

From the experience and good results obtained in the 1-D case [16] (see Section 5.4.2), the representative signal requires the capture of the general characteristics from those scenes contained in the hyperspectral cube. Given that all the spectral bands in a cube are acquired by the same instrument and under the same conditions, it is sensible

to think that an averaged or median image computed from all the spectral bands (Figure 6.3) may satisfactorily include the properties from the whole data set, becoming an adequate representative signal.

B. Workflow of F-2D-SSA

In the same terms as in the 1-D case, the difference introduced by the fast implementation F-2D-SSA is related to the SVD step, keeping the rest of the implementation workflow the same.

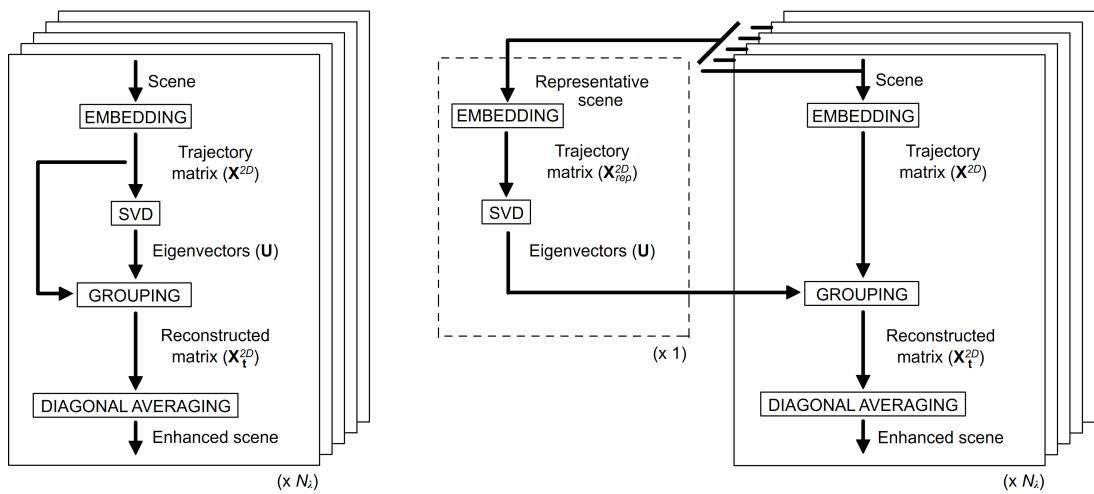


Fig. 6.4 Workflow for (left) 2D-SSA and (right) F-2D-SSA

Hence, the fast implementation (Figure 6.4) only requires a particular SVD computation, while the band-based flow consists of the embedding, grouping, and diagonal averaging steps. The grouping is carried out with the same set of eigenvectors \mathbf{U}_t by which all spectral scenes are reconstructed.

6.4 Analysis and Evaluation

After the explanation of both proposed methods, 2D-SSA and F-2D-SSA, where spatial domain is exploited for feature extraction, experimental settings are organised in this section, according to the general framework suggested in Section 2.3.1. Implemented again in the MATLAB [48] environment, several methods are studied and benchmarked [17], including the fast implementation, using 3 different data sets.

From this set of experiments, much higher classification accuracy is expected from both (F)-2D-SSA methods, with similar values to the 2D-EMD case and, therefore, surpassing the current state-of-the-art methods. On the other hand, the performance in terms of computational complexity is the key point when comparing 2D-EMD and 2D-SSA, where the fast implementation F-2D-SSA can further reduce the computation time and complexity, while maintaining the classification accuracy levels.

6.4.1 Experiments

Complete details about (A) data description, (B) data conditioning, (C) feature extraction methods evaluated, and (D) classifier employed are available in the following. The global experiments are similar to those undertaken in the 1D-SSA case, although now they include even more particular cases of study.

A. Data Description

It is found appropriate to employ 3 different HSI images from the 2 main sensors in this research, the Airborne Visible/InfraRed Imaging Spectrometer (AVIRIS) [30] and the Reflective Optics System Imaging Spectrometer (ROSIS) [32], including thus both natural and urban environments for land-cover classification. These images are:

1. 92AV3C [132] (Appendix A.1).
2. Pavia UA [36] (Appendix A.4).
3. Salinas C [36] (Appendix A.3).

B. Data Conditioning

With relation to the HSI images above, the common conditioning described in Appendix A is applied again to the different data sets respectively, where some bands are removed due to noise artifact and other similar issues. Additionally, for a proper benchmarking with other publications [18, 37], only 9 classes are evaluated in the 92AV3C data set, as this is claimed to provide more statistical significance. Although it is not reported in the present dissertation, similar results are also achieved using all the 16 original classes.

C. Feature Extraction

Several and comprehensive comparisons among many state-of-the-art methods for feature extraction and also data reduction are included in this stage, in the first place to clearly show the potential and general performance of the 2D-SSA method, and then, additionally, to study its fast implementation.

Initially, the main benchmarking includes the Baseline case (original spectral profiles), the previous 1D-SSA method [15], the 2D-EMD [18] (code in [138]), which is the main challenger, and the 2D-SSA proposal. This comparison is made under different conditions, with diverse configurations in each method to ensure a fair analysis (see Table 6.1), and reduced percentages of 5% and 10% in training the classifier.

Secondly, further benchmarking is carried out including several different state-of-the-art methods available in the literature to clearly show the advantage of the 2D-SSA method. This thorough benchmarking now includes methods such as Principal Component Analysis (PCA) [116], Independent Component Analysis (ICA) [69], Maximum Noise Fraction (MNF) [71], Extended MP (EMP) [104], and AFD [106], along with well-known spatial processing such as median filtering and morphological (opening and closing) operators, including also a combination with PCA (2D-SSA-PCA).

Table 6.1 Configuration for feature extraction methods in (F)-2D-SSA

Method	Parameters	Values adopted
Baseline	N/A	N/A
1D-SSA	Window size L EVG	5 and 10 1 st and 1-2 nd
2D-EMD	Stop threshold τ IMFG	0.2 1 st , 1-2 nd , 1-3 rd , 1-4 th
(F)-2D-SSA	Window size L^{2D} EVG	5×5 , 10×10 , 20×20 , 40×40 , 60×60 1 st , 1-2 nd , 1-5 th , 1-10 th

Finally, some experiments evaluate the fast implementation (F-2D-SSA) performance with relation to the 2D-SSA method under the same conditions. These experiments include a study on features similarity and new classification accuracy results. Once proved the expected similarity of both (F)-2D-SSA implementations, they are compared in terms of computational complexity, including Multiply Accumulates (MACs) and computation time.

D. Classification

The classifier stage is performed by Support Vector Machine (SVM) in accordance with the benchmarking publications. Implemented by means of the LIBSVM library [45], the SVM models are based on the Radial Basis Function (RBF) kernel, being trained by 2 different percentages, 5% and 10%, along the experiments.

Classification results from land-cover analysis are reported for performance comparisons, measured by the Overall Accuracy (OA) with standard deviation, and the McNemar's test of significance (where the Baseline method is taken as a reference), reporting the mean values from 10 repetitions. Additionally, Class by Class (CbC) and Average Accuracy (AA) values are also reported in some cases for further assessment.

On the other hand, the evaluation of 2D-EMD and 2D-SSA under a weaker classifier is also developed in the author's related publication [17]. A comparison of these feature extraction methods under a weaker classifier not so powerful as SVM proves the enhanced features and improved discrimination ability from the proposed 2D-SSA.

6.4.2 Results

Such a promising method as 2D-SSA requires comprehensive and in depth evaluations to appropriately assess its performance and derived advantages for feature extraction in hyperspectral remote sensing and Earth observation tasks.

Therefore, the summary of evaluations undertaken in the experimental sessions from this chapter comprises (A) main benchmarking in classification accuracy, (B) computation time comparison between 2D-EMD and 2D-SSA, and (C) comparison to other state-of-the-art methods. Then, following the introduction of the F-2D-SSA method, further evaluations include (D) features comparison between 2D-SSA and F-2D-SSA, (E) classification accuracy comparison for (F)-2D-SSA, and finally (F) computational complexity in (F)-2D-SSA. These are developed as follows.

A. Classification Accuracy

The values presented in Table 6.2, 6.3, and 6.4 for 92AV3C, Pavia UA, and Salinas C, respectively, show the main comparison under different conditions and configurations. The proposed 2D-SSA significantly beats the original features from the Baseline case

as well as the ones from the 1D-SSA method exploiting the spectral domain. This clearly shows that the spatial-domain exploitation leads to enhanced extraction with spatial structures preserved and noise mitigated. Accordingly, the classification accuracy for 92AV3C has massively increased from 81.3% and 85.6% to over 95.7% and 97.6%. For the other data sets, the already high accuracy from the Baseline case is increased as well, leading to values close to 99-100%, an excellent performance.

When comparing 2D-SSA to its main competitor, the 2D-EMD method, similar accuracy values are seen thanks to the spatial exploitation of hyperspectral cubes. However, the 2D-SSA method seems to generally provide slightly higher results. In addition, an important fact is that 2D-EMD appears extremely dependent on the IMFs components selected in the reconstruction. Actually, selecting only the first IMF leads to extreme deterioration in the classification accuracy, far below the Baseline values, which is unacceptable. However, the 2D-SSA improvements look much more reliable, with a consistent behavior under different conditions.

In fact, most configurations in 2D-SSA lead to high classification accuracy, surpassing the Baseline and 1D-SSA methods. Taking a closer look at Table 6.2, 6.3, and 6.4, the use of small L^{2D} with large number of components selected in EVG may provide the worst improvement, yet the Baseline case is still beaten. This is simply because the resulting features become more similar to the original ones, where reconstruction by all components would lead to the same Baseline results (see Section 5.4.2). All in all, the key point is to evade large EVG comprising too many components for a given window size L^{2D} .

Looking for further assessment, some other measurements such as the CbC and AA accuracy values are shown in Table 6.5, 6.6, and 6.7 for the best OA reached by every method under a training percentage of 10%. Again, these measurements support the outstanding results of 2D-SSA, where not only in OA, but also in AA and CbC accuracy terms performs great regardless of the Number of Samples (NoS) in each labelled class from the ground truth. With relation to the 2D-EMD method, the accuracy values are similar although 2D-SSA seems better for 9AV3C and Salinas C, while 2D-EMD presents higher values for urban Pavia UA. In conclusion, 2D-SSA and 2D-EMD seem to achieve similar classification results.

Table 6.2 Mean OA and McNemar’s test [\mathcal{L}] for 92AV3C

Parameters		OA (%) [\mathcal{L}]	
		Training 5%	Training 10%
N/A		Baseline	
		81.26 ± 0.94 [-0.00]	85.59 ± 0.63 [-0.00]
L	EVG	1D-SSA	
5	1 st	85.43 ± 0.95 [+11.0]	88.78 ± 0.52 [+9.24]
5	1-2 nd	83.42 ± 1.14 [+6.37]	88.02 ± 0.33 [+7.47]
10	1 st	85.32 ± 0.74 [+10.9]	88.68 ± 0.69 [+8.87]
10	1-2 nd	85.50 ± 0.93 [+11.4]	88.49 ± 0.57 [+8.57]
IMFG		2D-EMD	
	1 st	43.41 ± 1.20 [-50.4]	55.12 ± 1.10 [-42.3]
	1-2 nd	89.80 ± 1.42 [+17.6]	95.62 ± 0.51 [+23.8]
	1-3 rd	95.28 ± 0.45 [+31.7]	97.45 ± 0.34 [+29.5]
	1-4 th	94.02 ± 0.60 [+29.4]	96.11 ± 0.35 [+26.8]
L^{2D}	EVG	2D-SSA	
5 × 5	1 st	95.00 ± 1.07 [+30.3]	97.50 ± 0.58 [+29.1]
5 × 5	1-2 nd	93.23 ± 0.99 [+26.9]	96.03 ± 0.36 [+26.0]
5 × 5	1-5 th	89.07 ± 1.03 [+18.7]	92.99 ± 0.40 [+19.2]
5 × 5	1-10 th	84.85 ± 1.15 [+9.51]	89.49 ± 0.74 [+10.9]
10 × 10	1 st	95.71 ± 0.83 [+31.4]	97.59 ± 0.63 [+28.7]
10 × 10	1-2 nd	94.96 ± 0.80 [+29.9]	97.26 ± 0.55 [+28.3]
10 × 10	1-5 th	93.04 ± 0.93 [+26.0]	96.28 ± 0.53 [+26.3]
10 × 10	1-10 th	91.42 ± 0.92 [+23.2]	94.90 ± 0.46 [+23.3]
20 × 20	1 st	94.47 ± 0.67 [+27.9]	97.23 ± 0.61 [+27.5]
20 × 20	1-2 nd	94.43 ± 0.91 [+27.9]	97.36 ± 0.56 [+27.9]
20 × 20	1-5 th	94.90 ± 0.66 [+29.5]	97.43 ± 0.51 [+28.5]
20 × 20	1-10 th	93.70 ± 0.75 [+27.3]	96.91 ± 0.53 [+27.4]
40 × 40	1 st	94.43 ± 0.73 [+28.0]	97.14 ± 0.67 [+27.3]
40 × 40	1-2 nd	93.35 ± 0.94 [+25.3]	96.58 ± 0.45 [+25.8]
40 × 40	1-5 th	93.47 ± 1.36 [+25.8]	97.15 ± 0.56 [+27.4]
40 × 40	1-10 th	93.68 ± 0.82 [+26.6]	97.04 ± 0.60 [+27.1]
60 × 60	1 st	94.05 ± 0.77 [+26.5]	97.29 ± 0.47 [+27.4]
60 × 60	1-2 nd	92.94 ± 0.69 [+24.2]	96.55 ± 0.56 [+25.6]
60 × 60	1-5 th	94.13 ± 0.92 [+27.2]	97.29 ± 0.50 [+27.6]
60 × 60	1-10 th	93.02 ± 1.03 [+25.0]	96.72 ± 0.80 [+26.2]

Table 6.3 Mean OA and McNemar’s test [\mathcal{L}] for Pavia UA

Parameters		OA (%) [\mathcal{L}]	
		Training 5%	Training 10%
N/A		Baseline	
		95.83 ± 0.79 [-0.00]	96.67 ± 0.32 [-0.00]
<i>L</i> EVG		1D-SSA	
5	1 st	95.37 ± 0.86 [-2.44]	96.50 ± 0.31 [-0.88]
5	1-2 nd	95.53 ± 0.72 [-1.88]	96.60 ± 0.36 [-0.69]
10	1 st	95.21 ± 0.55 [-3.12]	96.15 ± 0.35 [-2.47]
10	1-2 nd	95.00 ± 0.84 [-4.14]	96.30 ± 0.52 [-2.12]
IMFG		2D-EMD	
	1 st	69.50 ± 1.75 [-38.1]	77.72 ± 1.63 [-30.8]
	1-2 nd	94.36 ± 0.73 [-4.16]	97.52 ± 0.34 [+2.92]
	1-3 rd	98.92 ± 0.34 [+11.6]	99.67 ± 0.11 [+12.5]
	1-4 th	99.53 ± 0.31 [+14.6]	99.80 ± 0.08 [+13.5]
<i>L</i> ^{2D} EVG		2D-SSA	
5 × 5	1 st	97.97 ± 0.38 [+7.31]	99.18 ± 0.28 [+9.92]
5 × 5	1-2 nd	98.21 ± 0.35 [+8.55]	98.99 ± 0.31 [+9.44]
5 × 5	1-5 th	97.57 ± 0.59 [+6.63]	98.90 ± 0.26 [+9.58]
5 × 5	1-10 th	97.21 ± 0.66 [+5.85]	98.18 ± 0.34 [+6.84]
10 × 10	1 st	97.77 ± 0.45 [+6.23]	98.92 ± 0.28 [+8.54]
10 × 10	1-2 nd	96.96 ± 0.64 [+3.56]	98.36 ± 0.39 [+6.20]
10 × 10	1-5 th	97.38 ± 0.64 [+5.19]	98.89 ± 0.44 [+8.76]
10 × 10	1-10 th	97.94 ± 0.44 [+7.59]	98.86 ± 0.29 [+9.02]
20 × 20	1 st	96.03 ± 0.75 [+0.56]	97.47 ± 0.47 [+2.64]
20 × 20	1-2 nd	97.07 ± 0.65 [+3.78]	98.43 ± 0.37 [+6.24]
20 × 20	1-5 th	97.20 ± 0.23 [+4.27]	98.51 ± 0.39 [+6.79]
20 × 20	1-10 th	96.67 ± 1.27 [+2.86]	98.76 ± 0.30 [+8.03]
40 × 40	1 st	95.91 ± 0.89 [+0.24]	97.64 ± 0.35 [+3.16]
40 × 40	1-2 nd	96.38 ± 1.04 [+1.68]	98.32 ± 0.53 [+5.87]
40 × 40	1-5 th	96.99 ± 0.40 [+3.54]	98.06 ± 0.42 [+4.92]
40 × 40	1-10 th	96.42 ± 0.65 [+1.74]	97.80 ± 0.45 [+3.94]
60 × 60	1 st	96.69 ± 0.42 [+2.46]	97.86 ± 0.45 [+4.02]
60 × 60	1-2 nd	96.38 ± 0.44 [+1.60]	98.23 ± 0.33 [+5.48]
60 × 60	1-5 th	96.35 ± 0.89 [+1.64]	97.98 ± 0.35 [+4.55]
60 × 60	1-10 th	96.78 ± 0.70 [+2.96]	98.08 ± 0.55 [+5.09]

Table 6.4 Mean OA and McNemar’s test [\mathcal{L}] for Salinas C

Parameters		OA (%) [\mathcal{L}]	
		Training 5%	Training 10%
N/A		Baseline	
		98.30 ± 0.20 [-0.00]	98.61 ± 0.12 [-0.00]
<i>L</i> EVG		1D-SSA	
5	1 st	98.46 ± 0.22 [+2.47]	98.76 ± 0.12 [+2.03]
5	1-2 nd	98.52 ± 0.15 [+3.41]	98.69 ± 0.09 [+1.21]
10	1 st	98.39 ± 0.28 [+1.66]	98.68 ± 0.09 [+0.78]
10	1-2 nd	98.42 ± 0.22 [+1.33]	98.76 ± 0.14 [+2.00]
IMFG		2D-EMD	
	1 st	68.58 ± 0.85 [-65.3]	76.42 ± 0.64 [-54.3]
	1-2 nd	94.56 ± 0.55 [-18.5]	97.56 ± 0.27 [-6.92]
	1-3 rd	99.54 ± 0.17 [+12.2]	99.78 ± 0.05 [+12.2]
	1-4 th	99.71 ± 0.14 [+13.8]	99.83 ± 0.04 [+12.7]
<i>L</i> ^{2D} EVG		2D-SSA	
5 × 5	1 st	99.51 ± 0.17 [+10.6]	99.77 ± 0.06 [+11.1]
5 × 5	1-2 nd	99.32 ± 0.13 [+9.45]	99.68 ± 0.07 [+10.3]
5 × 5	1-5 th	98.94 ± 0.19 [+6.33]	99.31 ± 0.18 [+7.35]
5 × 5	1-10 th	98.63 ± 0.12 [+4.01]	98.93 ± 0.18 [+3.64]
10 × 10	1 st	99.58 ± 0.26 [+11.2]	99.74 ± 0.12 [+10.8]
10 × 10	1-2 nd	99.34 ± 0.23 [+8.79]	99.69 ± 0.13 [+10.2]
10 × 10	1-5 th	99.44 ± 0.19 [+10.3]	99.77 ± 0.06 [+11.4]
10 × 10	1-10 th	99.06 ± 0.17 [+7.35]	99.47 ± 0.23 [+8.85]
20 × 20	1 st	99.62 ± 0.12 [+11.3]	99.81 ± 0.12 [+11.3]
20 × 20	1-2 nd	99.34 ± 0.15 [+8.55]	99.77 ± 0.12 [+11.0]
20 × 20	1-5 th	99.50 ± 0.24 [+10.5]	99.79 ± 0.08 [+11.3]
20 × 20	1-10 th	99.35 ± 0.13 [+9.17]	99.75 ± 0.05 [+10.9]
40 × 40	1 st	99.67 ± 0.17 [+12.0]	99.85 ± 0.08 [+12.1]
40 × 40	1-2 nd	99.81 ± 0.09 [+13.6]	99.93 ± 0.05 [+13.1]
40 × 40	1-5 th	99.46 ± 0.24 [+9.88]	99.81 ± 0.08 [+11.6]
40 × 40	1-10 th	99.56 ± 0.13 [+11.0]	99.77 ± 0.09 [+11.1]
60 × 60	1 st	99.63 ± 0.15 [+11.9]	99.90 ± 0.11 [+12.8]
60 × 60	1-2 nd	99.75 ± 0.15 [+13.0]	99.95 ± 0.05 [+13.5]
60 × 60	1-5 th	99.65 ± 0.20 [+12.1]	99.92 ± 0.05 [+13.0]
60 × 60	1-10 th	99.58 ± 0.17 [+11.1]	99.77 ± 0.07 [+11.1]

Table 6.5 Mean OA, CbC, and AA for 92AV3C (10% training)

Class	NoS	Baseline	1D-SSA	2D-EMD	2D-SSA
■ (2)	1434	80.71	84.81	95.53	96.35
■ (3)	834	72.03	80.99	95.96	97.72
■ (5)	497	89.98	92.73	96.38	96.76
■ (6)	747	97.23	97.77	99.36	97.92
■ (8)	489	99.00	99.11	99.45	99.18
■ (10)	968	76.83	83.54	94.73	95.74
■ (11)	2468	83.99	86.11	98.48	98.27
■ (12)	614	80.45	84.91	95.89	95.94
■ (14)	1294	98.27	98.41	99.90	99.27
AA (%)		86.50	89.82	97.30	97.46
OA (%)		85.59	88.78	97.45	97.59

Table 6.6 Mean OA, CbC, and AA for Pavia UA (10% training)

Class	NoS	Baseline	1D-SSA	2D-EMD	2D-SSA
■ (1)	310	81.94	81.72	99.86	96.70
■ (2)	957	97.21	97.07	100	100
■ (4)	154	96.23	96.30	99.28	94.49
■ (5)	698	99.67	99.71	99.62	100
■ (6)	2559	97.57	97.42	99.89	99.79
■ (7)	860	95.27	95.44	99.85	98.46
■ (8)	854	96.63	96.51	99.48	98.54
■ (9)	293	100	100	99.77	98.17
AA (%)		95.56	95.52	99.72	98.27
OA (%)		96.67	96.60	99.80	99.18

Table 6.7 Mean OA, CbC, and AA for Salinas C (10% training)

Class	NoS	Baseline	1D-SSA	2D-EMD	2D-SSA
■ (1)	240	95.32	95.56	100	100
■ (2)	3400	99.93	99.92	99.96	100
■ (3)	1957	99.71	99.80	99.88	99.96
■ (4)	599	99.13	98.42	99.09	99.70
■ (5)	1155	97.77	98.40	98.78	99.95
■ (6)	1414	99.99	99.99	99.98	100
■ (7)	848	99.62	99.65	99.93	99.99
■ (8)	5890	99.23	99.35	99.99	99.98
■ (15)	159	25.52	32.59	98.81	97.97
AA (%)		90.69	91.52	99.60	99.73
OA (%)		98.61	98.76	99.83	99.95

B. Computation Time for 2D-EMD and 2D-SSA

Being the 2D-EMD method the main competitor of the 2D-SSA proposal, it is important to evaluate not only the classification accuracy in land-cover analysis but also the computation time required for extracting the features. Therefore, yet the features from both methods can achieve similar accuracy values, 2D-SSA extraction is proved to be much faster, as shown in Table 6.8.

Table 6.8 Computation time for 2D-EMD and 2D-SSA (*best OAs)

Method	Parameters	Time (s)			
		92AV3C	Pavia UA	Salinas C	
2D-EMD	IMFG	1 st	322	196	394
		1-2 nd	635	365	765
		1-3 rd	* 936	529	1148
		1-4 th	1324	* 688	* 1506
2D-SSA	L^{2D}	5×5	20	* 11	21
		10×10	* 34	19	36
		20×20	81	42	84
		40×40	262	146	* 290
		60×60	535	308	* 590

On one hand, extracting features for the best classification cases from the 2D-EMD method requires 936, 688, and up to 1506 s for the 92AV3C, Pavia UA, and Salinas C data sets, respectively. On the other hand, obtaining features with similar or even better accuracy values with the 2D-SSA method takes only 34, 11, and 290 s for the respective data sets. Hence, by using 2D-SSA, similar classification accuracy is obtained while the time required for feature extraction is massively reduced to just few seconds, 1 or even 2 orders of magnitude faster.

The reason why 2D-SSA is much faster resides in its straightforward implementation, based on the SVD (or equivalent EVD), avoiding empirical iterations such as those undertaken in the 2D-EMD implementation. As a result, 2D-SSA goes a step beyond the current state of the art and provides a classification accuracy near 100% with features obtained in just few seconds, leading to faster data processing and allowing the implementation in portable and embedded systems for real-time applications, with high potential for onboard missions in military and remote sensing tasks.

C. Classification Accuracy for 2D-SSA and other Methods

After the main benchmarking against 2D-EMD, now the experiments include a wider comparison to other state-of-the-art methods, to definitely put in context the 2D-SSA proposal. All the methods compared here are divided into 2 main groups.

In the first place, it is possible to find methods maintaining the original dimensionality of features (N_λ), where 2D-SSA is benchmarked against a median filtering procedure, the AFD method [106], morphological opening and closing operators, and again 1D-SSA [15], 2D-EMD [18], and the Baseline case. In the second place, now data reduction is introduced along the feature extraction, where some classical methods such as PCA, ICA, and MNF are compared, including also the interesting EMP [104]. Additionally, 2D-SSA is combined with a posterior PCA computed in the spectral domain (2D-SSA-PCA), which is expected to prove benefits from combining 2D-SSA with other methods, leading in this case to a reduced number of features F .

Table 6.9 Mean OA and McNemar’s test [\mathcal{L}] for 92AV3C (best OAs)

Methods	OA (%) [\mathcal{L}] (F)	
	Training 5%	Training 10%
ORIGINAL DIMENSIONALITY OF FEATURES (200)		
Baseline	81.26 ± 0.94 [-0.00]	85.59 ± 0.63 [-0.00]
1D-SSA [15]	85.50 ± 0.93 [+11.4]	88.78 ± 0.52 [+9.24]
2D-EMD [18]	95.28 ± 0.45 [+31.7]	97.45 ± 0.34 [+29.5]
2D-SSA [17]	95.71 ± 0.83 [+31.4]	97.59 ± 0.63 [+28.7]
Median Filter	92.88 ± 0.29 [+25.4]	95.24 ± 0.46 [+23.4]
AFD [106]	95.11 ± 0.72 [+30.9]	96.66 ± 0.47 [+27.2]
M. Opening	94.07 ± 0.75 [+28.1]	96.40 ± 0.32 [+25.9]
M. Closing	92.51 ± 0.87 [+23.9]	95.41 ± 0.65 [+22.9]
DATA REDUCTION (dimensionality of features F)		
PCA	80.57 ± 0.85 [-1.59] (20)	84.19 ± 0.95 [-3.35] (20)
ICA	80.07 ± 1.21 [-2.69] (20)	83.72 ± 0.87 [-4.45] (20)
MNF	81.73 ± 1.04 [+1.13] (40)	85.94 ± 0.76 [+0.90] (10)
EMP [104]	94.83 ± 0.78 [+29.3] (34)	97.28 ± 0.34 [+28.0] (34)
2D-SSA-PCA	97.61 ± 0.69 [+35.5] (15)	99.01 ± 0.10 [+32.3] (20)

The benchmarking comparisons can be consulted in Table 6.9, 6.10, and 6.11, for the 92AV3C, Pavia UA, and Salinas C data sets, respectively, where each method provides its best result (highest classification OA) from several different configurations

evaluated, leading to a fair and significant comparison. More information is available in the author’s related publication [17].

Results for 92AV3C in Table 6.9 prove that 2D-SSA beats all the other approaches, including the interesting AFD and EMP techniques. Moreover, the joint 2D-SSA-PCA method still leads to further improvement in classification accuracy, from 95.71% and 97.59% to 97.61% and 99.01%, highlighting the benefits from exploiting both spatial and spectral domains.

With relation to the Pavia UA results in Table 6.10, even though 2D-SSA achieves OA values marginally below the ones from 2D-EMD, AFD, and EMP, the combination 2D-SSA-PCA produces the best result of all methods, where it actually requires a smaller number of features than EMP (20 instead of 34). Finally, OA values from the Salinas C data set (Table 6.11) again support the efficacy of both 2D-SSA and 2D-SSA-PCA, beating any other method and leading to almost 100% of accuracy, clearly showing the 2D-SSA potential in this context.

Table 6.10 Mean OA and McNemar’s test [\mathcal{L}] for Pavia UA (best OAs)

Methods	OA (%) [\mathcal{L}] (F)	
	Training 5%	Training 10%
ORIGINAL DIMENSIONALITY OF FEATURES (103)		
Baseline	95.83 ± 0.79 [-0.00]	96.67 ± 0.32 [-0.00]
1D-SSA [15]	95.53 ± 0.72 [-1.88]	96.60 ± 0.36 [-0.69]
2D-EMD [18]	99.53 ± 0.31 [+14.6]	99.80 ± 0.08 [+13.5]
2D-SSA [17]	98.21 ± 0.35 [+8.55]	99.18 ± 0.28 [+9.92]
Median Filter	98.77 ± 0.20 [+11.4]	99.25 ± 0.17 [+10.5]
AFD [106]	99.32 ± 0.28 [+13.0]	99.61 ± 0.23 [+12.2]
M. Opening	97.57 ± 0.35 [+6.72]	98.47 ± 0.31 [+6.97]
M. Closing	97.19 ± 0.45 [+5.11]	97.93 ± 0.35 [+5.08]
DATA REDUCTION (dimensionality of features F)		
PCA	94.29 ± 0.68 [-4.74] (15)	95.44 ± 0.34 [-4.46] (15)
ICA	94.59 ± 0.64 [-3.96] (15)	95.48 ± 0.41 [-4.30] (15)
MNF	94.54 ± 0.65 [-5.31] (15)	95.59 ± 0.32 [-4.72] (20)
EMP [104]	99.56 ± 0.68 [+14.1] (34)	99.88 ± 0.07 [+13.4] (34)
2D-SSA-PCA	99.58 ± 0.14 [+14.1] (20)	99.85 ± 0.07 [+13.4] (20)

Some extra comments can be derived from these comparisons, where it is obvious that the spectral-domain feature extraction (having PCA, ICA, MNF, and also 1D-

SSA as significant methods) offers a limited performance, especially when compared to those other methods considering the spatial information from the HSI cubes, such as median filtering or morphological operators. These methods provide considerable accuracy, yet performing worse than the particular AFD and EMP approaches. However, 2D-EMD and especially 2D-SSA are the methods showing higher classification OA, where 2D-SSA seems slightly better although being much faster in extracting the features.

Table 6.11 Mean OA and McNemar’s test [\mathcal{L}] for Salinas C (best OAs)

Methods	OA (%) [\mathcal{L}] (F)	
	Training 5%	Training 10%
ORIGINAL DIMENSIONALITY OF FEATURES (204)		
Baseline	98.30 \pm 0.20 [-0.00]	98.61 \pm 0.12 [-0.00]
1D-SSA [15]	98.52 \pm 0.15 [+3.41]	98.76 \pm 0.12 [+2.03]
2D-EMD [18]	99.71 \pm 0.14 [+13.8]	99.83 \pm 0.04 [+12.7]
2D-SSA [17]	99.81 \pm 0.09 [+13.6]	99.95 \pm 0.05 [+13.5]
Median Filter	99.57 \pm 0.09 [+11.8]	99.76 \pm 0.04 [+11.4]
AFD [106]	99.70 \pm 0.08 [+12.8]	99.79 \pm 0.06 [+11.7]
M. Opening	99.50 \pm 0.11 [+10.8]	99.74 \pm 0.12 [+11.1]
M. Closing	99.24 \pm 0.15 [+9.42]	99.47 \pm 0.10 [+9.04]
DATA REDUCTION (dimensionality of features F)		
PCA	98.60 \pm 0.16 [+3.81] (25)	98.82 \pm 0.09 [+2.37] (25)
ICA	98.57 \pm 0.18 [+3.41] (25)	98.81 \pm 0.07 [+2.38] (25)
MNF	98.08 \pm 0.29 [-2.67] (20)	98.40 \pm 0.08 [-2.89] (45)
EMP [104]	99.49 \pm 0.16 [+10.5] (19)	99.72 \pm 0.10 [+10.8] (19)
2D-SSA-PCA	99.83 \pm 0.16 [+14.0] (20)	99.92 \pm 0.06 [+12.9] (10)

All in all, it has been demonstrated that 2D-EMD and 2D-SSA provide the highest accuracy values among all the evaluated methods, covering spectral/spatial exploitation and original/reduced dimensionality of features. Moreover, the joint 2D-SSA-PCA method has shown extra efficacy in classification tasks, with reduced number of features, from hundreds to tens.

D. Comparison of Extracted Features in (F)-2D-SSA

Analogously to the 1-D case [16] in Section 5.3, the extracted features from the F-2D-SSA implementation are expected to be very similar to those from 2D-SSA. To

validate this assumption, features from both conventional and fast implementations, (F)-2D-SSA, are visually and quantitatively studied below.

The original and 2D-SSA-reconstructed spectral scenes ($L_r = 10$, $L_c = 10$, and $EVG=1^{st}$), including conventional and fast implementation (from both mean and median representative scenes), can be seen in Figure 6.5 for a spectral scene randomly selected (at 667 nm) from 92AV3C [132], where the difference among the 3 reconstructed cases seems imperceptible.

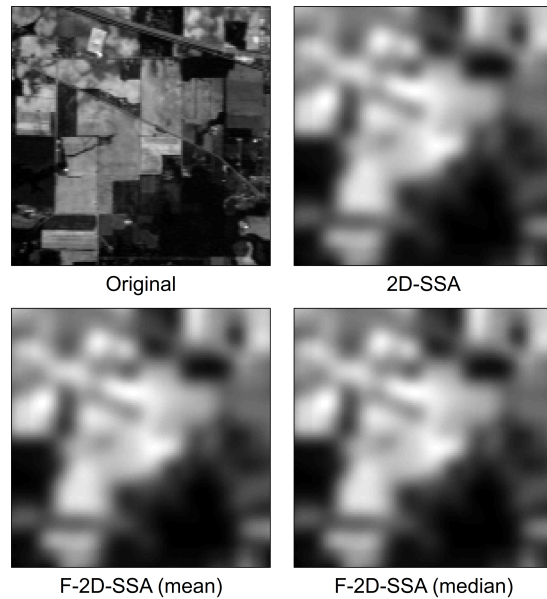


Fig. 6.5 Original and (F)-2D-SSA-reconstructed scenes

Leaving aside visual indicators, the difference among features from the conventional and fast implementations is quantitatively reported for a more detailed analysis. The cosine distance [136] is employed again, as in the 1-D case, as objective and adequate measurement.

Table 6.12 reports the mean cosine distance among all the original scenes and their respective (F)-2D-SSA implementations, where this is repeated for all possible configurations according to the L^{2D} and EVG parameters. From these results, the similarity of features between conventional and fast implementation is demonstrated, with a small dissimilarity with relation to the original features that is commonly found in the 3 implementations, which increases with larger window size L^{2D} and smaller EVG, in consistency with the 1-D case (see Section 5.4.2).

Table 6.12 Mean cosine similarity scores in (F)-2D-SSA for 92AV3C

Conventional 2D-SSA				
$L^{2D} \setminus \text{EVG}$	1st	1-2nd	1-5th	1-10th
5 × 5	99.8996	99.9345	99.9746	99.9917
10 × 10	99.7999	99.8553	99.9216	99.9536
20 × 20	99.6737	99.7383	99.8333	99.8857
40 × 40	99.5288	99.6150	99.7105	99.7793
60 × 60	99.4519	99.5417	99.6456	99.7249
F-2D-SSA (mean)				
$L^{2D} \setminus \text{EVG}$	1st	1-2nd	1-5th	1-10th
5 × 5	99.8995	99.9343	99.9745	99.9917
10 × 10	99.7996	99.8547	99.9212	99.9533
20 × 20	99.6728	99.7374	99.8326	99.8851
40 × 40	99.5313	99.5874	99.7075	99.7744
60 × 60	99.4586	99.5045	99.6226	99.6991
F-2D-SSA (median)				
$L^{2D} \setminus \text{EVG}$	1st	1-2nd	1-5th	1-10th
5 × 5	99.8995	99.9341	99.9742	99.9916
10 × 10	99.7998	99.8546	99.9208	99.9531
20 × 20	99.6731	99.7368	99.8324	99.8835
40 × 40	99.5279	99.6054	99.7062	99.7769
60 × 60	99.4391	99.5307	99.6359	99.7131

E. Classification Accuracy for (F)-2D-SSA

Providing very similar features, the (F)-2D-SSA implementations are now compared in terms of classification accuracy in land-cover analysis. Comprehensive results for all possible configurations are compared in Table 6.13 for the 92AV3C data set, showing OA and McNemar's test values, while complementary CbC and AA assessment is reported as well in Table 6.14. Similar findings are obtained for the other data sets.

It is not surprising that both fast implementations provide a classification accuracy similar to the conventional case. The OAs fluctuate around the conventional values in some degree, which seems related to the similarity scores. Taking the highest OAs, the fast implementations lead to 95.66% and 95.82%, in comparison to the 95.71% from the conventional case. Complementary CbC and AA values lead to the same conclusions. All in all, the global mean value from all possible configurations proves that, even though there is some fluctuation, this is quite reduced and the general behaviour from the fast implementations is validated.

Table 6.13 Mean OA and McNemar’s test [\mathcal{L}] in (F)-2D-SSA for 92AV3C

Parameters		2D-SSA	F-2D-SSA (mean)	F-2D-SSA (median)
L^{2D}	EVG			
5 × 5	1 st	95.00 [+30.3]	95.00 [+30.3]	95.01 [+30.3]
5 × 5	1-2 nd	93.23 [+26.9]	93.09 [+26.5]	93.09 [+26.4]
5 × 5	1-5 th	89.07 [+18.7]	89.88 [+20.3]	90.08 [+20.9]
5 × 5	1-10 th	84.85 [+9.51]	85.73 [+11.7]	85.71 [+11.8]
10 × 10	1 st	95.71 [+31.4]	95.66 [+31.3]	95.82 [+31.5]
10 × 10	1-2 nd	94.96 [+29.9]	95.22 [+30.4]	95.29 [+30.6]
10 × 10	1-5 th	93.04 [+26.0]	93.95 [+28.1]	93.98 [+28.1]
10 × 10	1-10 th	91.42 [+23.2]	92.08 [+24.7]	92.08 [+24.7]
20 × 20	1 st	94.47 [+27.9]	94.47 [+27.9]	94.33 [+27.6]
20 × 20	1-2 nd	94.43 [+27.9]	94.61 [+28.3]	94.36 [+27.6]
20 × 20	1-5 th	94.90 [+29.5]	95.40 [+30.6]	95.49 [+30.8]
20 × 20	1-10 th	93.70 [+27.3]	94.36 [+28.7]	94.52 [+29.1]
40 × 40	1 st	94.43 [+28.0]	94.23 [+27.4]	94.73 [+28.6]
40 × 40	1-2 nd	93.35 [+25.3]	94.07 [+27.1]	93.72 [+26.3]
40 × 40	1-5 th	93.47 [+25.8]	93.78 [+26.6]	93.68 [+26.3]
40 × 40	1-10 th	93.68 [+26.6]	94.75 [+28.9]	94.61 [+28.6]
60 × 60	1 st	94.05 [+26.5]	93.30 [+24.9]	92.59 [+23.1]
60 × 60	1-2 nd	92.94 [+24.2]	93.13 [+24.7]	92.85 [+23.9]
60 × 60	1-5 th	94.13 [+27.2]	93.59 [+26.0]	93.55 [+25.9]
60 × 60	1-10 th	93.02 [+25.0]	93.72 [+26.3]	93.22 [+25.4]
Global mean		93.19 [+25.9]	93.50 [+26.5]	93.44 [+26.4]

Table 6.14 Mean OA, CbC, and AA in (F)-2D-SSA for 92AV3C

Class	NoS	Baseline	2D-SSA	F-2D-SSA (mean)	F-2D-SSA (median)
■ (2)	1434	75.38	95.38	94.88	95.65
■ (3)	834	63.32	96.00	96.00	96.00
■ (5)	497	89.30	94.79	94.77	94.60
■ (6)	747	96.81	96.59	96.47	96.50
■ (8)	489	99.07	97.09	97.16	97.11
■ (10)	968	65.97	90.45	89.76	91.23
■ (11)	2468	81.10	96.54	96.92	96.60
■ (12)	614	69.97	93.86	93.84	93.67
■ (14)	1294	97.62	98.47	98.45	98.46
AA (%)		82.06	95.46	95.36	95.54
OA (%)		81.26	95.71	95.66	95.82

F. Computational Complexity

The F-2D-SSA method has been proved to achieve the same features and classification performance as the conventional case in practical terms. However, the main purpose of this fast implementation is to reduce the complexity by requiring only a single SVD computation instead of the hundreds initially demanded. This saving is shown in the following, where a brief description on the computational complexity required for the 2D-SSA method is included.

In the conventional 2D-SSA implementation, a 2-D image is reallocated into a 2-D trajectory matrix by means of a predefined window with size L^{2D} in the embedding step, but this reallocation procedure actually requires no MACs. Then, the SVD computation is carried out, where an equivalent EVD being applied to $\mathbf{X}^{2D}(\mathbf{X}^{2D})^\top$ results in less complexity than the one derived from SVD, with $((L^{2D})^2K^{2D} + (L^{2D})^3)$ instead of the $((L^{2D})^2K^{2D} + L^{2D}(K^{2D})^2 + (K^{2D})^3)$ suggested in [54, 137]. In the next step, the grouping, a selection of components is performed by Equation 6.11 with complexity $2L^{2D}K^{2D}T$. Then, in the last step, a diagonal averaging procedure is implemented with a complexity approximated to N_rN_c , analogously to [16].

According to the description above, a comparison of computational complexity can be undertaken between conventional and fast implementation of 2D-SSA, where in Table 6.15 complexity is expressed in terms of MACs and divided into the 4 steps. As already discussed, the savings are introduced in the SVD step, with costs being reduced N_λ times, i.e., the number of spectral bands (normally hundreds). Putting some values, Table 6.16 shows the number of MACs and saving factor globally achieved for a total of 4 different configurations, with a minimum saving of about 2 (half the initial cost) but also dramatically large savings superior to 100, depending on the configuration selected.

Table 6.15 Computational complexity (MACS) in (F)-2D-SSA

Step	2D-SSA	F-2D-SSA	Sav. fac.
Embedding	N/A	N/A	1
SVD	$((L^{2D})^2K^{2D} + (L^{2D})^3)N_\lambda$	$((L^{2D})^2K^{2D} + (L^{2D})^3)1$	N_λ
Grouping	$(2L^{2D}K^{2D}T)N_\lambda$	$(2L^{2D}K^{2D}T)N_\lambda$	1
Diag. Av.	$(N_rN_c)N_\lambda$	$(N_rN_c)N_\lambda$	1

Table 6.16 Number of MACS in (F)-2D-SSA for 92AV3C

L^{2D}	5×5	5×5	60×60	60×60
EVG	1 st	1-10 th	1 st	1-10 th
2D-SSA	2.69e9	4.48e9	2.85e13	2.86e13
F-2D-SSA	2.15e8	2.01e9	1.53e11	2.49e11
Saving factor	12.5	2.23	186	115

With such saving factors, not only the complexity is reduced in MACs terms but also in the computation time required for implementing the extraction of features, whether the mean or median approach is used. This fact is especially visible in those configurations with large saving factors, where the required time can be reduced up to 60%, going from 525 to 209 s for the 92AV3C data set as shown in Table 6.17.

Table 6.17 Computation time for (F)-2D-SSA for 92AV3C

Parameters		2D-SSA	F-2D-SSA	Reduction	
L^{2D}	EVG			(s)	(%)
5×5	1 st	19	18	1	5.26
5×5	1-2 nd	19	18	1	5.26
5×5	1-5 th	19	18	1	5.26
5×5	1-10 th	19	18	1	5.26
10×10	1 st	31	28	3	9.68
10×10	1-2 nd	31	28	3	9.68
10×10	1-5 th	32	28	4	12.5
10×10	1-10 th	32	28	4	12.5
20×20	1 st	69	58	11	15.9
20×20	1-2 nd	71	60	11	15.5
20×20	1-5 th	72	61	11	15.3
20×20	1-10 th	73	62	11	15.1
40×40	1 st	244	137	107	43.9
40×40	1-2 nd	251	142	109	43.4
40×40	1-5 th	255	144	111	43.5
40×40	1-10 th	257	145	112	43.6
60×60	1 st	496	194	302	60.9
60×60	1-2 nd	507	205	302	59.6
60×60	1-5 th	518	208	310	59.9
60×60	1-10 th	525	209	316	60.2

6.5 Summary

Feature extraction methods exploiting the spatial-domain information from HSI cubes have been introduced in recent years, with significant classification accuracy achieved. Following previous contributions from the present thesis, in particular the 1D-SSA method, its natural extension to the 2-D case is now introduced as 2D-SSA in hyperspectral remote sensing.

Working in the spatial domain, the 2D-SSA method is proved to achieve impressive high accuracy values around 99-100%, only comparable to the 2D-EMD method. Moreover, as the 2D-EMD method is based on empirical iterations, its implementation is found extremely complex and time-consuming, while the 2D-SSA method is simply based on the SVD, which leads to a reduced computation time in extracting features, 1 or even 2 orders of magnitude faster than the 2D-EMD case. Moreover, 2D-SSA complexity can be further reduced by a fast implementation analogous to the one introduced in the 1-D case. This fast implementation produces similar classification accuracy while further reduces the computational complexity, allowing potential use in portable devices, onboard spacecrafts (or other Earth observation platforms), and in real-time applications.

All in all, the 2D-SSA method has been proved to achieve the best classification results from a comprehensive state-of-the-art comparison, where the combination of the 2D-SSA method with a posterior PCA applied in the spectral domain (2D-SSA-PCA) provides an excellent performance even with a reduced number of features, since both spatial and spectral domains are exploited.

Chapter 7

Segmented-SAE for Improved Data Reduction

Although approaches based on Principal Component Analysis (PCA) and Singular Spectrum Analysis (SSA) are proposed in the previous chapters for improved feature extraction and data reduction, they rely mainly on Singular Value Decomposition (SVD) of the original or embedded data. In this chapter, however, learning-based approaches are explored for this purpose, using the Stacked AutoEncoders (SAEs) in the Deep Learning (DL) framework. In fact, the DL framework is derived from the Neural Network (NN) learning strategy, which has recently led to remarkable performance in many application areas [139–141]. In most of these works, SAEs are found to be useful in data abstraction for effective feature extraction [97, 98, 101].

Accordingly, SAEs have been proposed for feature extraction and data reduction in HyperSpectral Imaging (HSI), as the hidden nodes from deep layers in the DL algorithms are able to provide high levels of abstraction, resulting in features being reduced but including the main information from the original feature space. However, as the hidden nodes have to cope with hundreds of original features from the HSI cube, limited efficacy of feature extraction and extremely high computational cost become the bottleneck for its implementation in HSI.

To solve this challenging problem, a segmented procedure for the SAEs algorithm implementation, Segmented-SAE (S-SAE), is proposed in this chapter. By confronting smaller data segments, the complexity of the DL architectures is reduced, while the obtained features are improved due to the extraction of local information in the spectral

domain. This concept and following ideas open a wide range of possibilities for future research, with publications being developed by the author at the moment this dissertation is submitted.

In this chapter, followed by a short introduction in Section 7.1, the background of SAEs is presented in Section 7.2. In Section 7.3, the proposed S-SAE is discussed in detail. Results are presented in Section 7.4 with a brief summary in Section 7.5.

7.1 Introduction

Due to the challenges and motivations related to the research in hyperspectral remote sensing, many different methodologies regarding advanced Digital Signal Processing (DSP) have been studied, especially with relation to the feature extraction and data reduction stage [28, 87], where the accuracy in classifying pixels from an image is the main indicator for evaluations.

Consequently, it is not surprising that along with many other approaches, some DL methods are currently being introduced to remote sensing applications [97–101]. Based on NN architectures, which also includes convolutional NN and deep belief networks, the SAEs in a DL framework can lead to proper reduction in the dimensionality of features, as internal nodes in the network are able to capture and represent the original data in reduced elements by abstraction. The stimulating SAE method, however, requires remarkable complexity in its implementation. When thinking about the NN structure, it is easy to realise that the nodes forming the network have to deal with an excessive amount of original data, i.e., the hundreds of spectral bands available in each HSI pixel or sample. In addition, as the hidden units are expected to cope with all the spectral values in the same activation functions, the process of finding an adequate abstraction of features may be excessively complex.

In this chapter, the S-SAE is introduced, where the concept of spectral segmentation is exploited so that both reduction in complexity and extraction of local features can be reached. To this end, local and smaller SAEs are implemented in different segments or regions of the spectrum, leading to enhanced performance in terms of better classification accuracy and reduced complexity.

7.2 Stacked AutoEncoders (SAEs) in HSI

The concept underlying a basic AutoEncoder (AE) can be defined as an NN (also convolutional NN or deep belief network) architecture model, built by numerous interconnected nodes, in which an inputted signal, with dimensionality of features matching the number of nodes in the first or input layer, is reconstructed at the output of the model after going through an internal layer with reduced number of hidden nodes.

The global AE architecture is able to learn and pick up abstract features, so a reconstruction is feasible from the data contained in the reduced hidden nodes, providing an output signal matching the original input with some degree of accuracy. A scheme representing the basic AE architecture is available in Figure 7.1, where an input pixel from the hyperspectral cube, $\mathbf{p} \in \mathbb{R}^{N_\lambda}$, can see its dimensionality of features reduced to $F \ll N_\lambda$ elements, from which it can be reconstructed into $\hat{\mathbf{p}} \in \mathbb{R}^{N_\lambda}$.

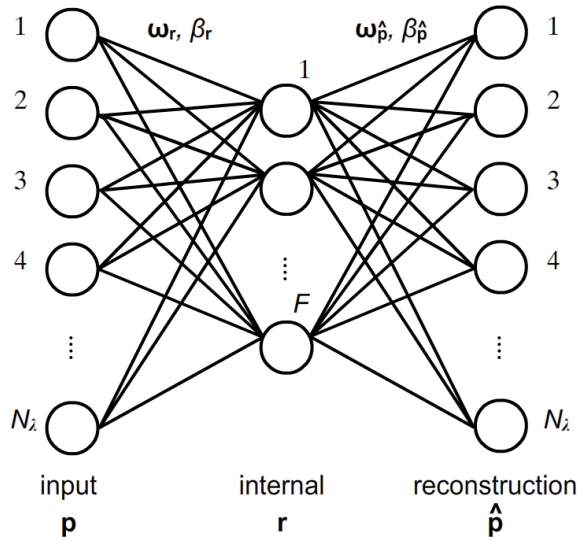


Fig. 7.1 Schematic representation of a basic AE

Mathematical formulation of AEs is indeed pretty simple and easy to understand. Basically, an AE needs to satisfy the following

$$\begin{aligned} \mathbf{r} &= f\{\boldsymbol{\omega}_r \mathbf{p} + \beta_r\} \\ \hat{\mathbf{p}} &= f\{\boldsymbol{\omega}_{\hat{\mathbf{p}}} \mathbf{r} + \beta_{\hat{\mathbf{p}}}\} \end{aligned} \quad (7.1)$$

obtaining the internal signal $\mathbf{r} \in \mathbb{R}^F$ from \mathbf{p} by means of the vector of weights $\boldsymbol{\omega}_r$ and bias value β_r . The $f\{\}$ operator denotes the activation function able to introduce

the nonlinearity in the network or AE architecture. In similar terms, the reconstructed signal $\hat{\mathbf{p}} \in \mathbb{R}^{N_\lambda}$ is calculated in this case from the internal \mathbf{r} thanks to the corresponding weights $\boldsymbol{\omega}_{\hat{\mathbf{p}}}$ and bias $\beta_{\hat{\mathbf{p}}}$. Finally, the training process can be expressed as

$$\underset{\boldsymbol{\omega}_{\mathbf{r}}, \boldsymbol{\omega}_{\hat{\mathbf{p}}}, \beta_{\mathbf{r}}, \beta_{\hat{\mathbf{p}}}}{\operatorname{argmin}} \{error\{\mathbf{p}, \hat{\mathbf{p}}\}\}, \quad (7.2)$$

simply denoting a minimisation of the difference between original and reconstructed signals. The error value usually depends on the application and related purposes, although no specific definitions are suggested.

Once the basic AE algorithm is clear, the extended SAE is formulated by expanding the AE concept and introducing several internal layers. Therefore, the new architecture presents similar input and output, where it is possible to find several intermediate layers in between. By introducing several layers, higher abstraction can be achieved through progressive levels. In Figure 7.2, the scheme representing a SAE with 2 internal layers is shown, where normally $F < E$.

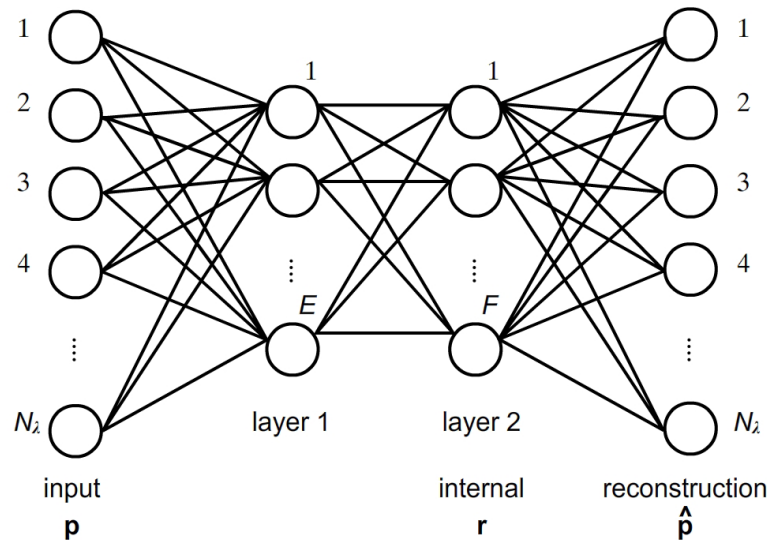


Fig. 7.2 Schematic representation of a SAE (2 internal layers)

Due to the abstraction level achieved by reduced features in the internal layers, SAEs can be introduced as a data reduction method in hyperspectral remote sensing, exploiting the spectral domain of the pixels. Hence, after training with a representative portion of pixels, each sample can be subject to a reduction in the dimensionality of features.

The training undertaken is based on an iterative process by which the weight (ω) and bias (β) values from different layers and nodes are continuously being updated. The error between input and output signals is used for the iterative update of these internal variables. Eventually, the error decreases enough to fulfil some criteria and the training process is completed.

An effective training demands a reduced error in the reconstruction, which is expected to lead to appropriate features from the deepest layer. In Figure 7.3, an original spectral profile from 92AV3C [132] is shown in comparison to its related reconstruction by a two-layer SAE.

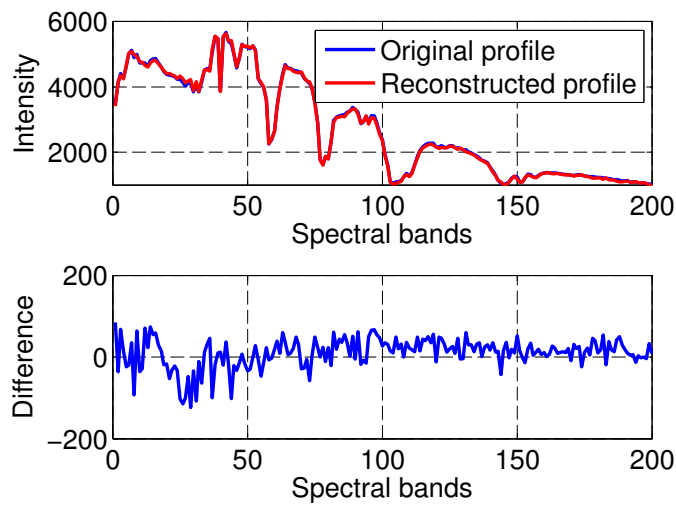


Fig. 7.3 Difference between original and SAE-reconstructed pixels

7.3 Proposed S-SAE

Regarding the potential of the SAE method for feature extraction and data reduction in HSI, and the computational drawbacks with relation to its implementation, the segmentation of the method is proposed, taking advantage of previous research and experience in the field [13, 62], leading to the S-SAE.

The conventional implementation of the SAE method presents 2 main problems to be addressed. On one hand, it treats all spectral bands equally, while it is common to find specific regions in the spectral domain with enhanced information. Therefore, this fact results in failure when trying to pick up the disparate contributions from each

band [13]. On the other hand, working with all spectral values simultaneously yields high computational complexity, as hidden nodes in the first layer have to deal with the original feature dimension N_λ . The number of interconnections inside the NN architecture seems therefore excessive and also may result in difficulties when looking for proper abstraction.

In order to solve these drawbacks, the S-SAE method proposes an implementation by parts, working into different segments of data. The segmentation concept has been already introduced to some feature extraction and data reduction methods, such as PCA [62, 116] (see Section 4.3.2). Even though this concept is simple, this is actually an advantage, as it has been proved to be effective and easy to implement.

The schematic representation of the S-SAE method can be found in Figure 7.4. The original profile \mathbf{p} is divided into G different segments or regions $\mathbf{p}_g | g \in [1, G]$ along its spectral content, where some considerations about the segmentation process are described later (Section 7.4.2). Independent SAEs are applied to each segment.

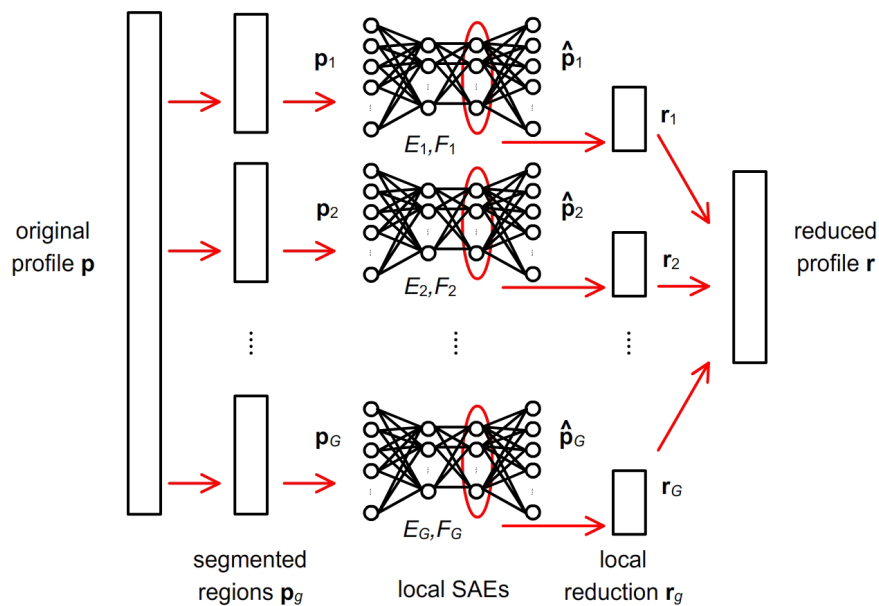


Fig. 7.4 Schematic representation of a S-SAE using two-layer SAEs

The independent SAEs have to deal with a smaller region of the spectrum as input, leading to reduced number of hidden nodes (E_g, F_g), presenting a simpler architecture, with less complexity involved, and focused on the feature extraction of a particular region, with potential enhancement in the extraction process.

Even though S-SAE consists of several independent SAEs, they present simpler NN architecture and, therefore, the global complexity is reduced. Additionally, the abstraction can be achieved more easily. At the output of the S-SAE network, the reduced features from each local region $\mathbf{r}_g | g \in [1, G]$ are concatenated $F = \sum_{g=1}^G F_g$ forming the final reduced feature vector.

7.4 Analysis and Evaluation

In the following experiments, the main objective is to compare the conventional SAE method to the S-SAE proposal, so the performance is evaluated in terms of classification accuracy and also computational complexity (number of interconnections). Moreover, some classical data reduction methods are included in the classification accuracy comparison, all under MATLAB [48] environment.

7.4.1 Experiments

Following the workflow defined in Section 2.3.1, information about (A) (B) the data employed, (C) the feature extraction methods implemented, and (D) classifier modelling are included in the experimental setup, being described as follows.

A. Data Description

In this case, 2 data sets (natural and urban) are selected for a brief analysis, having been acquired by different sensors. These are:

1. 92AV3C [132] (Appendix A.1).
2. Pavia CA [36] (Appendix A.5).

B. Data Conditioning

Some noisy spectral bands suggested in Appendix A are removed again for consistency. With relation to the number of labelled classes employed for land-cover classification, it is found appropriate to use all of them, i.e., 16 and 7 classes for 92AV3C and Pavia CA, respectively.

C. Feature Extraction

The 2 main methods to be compared here are SAE and S-SAE. Since most DL algorithms are generally based on empirical development, the configuration adopted for these 2 methods is not described here but in Section 7.4.2, as part of the results.

A wider comparison is made with some classical data reduction methods, including PCA [12], Independent Component Analysis (ICA) [69], and Maximum Noise Fraction (MNF) [71]. In this particular evaluation, the Baseline case (original feature space) is ignored, as only data reduction methods are compared. The number of extracted features F in all methods is 5, 10, 15, and 20.

D. Classification

Support Vector Machine (SVM) from LIBSVM [45] with Radial Basis Function (RBF) Gaussian kernel is implemented (training ratio of 5%) according to the considerations in Section 2.3.2. The performance is assessed in terms of Overall Accuracy (OA), Average Accuracy (AA), and Class by Class (CbC) accuracy values [20, 49], including McNemar's test of significance [50] having the PCA method as a reference. On the other hand, the complexity comparison between the SAE and the S-SAE methods is simply carried out by evaluating the total number of neural interconnections required.

7.4.2 Results

Due to the empirical nature of the DL related algorithms and methods, the first part of the experiments is focused on defining an (A) appropriate configuration to tune the SAE method. This basic configuration is also considered in S-SAE for fair comparisons. Then, subsequent (B) configuration of the S-SAE proposal is undertaken, with (C) classification accuracy and (D) complexity comparison between SAE and S-SAE.

A. Configuration for SAE

Firstly, the conventional SAE has to be properly tuned, so a good performance in terms of classification accuracy can be ensured. In order to find out adequate configuration parameters, the features obtained from several different SAE configurations are employed in classification tasks, leading to the results shown in Table 7.1.

From the obtained results, it is worth to highlight that larger layer depth or higher number of hidden units not necessarily means better features or classification OA (%), as already claimed in [97]. In fact, the best result for both data sets is obtained using only 2 layers, which indicates that under the current conditions it may be difficult to achieve a good level of abstraction. With relation to the number of hidden nodes, the intermediate case with 40 units performs the best.

Table 7.1 Mean OA for SAE configuration ($F = 10$)

Number of units (E)	Number of layers				
	2	3	4	5	6
92AV3C					
20	68.33	68.84	66.62	60.39	59.24
40	74.01	68.87	69.43	67.26	65.83
60	71.84	69.93	69.13	67.90	67.06
Pavia CA					
20	97.06	96.92	96.87	96.52	96.71
40	97.16	96.77	96.71	96.75	96.77
60	96.69	97.00	96.98	96.78	96.95

According to these considerations, the configuration for conventional SAE is defined by a two-layer scheme and 40 nodes in the first layer, implemented in the same terms for both data sets. This is described in Table 7.2, where the implementations are based on scaled conjugate gradient backpropagation, with a sigmoid activation function and a total of 2000 iterations (epochs) as a maximum limit during the training processes. These values are found appropriate for fast experiments and results.

Table 7.2 Configuration implemented for SAE

Region	Layer/nodes		Reduced features (F)
Original profile (N_λ)	1 st	$E=40$	5, 10, 15, 20
	2 nd	F	

B. Configuration for S-SAE

Once the conventional case is ready, subsequent configuration of the S-SAE method is required. The most important consideration in S-SAE is the number of segments G introduced and how to define them. Bearing in mind previous research on the seg-

mentation concept, this procedure can be related to the correlation values among the spectral bands in the hyperspectral cube.

As suggested in [62], the matrix correlation among the spectral values can be employed to work out the segmentation regions. Indeed, the correlation matrices obtained for 92AV3C and Pavia CA (Figure 7.5 and Figure 7.6, respectively), show a great visualisation of correlated regions (white=1 or -1, black=0). Hence, following the main diagonal regions and discarding the rest without loss of important information, an approximated segmentation is achieved, indicated in Table 7.3 for both data sets.

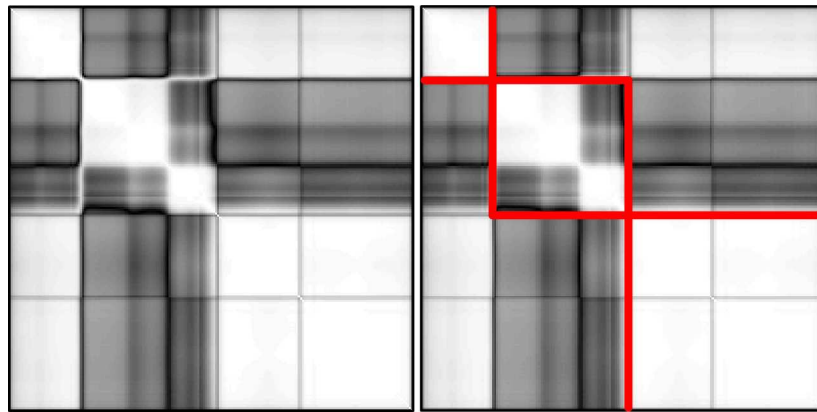


Fig. 7.5 Correlation matrix with selected regions for 92AV3C

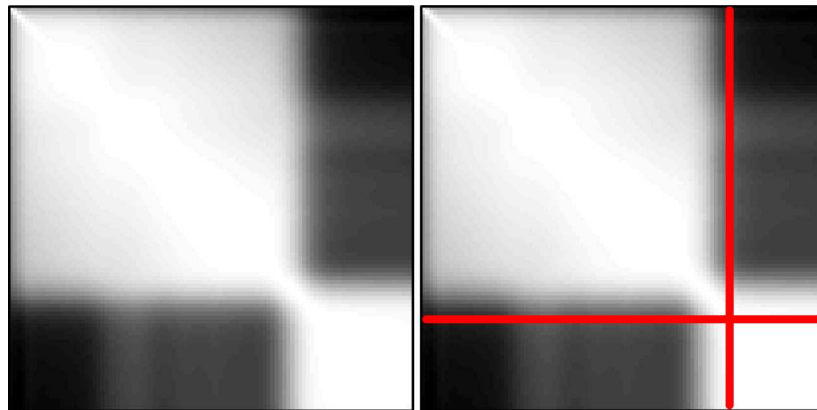


Fig. 7.6 Correlation matrix with selected regions for Pavia CA

The number of layers in each particular SAE is the same as in the conventional case, i.e., 2 for fair comparisons. Then, with relation to the number of nodes in the first layer (a total of 40 in the conventional case), these are equally divided into the

number of proposed segments (G). For the second layer, the total number of desired features (5, 10, 15, and 20) is also divided evenly among the number of segmented regions when it is possible, otherwise, remaining features are assigned to the widest region, where $F = \sum_{g=1}^G F_g$ in all cases.

Table 7.3 Configuration implemented for S-SAE

Region	Range	Layer/nodes		Features			
92AV3C							
Segment (N_λ) ₁	1-35	1 st 2 nd	$E_1=13$ F_1	1	3	5	6
Segment (N_λ) ₂	36-104	1 st 2 nd	$E_2=13$ F_2	2	3	5	7
Segment (N_λ) ₃	105-200	1 st 2 nd	$E_3=13$ F_3	2	4	5	7
Pavia CA							
Segment (N_λ) ₁	1-80	1 st 2 nd	$E_1=20$ F_1	3	5	8	10
Segment (N_λ) ₂	81-102	1 st 2 nd	$E_2=20$ F_2	2	5	7	10

C. Classification Accuracy

From the configurations proposed, both SAE and S-SAE are evaluated in land-cover classification, being also compared to the PCA, ICA, and the MNF data reduction methods. The main results are shown in Figure 7.7 and Figure 7.8, for the 92AV3C and the Pavia CA data sets, respectively. Moreover, Table 7.4 provides a summary with the best results achieved by every method.

Table 7.4 Mean OA and McNemar's test [\mathcal{L}] (F) (best OAs)

Method	92AV3C	Pavia CA
PCA	77.01 [-0.00] (15)	97.06 [-0.00] (5)
ICA	76.90 [-0.21] (20)	96.93 [-1.27] (5)
MNF	78.03 [+2.14] (10)	97.16 [+0.11] (5)
SAE	74.01 [-6.07] (10)	97.29 [+1.07] (15)
S-SAE	80.66 [+8.14] (20)	97.42 [+1.60] (5)

In the 92AV3C results, the conventional implementation of SAE seems to perform not so well as the rest of methods, especially the MNF approach. On the other hand,

in the Pavia CA case, the SAE method achieves the second best result. More importantly, the proposed S-SAE clearly beats the conventional implementation, proving the enhanced efficacy of features when obtained locally. In addition, the S-SAE proposal also outperforms the rest of methods.

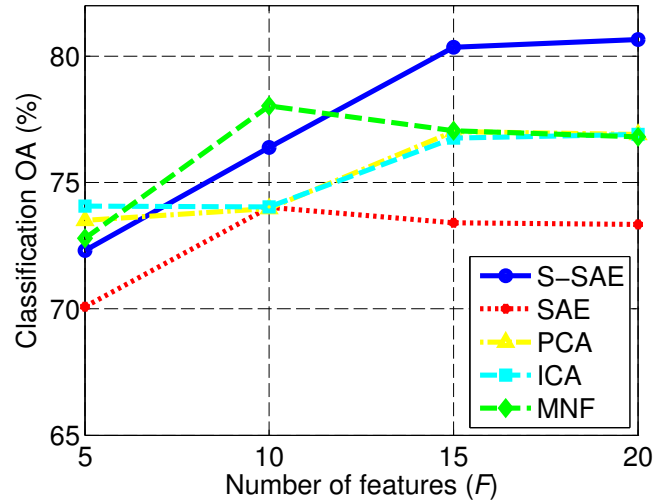


Fig. 7.7 Mean OA for 92AV3C ($F = 5$ to 20)

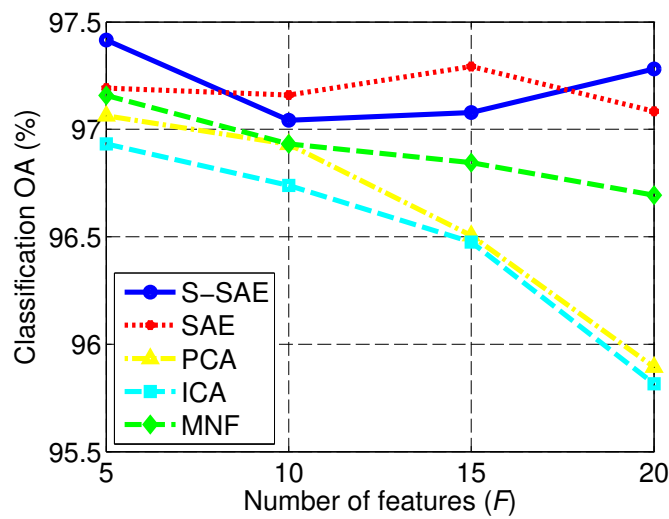


Fig. 7.8 Mean OA for Pavia CA ($F = 5$ to 20)

Complementary accuracy measurements, CbC and AA, are provided in Table 7.5 and Table 7.6 for the 92AV3C and the Pavia CA data sets, respectively. These values again support the S-SAE method, where similar or better accuracy is obtained for the individual labelled classes from the scenes.

Some issues have arisen from this set of results, as the MNF technique is detected to perform better in few classes, probably due to the noise avoidance related to that method. Moreover, SAE marginally outperforms the accuracy of the S-SAE method for some classes with small Number of Samples (NoS) available. Even though these issues have no negative impact on the proposed S-SAE, it can be interesting to further explore these cases in future.

Table 7.5 Mean OA, CbC, and AA for 92AV3C

Class	NoS	PCA	ICA	MNF	SAE	S-SAE
■ (1)	54	50.20	46.86	65.88	63.53	55.29
■ (2)	1434	74.58	77.83	72.10	65.62	78.91
■ (3)	834	60.88	59.14	70.27	56.69	68.17
■ (4)	234	46.26	45.32	48.42	43.60	54.96
■ (5)	497	88.39	88.26	86.95	86.55	88.92
■ (6)	747	92.50	93.10	90.27	89.61	93.07
■ (7)	26	67.50	60.83	65.00	60.42	59.17
■ (8)	489	97.87	96.96	93.00	94.57	96.01
■ (9)	20	24.21	18.95	34.21	38.95	36.32
■ (10)	968	62.30	62.81	72.46	67.31	74.00
■ (11)	2468	77.95	76.99	81.17	78.07	82.98
■ (12)	614	67.86	65.54	63.71	57.87	71.99
■ (13)	212	98.31	98.41	94.38	93.23	96.67
■ (14)	1294	93.87	92.05	94.27	92.10	93.69
■ (15)	380	51.50	57.48	35.98	35.93	44.88
■ (16)	95	81.33	78.22	87.11	81.00	83.00
AA (%)		70.97	69.92	72.20	69.07	73.63
OA (%)		77.01	76.90	78.03	74.01	80.66

Table 7.6 Mean OA, CbC, and AA for Pavia CA

Class	NoS	PCA	ICA	MNF	SAE	S-SAE
■ (1)	447	100	100	100	100	100
■ (2)	28	23.46	21.92	23.85	18.08	23.08
■ (3)	347	83.43	83.47	89.24	86.57	87.84
■ (4)	1213	96.23	95.80	95.82	96.07	96.43
■ (5)	3512	98.73	98.63	98.66	99.07	98.86
■ (6)	893	97.59	97.67	96.85	97.09	97.63
■ (9)	43	99.75	99.50	100	99.75	100
AA (%)		85.60	85.28	86.34	85.23	86.26
OA (%)		97.06	96.93	97.16	97.29	97.42

D. Computational Complexity

The local exploitation of different regions along the spectrum, with enhanced features extracted and better classification accuracy, is not the unique advantage of the proposed S-SAE technique. Thanks to this method, the global complexity in the NN architectures is also reduced.

Despite having not only one but several SAEs implemented, the total number of neural interconnections among all nodes is reduced in the S-SAE method. Now, for every individual SAE related to each spectral region, the number of hidden units and corresponding connections is limited, which contrasts with the conventional case, as a higher number of nodes interconnected to each other fallouts in multiplicity of these connections.

Taking a look at Figure 7.2, the total number of connections in the conventional SAE method can be easily obtained by the expression $(N_\lambda E + EF + FN_\lambda)$. In the S-SAE case, now the multiplicity of connections is reduced, although several SAEs have to be added together, leading to the expression $\sum_{g=1}^G (N_g E_g + E_g F_g + F_g N_g)$. Therefore, giving some values to these expressions, the total number of required interconnections for the SAE and the S-SAE methods is shown in Table 7.7 under different number of reduced features F .

Table 7.7 Number of neural interconnections in (S)-SAE

Data set	F	SAE	S-SAE	Reduction
92AV3C	5	9200	3030	67.07%
	10	10400	3426	67.06%
	15	11600	3795	67.28%
	20	12800	4225	66.99%
Pavia CA	5	4790	2424	49.39%
	10	5500	2750	50.00%
	15	6210	3134	49.53%
	20	6920	3460	50.00%

In general, the number of features selected involves no considerable effect in the achieved reduction percentage. However, it is possible to see a difference between the 2 data sets evaluated, where the reduction is about 67% for 92AV3C and 50% for Pavia CA. This difference can be explained by the number of segments or regions.

Being $(N_\lambda E + EF + FN_\lambda)$ the expression for the number of connections in conventional SAE, and considering a basic S-SAE case in which all 3 parameters N_λ , E , and F are simply divided by the number of segments G , then the new expression for connections in S-SAE is $G((N_\lambda E)/G^2 + (EF)/G^2 + (FN_\lambda)/G^2)$, or what is the same, $(N_\lambda E + EF + FN_\lambda)/G$, where the conventional complexity is merely divided by the number of regions or segments G . As $G = 3$ and $G = 2$ for the 92AV3C and the Pavia CA data sets, respectively, the reduction is obviously related to saving factors of 3 and 2, a remarkable reduction given by just few segmented regions.

7.5 Summary

In this chapter, some preliminary study using DL for feature extraction in HSI is investigated, where SAEs in particular are claimed to be an effective method for feature extraction and data reduction in hyperspectral remote sensing.

Given that the conventional implementation of the SAE method suffers from extremely high computational cost and low efficacy in feature extraction, the well-known concept of segmentation is introduced in this context. Actually, the proposed S-SAE technique applies the same algorithm but focusing on different regions along the spectrum of HSI pixels, working on them in an individual manner with local SAEs. By dividing the spectral domain into several few regions, the individual SAEs from the S-SAE method are found to provide locally extracted features with enhanced capabilities. Besides, the new NN architecture employing few local SAEs is able to reduce the global complexity in terms of node connections.

In summary, the introductory evaluation performed in this chapter demonstrates great potential of DL methods and related variants, pointing to future research lines in effective feature extraction in hyperspectral remote sensing and Earth observation.

Chapter 8

Conclusions and Future Work

8.1 Introduction

The present thesis has focused on methodologies for feature extraction and data reduction in hyperspectral remote sensing Earth observation. The contributions cover a wide range of aspects such as reduction in computational complexity, including onsite and real-time applications, and improved efficiency and efficacy in extracting features from the spectral- and also the spatial-domain information of hyperspectral cubes.

Feature extraction and data reduction in hyperspectral remote sensing is always a challenging problem and has been intensively investigated for decades. However, the problem remains unsolved, particularly due to the difficulty in processing the combined spectral and spatial data. Firstly, the computational complexity involved in many cases is not feasible for portable or embedded applications, where more efficient implementations are highly desired. Secondly, some widely used methods seem to provide limited performance in HyperSpectral Imaging (HSI), and variations from or extensions to them can be beneficial. Thirdly, most of the existing techniques focus mainly on the spectral-domain information, however, spatial-domain exploitation can lead to enhanced analysis, where methodologies with sensibly limited complexity are required. Finally, learning-based approaches from the Deep Learning (DL) framework, which seems promising, have not been properly assessed yet and need to be explored.

A total of 5 different contributions have been proposed in this thesis to solve these problems, which are summarised along with directions for future investigation.

8.2 Main Contributions

The main contributions of this thesis cover 5 different aspects of feature extraction and data reduction in hyperspectral remote sensing Earth observation, which include the Structured Covariance (SC) method for efficient computation, the Folded Principal Component Analysis (Fd-PCA) technique, 1-D Singular Spectrum Analysis (1D-SSA) and its fast implementation, extended 2D-SSA and its fast implementation, and the Segmented Stacked AutoEncoder (S-SAE) method. A detailed summary of these contributions is highlighted as follows:

1. In Chapter 3, a new method in HSI, SC [11], is proposed. By taking advantage of the 3-D structure of the HSI cube, different partitions of data are employed to sequentially construct a series of partial covariance matrices, which can be accumulated to obtain the overall covariance matrix. The SC implementation reduces memory requirements as the related mathematical operations involve matrices with smaller size. However, the overall complexity remains the same yet differently distributed. The different partitions along the rows, columns, pixels, or bands allow an optimum distribution of complexity that can be selected depending on the way how the data is acquired. In addition, the sequential SC schemes can lead to onsite and even real-time (on-the-fly) feature extraction in portable and embedded HSI devices while the data is acquired.
2. A new feature extraction method, Fd-PCA [13], is presented in Chapter 4. As a variation from PCA, samples are subject to a folding reallocating process and reorganised as a matrix before the conventional PCA is applied. By doing this, the folded samples involve matrices with smaller size, leading to a much reduced computational complexity and memory requirement. Additionally, the properties of the new derived covariance matrix are changed and seem to enable the extraction of local structures along with the global structures in conventional PCA. As a result, the new features are found to provide better discrimination ability and classification accuracy due to the extracted local and global structure information.

3. In Chapter 5, SSA, a recent technique for time series analysis, is applied to feature extraction in HSI. By adapting SSA to HSI and applying it to the spectral domain of cubes, noise is suppressed and samples are enhanced thus the classification accuracy is improved [15]. According to the evaluations, the characteristic of 1D-SSA is categorised into 3 different regions, i.e., noisy, stable, and lossy, where the performance depends on the configuration parameters L and Eigen-Value Grouping (EVG). Moreover, a fast implementation of 1D-SSA, Fast-1D-SSA (F-1D-SSA), is also proposed [16], where a unique set of eigenvectors is employed in the pixel-level implementation, leading to reduced computational complexity but similar features and classification accuracy.
4. A novel extension to 1D-SSA, the 2D-SSA method, is proposed and discussed in Chapter 6 for exploiting spatial information in HSI feature extraction. The 2D-SSA method [17] is able to extract local structures while avoiding noisy content, being able to produce outstanding classification accuracy, only comparable to the 2-D Empirical Mode Decomposition (2D-EMD) approach, but with much reduced complexity. Additionally, 2D-SSA can be combined with a basic PCA, namely 2D-SSA-PCA, where both spatial and spectral domains are conveniently exploited, which results in even higher classification accuracy. Furthermore, a fast implementation F-2D-SSA (journal publication under review) is also proposed, with similar efficacy in feature extraction but with much less complexity, including reductions of up to 60% for the computation time in some cases.
5. Finally, in Chapter 7 an introductory evaluation on SAEs is undertaken for HSI data reduction and feature extraction, where a new method called S-SAE is proposed (journal publication under review). As a variation of the SAE method, S-SAE enables extraction of local spectral information for improved classification and also reduced complexity in terms of much less neural connections needed. The preliminary assessment results have shown very promising performance and suggest that further investigation on this topic may lead to remarkable outcomes.

8.3 Future Work

Following the work derived from the present thesis, even though the contributions have achieved a certain level of success, several challenges are still there and can be translated into potential improvements and further investigation as summarised below:

1. The introduced SC implementations include both offline and real-time computations, which have been theoretically proved to work well through simulations. Therefore, the next step is to apply these techniques to specific real-application systems. In addition, more functionalities such as real-time calibration or correction procedures can be added as a further extension of these methodologies.
2. With relation to the Fd-PCA method, there are some issues to be evaluated that may improve the current algorithm, especially in dealing with uneven width of segments, in accordance with results of band clustering or band grouping. Furthermore, the folding procedure parameters may be better optimised by other criteria, which can be assessed using PCA or even the nonlinear-PCA [66].
3. Regarding the 1D-SSA technique, a common setting is utilised for all samples. This can be extended to spectral-based configurations that may lead to even better results, i.e., the parameters L and EVG can be adapted to the different spectral bands in the HSI cube. How to optimally tune these parameters can be further investigated.
4. For the 2D-SSA method, a similar investigation as in the 1-D case is expected with relation to the spectral-based configurations, where it is obvious that different spectral scenes would require different parameters in order to optimise the feature extraction. Moreover, combination of 2D-SSA with spectral-domain feature extraction methods is also a clear line to follow in the future.
5. Finally, as indicated by the promising results from preliminary assessments, the performance of the SAE-based methods is still under comprehensive investigation. It is foreseeable that the combination of DL, sparse representation, SSA, and other approaches such as compressive sensing [142, 143] will further advance the feature extraction and data reduction in HSI.

References

- [1] T. Qiao, J. Ren, C. Craigie, J. Zabalza, C. Maltin, and S. Marshall, “Quantitative prediction of beef quality using visible and NIR spectroscopy with large data samples under industry conditions,” *Journal of Applied Spectroscopy*, vol. 82, no. 1, pp. 137–144, Jan. 2015.
- [2] T. Kelman, J. Ren, and S. Marshall, “Effective classification of Chinese tea samples in hyperspectral imaging,” *Artificial Intelligence Research*, vol. 2, no. 4, pp. 87–96, Oct. 2013.
- [3] N. Neittaanmäki-Perttu, M. Grönroos, T. Tani, I. Pölönen, A. Ranki, O. Saksela, and E. Snellman, “Detecting field cancerization using a hyperspectral imaging system,” *Lasers in Surgery and Medicine*, vol. 45, no. 7, pp. 410–417, Sep. 2013.
- [4] Z. Liu, J. Yan, D. Zhang, and Q.-L. Li, “Automated tongue segmentation in hyperspectral images for medicine,” *Applied Optics*, vol. 46, no. 34, pp. 8328–8334, Dec. 2007.
- [5] G. Lu and B. Fei, “Medical hyperspectral imaging: a review,” *Journal of Biomedical Optics*, vol. 19, no. 1, pp. 010 901/1–010 901/23, Jan. 2014.
- [6] P.-Y. Sacré, C. D. Bleye, P.-F. Chavez, L. Netchacovitch, P. Hubert, and E. Ziemons, “Data processing of vibrational chemical imaging for pharmaceutical applications,” *Journal of Pharmaceutical and Biomedical Analysis*, vol. 101, pp. 123–140, Apr. 2014.
- [7] G. Reed, K. Savage, D. Edwards, and N. N. Daeid, “Hyperspectral imaging of gel pen inks: An emerging tool in document analysis,” *Science and Justice*, vol. 54, pp. 71–80, Sep. 2014.
- [8] K. Gill, J. Ren, S. Marshall, S. Karthick, and J. Gilchrist, “Quality-assured fingerprint image enhancement and extraction using hyperspectral imaging,” in *IET 4th International Conference on Imaging for Crime Detection and Prevention (ICDP)*, Nov. 2011.

- [9] B. Rafert, R. G. Sellar, E. Holbert, J. H. Blatt, D. W. Tyler, S. E. Durham, and H. D. Newby, "Hyperspectral imaging Fourier transform spectrometers for astronomical and remote sensing observations," in *Proceedings SPIE 2198, Instrumentation in Astronomy VIII*, Jun. 1994, pp. 338–349.
- [10] J. B. Rafert, "Before hyperspectral ruled the world," in *Hyperspectral Imaging and Applications Conference*, Oct. 2014.
- [11] J. Zabalza, J. Ren, J. Ren, Z. Liu, and S. Marshall, "Structured covariance principal component analysis for real-time onsite feature extraction and dimensionality reduction in hyperspectral imaging," *Applied Optics*, vol. 53, no. 20, pp. 4440–4449, Jul. 2014.
- [12] C. Rodarmel and J. Shan, "Principal component analysis for hyperspectral image classification," *Surveying and Land Information Science*, vol. 62, no. 2, pp. 115–122, 2002.
- [13] J. Zabalza, J. Ren, M. Yang, Y. Zhang, J. Wang, S. Marshall, and J. Han, "Novel Folded-PCA for improved feature extraction and data reduction with hyperspectral imaging and SAR in remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 93, pp. 112–122, Jul. 2014.
- [14] N. Golyandina and A. Zhigljavsky, *Singular Spectrum Analysis for Time Series*. Springer, 2013.
- [15] J. Zabalza, J. Ren, Z. Wang, S. Marshall, and J. Wang, "Singular spectrum analysis for effective feature extraction in hyperspectral imaging," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 11, pp. 1886–1890, Nov. 2014.
- [16] J. Zabalza, J. Ren, Z. Wang, H. Zhao, J. Wang, and S. Marshall, "Fast implementation of singular spectrum analysis for effective feature extraction in hyperspectral imaging," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, no. 99, Dec. 2014.
- [17] J. Zabalza, J. Ren, J. Zheng, J. Han, H. Zhao, S. Li, and S. Marshall, "Novel two-dimensional singular spectrum analysis for effective feature extraction and data classification in hyperspectral imaging," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4418–4433, Aug. 2015.
- [18] B. Demir and S. Ertürk, "Empirical mode decomposition of hyperspectral images for support vector machine classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4071–4084, Nov. 2010.

- [19] NASA Earth Observatory: Remote Sensing. [Online] Available at: <http://earthobservatory.nasa.gov/Features/RemoteSensing/>.
- [20] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis*. Springer, 2006.
- [21] T. L. Roush, "Mars: Remote sensing," in *Encyclopedia of Planetary Science*. Springer, 1997, pp. 459–461.
- [22] M. Montopoli, P. Tognolatti, F. Marzano, M. Pierdicca, and G. Perrotta, "Remote sensing of the Moon sub-surface from a spaceborne microwave radiometer aboard the European Student Moon Orbiter (ESMO)," in *IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2007, pp. 4451–4454.
- [23] EO Portal: Satellite Missions, Landsat-1. [Online] Available at: <https://directory.eoportal.org/web/eoportal/satellite-missions//landsat-1-3>.
- [24] ESA: Observing the Earth, Copernicus. [Online] Available at: http://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Overview4.
- [25] S. E. Craig, S. E. Lohrenz, Z. Lee, K. L. Mahoney, G. J. Kirkpatrick, O. M. Schofield, and R. G. Steward, "Use of hyperspectral remote sensing reflectance for detection and assessment of the harmful alga, *Karenia brevis*," *Applied Optics*, vol. 45, no. 21, pp. 5414–5425, Jul. 2006.
- [26] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, Jan. 1968.
- [27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, Mar. 1995.
- [28] X. Jia, B.-C. Kuo, and M. Crawford, "Feature mining for hyperspectral image classification," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 676–697, Mar. 2013.
- [29] M. Pal and G. Foody, "Feature selection for classification of hyperspectral data by SVM," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2297–2307, May 2010.
- [30] R. O. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J. Chovit, M. Solis, M. R. Olah, and O. Williams, "Imaging spectroscopy and the Airborne Visible/Infrared Imaging

- Spectrometer (AVIRIS),” *Remote Sensing of Environment*, vol. 65, no. 3, pp. 227–248, Jul. 1998.
- [31] Jet Propulsion Laboratory. [Online] Available at: <http://www.jpl.nasa.gov/>.
- [32] A. A. Mueller, A. Hausold, and P. Strobl, “HySens-DAIS/ROSIIS imaging spectrometers at DLR,” in *The 3rd EARSeL Workshop on Imaging Spectroscopy*, May 2003, pp. 3–14.
- [33] “The airborne imaging spectrometer ROSIS,” DLR Institute, Tech. Rep., 2000.
- [34] J. Pearlman, P. Barry, C. Segal, J. Shepanski, D. Beiso, and S. Carman, “Hyperion, a space-based imaging spectrometer,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 6, pp. 1160–1173, Jun. 2003.
- [35] R. Archibald and G. Fann, “Feature selection and classification of hyperspectral images with support vector machines,” *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 674–677, Oct. 2007.
- [36] Hyperspectral Remote Sensing Scenes, Computational Intelligence Group from the Basque University (UPV/EHU). [Online] Available at: http://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.
- [37] F. Melgani and L. Bruzzone, “Classification of hyperspectral remote sensing images with support vector machines,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.
- [38] J. Zabalza, J. Ren, C. Clemente, G. Di Caterina, and J. J. Soraghan, “Embedded SVM on TMS320C6713 for signal prediction in classification and regression applications,” in *IEEE European DSP Education and Research Conference (EDERC)*, Sep. 2012, pp. 90–94.
- [39] J. Zabalza, C. Clemente, G. Di Caterina, J. Ren, J. J. Soraghan, and S. Marshall, “Robust PCA micro-doppler classification using SVM on embedded systems,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 3, pp. 2304–2310, Jul. 2014.
- [40] B. Waske and J. A. Benediktsson, “Fusion of support vector machines for classification of multisensor data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 3858–3866, Dec. 2007.
- [41] I. Dópido, A. Villa, A. Plaza, and P. Gamba, “A comparative assessment of several processing chains for hyperspectral image classification: What features

- to use?” in *IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Jun. 2011.
- [42] M. Rojas, I. Dópido, A. Plaza, and P. Gamba, “Comparison of support vector machine-based processing chains for hyperspectral image classification,” in *Proceedings SPIE 7810, Satellite Data Compression, Communications, and Processing*, Aug. 2010.
- [43] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, Mar. 2002.
- [44] C.-W. Hsu and C.-J. Lin, BSVM library. [Online] Available at: <http://www.csie.ntu.edu.tw/~cjlin/bsvm/>.
- [45] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011, [Online] Available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [46] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, *A Practical Guide to Support Vector Classification*, Department of Computer Science, National Taiwan University, 2003.
- [47] C. Saunders, M. O. Stitson, J. Weston, L. Bottou, and A. Smola, *Support Vector Machine - Reference Manual*, Department of Computer Science, Royal Holloway, University of London, 1998.
- [48] MATLAB, version 8.00.0 (R2012b). Natick, Massachusetts: The MathWorks Inc., 2012.
- [49] R. G. Congalton and K. Green, *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*. CRC press, 2008.
- [50] G. Foody, “Thematic map comparison: evaluating the statistical significance of differences in classification accuracy,” *Photogrammetric Engineering and Remote Sensing*, vol. 70, no. 5, pp. 627–633, May 2004.
- [51] R. G. Pontius Jr and M. Millones, “Death to kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment,” *International Journal of Remote Sensing*, vol. 32, no. 15, pp. 4407–4429, Aug. 2011.
- [52] Y. Gao, P. Marpu, I. Niemeyer, D. M. Runfola, N. M. Giner, T. Hamill, and R. G. Pontius Jr, “Object-based classification with features extracted by a

- semi-automatic feature extraction algorithm - SEaTH,” *Geocarto International*, vol. 26, no. 3, pp. 211–226, Jun. 2011.
- [53] T. L. Sohl, B. M. Sleeter, K. L. Sayler, M. A. Bouchard, R. R. Reker, S. L. Bennett, R. R. Sleeter, R. L. Kanengieter, and Z. Zhu, “Spatially explicit land-use and land-cover scenarios for the Great Plains of the United States,” *Agriculture, Ecosystems and Environment*, vol. 153, pp. 1–15, Feb. 2012.
- [54] A. Korobeynikov, *Computation- and Space-Efficient Implementation of SSA*, Department of Statistical Modelling, Saint Petersburg State University, 2009.
- [55] J. E. Jackson, *A User’s Guide to Principal Components*. John Wiley & Sons, 1991.
- [56] R. Jošth, J. Antikainen, J. Havel, A. Herout, P. Zemčík, and M. Hauta-Kasari, “Real-time PCA calculation for spectral imaging (using SIMD and GP-GPU),” *Journal of Real-Time Image Processing*, vol. 7, no. 2, pp. 95–103, 2012.
- [57] M. Wang, D. Wang, W. Xu, B. Chen, and K. Guo, “Parallel computing of covariance matrix and its application on hyperspectral data process,” in *IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2012, pp. 4058–4061.
- [58] F. Vagni, “Survey of hyperspectral and multispectral imaging technologies,” Research and Technology Organisation, Tech. Rep., 2007.
- [59] F. Ndi, F. Adar, and S. H. Atzeni, “Spectral imaging,” HORIBA, Tech. Rep., 2011.
- [60] C. Wang, M. Menenti, and Z.-L. Li, “Modified principal component analysis (MPCA) for feature selection of hyperspectral imagery,” in *IEEE International Conference on Geoscience and Remote Sensing Symposium, (IGARSS)*, Jul. 2003, pp. 3781–3783.
- [61] P. M. Hall, D. Marshall, and R. R. Martin, “Incremental eigenanalysis for classification,” in *Proceedings of the British Machine Vision Conference*, Sep. 1998, pp. 286–295.
- [62] X. Jia and J. Richards, “Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 1, pp. 538–542, Jan. 1999.

- [63] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, Jan. 2004.
- [64] L. Wang, X. Wang, X. Zhang, and J. Feng, "The equivalence of two-dimensional PCA to line-based PCA," *Pattern Recognition Letters*, vol. 26, no. 1, pp. 57–60, Jan. 2005.
- [65] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- [66] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 3, pp. 447–451, May 2012.
- [67] A. Sharma and K. K. Paliwal, "Fast principal component analysis using fixed-point algorithm," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1151–1155, Jul. 2007.
- [68] Q. Du and J. E. Fowler, "Low-complexity principal component analysis for hyperspectral image compression," *The International Journal of High Performance Computing Applications*, vol. 22, no. 4, pp. 438–448, 2008.
- [69] K. J. Hyvarinen, A. and E. Oja, *Independent Component Analysis*. John Wiley & Sons, 2001.
- [70] J. Wang and C.-I. Chang, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 6, pp. 1586–1600, Jun. 2006.
- [71] A. Green, M. Berman, P. Switzer, and M. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 26, no. 1, pp. 65–74, Jan. 1988.
- [72] C.-I. Chang and Q. Du, "Interference and noise-adjusted principal components analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2387–2396, Sep. 1999.

- [73] S. Robila and L. Maciak, "Sequential and parallel feature extraction in hyperspectral data using nonnegative matrix factorization," in *IEEE Conference on Long Island Systems, Applications, and Technology (LISAT)*, May 2007, pp. 1–7.
- [74] L. O. Jiménez-Rodríguez, E. Arzuaga-Cruz, and M. Vélez-Reyes, "Unsupervised linear feature-extraction methods and their effects in the classification of high-dimensional data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 2, pp. 469–483, Feb. 2007.
- [75] B. Guo, S. Gunn, R. Damper, and J. Nelson, "Band selection for hyperspectral image classification using mutual information," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 522–526, Oct. 2006.
- [76] H. Zhang, J. Li, Y. Huang, and L. Zhang, "A nonlocal weighted joint sparse representation classification method for hyperspectral imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2056–2065, Jun. 2014.
- [77] C. Zhao, X. Li, J. Ren, and S. Marshall, "Improved sparse representation using adaptive spatial support for effective target detection in hyperspectral imagery," *International Journal of Remote Sensing*, vol. 34, no. 24, pp. 8669–8684, Oct. 2013.
- [78] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, Jul. 2002.
- [79] S. Marchesi and L. Bruzzone, "ICA and kernel ICA for change detection in multispectral remote sensing images," in *IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2009, pp. II980–II983.
- [80] Y. Chen, M. M. Crawford, and J. Ghosh, "Improved nonlinear manifold learning for land cover classification via intelligent landmark selection," in *IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS)*, Aug. 2006, pp. 545–548.
- [81] M. Crawford and W. Kim, "Manifold learning for multi-classifier systems via ensembles," in *Multiple Classifier Systems*. Springer, 2009, pp. 519–528.
- [82] J. Yin, C. Gao, and X. Jia, "Using Hurst and Lyapunov exponent for hyperspectral image feature extraction," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 4, pp. 705–709, Jul. 2012.

- [83] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
- [84] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 862–873, Mar. 2009.
- [85] C. Lee and D. A. Landgrebe, "Decision boundary feature extraction for neural networks," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 75–83, Jan. 1997.
- [86] B.-C. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 5, pp. 1096–1105, May 2004.
- [87] J. Ren, J. Zabalza, S. Marshall, and J. Zheng, "Effective feature extraction and data reduction in remote sensing using hyperspectral imaging," *IEEE Signal Processing Magazine*, vol. 31, no. 4, pp. 149–154, Jul. 2014.
- [88] L. M. Bruce, C. H. Koger, and J. Li, "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 10, pp. 2331–2338, Oct. 2002.
- [89] T. Qiao, J. Ren, C. Craigie, S. Marshall, and C. Maltin, "Effective compression of hyperspectral imagery using improved three dimensional discrete cosine transform," in *IEEE Millimeter Waves and THz Technology Workshop (UCMMT)*, Sep. 2013.
- [90] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.
- [91] N. E. Huang, S. R. Long, and Z. Shen, "The mechanism for frequency downshift in nonlinear wave evolution," *Advances in applied mechanics*, vol. 32, pp. 59–117, 1996.
- [92] H. Huang and J. Pan, "Speech pitch determination based on Hilbert-Huang transform," *Signal Processing*, vol. 86, no. 4, pp. 792–803, 2006.
- [93] N. Golyandina and D. Stepanov, "SSA-based approaches to analysis and forecast of multidimensional time series," in *Proceedings of the 5th St. Petersburg Workshop on Simulation*, 2005, pp. 193–198.

- [94] B. Hu, Q. Li, and A. Smith, "Noise reduction of hyperspectral data using singular spectral analysis," *International Journal of Remote Sensing*, vol. 30, no. 9, pp. 2277–2296, May 2009.
- [95] B. Yegnanarayana, *Artificial Neural Networks*. PHI Learning, 2009.
- [96] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2271–2282, May 2010.
- [97] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.
- [98] G. A. Licciardi and F. Del Frate, "Pixel unmixing in hyperspectral data by means of neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4163–4172, Nov. 2011.
- [99] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction of hyperspectral images," in *IEEE Workshop on Hyperspectral Image and Signal Processing (WHISPERS)*, Jun. 2014.
- [100] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 5132–5136.
- [101] M. Midhun, S. R. Nair, V. Prabhakar, and S. S. Kumar, "Deep model for classification of hyperspectral image using restricted Boltzmann machine," in *Proceedings of the International Conference on Interdisciplinary Advances in Applied Computing (ICONIAAC)*, Oct. 2014.
- [102] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: high confidence predictions for unrecognizable images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [103] F. Dell'Acqua, P. Gamba, A. Ferrari, J. Palmason, J. Benediktsson, and K. Arnason, "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 4, pp. 322–326, Oct. 2004.
- [104] J. Benediktsson, J. Palmason, and J. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 480–491, Mar. 2005.

- [105] M. Fauvel, J. Benediktsson, J. Chanussot, and J. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.
- [106] R. Phillips, C. Blinn, L. Watson, and R. Wynne, "An adaptive noise-filtering algorithm for AVIRIS data with implications for classification accuracy," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 9, pp. 3168–3179, Sep. 2009.
- [107] K. Perumal and R. Bhaskaran, "SVM-based effective land use classification system for multispectral remote sensing images," *International Journal of Computer Science and Information Security*, vol. 6, no. 2, pp. 97–105, 2009.
- [108] A. Pizurica and W. Philips, "Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 654–665, Mar. 2006.
- [109] B. Demir and S. Ertürk, "Improved hyperspectral image classification with noise reduction pre-process," in *IEEE European Signal Processing Conference (EU-SIPCO)*, Aug. 2008.
- [110] B. Demir, S. Ertürk, and M. Gullu, "Hyperspectral image classification using denoising of intrinsic mode functions," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 2, pp. 220–224, Mar. 2011.
- [111] A. Linderhed, "Adaptive image compression with wavelet packets and empirical mode decomposition," Ph.D. dissertation, Linköping University, Image Coding Group, 2004.
- [112] K. Usevich and N. Golyandina, "2D-extension of singular spectrum analysis: Algorithm and elements of theory," in *Matrix Methods: Theory, Algorithms, Applications*. World Scientific, 2010, pp. 449–474.
- [113] N. Golyandina and K. Usevich, "An algebraic view on finite rank in 2D-SSA," in *Proceedings of the 6th St. Petersburg Workshop on Simulation*, Jul. 2009, pp. 308–313.
- [114] L. J. Rodríguez-Aragón and Z. Anatoly, "Singular spectrum analysis for image processing," *Statistics and Its Interface*, vol. 3, no. 3, pp. 419–426, Mar. 2010.
- [115] I. V. Florinsky, "Chapter 6 - filtering," in *Digital Terrain Analysis in Soil Science and Geology*. Academic Press, 2012, pp. 103–132.

- [116] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, Jul. 2010.
- [117] Q. Du and J. Fowler, "Hyperspectral image compression using JPEG2000 and principal component analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 2, pp. 201–205, Apr. 2007.
- [118] M. Zhang and Q. Meng, "Citrus canker detection based on leaf images analysis," in *IEEE International Conference on Information Science and Engineering (ICISE)*, Dec. 2010, pp. 3584–3587.
- [119] D. Zhang and Z.-H. Zhou, "(2D) 2PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1, pp. 224–231, Dec. 2005.
- [120] Y. Zeng, D. Feng, and L. Xiong, "An algorithm of face recognition based on the variation of 2DPCA," *Journal of Computational Information Systems*, vol. 7, no. 1, pp. 303–310, Jan. 2011.
- [121] H. Kong, L. Wang, E. Teoh, X. Li, J. Wang, and R. Venkateswarlu, "Generalized 2D principal component analysis for face image representation and recognition," *Neural Networks*, vol. 18, no. 5-6, pp. 585–594, 2005.
- [122] E. Pang, S. Wang, M. Qu, R. Liu, C. Jia, and Z. Yu, "2D-SPP: a two-dimensional extension of sparsity preserving projections," *Journal of Information and Computational Science*, vol. 9, no. 13, pp. 3683–3692, Nov. 2012.
- [123] S. Yu, J. Bi, and J. Ye, "Probabilistic interpretations and extensions for a family of 2D PCA-style algorithms," in *Workshop on Data Mining Using Matrices Tensors (DMMT)*, Aug. 2008, pp. 1–7.
- [124] S. Wang and C.-I. Chang, "Variable-number variable-band selection for feature characterization in hyperspectral signatures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 9, pp. 2979–2992, Sep. 2007.
- [125] Y.-Q. Zhao, D. Zhang, and S. G. Kong, "Band-subset-based clustering and fusion for hyperspectral imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 2, pp. 747–756, Feb. 2011.
- [126] M. Andric, B. Bondzulich, and B. Zrnica, "The database of radar echoes from various targets with spectral analysis," in *IEEE Symposium on Neural Network Applications in Electrical Engineering (NEUREL)*, Sep. 2010, pp. 187–190.

- [127] Purdue's University Multispec Site: June 12, 1992 AVIRIS image North-South Flightline, Indiana. [Online] Available at: <https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html>.
- [128] N. Golyandina, V. Nekrutkin, and A. Zhigljavsky, *Analysis of Time Series Structure: SSA and Related Techniques*. CRC Press, 2001.
- [129] G. Rilling, P. Flandrin, P. Goncalves *et al.*, "On empirical mode decomposition and its algorithms," in *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Jun. 2003, pp. 8–11.
- [130] D. Broomhead and G. P. King, "Extracting qualitative dynamics from experimental data," *Physica D: Nonlinear Phenomena*, vol. 20, no. 2, pp. 217–236, 1986.
- [131] K. G. Broomhead D, "On the qualitative analysis of experimental dynamical systems," in *Nonlinear Phenomena and Chaos*. Adam Hilger Ltd, 1986, p. 113–144.
- [132] Purdue's University Multispec Site: AVIRIS image Indian Pine Test Site, Jun. 12, 1992. [Online] Available at: <https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html>.
- [133] J. Qin, T. F. Burks, M. A. Ritenour, and W. G. Bonn, "Detection of citrus canker using hyperspectral reflectance imaging with spectral information divergence," *Journal of Food Engineering*, vol. 93, no. 2, pp. 183–191, Jan. 2009.
- [134] S. Piqueras, J. Burger, R. Tauler, and A. de Juan, "Relevant aspects of quantification and sample heterogeneity in hyperspectral image resolution," *Chemometrics and Intelligent Laboratory Systems*, vol. 117, pp. 169–182, 2012.
- [135] G. Rilling, MATLAB codes for EMD. [Online] Available at: <http://perso.ens-lyon.fr/patrick.flandrin/emd.html>.
- [136] F. Van der Heijden, R. Duin, D. de Ridder, and D. M. Tax, *Classification, Parameter Estimation and State Estimation: An Engineering Approach using MATLAB*. John Wiley & Sons, 2004.
- [137] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, no. 5, pp. 403–420, 1970.
- [138] MATLAB Central File Exchange, Bidimensional EMD. [Online] Available at: <http://www.mathworks.com/matlabcentral/fileexchange/28761-bi-dimensional-empirical-mode-decomposition-bemd>.

- [139] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, Jan. 1998.
- [140] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, Jan. 2012.
- [141] N. Kasabov, "Evolving fuzzy neural networks for supervised/unsupervised on-line knowledge-based learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 31, no. 6, pp. 902–918, Dec. 2001.
- [142] T. Sun and K. Kelly, "Compressive sensing hyperspectral imager," in *Computational Optical Sensing and Imaging*, Oct. 2009, p. CTuA5.
- [143] C. Li, T. Sun, K. F. Kelly, and Y. Zhang, "A compressive sensing and unmixing scheme for hyperspectral data processing," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1200–1210, Mar. 2012.

Appendix A

Remote Sensing Data Sets

A.1 92AV3C (Indian Pines) from AVIRIS (NASA)

The 92AV3C data set, also known as Indian Pines in the literature, was acquired in Northwest Indiana, the United States, over an agricultural site. Indeed, it is a small portion or subscene [132] from the original image North-South Flightline [127], made by 145×145 pixels and 220 spectral bands (4 of the initial 224 were not valid) in the range 400 nm - 2500 nm.



















Fig. A.1 92AV3C data set with ground truth

This data set is normally employed for land-use purposes, containing a total of 16 labelled classes related to agriculture, forest, and perennial vegetation. Complete information about the labelled classes and their description is available in Figure A.1

and Table A.1, including the Number of Samples (NoS) or pixels available in each class. It is important to remark that there is an alternative version of this data set in [36], with slightly different ground truth (same classes but different NoS). Both versions are normally employed in the literature, leading to similar results and evaluations. In the present thesis, 92AV3C refers to [132] and the Indian Pines name is given to the version in [36].

Table A.1 92AV3C data set description

Source image		North-South Flightline [127]	
Spatial location		(376-520, 240-384) [132]	
Hyperspectral cube dimension		145 × 145 × 224	
Number of labelled pixels		10366	
Number of background pixels		10659	
Number of classes		16	
Colour	Code	Class name	NoS
	1	Alfalfa	54
	2	Corn-notill	1434
	3	Corn-mintill	834
	4	Corn	234
	5	Grass/Pasture	497
	6	Grass/Trees	747
	7	Grass/Pasture-mowed	26
	8	Hay-windrowed	489
	9	Oats	20
	10	Soybean-notill	968
	11	Soybean-mintill	2468
	12	Soybean-clean	614
	13	Wheat	212
	14	Woods	1294
	15	Building/Grass/Trees/Drives	380
	16	Stone/Steel towers	95

Widely used conditioning is applied to this data set, where it is really common to discard some spectral bands in the regions (104-108, 150-163, 220) [18, 35], resulting in 200 bands for analysis. It is also usual to avoid some labelled classes with few NoS in the ground truth [18, 29], claimed to provide higher statistical significance. Therefore, in some experiments only the labelled classes with code 2, 3, 5, 6, 8, 10, 11, 12, and 14 are employed.

A.2 Indian Pines B from AVIRIS (NASA)

Indian Pines B is another subscene from the original North-South Flightline [127]. Given that the widely used 92AV3C (see Appendix A.1) data set has been proved to be highly adequate for hyperspectral remote sensing experiments, it is found appropriate to extract a different portion of the same original image [127] for complementary analysis in the present dissertation (see Figure A.2).

The spatial location from which to obtain this subscene is defined by the pixels in (902-1051, 201-350). Therefore, it is a 150×150 portion with same spectral bands (200), range and resolution (20 m in the spatial and 10 nm in the spectral domain) than the 92AV3C or Indian Pines. However, a total of 20 classes are now included in the ground truth, mostly related again to agriculture and vegetation (see Table A.2). The class codes are referred to those used in the original North-South Flightline image, unlike the 92AV3C case. The name of some classes is not clearly provided and may differ from other studies.

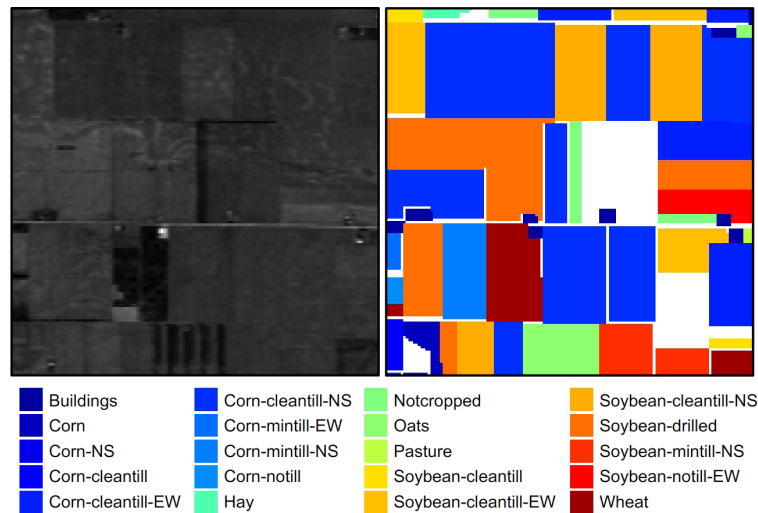






















Fig. A.2 Indian Pines B data set with ground truth

Data conditioning for the Indian Pines B image is quite simple, basically removing some spectral bands in the regions (104-108, 150-163, 220) in the same way as it is suggested for the 92AV3C image [18, 35], given that both subscenes come from the same source image [127].

Table A.2 Indian Pines B data set description

Source image		North-South Flightline [127]	
Spatial location		(902-1051, 201-350)	
Hyperspectral cube dimension		150 × 150 × 224	
Number of labelled pixels		19423	
Number of background pixels		3077	
Number of classes		20	
Colour	Code	Class name	NoS
	2	Buildings	305
	4	Corn	196
	7	Corn-NS	10
	8	Corn-cleantill	147
	9	Corn-cleantill-EW	1485
	10	Corn-cleantill-NS	6666
	14	Corn-mintill-EW	90
	15	Corn-mintill-NS	702
	16	Corn-notill	77
	26	Hay	69
	29	Notcropped	386
	30	Oats	686
	33	Pasture	24
	38	Soybean-cleantill	162
	40	Soybean-cleantill-EW	1206
	41	Soybean-cleantill-NS	1968
	44	Soybean-drilled	3041
	48	Soybean-mintill-NS	704
	50	Soybean-notill-EW	426
	56	Wheat	1073

A.3 Salinas C from AVIRIS (NASA)

Salinas C is a subscene from the original image in [36] collected over Salinas Valley (see Figure A.3) in California, the United States. It presents a size of 150×150 pixels, with 224 spectral bands and a spatial resolution of 3.7 m according to [36]. Its ground truth covers 9 labelled classes related to vegetation (see Table A.3). Spectral bands in (108-112, 154-167, 224) are removed for data conditioning as suggested by [36].

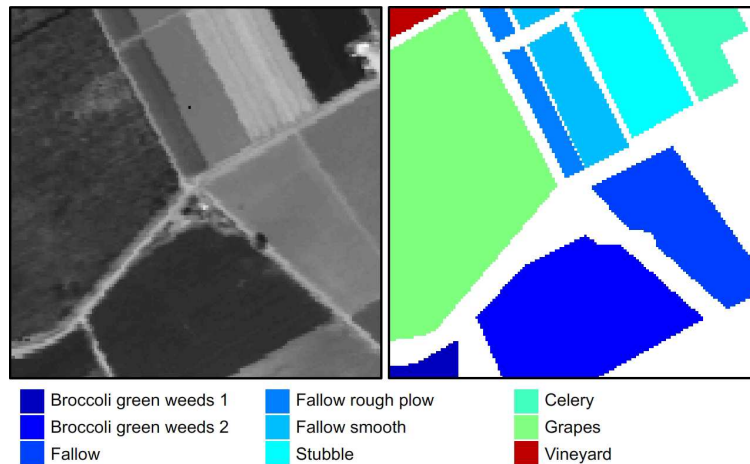


Fig. A.3 Salinas C data set with ground truth

Table A.3 Salinas C data set description

Source image	Salinas Valley [36]		
Spatial location	(91-240, 61-210)		
Hyperspectral cube dimension	$150 \times 150 \times 224$		
Number of labelled pixels	15662		
Number of background pixels	6838		
Number of classes	9		
Colour	Code	Class name	NoS
■	1	Broccoli green weeds 1	240
■	2	Broccoli green weeds 2	3400
■	3	Fallow	1957
■	4	Fallow rough plow	599
■	5	Fallow smooth	1155
■	6	Stubble	1414
■	7	Celery	848
■	8	Grapes	5890
■	15	Vineyard	159

A.4 Pavia University A (Pavia UA) from ROSIS (DLR)

Pavia University image [36] was acquired over Pavia, in North Italy, resulting in a scene with 103 spectral bands valid for analysis and considerable spatial resolution of 1.3 m. A 150×150 portion is extracted for the experiments, where 8 classes are found in the ground truth. These refer to urban elements such as asphalt, bitumen, or shadows (Figure A.4, Table A.4). For this data set, most related publications in the literature employ the available 103 spectral bands, so no particular conditioning is carried out.

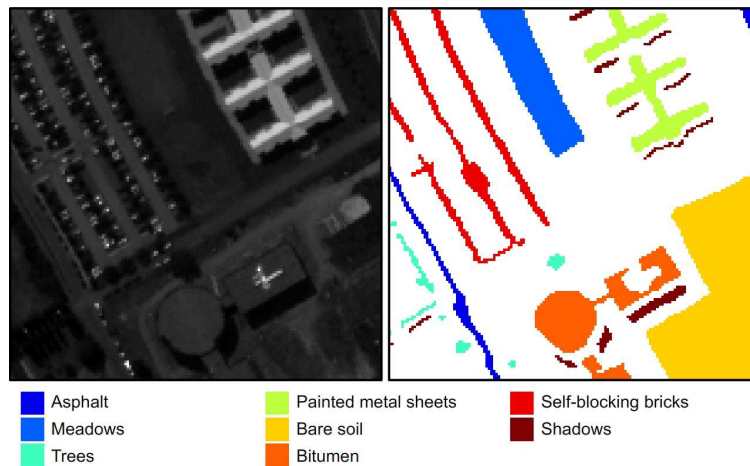










Fig. A.4 Pavia UA data set with ground truth

Table A.4 Pavia UA data set description

Source image		Pavia University [36]	
Spatial location		(201-350, 51-200)	
Hyperspectral cube dimension		$150 \times 150 \times 115$	
Number of labelled pixels		6685	
Number of background pixels		15815	
Number of classes		8	
Colour	Code	Class name	NoS
	1	Asphalt	310
	2	Meadows	957
	4	Trees	154
	5	Painted metal sheets	698
	6	Bare soil	2559
	7	Bitumen	860
	8	Self-blocking bricks	854
	9	Shadows	293

A.5 Pavia Centre A (Pavia CA) from ROSIS (DLR)

Pavia CA (Figure A.5) is a subscene extracted from the original Pavia Centre image [36], with 150×150 spatial size and a total of 102 spectral bands available. Its ground truth includes 7 labelled classes such as water, trees or asphalt (see Table A.5). Similarly to the Pavia UA case (see Appendix A.4), no particular conditioning is applied to this data, as the 102 available bands are commonly employed for land-cover analysis.

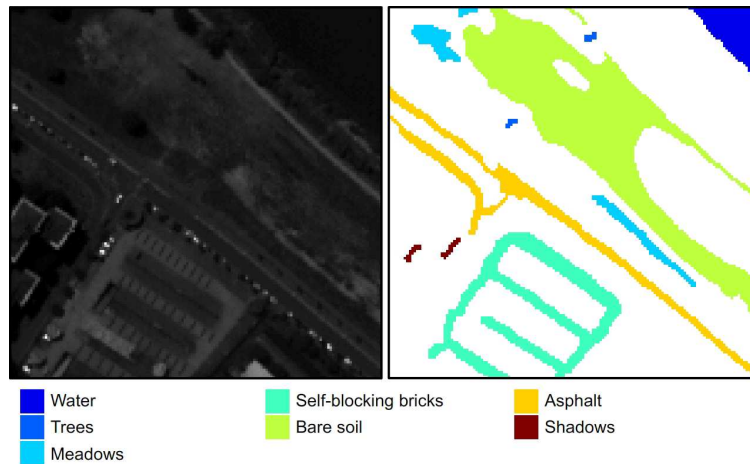


Fig. A.5 Pavia CA data set with ground truth

Table A.5 Pavia CA data set description

Source image	Pavia Centre [36]		
Spatial location	(251-400, 401-550)		
Hyperspectral cube dimension	$150 \times 150 \times 115$		
Number of labelled pixels	6483		
Number of background pixels	16017		
Number of classes	7		
Colour	Code	Class name	NoS
■	1	Water	447
■	2	Trees	28
■	3	Meadows	347
■	4	Self-blocking bricks	1213
■	5	Bare soil	3512
■	6	Asphalt	893
■	9	Shadows	43

A.6 Botswana A from HYPERION (NASA)

The Botswana A image (see Figure A.6) is a 300×75 pixels subscene extracted from the original Botswana data set [36], an image taken over the Okavango Delta, in Botswana. It consists of 242 spectral bands and ground truth with 5 labelled classes related to acacia and mopane vegetation, as shown in Table A.6.

In the data conditioning, several spectral bands are removed due to water absorption regions and noisy content, being available for analysis only 145 bands. These are in the regions (10-55, 82-97, 102-119, 134-164, 187-220) [36].

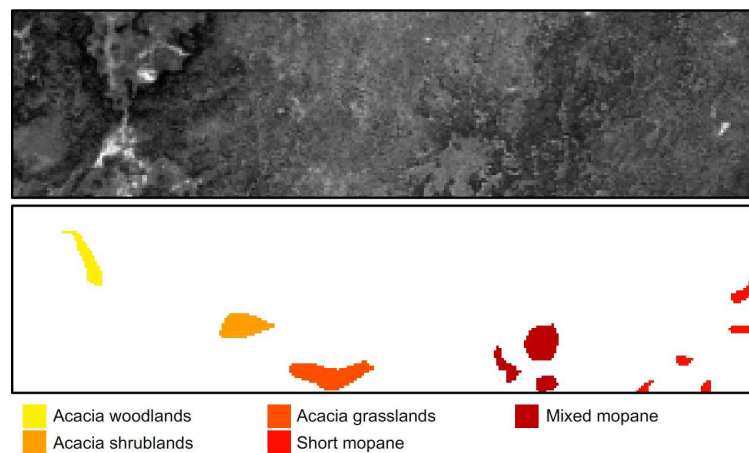


Fig. A.6 Botswana A data set with ground truth (rotated 90° clockwise)

Table A.6 Botswana A data set description

Source image		Botswana [36]	
Spatial location		(291-590, 176-250)	
Hyperspectral cube dimension		$300 \times 75 \times 242$	
Number of labelled pixels		905	
Number of background pixels		21595	
Number of classes		5	
Colour	Code	Class name	NoS
■	9	Acacia woodlands	104
■	10	Acacia shrublands	154
■	11	Acacia grasslands	233
■	12	Short mopane	146
■	13	Mixed mopane	268

Appendix B

Publications by the Author

B.1 Journal Publications

1. J. Zabalza, J. Ren, J. Ren, Z. Liu, and S. Marshall, “Structured covariance principal component analysis for real-time onsite feature extraction and dimensionality reduction in hyperspectral imaging,” *Applied Optics*, vol. 53, no. 20, pp. 4440–4449, Jul. 2014.
2. J. Zabalza, J. Ren, M. Yang, Y. Zhang, J. Wang, S. Marshall, and J. Han, “Novel Folded-PCA for improved feature extraction and data reduction with hyperspectral imaging and SAR in remote sensing,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 93, pp. 112–122, Jul. 2014.
3. J. Zabalza, J. Ren, Z. Wang, S. Marshall, and J. Wang, “Singular spectrum analysis for effective feature extraction in hyperspectral imaging,” *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 11, pp. 1886–1890, Nov. 2014.
4. J. Zabalza, J. Ren, Z. Wang, H. Zhao, J. Wang, and S. Marshall, “Fast implementation of singular spectrum analysis for effective feature extraction in hyperspectral imaging,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, no. 99, Dec. 2014.

5. J. Zabalza, J. Ren, J. Zheng, J. Han, H. Zhao, S. Li, and S. Marshall, "Novel two-dimensional singular spectrum analysis for effective feature extraction and data classification in hyperspectral imaging," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4418–4433, Aug. 2015.
6. J. Zabalza, C. Clemente, G. Di Caterina, J. Ren, J. J. Soraghan, and S. Marshall, "Robust PCA micro-doppler classification using SVM on embedded systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 3, pp. 2304–2310, Jul. 2014.
7. J. Ren, J. Zabalza, S. Marshall, and J. Zheng, "Effective feature extraction and data reduction in remote sensing using hyperspectral imaging," *IEEE Signal Processing Magazine*, vol. 31, no. 4, pp. 149–154, Jul. 2014.
8. T. Qiao, J. Ren, C. Craigie, J. Zabalza, C. Maltin, and S. Marshall, "Quantitative prediction of beef quality using visible and NIR spectroscopy with large data samples under industry conditions," *Journal of Applied Spectroscopy*, vol. 82, no. 1, pp. 137–144, Jan. 2015.
9. T. Qiao, J. Ren, C. Craigie, J. Zabalza, C. Maltin, and S. Marshall, "Singular spectrum analysis for improving hyperspectral imaging based beef eating quality evaluation," *Computers and Electronics in Agriculture*, vol. 115, pp. 21–25, Jul. 2015.

B.2 Journal Publications under Consideration

1. J. Zabalza, J. Ren, and S. Marshall, "Fast implementation of two-dimensional singular spectrum analysis for effective feature extraction in hyperspectral imaging," *ISPRS Journal of Photogrammetry and Remote Sensing* (2015).
2. J. Zabalza, J. Ren, and S. Marshall, "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imagery," *Neural Computing* (2015).
3. T. Qiao, J. Ren, J. Zabalza, et al, "Curvelet transform and singular spectrum analysis for improving hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing* (2015).
4. T. Qiao, J. Ren, J. Zabalza, et al, "Visible hyperspectral imaging for lamb quality prediction," *Technisches Messen* (2015).
5. J. B. Rafert, J. Zabalza, J. Ren, and S. Marshall, "Singular spectrum analysis: a note on data processing for Fourier transform hyperspectral imagers," (under preparation).

B.3 Conference Publications

1. J. Zabalza, J. Ren, C. Clemente, G. Di Caterina, and J. J. Soraghan, "Embedded SVM on TMS320C6713 for signal prediction in classification and regression applications," in *IEEE European DSP Education and Research Conference (EDERC)*, Sep. 2012, pp. 90–94.
2. J. Zabalza, J. Ren, and S. Marshall, "Singular spectrum analysis for effective noise removal and improved data classification in hyperspectral imaging," in *IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Jun. 2014.
3. J. Zabalza, J. Ren, and S. Marshall, "'On the fly' dimensionality reduction for hyperspectral image acquisition," in *IEEE European Signal Processing Conference (EUSIPCO)*, Sep. 2015.
4. T. Qiao, J. Ren, J. Zabalza, and S. Marshall, "Prediction of lamb eating quality using hyperspectral imaging," in *Optical Characterization of Materials (OCM) Conference*, Mar. 2015, p. 15.
5. T. Qiao, J. Ren, C. Craigie, J. Zabalza, C. Maltin and S. Marshall, "Comparison between near infrared spectroscopy and hyperspectral imaging in predicting beef eating quality," in *Hyperspectral Imaging and Applications Conference*, Oct. 2014.
6. S. Marshall, T. Kelman, T. Qiao, P. Murray, and J. Zabalza, "Hyperspectral imaging for food applications," in *IEEE European Signal Processing Conference (EUSIPCO)*, Sep. 2015.