Spectral-spatial classification techniques for hyperspectral imagery

Author:
Gao, Qishuo

Publication Date:
2019

DOI:
https://doi.org/10.26190/unsworks/3742

Spectral-Spatial Classification Techniques for Hyperspectral Imagery

Qishuo Gao

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Civil and Environmental Engineering

Faculty of Engineering

June 2019

UNSW SYDNEY | Australia's Global University

Thesis/Dissertation Sheet

Surname/Family Name	:	Gao
Given Name/s	:	Qishuo
Abbreviation for degree as give in the University calendar	:	PhD
Faculty	:	Faculty of Engineering
School	:	School of Civil and Environmental Engineering
Thesis Title	:	Spectral-Spatial Classification Techniques for Hyperspectral Imagery

Abstract 350 words maximum: (PLEASE TYPE)

Hyperspectral image (HSI) classification plays an important role in a variety of applications such as land-use classification, mineral identification, climate change detection, and urban planning. Many classifiers have been developed in recent decades; however, the extraction of efficient features is still a challenging issue because of some problems, such as *Hughes phenomenon* and limited number of training samples. This thesis investigates four efficient techniques for HSI classification that take advantage of both spectral and spatial information to overcome the limitations of traditional classifiers.

This study investigates HSI classification from different perspectives. Firstly, a framework that integrates two promising techniques: a joint sparse model and a discontinuity preserving relaxation algorithm, is proposed to perform the classification task. Secondly, this study develops a novel neighbour selection strategy for joint sparse models, and a multi-level joint sparse model is constructed to fully exploit spectral-spatial information for HSI classification based on different parameters used. This method can overcome the oversmoothing effect of the first technique. Thirdly, an extension of the second approach is developed in this study based on a multi-scale conservative smoothing scheme and adaptive sparse representation. This method can automatically overcome the oversmoothing effect as well as exploit the correlations among features extracted from different perspectives. The last approach solves the HSI classification task with multiple feature learning and a convolutional neural network (CNN). This method not only takes advantage of the CNN capability for enhanced feature extraction, but also fully and jointly exploits the spectral and spatial information.

This thesis exploits the spectral-spatial information of HSIs from four different perspectives: integration of different techniques, multi-level-based, multi-scale-based, and multi-number-based. Experimental results demonstrate that exploiting spatial information from multiple perspectives can boost the classification accuracies of single perspective-based methods. This study also suggests that the multiple perspectives-based methods can reduce the negative impacts of limited training samples and *Hughes phenomenon* of conventional classifiers in HSI classification.

FOR OFFICE USE ONLY Date of completion of requirements for Award:

COPYRIGHT STATEMENT

'I hereby grant the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.
I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstract International (this is applicable to doctoral theses only).
I have either used no substantial portions of copyright material in my thesis or I have obtained permission to use copyright material; where permission has not been granted I have applied/will apply for a partial restriction of the digital copy of my thesis or dissertation.'

Signed .....

Date 22/06/2019

AUTHENTICITY STATEMENT

'I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.'

Signed .....

Date 22/06/2019

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed .....

Date .....26/03/2019.....

INCLUSION OF PUBLICATIONS STATEMENT

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in their thesis in lieu of a Chapter if:

- The student contributed greater than 50% of the content in the publication and is the "primary author", ie. the student was responsible primarily for the planning, execution and preparation of the work for publication
- The student has approval to include the publication in their thesis in lieu of a Chapter from their supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis

Please indicate whether this thesis contains published material or not.

☐ *This thesis contains no publications, either published or submitted for publication (if this box is checked, you may delete all the material on page 2)*

☐ *Some of the work described in this thesis has been published and it has been documented in the relevant Chapters with acknowledgement (if this box is checked, you may delete all the material on page 2)*

☒ *This thesis has publications (either published or submitted for publication) incorporated into it in lieu of a chapter and the details are presented below*

CANDIDATE'S DECLARATION

I declare that:

- I have complied with the Thesis Examination Procedure
- where I have used a publication in lieu of a Chapter, the listed publication(s) below meet(s) the requirements to be included in the thesis.

Name	Signature	Date (dd/mm/yy)
Qishuo Gao		26/03/2019

Postgraduate Coordinator's Declaration (to be filled in where publications are used in lieu of Chapters)

I declare that:

- the information below is accurate
- where listed publication(s) have been used in lieu of Chapter(s), their use complies with the Thesis Examination Procedure
- the minimum requirements for the format of the thesis have been met.

PGC's Name	PGC's Signature	Date (dd/mm/yy)
Adrian Russell		27/3/19

For each publication incorporated into the thesis in lieu of a Chapter, provide all of the requested details and signatures required

Details of publication #1: *Full title:* Hyperspectral Image Classification Using Joint Sparse Model and Discontinuity Preserving Relaxation *Authors:* Qishuo Gao, Samsung Lim and Xiuping Jia *Journal or book name:* IEEE Geoscience and Remote Sensing Letters *Volume/page numbers:* Volume: 15, pp.78-82. *Date accepted/ published:* 07 December 2017						
Status	*Published*	✓	*Accepted and In press*		*In progress (submitted)*	
The Candidate's Contribution to the Work The candidate designed the methodology, implemented the experiments and drafted the manuscript.						
Location of the work in the thesis and/or how the work is incorporated in the thesis: Chapter 4						
Primary Supervisor's Declaration I declare that: • the information above is accurate • this has been discussed with the PGC and it is agreed that this publication can be included in this thesis in lieu of a Chapter • All of the co-authors of the publication have reviewed the above information and have agreed to its veracity by signing a 'Co-Author Authorisation' form.						
Supervisor's name *Samsung Lim*	*Supervisor's signatu*				*Date (dd/mm/yy)* 20/03/2019	

Details of publication #2: *Full title:* Improved Joint Sparse Models for Hyperspectral Image Classification Based on a Novel Neighbour Selection Strategy. *Authors:* Qishuo Gao, Samsung Lim and Xiuping Jia *Journal or book name:* Remote Sensing *Volume/page numbers:* Volume: 10, pp:905 *Date accepted/ published:* 8 Jun 2018						
Status	*Published*	✓	*Accepted and In press*		*In progress (submitted)*	
The Candidate's Contribution to the Work The candidate designed the methodology, implemented the experiments and drafted the manuscript.						
Location of the work in the thesis and/or how the work is incorporated in the thesis: Chapter 5						
Primary Supervisor's Declaration I declare that: • the information above is accurate • this has been discussed with the PGC and it is agreed that this publication can be included in this thesis in lieu of a Chapter • All of the co-authors of the publication have reviewed the above information and have agreed to its veracity by signing a 'Co-Author Authorisation' form.						
Supervisor's name *Samsung Lim*	*Supervisor's signature*				*Date (dd/mm/yy)* 20/03/2019	

Details of publication #3:
Full title: Hyperspectral Image Classification Using Convolutional Neural Networks and Multiple Feature Learning.
Authors: Qishuo Gao, Samsung Lim and Xiuping Jia
Journal or book name: Remote Sensing
Volume/page numbers: Volume: 10, pp: 299
Date accepted/ published: 15 Feb 2018

Status	*Published*	✓	*Accepted and In press*		*In progress (submitted)*	

The Candidate's Contribution to the Work
The candidate designed the methodology, implemented the experiments and drafted the manuscript.

Location of the work in the thesis and/or how the work is incorporated in the thesis:
Chapter 6

Primary Supervisor's Declaration
I declare that:
- the information above is accurate
- this has been discussed with the PGC and it is agreed that this publication can be included in this thesis in lieu of a Chapter
- All of the co-authors of the publication have reviewed the above information and have agreed to its veracity by signing a 'Co-Author Authorisation' form.

Supervisor's name	*Supervisor's signature*	*Date (dd/mm/yy)*
Samsung Lim		26/03/2019

Details of publication #4:
Full title: Spectral-Spatial Hyperspectral Image Classification using a Multi-scale Conservative Smoothing Scheme and Adaptive Sparse Representation
Authors: Qishuo Gao, Samsung Lim and Xiuping Jia
Journal or book name: IEEE Transactions on Geoscience and Remote Sensing
Volume/page numbers:
Date accepted/ published:

Status	*Published*		*Accepted and In press*		*In progress (submitted)*	✓

The Candidate's Contribution to the Work
The candidate designed the methodology, implemented the experiments and drafted the manuscript.

Location of the work in the thesis and/or how the work is incorporated in the thesis:
Chapter 7

Primary Supervisor's Declaration
I declare that:
- the information above is accurate
- this has been discussed with the PGC and it is agreed that this publication can be included in this thesis in lieu of a Chapter
- All of the co-authors of the publication have reviewed the above information and have agreed to its veracity by signing a 'Co-Author Authorisation' form.

Supervisor's name	*Supervisor's signature*	*Date (dd/mm/yy)*
Samsung Lim		26/03/2019

Acknowledgements

First of all, I would like to sincerely acknowledge my supervisor, Prof. Samsung Lim, for his advice, guidance, encouragement and constant support throughout my entire Ph.D. study. I am also deeply grateful to my joint-supervisor, Prof. Xiuping Jia, for her continuous support and invaluable comments from time to time. They are of the best teachers I have ever had. This study would not be completed without their support. I also want to thank the anonymous reviewers for their careful reading and helpful comments which significantly helped in improving the journal papers.

I extend my sincere thanks to all the staff members of School of Civil and Environmental Engineering for their support and assistances. I would like to thank my colleagues for making the office life joyous. I thank my friends for accompanying me and helping me get through all the tough time.

I would like to express my gratitude to my parents and my sisters for their boundless love, support and encouragement throughout my whole life. Most importantly, I specially want to express my thanks to my husband, Jianxu, for all his support, understanding, patience and encouragement.

List of Publications

The publications which have been published in journals and presented in the conference proceedings during the period of my PhD study are listed as follows.

Journal Papers:

Gao, Q., Lim, S. and Jia, X., 2019. Spectral-Spatial Hyperspectral Image Classification using a Multi-scale Conservative Smoothing Scheme and Adaptive Sparse Classification. *IEEE Transactions on Geoscience and Remote Sensing* (DOI: 10.1109/TGRS.2019.2915809).

Gao, Q., and Lim, S., 2019. A Probabilistic Fusion of a Support Vector Machine and a Joint Sparsity Model for Hyperspectral Imagery Classification. *GIScience &Remote Sensing* (DOI:10.1080/15481603.2019.1623003).

Gao, Q., Lim, S. and Jia, X., 2018. Hyperspectral Image Classification Using Joint Sparse Model and Discontinuity Preserving Relaxation. *IEEE Geoscience and Remote Sensing Letters*, 15(1), pp.78-82.

Gao, Q., Lim, S. and Jia, X., 2018. Improved Joint Sparse Models for Hyperspectral Image Classification Based on a Novel Neighbour Selection Strategy. *Remote Sensing*, 10(6), pp.905.

Gao, Q., Lim, S. and Jia, X., 2018. Hyperspectral Image Classification Using Convolutional Neural Networks and Multiple Feature Learning. *Remote Sensing*, 10(2), pp.299.

Conference Papers:

Gao, Q. and Lim, S., 2018, July. Hyperspectral Image Classification Based on a Convolutional Neural Network and Discontinuity Preserving Relaxation. In 2018 IEEE International Geoscience and Remote Sensing Symposium (pp. 3591-3594). IEEE.

Gao, Q., Lim, S. and Jia, X., 2017, September. Classification of Hyperspectral Imagery Based on Dictionary Learning and Extended Multi-attribute Profiles. In International Conference on Image and Graphics (pp. 358-369). Springer, Cham.

Abstract

Hyperspectral image (HSI) classification plays an important role in a variety of applications such as land-use classification, mineral identification, climate change detection, and urban planning. Many classifiers have been developed in recent decades; however, the extraction of efficient features is still a challenging issue because of some problems, such as *Hughes phenomenon* and limited training samples. This thesis investigates several effective techniques for HSI classification that take advantage of both spectral and spatial information to overcome the limitations of traditional classifiers.

Firstly, this thesis presents a HSI classification framework that integrates two promising techniques, a joint sparsity model and a discontinuity preserving relaxation algorithm. The joint sparse model is firstly applied to obtain a posteriori probability distribution of pixels and then the discontinuity preserving relaxation method is used to further improve the classification results. The joint sparsity model ensures the classification accuracy in most homogenous areas, while the relaxation method smooths the result without blurring the class boundaries by estimating discontinuities in the original image. Experiments reveal that this integrated approach can take advantage of both methods to improve the classification of hyperspectral data sets.

Secondly, a novel technique based on a multi-level joint sparsity model is constructed to fully exploit spectral-spatial information for HSI classification. An adaptive local neighbour selection strategy is developed, which computes weights based on the spectral distances between pixels and uses the labels of the training data as *a priori* information. Structural similarity between the central pixel and its neighbours can be exploited in a sensible way by considering the different contributions of each spectral band. The selected parameters can be used to generate multiple joint sparse matrices at different levels and the efficient performance of multi-level joint sparse optimization improves the classification results. This study shows that extracting spatial information at multiple levels can produce more useful information for HSI classification.

Thirdly, a new classifier is developed, based on a multi-scale conservative smoothing scheme and adaptive sparse representation, to enable efficient spectral-spatial HSI classification. A multi-scale conservative smoothing algorithm is proposed to reduce noise and extract spatial structure information from coarse to fine levels. Over-smoothing is automatically prevented by imposing a weighting scheme on the neighbouring pixels used for smoothing, where the contributions of dissimilar neighbours are suppressed. Subsequently, an adaptive sparse representation is introduced to integrate the characteristics of different perspectives from the series of enhanced HSIs. From this representation, the sparse coefficients of a given unknown pixel can be obtained and used for classification. Extensive experiments conducted on three well-known data sets demonstrate that the proposed approach can achieve superior performance in terms of classification accuracy.

Fourthly, this study proposes a novel HSI classification framework based on a multiple feature learning convolutional neural network (CNN). We built a novel CNN architecture that uses various features extracted from raw imagery. The network generates the corresponding relevant feature maps, and those maps are fed into a concatenating layer to form a joint feature map. The obtained joint feature map is then fed to subsequent layers to predict the final label for each hyperspectral pixel. Experimental results show that this CNN-based multi-feature learning framework has significantly improved classification accuracy.

The proposed four methods use both spectral and spatial information to improve HSI classification performance. The studies demonstrate that exploiting spatial information from multiple perspectives can boost the classification accuracies of single perspective-based methods. The first approach integrates different methods, the second constructs a multi-level sparsity matrix for the test pixel, the third applies a multi-scale conservative smoothing scheme on the HSIs and the fourth extracts multiple features prior to the classification. Experimental results show that the techniques developed in this thesis can classify HSIs efficiently and effectively, and overcome the limitations of conventional algorithms.

Abbreviations

AA: Average Accuracy

AE: Autoencoder

AF: Attribute Filter

AP: Attribute Profile

AVIRIS: Airborne Visible/Infrared Imaging Spectrometer

BPT: Binary Partition Tree

CASI: Compact Airborne Spectrographic Imager

CNN: Convolutional Neural Network

CR: Continuous Relaxation

CRF: Conditional Random Field

DAG: Directed Acyclic Graphs

DBN: Deep Belief Network

DPR: Discontinuity Preserving Relaxation

EAP: Extended Attribute Profile

EEP: Extended Extinction Profile

EF: Extinction Filters

ELM: Extreme Learning Machine

EM: Expectation Maximization

EMAP: Extended Multi-Attribute Profile

EMEP: Extended Multi- Extinction Profile

EMP: Extended Morphological Profile

EP: Extinction Profile

ESRM: Extended Sparse Representation Model

GRSS: IEEE Geoscience and Remote Sensing Society

HSeg: Hierarchical Segmentation

HSI: Hyperspectral Image

ICA: Independent Component Analysis

JSM: Joint Sparse Model

JSRC: Joint Sparse Representation Classification

LiDAR: Light Detection and Ranging

LP: Low Pass

MAP: Maximum a Posterior Probability

MASR: Multiscale Adaptive Sparse Representation

MFASR: Multiple Feature Learning Using Adaptive Sparse Representation

MF-JSRC: Multiple Feature Learning Using Joint Sparse Representation Classification

MLR: Multinomial Logistic Regression

MM: Mathematical Morphology

MNFL: Multiple Nonlinear Feature Learning

MP: Morphological Profile

MRF: Markov Random Field

NLW-JSRC: Nonlocal Weighted Joint Sparse Representation

NP-hard: Nondeterministic Polynomial-Time Hard

OA: Overall Accuracy

OMP: Orthogonal Matching Pursuit

PC: Principle Components

PR: Probabilistic Relaxation

RADAR: Radio Detection and Ranging

ReLU: Rectified Linear Unit

RF: Random Field

RGF: Rolling Guidance Filter

RNN: Recurrent Neural Network

ROSIS: Reflective Optics Spectrographic Imaging System

SAE: Stacked Autoencoders

SDAE: Stacked Denoise Autoencoder

SE: Structuring Element

SOMP: Simultaneous Orthogonal Matching Pursuit

SR: Sparse Representation

SRC: Sparsity Representation Classification

SVM: Support Vector Machine

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Introduction of Optical Remote Sensing Imaging

Remote sensing refers to the acquisition of information concerning targets on Earth's surface through the use of satellite- or aircraft-based sensors [1]. Broadly, there are two types of remote sensing systems: *active* and *passive*. Active remote sensing systems emit energy to scan objects, measuring the radiation reflected or back-scattered by the target. Examples of active remote sensing applications are Light Detection and Ranging (LiDAR) and Radio Detection and Ranging (RADAR). In passive remote sensing, also called *optical remote sensing*, instruments detect natural energy that is reflected or emitted from the observed scene. Generally, solar radiation is the most common energy reflected by passive instruments (see Fig. 1.1), and optical remote sensors can measure the solar reflectance in a wide optical wavelength ranging from 400 nm to 2500 nm [2]. Objects can be recognized by their spectral reflectance signatures; different materials have different absorption characteristics at different wavelengths. Each wavelength range has its own contribution to a measurement.

Remote sensing systems have been undergoing a technology revolution, and high-resolution sensors have been available since the late 1980s. The *resolution* of imagery provides different potential details, and in the context of high-resolution images, resolution can have different meanings [3]. *Spatial resolution* is the size of the smallest features that can be detected by an imaging system, and is usually measured in terms of the so-called *pixel*. *Spectral resolution* represents the ability of optical sensors to resolve features in specific wavelengths of the electromagnetic spectrum; as spectral resolution becomes finer, the bandwidth becomes narrower. The resolutions of different optical sensors are designed differently based on the types of intended tasks. For example, high spectral resolution remote sensors are needed for generating panchromatic, multispectral, and hyperspectral images (Fig. 1.2).

Panchromatic images are acquired by satellites such as Landsat and SPOT 6/7; they use a single spectral channel and usually have a wavelength range in the visible band. A single intensity value is rendered for each pixel, which can be visualized in a greyscale

image. Commonly, this value records combined information from the red, green, and blue visible bands.

Multispectral images can be acquired by satellites such as Landsat, SPOT, HRV-XS, Wordview-2, and Worldview-3. Multispectral images have more than three spectral bands and contain multiple spectral signatures for targets. Their wavelength ranges can cover both the visible and infrared bands of the electromagnetic spectrum. . Multispectral sensors typically provide images with less than 15 bands.

Hyperspectral imaging systems have been widely applied in the remote sensing community since the technology became available in the late 1980s. These sensors (e.g. Hyperion on the EO1 satellite) are able to capture data from numerous and narrower spectral channels. They can include wavelengths from the visible to the near-infrared bands of the electromagnetic spectrum. Hyperspectral images usually have more than 100 bands, and this rich spectral information presents the possibility of identifying targets and increasing understanding of the Earth's surface. Each pixel in a hyperspectral image can be represented as a discrete spectrum that contains the reflected solar radiance of objects in a given region. The representation of a hyperspectral data set is similar to that of multispectral data, with the spectral response of a pixel being expressed as a function of spectral bands in the spectral space representation (Fig. 1.3) [1]. This thesis focuses on hyperspectral image analysis.

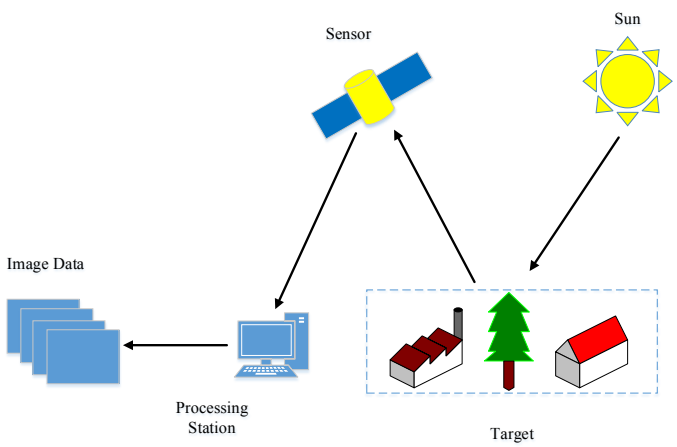


Fig. 1.1. An overview of optical remote sensing system.

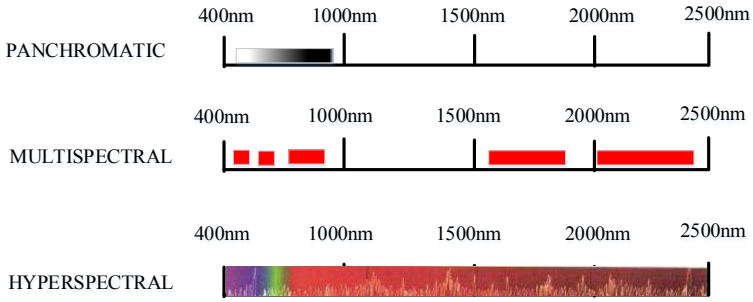


Fig. 1.2. Three types of high spectral resolution remote sensing images.

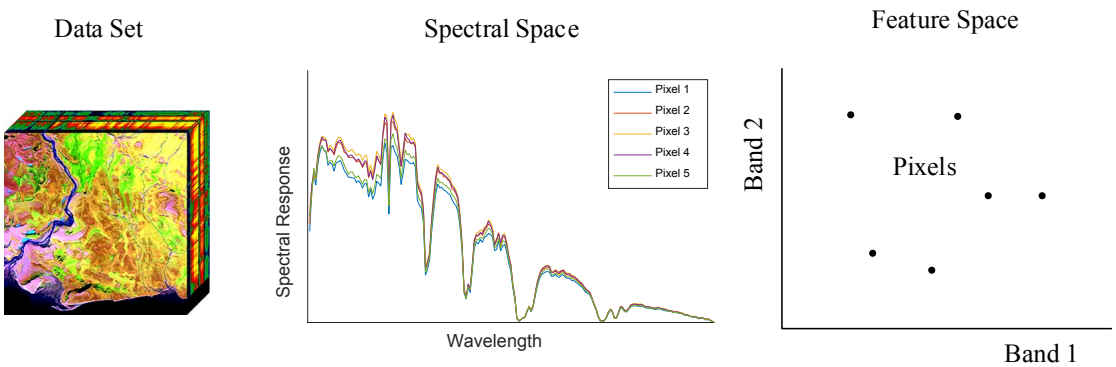


Fig. 1.3. An illustration of hyperspectral data set, and its spectral and feature representations. Illustrations adapted from [4].

1.2 Hyperspectral Image Classification

Advanced hyperspectral imaging systems are able to provide hyperspectral data with high spectral resolution. The abundant and subtle information provided by hyperspectral data makes it possible to distinguish and identify various materials in an image. This capability has led to the wide use of hyperspectral imagery in remote sensing society for applications such as land-cover classification, urban planning, mineral mapping, climate change detection, military surveillance, and species monitoring [5]. A quantitative analysis of hyperspectral images is required to extract the information about the scenes accurately, and the classification of hyperspectral images can be performed on the basis of the quantitative analysis.

The classification of hyperspectral images is essential for analysing and interpreting the contained data. The term *classification* denotes the process that labels each pixel with a

set of classes according to its spectral characteristics. Unsupervised classification groups spectrally similar pixels into a cluster without using prior knowledge, while supervised classification assigns unknown pixels based on the characteristics of *training samples* selected from each class prior to the experiment; supervised classification is vitally important for analysing hyperspectral data [6, 7]. This thesis focuses mainly on the supervised classification of hyperspectral data.

The first attempts at supervised classification of hyperspectral data used techniques designed for panchromatic and multispectral images. However, the results were not satisfactory due to the problems including the high dimensionality and redundancy of data, spatial distortions, and limited numbers of training samples.

1.3 Difficulties in Hyperspectral Image Classification

The special properties of hyperspectral data present many opportunities as well as challenges to develop reliable, high performance classification techniques. This section introduces some of the difficulties in processing hyperspectral images.

1.3.1 Limited Training Samples

An adequate number of training samples is essential for image classification. However, training samples for hyperspectral data are usually collected and labelled manually based on fine spatial resolution satellite images or field measurements, which is extremely expensive and/or time consuming. Moreover, the selection and labelling of training samples becomes more difficult as the study area is more complex and heterogeneous. The presence of mixed pixels in images with coarse spatial resolution also makes classification more challenging.

Furthermore, hyperspectral images feature inherently nonlinear relations between the acquired data and the corresponding materials. These nonlinear relations can be caused by several factors such as interfering scattering from other materials, atmospheric distortions and intraclass variations among similar classes. When nonlinear relations are present, the class patterns learned from limited training samples may not be reliable, and therefore the resultant classification accuracy may be not acceptable.

From the perspective of mathematical statistics and pattern recognition, the combination of limited training samples and a large number of bands impairs the reliability of conventional classification methods, especially when the calculation of a transformation matrix is involved. Sufficient quantity of training samples is essential for machine learning algorithms to learn highly reliable class patterns.

Therefore, limitations in the availability of training samples make the processing of hyperspectral images an extreme challenge.

1.3.2 Hughes Phenomenon

Hughes phenomenon which is also known as the "*curse of dimensionality*", and tends to occur when the number of training samples is too small compared to the dimensionality of data features [8, 9]. The feature space increases very rapidly with the increasing feature dimensionality, and the resultant sparsity is problematic for statistical methods. To achieve high accuracy, the amount of reference data should grow exponentially with the dimensionality. However, when classifying hyperspectral data, the number of available training samples is usually limited while the dimensionality of data features (i.e. spectral bands) is inherently high. Therefore, the performance of conventional classification methods may be limited by the Hughes phenomenon.

1.3.3 Feature Reduction

There are two aspects of feature reduction relevant to hyperspectral image analysis, *feature extraction* and *feature reduction*.

Each pixel of a hyperspectral image can be denoted as a vector, and the length of this vector is the number of spectral bands. Each band is seen as a feature of this pixel vector. As discussed in Section 1.3.2, the use of too many features in combination with a limited number of training samples may decrease classification accuracy. In addition, the narrowness of spectral bands creates redundancy. Therefore, it is important to select only the features that are most useful for separating different classes. This selection is referred to as *feature extraction*, and many approaches [10, 11] have been implemented for and successfully applied to hyperspectral image classification. Feature extraction

can enhance the separability of classes in a lower dimensionality feature space due to the different capabilities in classes.

Feature selection is used to identify an optimal subset of features from the original spectral bands, and is a very important step in hyperspectral image analysis. Feature selection is usually applied as a preprocessing step, and is dependent on the properties of classifiers. In the last decade, a variety of promising methods have been applied to feature selection, most of which can be categorized into three types: filters, wrappers, and embedded approaches [12]. All of these approaches define a criterion by which to evaluate the discriminating power of the selected features. Different subsets may be selected as suitable for different classifiers.

As described above, feature reduction can prevent the Hughes phenomenon and feature selection can be used to enhance class separability. However, both of these processes are time consuming, and class statistics cannot be estimated properly when given limited training samples. In addition, feature selection may not be optimal when biased class statistics render the separability measure unreliable.

1.4 Classification Techniques

Hyperspectral images have many unique properties. The wavelengths covered range from 400 to 2,000 nm and include hundreds of spectral channels at a very narrow spectral resolution (i.e. 10 nm). The extremely rich spectral attributes of hyperspectral data provide the potential for discriminating many classes in detail. Supervised classification aims to assign each pixel to a set of classes based upon patterns defined using prior selected training samples. Key problems involved in supervised classification are processing the huge amount of available hyperspectral data and classifying it with a high degree of accuracy. A considerable array of classification techniques has been developed; in general, they can be grouped into spectral-based or spectral-spatial classification methods.

1.4.1 Spectral-Based Classification

Traditional classifiers typically classify images based on the rich spectral information provided by the numerous bands. These methods only distinguish pixels by their spectral profiles, and thus can be referred to as "*spectral-based*" classifiers. Examples of such classifiers that have been developed for and applied to hyperspectral image classification are: support vector machines [13], maximum likelihood classifiers [14], multinomial logistic regression [15], neural networks [14, 16], and Fisher discrimination classifiers [17]. Spectral-based classifiers are easy to implement and their learning is not complicated. Moreover, their computational complexity is relatively low. Fig. 1.4 shows an example classification of the AVIRIS Indian Pines data set using a support vector machine.

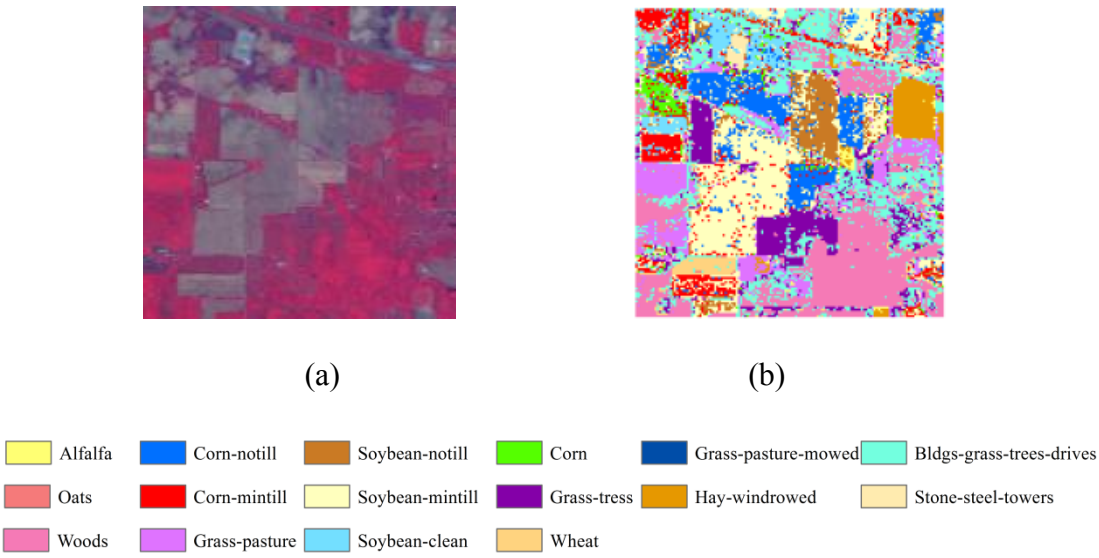


Fig. 1.4. An example of spectral-based classification of AVIRIS Indian Pines data using support vector machine: (a) The false colour image; (b) Classification results. Illustration taken from [18].

Although spectral-based classifiers make full use of the spectral information in hyperspectral data, there are some concerns with their application.

The first concern is the presence of different types of noise and uncertainty during classification. Noise and scattering from other objects is inherent in the acquisition of hyperspectral data, and can result in spectral profiles for objects of the same type being

very different while those of different types may not be distinguishable from each other. Intra-class variability can also lead to distinct differences in the spectral characteristics of similar classes. These uncertainties will result in a low performance classification.

Meanwhile, neighbourhood pixels in hyperspectral images are highly correlated, with the contextual and textural structures being more evident in high-resolution images. Considering the interactions between pixels that are spatial neighbours can reduce uncertainties in labelling, improve the discrimination power of classification methods, and help alleviate the "salt and pepper" appearance of the classification maps.

1.4.2 Spectral-Spatial Classification

To counter the inherent difficulties in spectral-based classification of hyperspectral data and boost classification accuracy, spatial and contextual information can be exploited. Various techniques have been designed to incorporate spatial and contextual information during classification; these approaches are referred as "*spectral-spatial*" classifiers. In this thesis, we categorize these techniques into three groups: fixed-size neighbourhood-based approaches, the adaptive neighbourhood-based approaches, and combinations of different systems [19].

Methodologies using a fixed-size neighbourhood system extract spatial and contextual information within a neighbourhood with given size. In recent years, a number of such techniques have been developed. Markov random fields [20] and conditional random fields [21] calculate the spatial interactions within a local neighbourhood region. Joint sparse models [22] incorporate spatial information using a predefined local region size. The latest 2-dimensional convolutional neural networks [23] use as input subsets of neighbouring pixels with the main pixel of interest located in the centre. Fig. 1.5 illustrates an example of classification of AVIRIS Indian Pines data with a joint sparse model.

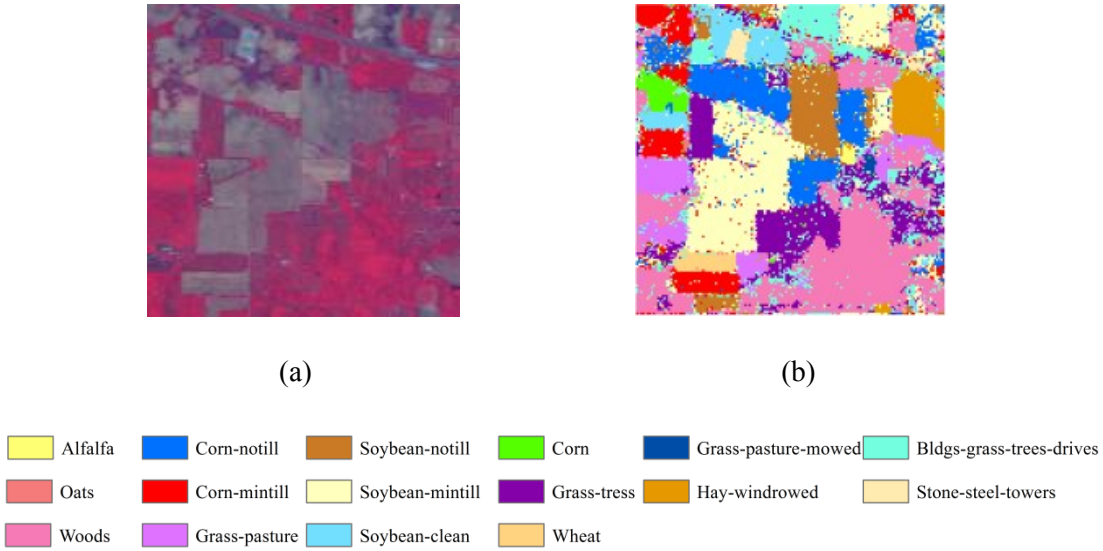


Fig. 1.5. An example of spectral-spatial classification of AVIRIS Indian Pines data using a joint sparse model: (a) The false colour image; (b) Classification results. Illustration taken from [18].

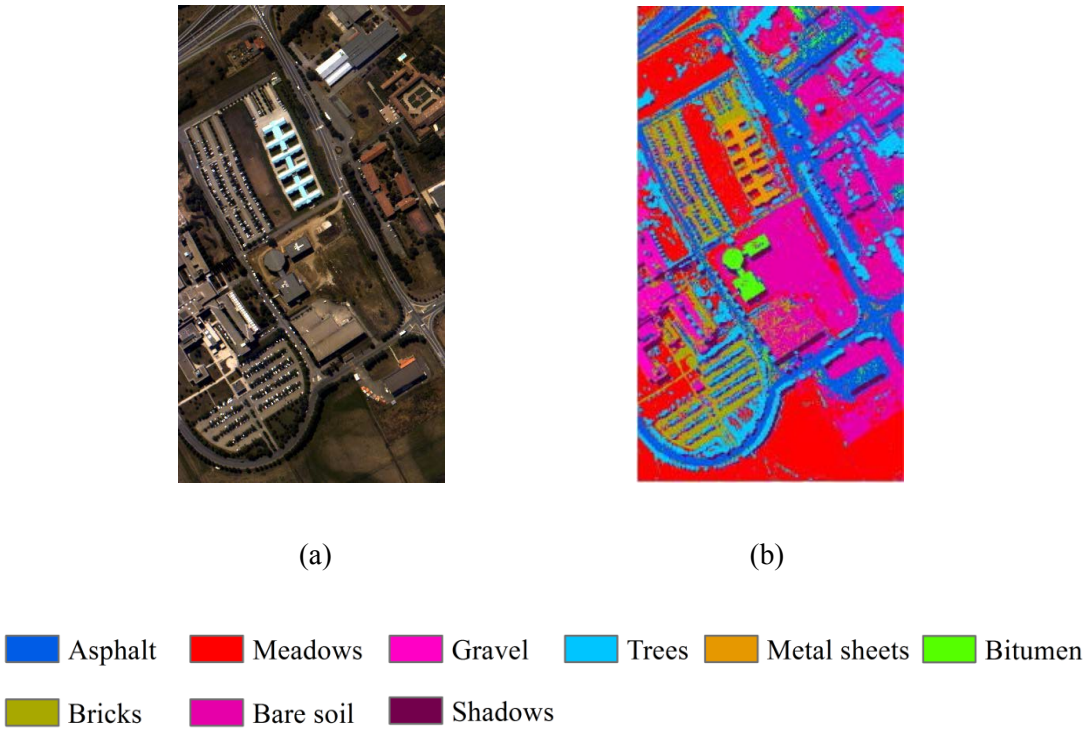


Fig. 1.6. An example of spectral-spatial classification of ROSIS University of Pavia data using a mathematical morphology-based method: (a) The false colour image; (b) Classification results. Illustration taken from [18].

Adaptive neighbourhood approaches extract spatial and contextual information within adaptively changed spatial regions. Mathematical morphology-based methods, such as morphological profiles [24], attribute profiles [25], and extinction profiles [26] extract the structural features of images based on a set of criteria or filters. Segmentation-based methods [27, 28] partition an image into multiple segments, with the pixels in any one segment being assigned to the same label on the assumption that they share similar characteristics. Such segmentation methods usually locate a set of contours (i.e. boundaries) which are used to partition the whole image. Fig. 1.6 illustrates an example of classification of ROSIS University of Pavia data with a mathematical morphology-based method.

There are also some methodological methods which combine multiple systems. For example, morphological features can be extracted and then classified by a joint sparse model or convolutional neural network. Markov random fields can also be used in post-processing to refine segmentation results.

Spatial resolution determines the level of observed spatial detail, and while high levels of detail are desirable, they also bring challenges for spectral-spatial classification. Higher resolution makes the acquisition of training samples much more expensive and time consuming. In images with fine spatial resolution, the shadow problem may compromise classification accuracy, and intra-spectral variation may become more of a challenge [29]. The combination of spectral and spatial information is very valuable for understanding and interpreting ground cover; however, how to make full use of and select suitable classification algorithms for spectral-spatial information given limited training samples remains an active area of research.

1.5 Classification Accuracy Assessment

Selecting the most suitable classification method requires assessments of classification accuracy. In general, for supervised classification, it is assumed that the difference between classification results and the reference data (also known as *groundtruth*) is caused by classification error. In other words, classification error is the discrepancy between groundtruth and the thematic map generated by a classification method.

Currently, the most widely used accuracy assessment for hyperspectral data classification is the confusion matrix [30]. This matrix provides a cross-tabulation of the obtained labels at specific locations against corresponding realities. Many assessment metrics can be derived from a confusion matrix, including overall accuracy (OA), average accuracy (AA), and the kappa coefficient. Overall accuracy is the percentage of pixels correctly classified for the whole data set, which can be interpreted easily. Average accuracy focuses on individual classes, and is computed by averaging the accuracies for each class (i.e. mean class-wise accuracy). Cohen's kappa coefficient has also been used as a standard assessment measure of classification accuracy since 1999 [31]. It measures the agreement between two evaluators who (each) classify items into mutually exclusive categories. Table 1.1 shows a confusion matrix, and Equation (1.1) shows the OA, AA, and kappa coefficient derived from it.

In this thesis, OA, AA, and the kappa coefficient will be used as the metrics for assessing classification accuracy.

Table 1.1 The Confusion Matrix with $q = 4$ Classes.

	A	B	C	D	$\sum$
A	n_{AA}	n_{AB}	n_{AC}	n_{AD}	n_{A+}
B	n_{BA}	n_{BB}	n_{BC}	n_{BD}	n_{B+}
C	n_{CA}	n_{CB}	n_{CC}	n_{CD}	n_{C+}
D	n_{DA}	n_{DB}	n_{DC}	n_{DD}	n_{D+}
$\sum$	n_{+A}	n_{+B}	n_{+C}	n_{+D}	n

$$\text{Overall Accuracy} = \frac{\sum_{k=1}^{q} n_{kk}}{n} \times 100$$

$$\text{Avearage Accuracy} = \frac{\sum_{k=1}^{q} \frac{n_{kk}}{n_{+k}}}{q} \times 100 \qquad (1.1)$$

$$\text{Kappa Coefficient} = \frac{n\sum_{k=1}^{q} n_{kk} - \sum_{k=1}^{q} n_{k+} n_{+k}}{n^2 - \sum_{k=1}^{q} n_{k+} n_{+k}}$$

1.6 Research Objectives

As discussed in Section 1.4.2, recently developed methodological approaches to classifying hyperspectral data tend to incorporate spatial and contextual information in order to improve classification accuracy. However, they still have a few limitations.

Firstly, fixed-size neighbourhood-based techniques, such as joint sparse models and 2-dimensional convolutional neural networks, are sensitive to the selected region scale. If an oversized neighbourhood area is selected for a specific test pixel, the accuracy tends to decrease, while when the selected area is too small, sufficient contextual properties cannot be included. Hence, choosing an optimal region scale is critical, but also difficult. In Bayesian image analysis, Markov random fields and conditional random fields have been widely used as probabilistic graphical models to provide spatial-contextual models for prior distribution. These methods should be in conjunction with suitable soft classification methods that can produce the reliable posterior probabilistic results for each pixel.

Adaptive neighbourhood systems are also faced with some limitations. For example, mathematical morphology-based methods use shallow handcrafted features to characterize spatial and contextual information at the feature extraction stage. This requires the adjustment of a number of threshold values, which is complicated. In segmentation based methods, it is difficult to select the most meaningful objects by which to segment. Both over-segmentation and under-segmentation lead to compromised classification accuracy, and insufficient training samples further limit the performance of this type of object-oriented classification method.

The main objective of this thesis is to develop efficient and feasible methods for fully exploiting spectral and spatial information in hyperspectral data classification, and thereby overcome the limitations of conventional algorithms.

1.7 Thesis Structure

This thesis is organized into nine chapters. A brief overview of the thesis structure as follows:

Chapter 1 presents an introduction of hyperspectral imagery and the difficulties related to hyperspectral image classification. A detailed literature review on the latest advances in hyperspectral image classification is presented in Chapter 2. Chapter 3 briefly introduces data sets used in this thesis. The contributions of this thesis are detailed from Chapter 4 to Chapter 7.

Chapter 4 introduces a framework that integrates a joint sparse model and a discontinuity preserving relaxation algorithm. The joint sparse model is firstly used in a probabilistic sense, to obtain the probability scores of each pixel. The resulting probabilistic distribution map is refined by the discontinuity preserving relaxation scheme. This two-step framework leverages spatial information in both steps, and classification accuracy in most homogenous areas is guaranteed. The method is evaluated on two famous data sets and compared with several well-known classifiers.

In Chapter 5, a novel technique for the hyperspectral image classification based on a multi-level joint sparsity model is constructed to fully exploit spectral-spatial information. An adaptive local neighbour selection strategy is developed that computes weights based on the distances between pixels and uses the labels of training data as *a priori* information. Structural similarity between the central pixel and its neighbours is exploited in a sensible way by considering the different contributions of each spectral band. Multiple joint sparse matrices can be generated on different levels based on the selected parameters, and multi-level joint sparse optimization can be performed efficiently to improve the classification results. The proposed method is compared with some baseline approaches using real hyperspectral data sets.

In Chapter 6, a new classifier based on a multi-scale conservative smoothing scheme and adaptive sparse representation is developed for efficient spectral-spatial HSI classification. A multi-scale conservative smoothing algorithm is proposed to reduce noise and extract spatial structure information across coarse and fine levels. Over-smoothing is automatically prevented through imposing a weighting scheme on the neighbouring pixels used for smoothing, where the contributions from dissimilar neighbours are suppressed. Finally, an adaptive sparse representation is introduced to integrate the characteristics of different perspectives from a series of enhanced

hyperspectral images. From this representation, sparse coefficients can be obtained for a given unknown pixel and used for classification.

Chapter 7 introduces a novel HSI classification framework based on a multiple feature learning convolutional neural network. We built a convolutional neural network with novel architecture that uses various features extracted from raw imagery as its input. From this input, the network generates the corresponding relevant feature maps, which are fed into a concatenating layer to form a joint feature map. The joint map is then fed to subsequent layers in order to predict the final labels for each hyperspectral pixel. Experiments conducted on three well-known data sets show that this framework significantly improves classification accuracy.

Finally, Chapter 8 presents some discussions of the proposed methods. Chapter 9 draws together the conclusions of this thesis and suggests some directions for future research.

Chapter 2 Latest Advances in Hyperspectral Image Classification

As discussed in Chapter 1, there are several types of methodological approaches for extracting spatial information and contextual information from hyperspectral data. As hyperspectral classification techniques have developed rapidly over recent years, this thesis attempts to systematically review the latest advances specifically with respect to spectral-spatial hyperspectral image (HSI) classification. They are reviewed in five branches: mathematical morphological-based approaches, probabilistic graphical methods, segmentation methods, SR models, and deep learning-based techniques.

2.1 Mathematical Morphological-Based Classifiers

Mathematical morphology (MM) was firstly introduced in the report of [28], and can be applied to many image processing problems such as image segmentation and image enhancement. A morphological transformation-based technique, namely *morphological profiles* (MPs), has been extensively used for image analysis since it was firstly introduced in [28]; this method mainly focuses on extracting structural information from images. MPs are constructed by applying a set of opening and closing operations with *structuring elements* (SEs) of increasing sizes to a single-band image/panchromatic image. This method was then generalized as *extended MPs* (EMPs) in [32], and has been successfully used to extract contextual information for hyperspectral data [24, 33] in a multivariate manner. In [34], multiple SEs are used to generate EMPs, and are then integrated with a multiple kernel learning method to present spectral-spatial information from hyperspectral images. EMPs have been used in the classification of hyperspectral data, for which some promising results can be found in [32].

Although MPs and their extensions have achieved some remarkable performance in hyperspectral image classification, they have some limitations. For instance, SE size is fixed, and information can only be extracted for existing objects. Other characteristics of regions should also be considered when analysing hyperspectral data.

Morphological *attribute profiles* (APs) [35] were proposed to overcome the shortcomings of MPs. APs are a generalization of MPs that use a series of *attribute filters* (AFs) for the multilevel extraction of information. AFs are more general than operators by reconstruction due to their capacity to transform images based on attributes other than the shape and size of the SE used in MP construction. APs offered more flexible and informative solutions for image representation, and were subsequently generalized to *extended multi-APs* (EMAP) and *extended APs* (EAPs) [36] by sequencing the APs with different types of attributes.

In [25, 37], independent component analysis (ICA) was first applied to hyperspectral images to generate feature maps, and different attributes were applied to each feature map, producing EMAPs. The authors proposed an automatic procedure for determining the values of EMAP filter parameters, and tuning of these parameters was achieved with a genetic algorithm [38]. Similar work was done in [39], which presented an efficient means of automatically building an EAP from the standard deviation attribute based on class-specific statistics. In [40], EMAPs were based on hyperspectral features derived from both supervised and unsupervised feature reduction techniques, and a random field (RF) and support vector machine (SVM) were used to classify the EMAPS. In [41], EMAPs were integrated with a novel composite kernel that exhibits great flexibility in combining spectral and spatial information for hyperspectral image classification. In [42], random subspace ensemble techniques were applied to EMAPs features to reduce the impact of the curse of dimensionality and improve classification accuracy. In the report of [43], the authors proposed a Bayesian maximum *a posterior* formulation to compute class-specific probability scores based on EMAPs as the prior to a Markov random field. AP-based techniques provide very efficient tools for the presentation of spatial and contextual information in the classification of hyperspectral data, and we refer to [44-46] for a detailed survey about APs and their extensions with applications to hyperspectral image classification.

Very recently, another successful variation of MPs, *extinction profiles* (EPs), were proposed in [47]. In addition, generalizations of EPs, e.g. *extended EPs* (EEPs) and *extended multi-EPs* (EMEPs) [26], have been introduced and utilized for the efficient extraction of hyperspectral features. EMEPs have been integrated with several

classifiers to perform spectral-spatial hyperspectral classification. In [48], EMEPs are firstly extracted and then classified using a random forest. In [49], EPs were extracted from three independent components of a hyperspectral image, three groups of EPs were constructed, and a composite kernel was applied within each EP to explore spatial information. In [50, 51], EPs containing different attributes were derived from both hyperspectral and LiDAR data, and then the features were fused to provide the input for a deep learning-based classifier. In the report of [52], EMEPs were extracted from images associated with the first component extracted through ICA, and then classified by a random forest ensemble-based classifier. All of these EP-based techniques have achieved very competitive results for classifying various real hyperspectral data sets.

2.2 Probabilistic Graphical Models

Probabilistic relaxation is considered an efficient way to incorporate spatial information in image processing. It characterizes neighbourhood information based on the class-wise probabilistic scores of each pixel, obtained using other probabilistic classifiers. The most popular relaxation methodology for considering neighbouring information in the classification stage is the probabilistic graphical model. These models treat each pixel as a graphical node, and use graphical edges to describe the connections between each pair of neighbouring pixels. This topological description allows spatial dependencies within an image to be captured in a probabilistic sense. Each node has a function describing the potential of it belonging to each class, and each edge has a function describing the relationship between the neighbouring pixels is connects.

Various graphical models that rely on neighbouring systems have been applied for remote sensing image classification. The most frequently used systems are the four neighbourhood system and eight neighbourhood system; larger neighbourhood systems bring exponentially increase of computational complexity.

Markov Random Fields (MRFs) are perhaps the most popular and widely used probabilistic graphical models for hyperspectral image classification. MRFs characterize spatial and contextual information for prior distribution in a Bayesian rule. Let $\mathbf{X} = \{\mathbf{x}_i\}$ be the observed data from an image where $\mathbf{x}_i$ is the $i-th$ pixel of the image,

and let $\mathbf{Y} = \{y_i\}$ be the class labels corresponding to the pixels. According to Bayesian theory, the posterior probabilities of each class for a pixel can be expressed as a function of class prior probabilities and class likelihood [53]:

$$P(\mathbf{Y} \mid \mathbf{X}) \propto P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X} \mid \mathbf{Y}) P(\mathbf{Y}). \tag{2.1}$$

Based on the Hammersley-Clifford theorem and given the assumption that the observed data is conditionally independent, the posterior distribution $P(\mathbf{Y} \mid \mathbf{X})$ can be defined as a Gibbs distribution proportional to $\exp[-U(\mathbf{Y} \mid \mathbf{X})]$, where U is an energy function. For image classification, the MRF can be conducted as a functional form of energy [54]:

$$U(\mathbf{Y} \mid \mathbf{X}) = \sum_{i=1}^{n} D_i(\boldsymbol{x}_i, y_i) + \lambda \sum_{i=1}^{n} \sum_{j \in N(i)} V_{ij}(y_i, y_j). \tag{2.2}$$

where $D_i(\boldsymbol{x}_i, y_i)$ is a spectral energy term observed from the data, $V_{ij}(y_i, y_j)$ is a spatial term associated with the spatial relationship between the $i\text{-}th$ and $j\text{-}th$ pixels; $N(i)$ is the neighbourhood system for the $i\text{-}th$ pixel used in the model, for example the four neighbourhood system with $i\text{-}th$ pixel centred; and λ is a parameter for adjusting the relative contributions of the two terms. $D_i(\boldsymbol{x}_i, y_i)$ and $V_{ij}(y_i, y_j)$ are also known as *unary* and *pairwise* potentials, respectively. For hyperspectral image classification, given hyperspectral data $\mathbf{X}$, the minimization of energy $U(\mathbf{Y} \mid \mathbf{X})$ with respect to $\mathbf{Y}$ is equal to the Bayesian function established by Equation (2.1).

In general, unary potentials can be modelled as the negative class-conditional log-likelihood of observations from the original data, and the spatial term determines the form of the MRF model. There are different ways to efficiently conduct the modelling; for example, in [55], a Potts model was used to explore the spatial relationships between neighbouring pixels, and the abundance vectors obtained by an unmixing method were used to model the spectral energy term. In [20], an MRF was implemented in a *maximum a posterior* manner to model the local spatial correlations of neighbouring pixels, and optimization was achieved using a min-cut-based algorithm. In [56], an MRF in a Bayesian framework was applied for the pixel-wise classification of

hyperspectral imagery based on the conditional probabilities derived from a sparse representation model, and this model was optimized using a graph-cut-based α - *expansion* algorithm. An adaptive MRF was proposed to characterize spatial information for the classification of hyperspectral imagery in [57], and a relative homogeneity index for each pixel was adopted to determine the tradeoff between the two contributions in Equation (2.2). In [58], the authors analysed the relationship between a MRF decision rule and a SVM-kernel expansion, and established an integration model for hyperspectral image classification. When using a MRF model, the acquisition of class-conditional log-likelihood can be grouped into two categories: parametric [55, 57, 59-61] and non-parametric [20, 62, 63]. Spatial behaviours can be constructed as favouring edge-preserving, smoothing, isotropic, or anisotropic [19, 61, 62]. Some advanced MRFs have been integrated in multiscale [62], multiresolution [64], segmentation [65, 66], and hierarchical structures [65] for spectral-spatial hyperspectral image classification.

The optimization of MRF models often relies on energy-minimization algorithms; for example, the graph-cut-based [54, 67, 68] and belief propagation-based [69, 70] methods have been very popular in the literature. MRFs model the contextual information of pixels in the labelling stage under the assumption that the observed data is conditionally independent. However, this assumption neglects contextual information in the observed data of a given class. This problem has been addressed through the application of conditional random fields (CRFs) to integrate contextual information for image classification. CRFs directly model posterior probabilities as a Gibbs distribution and avoid explicating the modelling of likelihood.

CRFs have been used as a probabilistic graphical model for the spectral-spatial classification of hyperspectral images in conjunction with many approaches. In [71], multinomial logistic regression (MLR) was used to define pixel-wise (unary) potentials, and employed a CRF to capture the underlying patterns in both labels and observed data. In [72], a multiclass boosted rotation forest method was adopted to provide the posterior probabilities that served as unary potentials for an eight-connected CRF, and an α - *expansion* algorithm was used to solve the optimization problem. In [73], the outputs

of a superpixel-based sparse representation and a patch-based sparse representation were combined to provide the unary potentials for a hierarchical CRF; this method was able to simultaneously consider information from a variety of neighbouring pixels and detect boundary areas. In [74], Gaussian processes were applied to obtain the unary potential of each spectral vector, and used CRFs to incorporate spatial neighbourhood information from the observed data.

MRFs and CRFs are distinct from conventional classifiers in that they model non-identically distributed pixels in a Bayesian framework, which allows the output to exhibit dependency structures. They have been widely used to integrate spatial and contextual information in the classification of hyperspectral images, and in some cases have achieved state-of-the-art performance.

2.3 Segmentation Based Classification

Segmentation is another important method for spectral-spatial classification of hyperspectral data. Segmentation-based methodology partitions an image into several segments, each of which is treated as a superpixel and classified. Some methods carry out segmentation of an image according to criteria such as intensity values or textual properties. After segmentation, classification is performed. Object-based classification approaches are often applied in this context. For each region in a segmentation map, all pixels will be assigned to the most frequent classes based on some pixel-wise classification results. This procedure is known as *plurality voting* or *majority voting*.

A number of different segmentation techniques have been proposed for the classification of hyperspectral imagery. Among the most representative methods are watershed segmentation, expectation maximization (EM)-based segmentation, and hierarchical segmentation (HSeg).

Watershed segmentation is a morphological transformation which partitions an image into several regions in a topographic manner [7]. One single-band image can be divided into several catchment basins according to watershed lines, and each basin is related to one *minimum* in the image. The application of watershed segmentation to hyperspectral

image classification is not straightforward. Usually, the watershed algorithm is applied to a gradient function, and only one band gradient is computed for a multiband image. Then, each watershed pixel is assigned to the region that has the "closest" median [75]. Watershed segmentation was first extended to hyperspectral image classification by Tarabalka et al. in [76], with a pixel-wise SVM used to classify regions and majority voting applied within regions.

EM-based segmentation is a partitional clustering approach that groups pixels into different clusters based on spectral similarities. This method does not consider spatial locations or neighbour relationships. In this method, an initial number of clusters should be defined, which is usually set as the number of classes. Then each pixel is modelled by a Gaussian probability density function [7]:

$$p(\mathbf{x}) = \sum_{c=1}^{C} \omega_c \phi_c(\mathbf{x}; \mathbf{u}_c, \mathbf{\Sigma}_c) \tag{2.3}$$

where C denotes the number of clusters, ω_c represents the weight parameter of the $c\text{-}th$ cluster, and $\phi(\mu, \Sigma)$ is a Gaussian density function with mean μ and covariance matrix Σ. The partitioning of C clusters can be obtained by optimizing this function, and adjacent pixels can be assigned to either the neighbouring regions or the disjoint regions. Partitional clustering was applied for segmentation-based hyperspectral image classification in [77, 78].

The HSeg method considers both spectral similarities and the spatial adjacency of pixels. Its segmentation process combines region growing with unsupervised classification; region growing produces spatially connected regions, while unsupervised classification groups similar regions that are not spatially connected. The main steps of typical HSeg methods can be summarized as [7]: 1) Compute the dissimilarity criterion for all pairs of adjacent regions. This can be done by different methods, such as spectra angel mapper (SAM), which can be applied to calculate the spectral similarity between two vectors; 2) Merge spatially adjacent regions according to the smallest criterion calculated in step 1; 3) Merge spatially nonadjacent regions according to a weight parameter set prior to implementation; and 4) Repeat all steps until convergence is

obtained. However, a particular object in this pipeline can be represented by several regions or can be merged with other objects in one region based on different criteria. Therefore, manual selection of segmentation levels is important for HSeg algorithms. The import of hierarchical segmentation for segmentation-based hyperspectral image classification can be seen in [65, 79-82].

Although the abovementioned segmentation methods have successfully integrated spectral and spatial information for hyperspectral image classification, they remain limited in some respects. Most segmentation methods are dependent on the parameters selected and on the degree of region homogeneity. Automatic segmentation is a topic that has attracted a lot of attention. In [83], an automatic marker-based segmentation method was proposed for the classification of hyperspectral data. *Markers* were defined as the most representative pixels for spatial objects, and chosen from the probabilistic classification results based on some pixel-wise classification results. Then a region growing method was applied to derive a classification map. A constrained marker-controlled segmentation method for classifying hyperspectral imagery was also proposed in [82]. In that approach, markers were automatically selected using probabilistic classification distributions, and then a constrained HSeg method was used to compute the classification results.

Automatic segmentation has also been performed by exploiting region-based image representations. These approaches can provide a hierarchical structure for regions at different scales. One such model is a binary partition tree (BPT), which can be interpreted as a hierarchical partition of an image: the nodes represent image regions while branches denote the relationships among regions. The root node is the entire image, and each of the following levels partitions the image into two progressively smaller non-overlapping regions. A BPT is constructed in a bottom-up manner, and it iteratively groups similar pairs of regions. Once the tree is constructed, image segmentation can be performed through cutting horizontally across the branches.

In [84], a BPT was proposed for defining a hierarchical representation of a given hyperspectral image. At first, the authors constructed a BPT to iteratively clusterobjects of interest as nodes, and used a SVM to classify those nodes. The classification map

was generated by pruning the tree in a bottom-up sequence. In [85], the results of linear spectral unmixing were used to minimize reconstruction error, and then a BPT was implemented for the segmentation. BPT models have been also applied in a multi-object manner in [86, 87] for the segmentation of remote sensing images.

A MRF-based graph cut method can also be used as a segmentation method for classification. In [66, 88], a MLR algorithm was applied for computing the posterior probability distributions of pixels as the MRF prior, and a maximum *a posterior* segmentation was implemented for deriving the classification results.

2.4 Sparsity Representation Classification

Sparse representation (SR) is a promising tool for solving many image processing problems such as denoising [89], fusion [90] and image compression [91]. SR assumes that a natural signal can be linearly expressed using a few coefficients from a so-called dictionary [92]. SR has been extended to the classification of hyperspectral images based on the assumption that, despite their high-dimensional characteristics, the pixels of a given class usually lie in a low-dimensional subspace[93]. This enables a test pixel with an unknown label to be linearly represented by a few elements, and for the label to be determined after the coefficient vectors are recovered from a training dictionary.

2.4.1 Introduction of Sparse Representation Classification

For the sparse representation classification (SRC) model, assume that there are N training pixels belonging to C classes, and $\mathbf{x}$ is a L dimensional pixel. Let $\mathbf{D}$ be the dictionary learnt by training samples, and $\mathbf{x}$ can be linearly represented by the dictionary $\mathbf{D}$:

$$\begin{aligned}\mathbf{x} &= [\mathbf{D}_1, \mathbf{D}_2 \ldots \mathbf{D}_c] \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_c \end{bmatrix} \\ &= \mathbf{Dr}.\end{aligned} \tag{2.4}$$

where $\mathbf{D}_c \in \mathbb{R}^{L\times N_c}$ ($N_1 + ...N_c + ... + N_C = N$) is the sub-dictionary for the *c-th* class and $r_c \in \mathbb{R}^{N_c\times 1}$ is the set of sparse coefficients corresponding to $\mathbf{D}_c$. In an ideal situation, if $\mathbf{x}$ belongs to the *c-th* class, then $r_j = 0$, $\forall j = 1...C, j \neq c$ .The label of $\mathbf{x}$ can be directly determined from the recovered sparse coefficients and reconstruction error. The class label of test sample $\mathbf{x}$ can be obtained according to the minimum residual between the pixel and the reconstruction vector:

$$Class(\mathbf{x}) = \arg\min_{c=1,2...C} \left\| \mathbf{x} - \mathbf{D}_c r_c \right\|_2 . \tag{2.5}$$

2.4.2 Introduction of the Joint Sparse Model

As spatial information is very important for hyperspectral image classification, it is essential to also embed spatial contextual information into the SR model. Chen et al. [22] proposed using a joint sparse model (JSM) to exploit the correlations between neighbouring pixels and a center pixel. Given a patch of $\sqrt{W} \times \sqrt{W}$ pixels where W is a square number, let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 ... \mathbf{x}_W]$ be the joint signal matrix consisting of all the neighbouring pixels in this patch. In other words, the test pixel is located at the centre of the selected region and the remaining pixels in $\mathbf{X}$ are its neighbours. According to the principles in [22], $\mathbf{X}$ can be expressed as:

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_1, \mathbf{x}_2 ... \mathbf{x}_W] = [\mathbf{D}\mathbf{r}^1, \mathbf{D}\mathbf{r}^2 ... \mathbf{D}\mathbf{r}^W] \\ &= \mathbf{D}[\mathbf{r}^1, \mathbf{r}^2 ... \mathbf{r}^W] = \mathbf{DR}. \end{aligned} \tag{2.6}$$

where $\mathbf{R} = [\mathbf{r}^1, \mathbf{r}^2 ... \mathbf{r}^W] \in \mathbb{R}^{N\times W}$ is the sparsity matrix, and the selected atoms in dictionary $\mathbf{D}$ are determined by the nonzero coefficients in $\mathbf{R}$. Therefore, the common sparsity pattern for pixels can be recognized by enforcing the indices of nonzero atoms in the sparsity coefficient matrix. The label of $\mathbf{x}$ can be directly determined from the recovered sparse coefficients and reconstruction error:

$$Class(\mathbf{x}) = \arg\min_{c=1,2...C} \left\| \mathbf{X} - \mathbf{D}_c \mathbf{R}_c \right\|_2 . \tag{2.7}$$

where $\mathbf{R}_c$ represents the reconstruction residual corresponding to the *c-th* class. Compared to a pixel-based SRC, JSM can achieve a better classification result by incorporating the contextual information of neighbouring pixels. However, different areas need different region scales, and even though neighbouring pixels tend to have similar spectral signatures, there are some less correlated pixels may exist in one local patch due to the spectrally heterogeneous features in hyperspectral scenes.

2.4.3 Overview of JSM in Hyperspectral Image Classification

Recently, better performance has been achieved with JSM compared to pixel-wise SR methods [94]. A number of spectral-spatial classifiers based on JSM have been successfully applied for the classification of hyperspectral data. Several methods have been attempted for obtaining the more reliable joint matrix. A k-nearest neighbour selection method was applied prior to the JSM in [95, 96] to determine the importance of each neighbouring pixel in a given neighbourhood, and a Gaussian weighted function was used as the selection criterion. Similar work was done in [97], where reliable neighbours were chosen for the test pixels and a sub-dictionary extracted from the original dictionary based on spectral similarities between the test pixel and dictionary atoms. In [98], a shape-adaptive region was constructed for each test pixel based on the first principal component extracted from a PCA, and a shape adaptive region was used in JSM to obtain the final classification results for each pixel. Chen et al. [99] proposed to project samples into a high-dimensional feature space to improve class-wise separability, and JSM was used to classify features from that new dimensional space.

Very recently, proposals have been put forth to incorporate different types of features into the sparse models. In [100], JSM was used to classify AP features extracted from hyperspectral images. In [101], several features (e.g. spatial features, Gabor textures, local binary patterns, and MPs) were extracted and transformed nonlinearly into a low-dimensional kernel space before being classified by a SRC. In [102], both spectral features and prior extracted spatial features (e.g. shape and texture) were fed into JSM to acquire the respective representation vectors, and then a joint sparsity ℓ_0 norm was applied on the coefficients to impose a common sparsity upon them.

Other classification techniques have also been integrated with sparse models to obtain more accurate classification results. SR was used to produce probabilistic results in [73], and as prior to a CRF for the classification of hyperspectral images. Li et al. [103] proposed a superpixel-based JSM for hyperspectral image classification. In this method, superpixels were obtained based on some shape and structure criteria, and the superpixel containing the test pixel was classified rather than constructing a joint matrix based on a fixed-sized window. Similar work was performed in [104]. In [105], a segmentation method was used to partition an image into several homogenous regions; for each region, all pixels within the region were simultaneously coded in a SRC model to enforce the same sparsity on them. In [106], an extreme learning machine (ELM) was trained and used to determine whether pixels were classified or not based on a criterion. Pixels that were not classified by the ELM were subsequently processed by a SRC using a sub-dictionary extracted by the ELM. In [107], correlation coefficients and a joint sparse model were fused to achieve better performance over the model alone. In this method, correlation coefficients were first calculated between training and test samples, and a JSM was used to compute the residuals. Finally, the correlation coefficients and the residuals were fused to perform the final classification.

Multi-feature learning has attracted a lot of attention in the area of image processing, and it has very recently been extended to hyperspectral image classification. In [93], the authors proposed to extract complementary features in a multiscale fashion, and those multiscale features were classified by a JSM with an adaptive norm applied. This work was also extended to create a multi-feature learning adaptive sparse representation for the classification of hyperspectral data in [98]. In [103], class-level sparsity was exploited for multi-feature fusion prior to sparse learning, and the correlation and discrimination among different classes were considered during the dictionary learning procedure.

All of the abovementioned approaches have achieved state-of-the-art performance; however, how best to extract complementary information for a JSM is still an open question. Given this, we establish three novel frameworks based on JSM for the classification of hyperspectral images. These frameworks are presented in Chapters 4-6.

2.5 Deep Learning-Based Classifiers

Deep learning, which exploits non-linear transformation of data via several layers, has attracted a lot of attentions in the field of machine learning. In the context of feature extraction, deep learning automatically extracts invariant and discriminative features from a hierarchy of hidden layers. Deep learning-based methods have been adapted to hyperspectral image classification and recently shown to outperform many conventional approaches. Commonly, given a proper architecture and sufficient training samples, a deeper network can extract more abstract and robust features than shallow ones.

In the case of hyperspectral image classification, deep learning-based methods—e.g. autoencoders (AEs), stacked autoencoders (SAEs), deep belief networks (DBNs), recurrent neural networks (RNNs), and convolutional neural networks (CNNs)—have been demonstrated to be very efficient in extracting robust and invariant features. The input of a deep learning model can be a single spectrum or a neighbourhood region for the selected pixel. Spatial information can be incorporated by involving the neighbouring pixels of the pixel to be classified. However, a large number of parameters (i.e. weights) needed to be tuned during the training process, which means that models given an insufficient number of training samples may face the "overfitting" problem. In the development of deep learning models for hyperspectral data, several attempts have been made to improve classification performance.

AE has been conventionally applied as an unsupervised pixel-wise approach for extracting features and reducing the dimensionality of images. In the context of hyperspectral image classification, AE and its extension SAE are used in both unsupervised and supervised manners. In [108], a marginal SAE was applied to extract discriminant features from samples in homogenous regions that were partitioned by a segmentation method prior to the training stage. In [109], the authors proposed to use a set of SAEs to extract features from different segments of the spectrum, and then those reduced features were concatenated into a single feature that was used for the final classification. Segmented SAEs help reduce computational complexity without compromising accuracy. In [110], a stacked denoise autoencoder (SDAE) was used to train a robust and discriminative network for the high performance classification of

hyperspectral data. In [111], a single-layer AE and a multi-layer SAE were applied to extract shallow and deep features respectively and then the two kinds of features were combined for classification. In [112], AE was used to extract deep features from hyperspectral data, which were then classified by a SVM. Other work based on AEs and SAEs can be found in [113, 114].

DBNs have also been extended to hyperspectral image classification. For example, a three layer DBN was used in [115] to extract deep features from the original spectrum and their neighbours, and then the features were classified using a logistic regression classifier. Similar work has been done in [116, 117]. To deal with the problem of limited training samples, the authors of [118] proposed using a diversified DBN to regularize the pretraining and fine-tuning procedures; this regularization was achieved by introducing several latent factors obtained through a recursive greedy method. In [119, 120], DBNs have also been applied for the classification of hyperspectral images.

In RNNs, the network is a graph in a temporal sequence. In the context of hyperspectral image classification, RNNs are usually used to characterize sequential information of the spectrum and in a band-to-band fashion. In [121], the authors processed hyperspectral pixels with a RNN in a sequence to capture the sequential properties of the data, and also applied a new activation function to speed the training process.

Notably, the classification accuracy of deeper networks tends to decrease with the limited training samples available for hyperspectral imagery. This problem is more serious for fully connected models such as AEs and DBNs [19]. In CNNs, the number of parameters is reduced by the properties of shared weights and local connections, which makes it feasible to obtain high classification accuracy for hyperspectral data even when limited training samples are available. Given this, we would like to highlight CNNs as a promising and powerful technique for the classification of hyperspectral image; this thesis mainly focuses on CNNs.

2.5.1 Introduction of CNN

A complete CNN contains many layers, including convolutional layers, down-sampling layers, and some activation layers. An end-to-end CNN maps the input pixel vectors to

the labels or at least the respective potentials of the inputs belonging to each of the classes (i.e. probabilistic distribution).

CNNs are widely initialized using batch normalization, which imposes the zero means and unit variance on the inputs. In this thesis, batch normalization is used for all CNNs constructed.

Suppose $\mathbf{x}$ is a vector of pixels of the input image $\mathbf{X}$ for a layer, and an individual neuron performs an operation on $\mathbf{x}$ to produce an output $\mathbf{a}$. The neuron function can be defined as follows:

$$\mathbf{a} = \sigma(f\mathbf{x} + \mathbf{b}). \tag{2.8}$$

where f is a weight filter, $\mathbf{b}$ is a bias, and $\sigma(\bullet)$ is an activation function, usually a nonlinear function. Each neuron is typically associated with a specific spatial location (i, j) and a dimension d. This means that the convolutional block is implemented on all locations throughout the spectral dimensionality.

For each layer, at least one activation function is applied; the most frequently used activation functions are the sigmoid function and the rectified linear unit (ReLU) [122]. ReLU is applied throughout this thesis; it retains positive inputs while returning 0 for negative inputs:

$$\sigma(\mathbf{x}) = \max(0, \mathbf{x}). \tag{2.9}$$

It is common to stack the outputs of the previous layer and feed them to the next layer. A typical CNN has a hierarchical structure with multiple convolutional layers stacked sequentially. In addition to the convolutional layers, some down-sampling/pooling functions are usually applied on the layers to increase the receptive field of the neurons. Typical pooling functions adopted in CNNs are max-pooling and mean-pooling, which returns the maximum values and mean values of the inputs, respectively.

In order to reduce feature dimensionality and prevent overfitting, dropout is sometimes used in CNNs, particularly those with deep architecture. A dropout function randomly

drops some hidden neurons based on a predefined threshold through setting their values to zero. By doing so, the dropped neurons do not contribute to the next layer and are used in the back-propagation optimization.

The main role of CNNs in image classification is to predict the class labels of test pixels by minimizing a loss function $\mathcal{L}$. A commonly used log-loss function is applied throughout this thesis:

$$\mathcal{L}(\mathbf{x}, c) = -\log \mathbf{x}. \tag{2.10}$$

where $\mathbf{x}_c$ denotes the true label values. In this chapter, a softmax function is applied to the top layer to produce the output with a probability distribution i.e. $\mathbf{x}_k = p(k), k = 1,...,C$. Once $\mathcal{L}$ is applied, weights and biases are determined by minimizing loss. Optimization is performed by a gradient descent algorithm.

2.5.2 Overview of CNNs in Hyperspectral Image Classification

CNN inputs can be 1-dimensional, 2-dimensional, and 3-dimensional. CNNs with 1-dimensional inputs directly classify the images in the spectral domain; those with 2-dimensional inputs extract features from neighbouring pixels and use the neighbours of the pixel to be classified as input; and CNNs with 3-dimensional inputs extract complex features from both spectral and spatial domains. CNNs that consider spatial information can achieve better performance in terms of classification accuracy. In [123], hyperspectral image features were extracted by a five-layer CNN, and convolutional kernels were applied in a 1-dimensional manner. In [23], an end-to-end network was proposed for hyperspectral image classification that optimized the parameters of CNN layers to alleviate overfitting. In [124], the authors argued that effective joint exploitation of spectral and spatial information can be realized by a contextual CNN designed to accept a multi-scale input and to use a fully convolutional structure for the classification task. In [125], a 3-dimensional CNN model was proposed to balance the insufficient number of training samples with high data dimensionality, and a ℓ_2 regularization was employed to avoid overfitting. The authors of [126] proposed a multi-feature learning-based CNN to fully leverage spectral-spatial information through

inputting different types of features. In [127], a 3-dimensional CNN was extended to a deformable CNN, and the network allowed the sampling locations to be adaptively changed based on different spatial contexts. In [128], a convolutional subnetwork was used to extract abstract features from the raw data, and a deconvolutional subnetwork to encode them. The final learning was realized by a residual learning method. It should be noted that in this framework, the CNNs were used as an unsupervised learning technique. Some other promising work based on CNNs can be found in [129-132].

CNNs can also be integrated with other techniques to further boost classification accuracy. In [50], a high classification accuracy was obtained by prior extraction of the extinction features and using them as the inputs for a deep CNN. In [133], Gabor filtering was applied to extract spatial information and a CNN was adopted for further processing of the extracted features. In [134], features extracted by a deep CNN were used as the initial dictionary of a SR model. In [135], the superpixel segmentation method was integrated with a CNN, and in [136], a transferring technique was applied prior to a CNN. Finally, active learning was combined with CNNs to avoid overfitting and the curse of dimensionality for high performance hyperspectral data classification in [137].

CNNs have attracted a lot attention of researchers in image processing due to their superior performance over other fully connected networks. As discussed above, CNNs have been used in classification of hyperspectral images and achieved some competitive results. However, the application of CNNs to hyperspectral image classification is still in early stages, and CNNs embrace a wide range of structures. In this thesis, we have developed a simple but effective framework based on a CNN and multiple feature learning to achieve high accuracy of hyperspectral image classification. The details of the proposed approach are explained in Chapter 7.

2.6 Summary

In this chapter, recent advances in five branches of spectral-spatial classification of hyperspectral images have been reviewed: mathematical morphological-based

approaches, probabilistic graphical methods (i.e. MRFs and CRFs), segmentation methods, SR models, and deep learning-based techniques.

Mathematical morphological approaches are used in the feature extraction stage, and classification in these approaches should be fulfilled by a classifier such as MLR, SVM, or Random Filed (RF). APs and EPs can provide discriminative and variant features for classification in an unsupervised manner, but the definition of parameters is of great importance. APs and EPs can provide very accurate classification results when integrated with a competitive classifier, for example a CNN.

MRFs and CRFs have been demonstrated great promise for the characterization of spatial and contextual information from hyperspectral data. Furthermore, they have been proven to be more powerful when integrated with kernel classifiers (e.g. SVMs) and energy minimization approaches. The probabilistic graphical models should be extended to increase their flexibility, and their integration with recent CNNs may allow more flexible and robust characterization of spatial information.

Segmentation methods extract information based on homogeneous regions that are partitioned by some criterion, e.g. shape, size, or texture. The greatest potential of segmentation methods is in integrating them with other classifiers. Since CNNs have been proven to be one of the most powerful classification tools, the integration of CNNs and segmentation methods may be a forthcoming hot topic in hyperspectral image classification.

SR-based methods are an important branch of hyperspectral image classification, and their most important aspect is the incorporation of spatial information. Models based on JSM have achieved competitive results. JSM can be designed as an end-to-end framework that maps the raw spectral vectors to labels via dictionary learning. JSM belongs to the fixed-size neighbourhood system, and its variations focus on extracting representative and robust spectral-spatial features from a fixed-size region. In particular, the extraction of multilevel features in the region is one of the greatest JSM successes to date. The feature extraction prior to sparse learning can help further improve classification accuracy and solve the curse of dimensionality.

Although the advent of deep learning has led to a paradigm shift in image processing, its application to hyperspectral image classification remains in the early stage. In addition, the various forms of deep learning bring challenges in constructing effective deep models when provided with limited training samples. The design of a generative and robust network is an active topic of development for hyperspectral image classification. Notably, deep learning models can be combined with other techniques to obtain better classification results.

By following this review of the latest advances in hyperspectral image classification, the work presented in the remainder of the thesis is guided by the above remarks and considerations.

Chapter 3 Data Sets

In this chapter, we briefly introduce several data sets used to experimentally evaluate methods. In this thesis, four benchmark data sets[1] are used to examine the proposed methods and comparative approaches. Three are well-known and frequently used for evaluating HSI classification performance. The fourth is a recent data set released for the 2013 Data Fusion Contest of the IEEE Geoscience and Remote Sensing Society (GRSS) [138]. These data sets have different characteristics and contexts in terms of spatial and spectral resolution.

The first data set is the well-known airborne visible/infrared imaging spectrometer (AVIRIS) Indian Pines scene. It was collected over northwestern Indiana, United States in Jun 1992. The scene was captured over an agricultural site, includes 145×145 pixels, and its spatial resolution is 20 metres per pixel. A total of 220 spectral channels were included in the original data set; however, 20 of those (104-108, 150-163, and 220) are water absorption bands that are usually removed prior to experiments. The image contains 16 mutually exclusive classes. This data set is frequently used in the hyperspectral analysis community because its many mixed pixels and unbalanced number of samples per class constitute a challenging classification problem. Fig. 3.1 shows a false colour composite of the image and the 16 groundtruth classes of interest.

The second data set was acquired by the reflective optics spectrographic imaging system (ROSIS) instrument over the urban area of the University of Pavia, Pavia, Italy. The flight was operated by the Deutschen Zentrum for Luftund Raumfahrt (DLR, the German Aerospace Agency) in the framework of the Hysens project, which was managed and sponsored by the European Commission. The scene has a high spatial resolution of 1.3 metres per pixel, and it contains 610×340 pixels. With water

[1] We would like to thank Prof. D. Landgrebe from Purdue University for providing the free downloads of the hyperspectral AVIRIS data set, Prof. Paolo Gamba from the Telecommunications and Remote Sensing Laboratory for providing the Pavia University data set, the California Institute of Technology for providing the Salinas data set, the Hyperspectral Image Analysis group and the NSF Funded Center for Airborne Laser Mapping (NCALM) at the University of Houston for providing the Houston data set, and the IEEE GRSS Data Fusion Technical Committee for organizing the 2013 Data Fusion Contest.

absorption bands removed, 103 bands are used in the thesis. The wavelengths range from 0.43 to 0.86 μm. This data set has nine classes including urban, vegetation, and soil features. Fig. 3.2 shows a false colour composite of the image and the nine groundtruth classes of interest.

The third image used in this thesis is the AVIRIS Salinas image, acquired over the Valley of Salinas, southern California, United States. The image is of 512×217 pixels with 224 spectral bands. Twenty water absorption bands (108-112, 154-167, and 224) are removed prior to analysis. Salinas has a 3.7 metres per pixel spatial resolution and 16 mutually exclusive classes. This scene includes vegetables, bare soil, and vineyard fields. Due to the presence of spectral similarity in the available classes, this data set has been widely used as a benchmark for HSI classification. Fig. 3.3 shows a false colour composite of the image and the groundtruth.

The fourth data set, named *grss_dfc_2013* [138], was captured by the Compact Airborne Spectrographic Imager (CASI) over the test site of the University of Houston, Texas, United States in June 2012. The image is of 349×1905 pixels with a spatial resolution of 2.5 metres per pixel. It consists of 144 spectral channels ranging from 0.38 to 1.05 μm. It has 15 classes, and is a mostly urban scene. Fig. 3.4 shows a false colour composite of the image and the groundtruth.

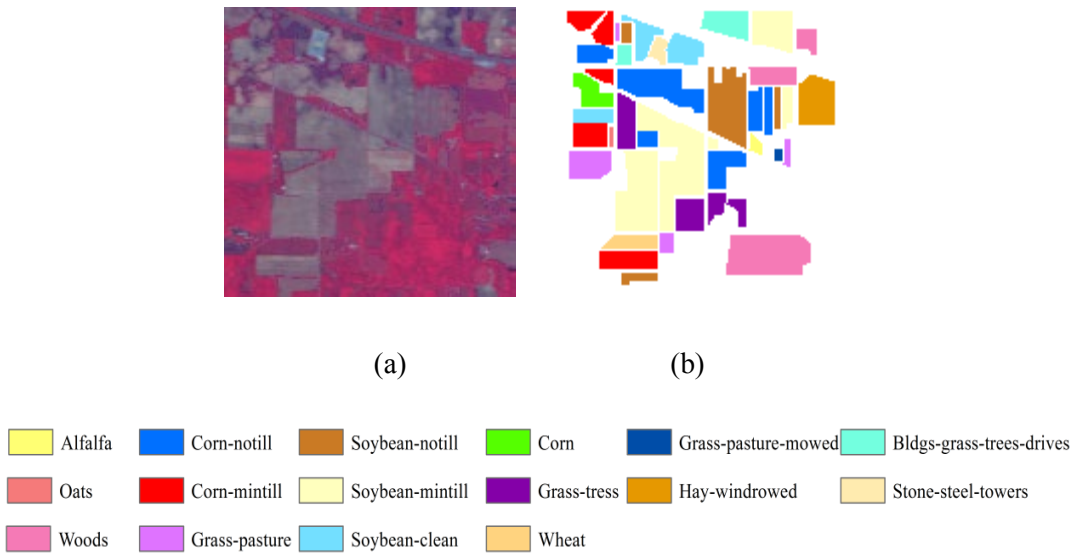
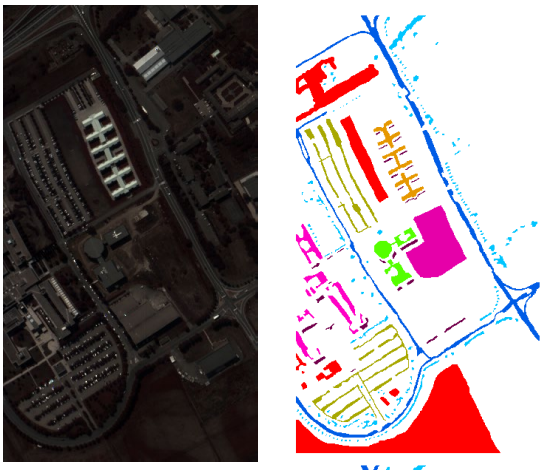


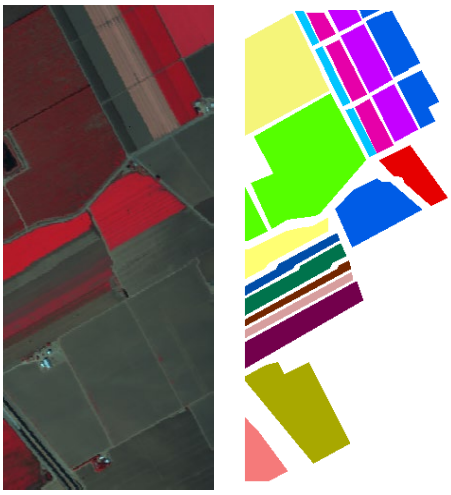
Fig. 3.1. AVIRIS Indian Pines data set: (a) False colour composition. (b) Groundtruth.



(a) (b)

Asphalt Meadows Gravel Trees Metal sheets Bitumen
Bricks Bare soil Shadows

Fig. 3.2. ROSIS University of Pavia data set: (a) False colour composition. (b) Groundtruth.

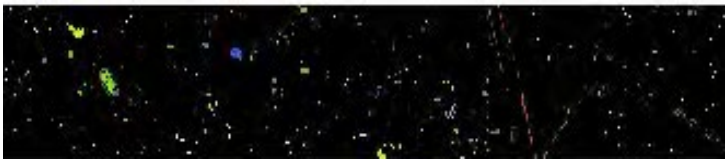


(a) (b)

Weeds_1 Weeds_2 Fallow Fallow plow Fallow smooth Stubble
Celery Grapes Soil Corn Lettuce 4wk Lettuce 5wk
Lettuce 6wk Lettuce 7wk Vinyard untrained Vinyard trellis

Fig. 3.3. AVIRIS Salinas data set: (a) False colour composition. (b) Groundtruth.

(a)



(b)

Healthy Grass Stressed Grass Synthetic Grass Residential Commercial
Parking Lot 1 Parking Lot 2 Tennis Court Running Track Highway
Tree Soil Water Road Railway

Fig. 3.4. CASI Houston University data set: (a) False colour composition. (b) Groundtruth.

Chapter 4 HSI Classification Using JSM and DPR

4.1 Introduction

As discussed in Chapter 2, in the past few years, the concept of signal sparsity has been applied to HSI analysis. The basic assumption is that a linear representation of natural signals in terms of a few atoms can carry the most important information [139]. For HSI, pixels from the same class would be placed in a low-dimensional feature subspace. According to SR, they can be represented by a subset of atoms in a dictionary. Therefore, SR-based methods have been proposed for HSI classification to highlight the differences between pixels from different classes. One of the well-known characteristics of HSI is that the neighbouring pixels tend to have similar contextual properties and are likely to belong to the same class [140]. JSM has also been proposed [22] to exploit the spatial information. JSM assumes that the classification result would be improved by incorporating the neighbouring information of the test pixel. However, this method assigns the equal weight to each neighbouring pixel of the test pixel, which is inappropriate for the heterogeneous areas, especially around class boundaries [93]. In other words, JSM can perform very well for the homogeneous areas but overestimate the contributions of pixels around the class boundaries. Some strategies that use different weights of neighbouring pixels have been presented to resolve this problem [97]; however, it is very difficult to determine the optimal neighbourhood size for a test pixel.

Another category of strategies to use the neighbouring information for HSI is relaxation-based approaches. They use the morphological filters or comprise relaxation approaches to integrate the spatial context of neighbouring pixels. These methods can remove the noise and enhance the quality of classification by correcting both spectral and spatial distortions. They can be used as pre-processing methods prior to classification process to reduce the noisy level of images. In [141], a diffusion algorithm was adopted to reduce the variability of the image, while preserving the boundary of HSI object. Also, the methods can be utilized to detect the spatial properties before a classification. The set of relaxation methods can also be implemented as a post-

processing step after a probabilistic pixel-wise classification of the original HSIs [142]. Continuous relaxation (CR) and probabilistic relaxation (PR) methods are widely used in such a way. Actually, MRF is one kind of PR strategies, and many methods have been proposed based on this [143, 144]. However, in contrast to the improvement in preserving the edges, the classification accuracy tends to decrease in some homogenous areas [145] due to the oversmoothing effect of images.

In [146], a relaxation approach was proposed to accurately preserve the boundaries among different classes, and this method relies on the discontinuity characteristics of the HSI cube. In the report, a MLR based approach was applied to obtain a pixel-wise probability, and the proposed relaxation scheme was applied to learn the final result.

In this chapter, we propose a novel framework, namely, joint sparsity-based discontinuity preserving relaxation (JSDPR) which takes the spectral and spatial information into account in every step of classification by integrating a JSM and discontinuity preserving relaxation (DPR). The discontinuity preserving method incorporates the contextual information into the probability distribution obtained by the JSM to further improve the accuracy. This probability relaxation-based approach consists of two steps: (1) JSM is implemented to obtain a posteriori probability distribution and (2) the DPR is used to compute the final class-wise probability and derive the class labels for test pixels. The main contribution of this method is the integration of the JSM and the DPR where the JSM can ensure the classification accuracy in most homogenous areas and the DPR smooths the result without blurring the boundaries by estimating the discontinuities of the original image, which can further improve the performance. For illustrative purposes, Fig. 4.1 shows the framework of the proposed method.

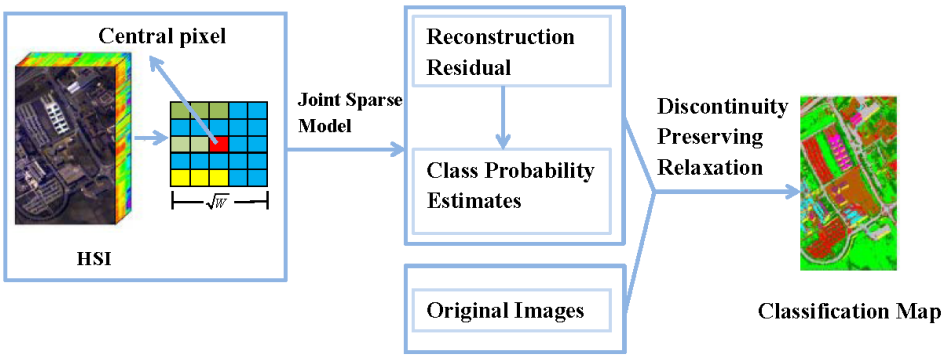


Fig. 4.1. The outline of the proposed method.

Section 4.2 provides a description of the SR and JSM, and also the estimation of the probabilistic distribution of test pixels from JSM model. The principle of DPR is also introduced in Section 4.2. In Section 4.3, the experimental results of the proposed framework on two data sets are delivered. Section 4.4 concludes this chapter. The work of this chapter has been published in IEEE Geoscience and Remote Sensing Letters [147].

4.2 Proposed Framework

4.2.1 Sparsity Representation Classification Model

For the SRC model, assume that there are N training pixels belonging to C classes, and $\mathbf{x}$ is a L dimensional pixel. Let $\mathbf{D}$ be the dictionary learnt by training samples, and $\mathbf{x}$ can be linearly represented by the combination of $\mathbf{D}$:

$$\begin{aligned}\mathbf{x} &= [\mathbf{D}_1, \mathbf{D}_2 ... \mathbf{D}_C] \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_C \end{bmatrix} \\ &= \mathbf{Dr}.\end{aligned} \tag{4.1}$$

where $\mathbf{D}_c \in \mathbb{R}^{L \times N_c}$ ($N_1 + ... N_c + ... + N_C = N$) is the sub-dictionary for the *c-th* class, $r_c \in \mathbb{R}^{N_c \times 1}$ is the sparse coefficients corresponding to $\mathbf{D}_c$. In an ideal situation, if $\mathbf{x}$ belongs to the *c-th* class, then $r_j = 0$, $\forall j = 1...C, j \neq c$. Given the dictionary $\mathbf{D}$, coefficient vectors can be recovered by solving the optimization problem:

$$\begin{aligned} &\hat{\mathbf{r}} = \arg\min_{\mathbf{r}} \|\mathbf{r}\|_0 \\ &\text{subject to} \quad \mathbf{Dr} = \mathbf{x}. \end{aligned} \tag{4.2}$$

Considering empirical error tolerance σ, Equation (4.2) can be relaxed with the following inequality:

$$\begin{aligned} &\hat{\mathbf{r}} = \arg\min_{\mathbf{r}} \|\mathbf{r}\|_0 \\ &\text{subject to} \quad \|\mathbf{Dr} - \mathbf{x}\|_2 \le \sigma. \end{aligned} \tag{4.3}$$

Equation (4.3) can also be replaced by a sparse objective function:

$$\begin{aligned} &\hat{\mathbf{r}} = \arg\min_{\mathbf{r}} \|\mathbf{x} - \mathbf{Dr}\|_2 \\ &\text{subject to} \quad \|\mathbf{r}\|_0 \le P. \end{aligned} \tag{4.4}$$

where P is a predefined sparsity parameter corresponding to the number of zero entries in $\mathbf{r}$. This nondeterministic polynomial-time hard (NP-hard) problem can be optimized by greedy pursuit algorithms. Orthogonal Matching Pursuit (OMP) [148] is a typical algorithm that solves this NP-hard problem, in which the residual is always orthogonal to the span of the already selected atoms, and $\mathbf{r}$ is updated by the residual in each iteration. This problem can also be relaxed to a basis pursuit problem by replacing the ℓ_0 norm with other forms of regularization as follows:

$$\hat{\mathbf{r}} = \arg\min_{\mathbf{r}} \|\mathbf{x} - \mathbf{Dr}\|_2 + \lambda \|\mathbf{r}\|_q . \tag{4.5}$$

where λ is a regularization parameter, and the norm is l_1 and l_2 when $q = 1$ and $q = 2$ respectively. Normally, ℓ_1 norm is more effective in solving the convex optimization problems than ℓ_0 norm is, and ℓ_2 norm can avoid the overfitting issue. The detailed procedure to solve this convex problem can be found in [149].

The label of $\mathbf{x}$ can be directly determined by the recovered sparse coefficients and reconstruction error. Let e represent the residual error between the test sample and the reconstruction term by sparse representation:

$$e_c = \left\| \mathbf{x} - \mathbf{D}_c \hat{\mathbf{r}}_c \right\|_2 \qquad c = 1, 2 \ldots C. \tag{4.6}$$

where $\mathbf{r}_c$ represents the residual computed by dictionary and an optimal sparse coefficient for the *c-th* class. Then the class label of test sample $\mathbf{x}$ can be obtained according to the minimum residual:

$$Class(\mathbf{x}) = \arg\min_{c=1,2\ldots C} e_c. \tag{4.7}$$

4.2.2 Joint Sparse Model

In this section, we will briefly introduce the JSM and the estimation of probabilistic distribution from JSM.

Because neighbouring pixels tend to have similar contextual properties in HSI, a JSM uses the neighbouring information of test pixels and reduce the negative impact of common SR-based classifiers. Let $\mathbf{x}_i \in \mathbb{R}^{L\times 1}$ be a test pixel with L denoting the number of spectral bands. Assume that the test pixel is located at the center of a neighbourhood defined by a window size of $\sqrt{W} \times \sqrt{W}$, where W is a square number. Let $\mathbf{X}_i = [\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_W]$ be the pixels with similar contextual properties within the specified neighbourhood. Then, $\mathbf{x}_i$ can be expressed as:

$$\begin{aligned} \mathbf{X}_i &= [\mathbf{D}\mathbf{r}^1, \mathbf{D}\mathbf{r}^2 \ldots \mathbf{D}\mathbf{r}^W] \\ &= \mathbf{D}[\mathbf{r}^1, \mathbf{r}^2 \ldots \mathbf{r}^W] = \mathbf{D}\mathbf{R}_i. \end{aligned} \tag{4.8}$$

where $\mathbf{D}$ is the dictionary and $\mathbf{R}_i = [\mathbf{r}^1, \mathbf{r}^2 \ldots \mathbf{r}^W] \in \mathbb{R}^{n\times W}$ with n being the number of atoms in the dictionary, represents a set of sparsity coefficient vectors.

Given the dictionary $\mathbf{D}$, $\mathbf{R}_i$ can be optimized by solving the following objective function:

$$\begin{aligned} &\hat{\mathbf{R}}_i = \arg\min \left\| \mathbf{X}_i - \mathbf{D}\mathbf{R}_i \right\|_F, \\ &\text{subject to} \quad \left\| \mathbf{R}_\mathrm{i} \right\|_{row,0} \le K. \end{aligned} \tag{4.9}$$

where $\|\cdot\|_F$ is the Frobenius norm, and $\|\mathbf{R}_i\|_{row,0}$ represents the number of nonzero rows in $\mathbf{R}_i$ with K being the upper bound of the sparsity level. Equation (4.9) is an NP-hard problem. It can be solved by simultaneous greedy algorithms or polynomial time methods. The Simultaneous Orthogonal Matching Pursuit (SOMP) [22] is a generalized OMP algorithm in which the elements in the dictionary are sequentially selected, and the residual is sequentially updated. In this chapter, we use the SOMP to solve the optimization model of Equation (4.9).

If the sparse coefficient matrix $\mathbf{R}_i$ is known, the probability distribution associated with each label can be computed with respect to residuals. We define $\mathbf{E}_i = [e_{i,1},...,e_{i,c},...,e_{i,C}]$ as the residuals corresponding to each class for the test pixel $\mathbf{x}_i$:

$$e_{i,c} = \|\mathbf{x}_i - \mathbf{D}_c\hat{\mathbf{r}}_c\|_2 \qquad c = 1,2...C. \tag{4.10}$$

where $\mathbf{D}_c$ and $\mathbf{r}_c$ are the dictionary and coefficient vector of the *c*-th class, respectively. Considering the residual, the test pixel can be assigned to the class corresponding to the smallest value in the JSM. This is because the class corresponding to the least residual is most likely to be labeled than the others. According to the principle that the probability can be defined as inversely proportional to the residual [150]:

$$p_{i,c} = \frac{1}{e_{i,c}\sigma}. \tag{4.11}$$

where $p_{i,c}$ refers to as a posteriori probability to assign class c for the test pixel $\mathbf{x}_i$, we define $\mathbf{p}_i = [p_{i,1},...,p_{i,c},...,p_{i,C}]^T$ as the probability set for labeling the test pixel $\mathbf{x}_i$. σ is a constant for the normalization of the probability.

4.2.3 Discontinuity Preserving Relaxation

To further improve the classification accuracy, a discontinuity preserving relaxation (DPR) [146] method is used to make the final decision in this chapter. The DPR method aims to find a balance between the adjustment of noisy classification and smoothness of

classification, which is realized by imposing a weight parameter on two different terms that are corresponding to noise and smoothness, respectively.

Let $\mathbf{p} = [\mathbf{p}_1, ..., \mathbf{p}_i, ..., \mathbf{p}_N] \in \mathbb{R}^{C \times N}$ (where N is the number of samples) be the C -dimensional vectors for all the samples, $\boldsymbol{\theta}_i = [\boldsymbol{\theta}_{i,1}, ..., \boldsymbol{\theta}_{i,c}, ..., \boldsymbol{\theta}_{i,C}]^T$ be the final vector of probability computed from the DPR method, and $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_i, ..., \boldsymbol{\theta}_N] \in \mathbb{R}^{C \times N}$ be the probability matrix for all the samples. The DPR method can be realized by solving the following optimization function:

$$\begin{aligned} &\min_{\theta} (1-\lambda) \left\| \boldsymbol{\theta} - \mathbf{p} \right\|^2 + \lambda \sum_i \sum_{j \in S_i} \delta_j \left\| \boldsymbol{\theta}_j - \boldsymbol{\theta}_i \right\|^2, \\ &\text{subject to:} \quad \boldsymbol{\theta}_i \geq 0 \quad 1^T \boldsymbol{\theta}_i = 1. \end{aligned} \tag{4.12}$$

where $0 \leq \lambda \leq 1$ denotes the weight value that controls the different impacts of the two terms in Equation (4.12), and S_i is the neighbourhood of the test pixel $\mathbf{x}_i$. It should be noted that λ measures the misfit and smoothness level of the data. In other words, if λ is large, no discontinuities exist among the chosen pixels. δ_j is the value at location $j \in S_i$ which is obtained by a Sobel filter:

$$\delta = \exp\left(-\sum_{i=1}^{L} sobel(\mathbf{I}^{(i)}) \right). \tag{4.13}$$

where $sobel()$ is the Sobel operator that produces an output of 0 or 1, and $\mathbf{I}$ is the original image cube.

The first term of Equation (4.12) measures the data misfit, and the second term promotes smoothness according to the weight of δ_j , which also means that it specifically models the pixels around the class boundaries. The DPR can be applied to the spatially homogenous areas by exploiting the correlation between the neighbouring pixels. The class boundaries in the original image are firstly detected by the Sobel filter, and then the DPR smooths the homogenous areas without crossing the boundaries so that it can help preserve the discontinuities in the original image. The objective function

of Equation (4.12) is convex, and the projected iterative Gauss-Seidel is applied to solve this problem. After $\boldsymbol{\theta}_i = [\boldsymbol{\theta}_{i,1},...,\boldsymbol{\theta}_{i,c},...,\boldsymbol{\theta}_{i,C}]^T$ is recovered, $\mathbf{x}_i$ can be assigned to the class that has the maximum probability:

$$Class(\mathbf{x}_i) = \arg\max_{c=1,2...C} \boldsymbol{\theta}_{i,c}. \tag{4.14}$$

4.3 Experimental Results and Discussion

The effectiveness of the proposed method (referred to as JSDPR) is verified with Indian Pines and University of Pavia data sets. JSM [22] (based on SOMP), the extended sparse representation model in the study of Li et al. [150] (referred to as ESRM), and a nonlocal weighted joint sparse representation model (NLW-JSRC) [97] are tested for comparison purpose in this section.

In the experiments, the sparsity level for JSM and JSDPR is set as between 5 and 80 empirically, and the window size is chosen from 3×3 to 19×19. The optimal values are chosen in this article. λ is set as 0.85, and S_i is set as a neighbourhood of eight. OA, AA, and kappa coefficient (k) are calculated to validate the quality of the results. Each result in this section is an average performance over 10 rounds of experiments.

4.3.1 AVIRIS Indian Pines Data Set

The class information and classification results are shown in Table 4.1. Classification maps are presented in Fig. 4.2. All the tested methods using the spatial information perform well on the data set. As shown in Table 4.1 that the proposed JSDPR achieves the best result. The result of NLW-JSRC confirms the effectiveness of the strategy that assigns different weights to the neighbouring pixels.

From Fig. 4.2, one can conclude that JSDPR performs better than the other methods in leading to more homogeneous areas in the classification maps. In Fig. 4.2, JSDPR exhibits better performance in the task of preserving class boundaries (e.g., alfalfa, woods, and buildings-grass-trees) than the other algorithms. For classes such as alfalfa, grass/pasture-owed and oats which have small training data sets, the proposed method

(JSDPR) produces 92.60%, 100%, and 100% accuracies, respectively. Especially for grass/pasture-owed and oats classes, the improvements are 23.08% and 10% higher than JSM, respectively. This is remarkable, and the same conclusion can be made after comparing JSDPR with the other methods. On the other hand, it should be noted that ESRM preserves the class boundaries better than NLW-JSRC; however, it misclassifies more labels. Because of the smoothness effect by both JSM and DPR, some areas may be oversmoothed, which can be observed from the classification maps

4.3.2 ROSIS Urban Data Set: University of Pavia

Class information of the University of Pavia image and the quantitative results obtained by various different classifiers are described in Table 4.2. The classification maps are displayed in Fig. 4.3. It can be seen that the proposed JSDPR yields the best accuracy for most classes for the University of Pavia image. From Fig. 4.3, one can conclude that the proposed JSDPR obviously smooths the homogeneous areas and preserves the discontinuities.

Table 4.1. Class Information and Classification Accuracies (%) for the Indian Pines Image.

Class	Class Name	Train	Test	JSM	ESRM	NLW-JSRC	JSDPR
1	Alfalfa	6	40	85.19	**93.48**	88.89	92.60
2	Corn-no till	129	1299	93.65	88.39	90.66	**94.63**
3	Corn-min till	83	747	94.36	91.25	90.53	**99.88**
4	Corn	24	213	94.44	94.35	**96.15**	**96.15**
5	Grass/trees	48	435	93.16	95.44	**96.78**	93.36
6	Grass/pasture	73	657	93.98	97.51	**97.99**	**97.99**
7	Grass/pasture-mowed	5	23	76.92	**100.00**	84.62	**100.00**
8	Hay-windrowed	48	430	99.80	98.34	**100.00**	99.80
9	Oats	4	16	90.00	65.00	90.00	**100.00**
10	Soybeans-no till	97	875	93.60	92.02	95.35	**97.83**
11	Soybeans-min till	196	2259	95.87	88.83	95.71	**97.41**
12	Soybeans-clean till	59	534	91.53	92.94	**96.42**	91.37
13	Wheat	21	184	92.92	99.02	92.45	**99.53**
14	Woods	114	1151	99.61	95.25	99.92	**100.00**
15	Buildings-grass-trees	39	347	94.21	92.76	91.58	**98.95**
16	Stone-steel towers	12	81	80.00	**100.00**	90.53	90.53
	OA			94.98	92.25	95.22	**97.18**
	AA			91.83	92.79	93.60	**96.88**
	k			94.25	91.20	94.55	**96.79**

Table 4.2. Class information and Classification Accuracies (%) for the University of Pavia Image.

Class	Class Name	Train	Test	JSM	ESRM	NLW-JSRC	JSDPR
1	Asphalt	250	6381	86.83	76.61	**91.36**	89.64
2	Meadows	250	18399	96.72	96.63	97.31	**99.02**
3	Gravel	250	1849	97.95	99.33	99.24	**99.81**
4	Trees	250	2814	**93.64**	95.59	93.41	93.60
5	Meta sheets	250	1095	97.40	**99.93**	99.48	**99.93**
6	Bare soil	250	4779	99.86	93.22	99.52	**100.00**
7	Bitumen	250	1080	99.40	**100.00**	97.22	**100.00**
8	Bricks	250	3432	96.47	94.05	96.50	**97.99**
9	Shadows	250	697	81.10	**78.67**	67.90	77.61
	OA			95.14	92.77	95.81	**96.83**
	AA			94.37	92.67	93.55	**95.29**
	k			93.61	90.48	94.48	**95.81**

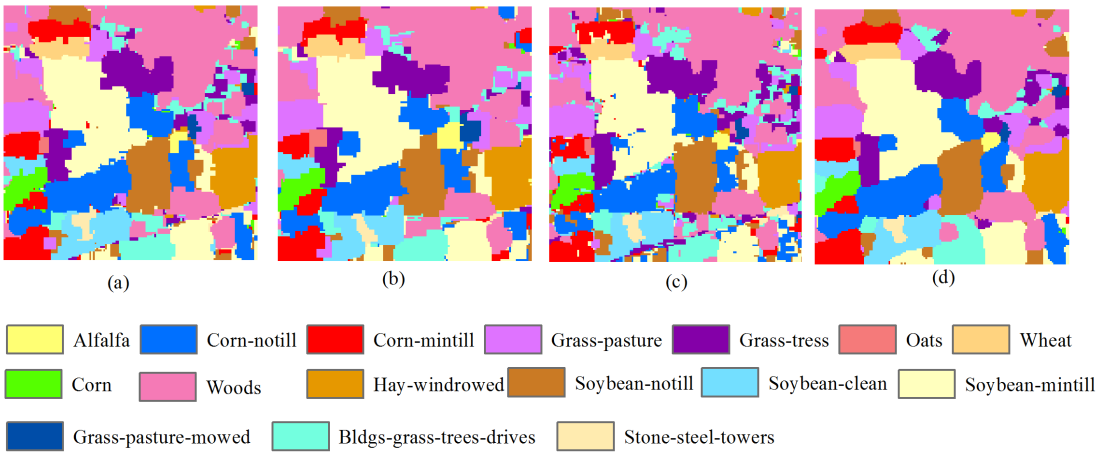


Fig. 4.2. Classification maps of the Indian Pines data set: (a) JSM; (b) ESRM; (c) NLW-JSRC; (d) JSDPR.

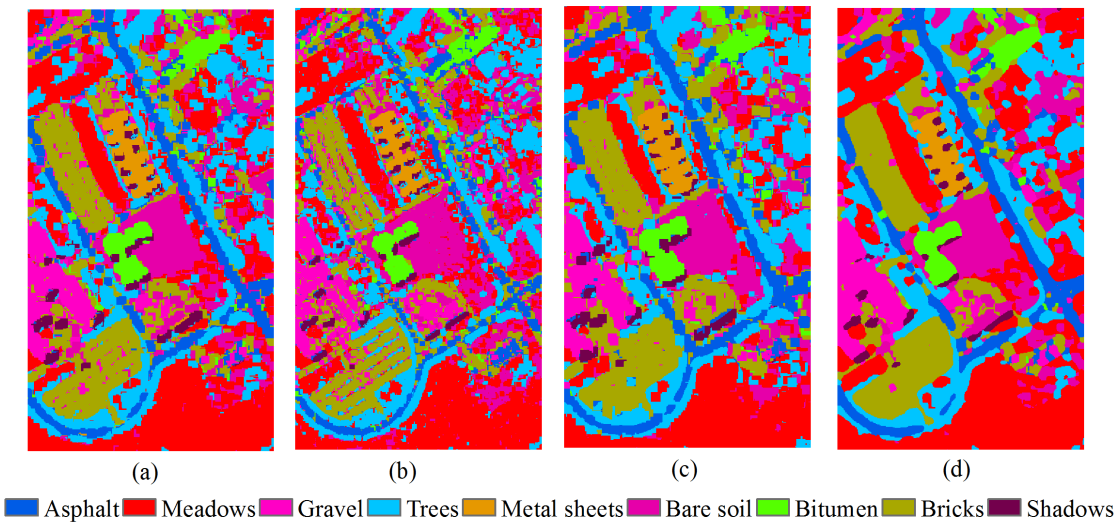


Fig. 4.3. Classification maps of the University of Pavia data set: (a) JSM; (b) ESRM; (c) NLW-JSRC; (d) JSDPR.

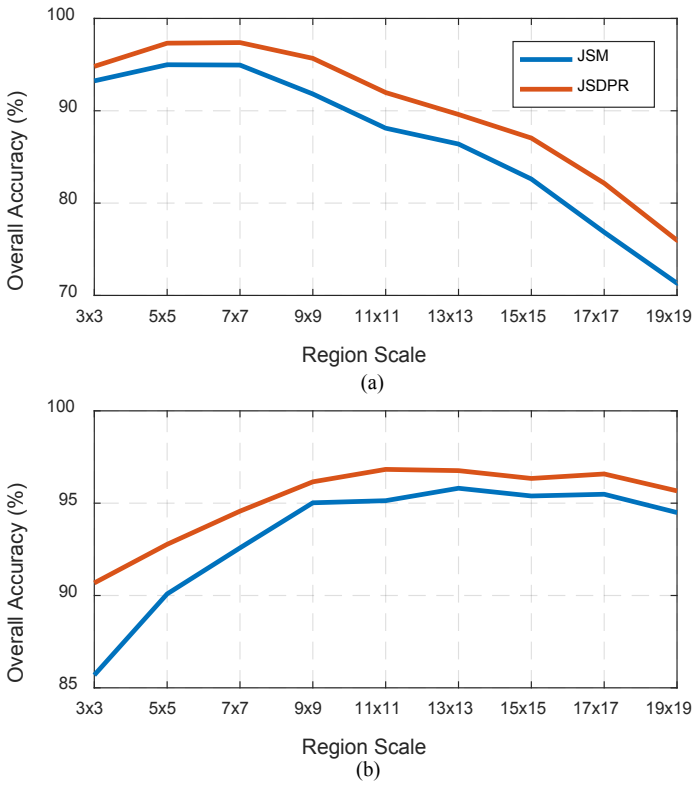


Fig. 4.4. The effect of window sizes on accuracies obtained by JSM and JSDPR for two different data sets: (a) Indian Pines; (b) University of Pavia.

4.3.3 Parameter Analysis

We also demonstrate the effect of window sizes on the accuracies obtained by JSDPR and JSM. In this experiment, the training and test data sets are chosen as the same as the previous experiments and the window size varies from 3×3 to 19×19.

It can be observed from Fig. 4.4 that both JSM and JSDPR produce the best accuracy when the window size is selected as 7×7 for the Indian Pines data set, and 11×11 for the University of Pavia data set, respectively. Because the number of neighbours in the DPR procedure is fixed, the accuracies of JSM and JSDPR show similar trends to each other as the window size increases. As can be seen in Fig. 4.4, JSDPR performs better than JSM in all cases. Because of the different spatial resolution and the number of pixels in the homogeneous area, the optimal window size varies from image to image. Because the Indian Pines image has a low resolution, a smaller window size is appropriate for the JSM process. A larger window size is found optimal for the ROSIS image because of its high spatial resolution. The accuracy decreases if the window size increases further than the optimal size because more uncorrelated pixels can be included in the process.

Finally, Table 4.3 shows the run time averaged over ten repeated experiments of the adopted classifiers and the proposed methods for the classification of the two data sets.

Table 4.3. Run Time (Minutes) of All the Classifiers for the Classification of Two Data Sets.

	JSM	ESRM	NLW-JSRC	JSDPR
Indian Pines	1.5	4.7	2.7	3.5
University of Pavia	10.8	13.6	11.2	11.8

4.4 Summary

In this chapter, we introduced a novel framework based on a JSM and a DPR method. Based on the assumption that the neighbouring pixels tend to have similar contextual properties, the main steps of the proposed framework were developed as follows: (1) distribute posterior probabilities to the test pixels by the JSM that considers the structural similarities between neighbouring pixels and the test pixels, and 2) apply the

DPR method that can help locally smooth the classification maps and preserve the class boundaries. The proposed method is proven to preserve the class boundaries while smoothing the homogenous areas. The experiments indicate that the proposed framework can produce a competitive accuracy when compared with known state-of-the-art classification methods.

The proposed method has some limitations. Firstly, JSM and the adopted relaxation method both rely on the neighbourhood system predefined, which cannot easily capture the distinct characterises and contexts within the window. In addition, the proposed framework may provide an oversmoothed classification map due to the selection of neighbourhood system. This analysis motivates us to investigate new directions to prevent the oversmoothing effect. In Chapter 5 and Chapter 6, new models will be presented to well exploit the spectral-spatial information as well as reduce the oversmoothing effect.

Chapter 5 A Novel Neighbour Selection Strategy for HSI Classification

5.1 Introduction

As discussed in Chapter 2 and Chapter 4, JSM is sensitive to the selected region scale because near-edge areas require a small region scale and homogenous areas need a large region scale. Some experiments have shown that, if an oversized area is selected for a specific test pixel, the accuracy tends to decrease [151]. If the scale is too small, then insufficient contextual properties are included; hence it is difficult to choose an optimal region scale for JSM.

Chapter 4 proposed to integrate JSM with DPR to improve HSI classification performance; however, the homogeneous areas tend to be over-smoothed due to the large neighbourhood selected for JSM and DPR. This chapter is also an improvement of JSM, and aims to exploit sufficient spatial information in a given neighbourhood without causing oversmoothing effect.

For a given specific area, distinct structures and characteristics as well as some irrelevant information will exhibit, however, some pixels with different spectral structures of the test pixel also exist in this region. If a strategy aims to find the most similar pixels to the test pixel and reject the dissimilar neighbouring pixels, information of correlated spatial context should be more representative for classification. Hence, we propose an adaptive neighbour selection strategy which computes the weights based on distances between pixels, with the labels of training data as a priori information. The structural similarity between the central pixel and its neighbours can be exploited in a more sensible way by considering the different contribution of each spectral band. Based on this, a novel joint sparse model-based classification approach, namely ‘adaptive weighted joint sparse model’ (AJSM) is proposed in this chapter. Moreover, we propose a novel classification method namely ‘multi-level joint sparse representation model’ (MLSR), in order to take advantage of the correlations among neighbouring pixels in a region. The procedures of MLSR are summarized as: 1) Local

matrices are obtained by the proposed adaptive neighbour selection strategy. Different thresholds distances can result in different local matrices corresponding to different levels; therefore 2) Different joint sparse representations of the test pixel from different levels can be constructed. Since the pixels with similar distances can be simultaneously sparsely represented by the features in the same subspace, MLSR is designed to learn the dictionary for each joint sparse model separately; 3) A simultaneous orthogonal matching pursuit (SOMP) algorithm is employed to learn the multi-level classification task.

The weight matrix for AJSM and MLSR is constructed by the ratio of the between-class and within-class distances with the consideration of a priori label information. This alleviates the negative impact when we classify the mixed pixels and similar pixels. In addition, the proposed MLSR performs on one region scale with different levels, and the sparse coding procedures at different levels are independent with each other. To sum up the main advantage of the proposed multi-level method, various parameter values can generate multiple sparse models to represent the different inner contextual structures among pixels, thereby improve the HSI classification accuracy.

The remainder of this chapter is organized as follows. Section 5.2 describes the proposed methods in detail for HSI classification. Experimental results on three benchmark data sets are presented in Section 5.3. Finally, a conclusion is provided in Section 5.4. The work of this chapter has been published in Remote Sensing [18].

5.2 Proposed Methods

We introduce an adaptive weight joint sparse model (AJSM) and a multi-level joint sparse representation model (MLSR) for HSI classification in this section. Multiple local signal matrices are constructed using different parameters to realize the similarity learning in MLSR. In fact, AJSM is a simple form of MLSR. The proposed AJSM is expected to improve the classification accuracy in these areas by not taking all the neighbouring pixels to construct the joint sparse matrix. And MLSR improves the classification results by selecting the neighbour pixels from various levels using the proposed adaptive neighbour selection strategy.

To better understand the procedure of the proposed method, a flowchart is shown in Fig. 5.1 where each component of the method is explained in detail in the following sections.

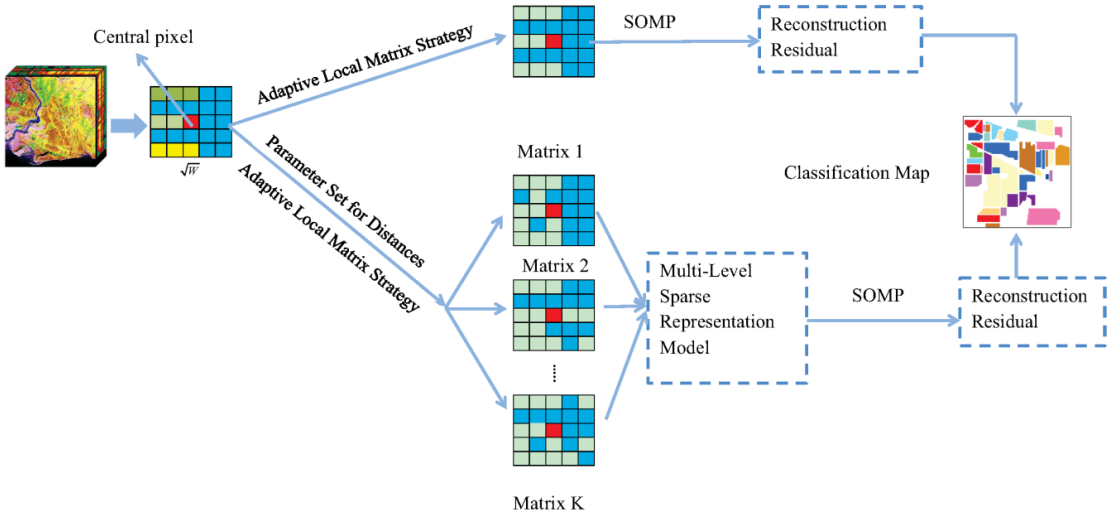


Fig. 5.1. Flowchart of the proposed AJSM and MLSR methods.

5.2.1 Adaptive Local Signal Matrix

In order to select reasonable neighbours to construct the joint matrix, the weighted Euclidean distances between the test pixel and its neighbours are used. We first select a region with a window size $\sqrt{W} \times \sqrt{W}$, which is centered at the test pixel $\mathbf{x}_i$. Different weights are given to each spectral band according to their contribution to the whole spectral characteristics. The weighting strategy is described as follows:

$$
\begin{aligned}
A < \mathbf{x}_i, \mathbf{x}_j > &= \sqrt{\sum_{l=1}^{L} \mathbf{w}_l (\mathbf{x}_{il} - \mathbf{x}_{jl})^2} \\
\mathbf{w}_l &= \frac{\exp(\alpha \mathbf{I}_l)}{\sum_{l=1}^{L} \exp(\alpha \mathbf{I}_l)} \\
\mathbf{I}_l &= \frac{\sum_{i=1} \sum_{c} In(y_i = c)(\overline{\mathbf{x}}_{cl} - \overline{\mathbf{x}}_l)^2}{\sum_{i=1} \sum_{c} In(y_i = c)(\overline{\mathbf{x}}_{il} - \overline{\mathbf{x}}_{cl})^2}.
\end{aligned}
\tag{5.1}
$$

where $A < \mathbf{x}_i, \mathbf{x}_j >$ is the weight distance between pixels $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathbf{w}_l$ is the weight for the *l-th* feature, and $\mathbf{w}_l$ is determined by training samples from different classes. α is a positive parameter that controls the influence of a class-specific distance $\mathbf{I}_l$. If $\alpha = 0$, the distance between two pixels decreases to the equal weight Euclidean distance. If α is large enough, the change will be reflected on $\mathbf{I}$. $In(\cdot)$ denotes an indicator function which takes between-class and within-class distances into account. $\overline{\mathbf{x}}_{cl}$ is the average of the *c-th* class of the *l-th* feature, and $\overline{\mathbf{x}}_l$ represents the average of all training samples of the *l-th* feature; y_i represents the label of pixel $\mathbf{x}_i$.

The pixels with a predefined distance can be selected as similar neighbours according to this method. In other words, this adaptive neighbour selection strategy can identify the samples with similar characteristics to form a group. The superiority of this weight strategy over other weighting schemes is that it considers the spectral similarities at a pixel level, and the discriminative information among different groups can be obtained from training samples.

5.2.2 Adaptive Weight Joint Sparse Model

The goal of Equation (5.1) is to find the optimal samples to reconstruct the central pixel. Once the appropriate weights are assigned to each spectral band, the weight distances between the test pixel and its neighbouring pixels can be evaluated. Based on the top *N*-nearest strategy, *N* nearest neighbouring pixels can be chosen as the adaptive weight joint sparse matrix to relax the joint sparse model as described in Chapter 4. Here we define $\mathbf{S}_N$ as the weight matrix chosen from the original joint sparse matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 ... \mathbf{x}_W]$. In other words, *N* nearest pixels are selected from the *W* pixels based on the previous adaptive weight scheme. The adaptive weight joint sparse model can be expressed as:

$$\begin{aligned} &\widehat{\mathbf{R}} = \arg\min \left\| \mathbf{S}_N - \mathbf{DR} \right\|_F \\ &\text{subject to} \quad \left\| \mathbf{R} \right\|_{row,0} \le P. \end{aligned} \tag{5.2}$$

The label of central pixel can be identified by minimizing the class residual:

$$Class(\mathbf{x}) = \arg\min_{c=1,2...C} \left\| \mathbf{S}_N - \mathbf{D}_c \mathbf{R}_c \right\|_F . \tag{5.3}$$

The procedure of AJSM is summarized below in Algorithm 5.1.

Algorithm 5.1. The implementation of AJSM.

Input: training data sets belonging to the *c-th* class: $\mathbf{X}_c$, region scale: W, top number of nearest neighbours: N, test data sets $\mathbf{X}_T$.

Initialization: initialize dictionary $\mathbf{D}$ with training samples, and normalize the columns of $\mathbf{D}$ to have unit ℓ_2 norm.

1. Compute the $\mathbf{w}_l$ for each spectral band according to Equation (5.1);

2. For each test pixel in $\mathbf{x}_i \in \mathbf{X}_T$:

Construct the weight matrix $\mathbf{S}_\mathrm{N}$ according to Equation (5.1);and normalize the columns of $\mathbf{S}_\mathrm{N}$ to have a unit ℓ_2 norm;

Calculate the sparse coefficient matrix $\mathbf{R}$ and dictionary $\mathbf{D}$ from Equation (5.2) using SOMP;

Determine the class label y_t for each test pixel $\mathbf{x}_i \in \mathbf{X}_T$ by Equation (5.3).

Output: 2- dimensional classification map.

It has been identified that neighbouring pixels consist of different types of materials in the heterogeneous areas in HSI. JSM cannot perform well on such areas due to its definition of neighbouring pixels which tend to have similar labels. The proposed AJSM is expected to improve the classification accuracy in these areas by not taking all the neighbouring pixels to construct the joint sparse matrix.

5.2.3 Multi-level Weighted Joint Sparse Model

The neighbour pixels selected from a fixed scale using single level criteria as seen in JSM and AJSM may not contain the complementary and accurate information, and the neighbouring pixels selected from different levels of criteria can help represent the data

wholly. Herewith we propose a multi-level weighted joint sparse model to fully integrate the neighbour information as well as to avoid the outliers dominating in sparse coding. For a test pixel, its neighbour pixels are selected by the proposed adaptive neighbour selection strategy with different levels of distance thresholds . Then the multiple joint sparse matrices are constructed by the corresponding neighbour pixels with different distance threshold level values. The details of this method are described as follows.

Assume that $\mathbf{S}_{i,k}$ is the *k-th* joint sparse matrix constructed for pixel x_i. Here we define $\mathbf{S}_{i,k} = [\varpi < \mathbf{x}_i, \mathbf{x}_{i,1} > \mathbf{x}_{i,1}, ..., \varpi < \mathbf{x}_i, \mathbf{x}_{i,j} > \mathbf{x}_{i,j}, ..., \varpi < \mathbf{x}_i, \mathbf{x}_{i,W} > \mathbf{x}_{i,W}]$, where $\varpi < \mathbf{x}_i, \mathbf{x}_{i,j} >$ is a function that determines if pixel $\mathbf{x}_{i,j}$ can be preserved to reconstruct $\mathbf{x}_i$, $\mathbf{x}_{i,j}$ is the *j-th* sample in the given region which is restricted by the scale $\sqrt{W} \times \sqrt{W}$. In Equation (5.1), $A < \mathbf{x}_i, \mathbf{x}_{i,j} >$ is a monotonously increasing function of the weighted distances. Although there are many ways to define $\varpi < \mathbf{x}_i, \mathbf{x}_{i,j} >$, we define it as a piecewise constant to simplify the selection of different joint sparse matrices as follows:

$$\varpi < \mathbf{x}_i, \mathbf{x}_{i,j} >= \begin{cases} 1, & A < \mathbf{x}_i, \mathbf{x}_{i,j} > \leq \varepsilon \\ 0, & A < \mathbf{x}_i, \mathbf{x}_{i,j} > > \varepsilon \end{cases}. \tag{5.4}$$

where ε is a threshold controlling the value of the corresponding element in $\mathbf{S}_{i,k}$. According to Equation (5.4), when a pixel in $\mathbf{S}_{i,k}$ has the corresponding weighted distance with the test pixel $\mathbf{x}_i$: $A < \mathbf{x}_i, \mathbf{x}_{i,j} > > \varepsilon$, it will not be selected in the joint sparse model. Otherwise, if $A < \mathbf{x}_i, \mathbf{x}_{i,j} > \leq \varepsilon$, the corresponding term will be selected to reconstruct the test pixel. In other words, $\mathbf{S}_{i,k}$ is constructed by the terms that have the weighted distances less than ε between itself and the test pixel $\mathbf{x}_i$.

By using the proposed scheme, we can generate different patches with various values of ε:

- $\varepsilon = 0$: This is an independent set. In this situation, only the central pixel itself is selected. This means that the joint sparse model becomes a pixel-wise sparse representation model.
- $\varepsilon \geq 1$: Because $A < \mathbf{x}_i, \mathbf{x}_{i,j} > \leq 1$ in this situation, all the neighbours of the test pixel in the given area are selected.
- $0 < \varepsilon < 1$: The sparsity representation of $\mathbf{x}_{i,j}$ is satisfied. A smaller number of pixels are selected for the reconstruction of $\mathbf{x}_i$.

As described above, for each test pixel $\mathbf{x}_i$, when different parameters of $\{\varepsilon_1, ..., \varepsilon_k, ..., \varepsilon_K\}$ are applied, *K* different patches can be generated to represent this pixel with the inner contextual information involved. Our next task is to construct the multi-level joint sparse representation model for the test pixel.

5.2.4 Multi-level Joint Sparse Representation

Herein we extend JSM to a multi-level version for the classification task. After *K* different patches are constructed for each pixel, the patches for the test pixel can be arranged as a feature matrix: $\mathbf{S}_i = [\mathbf{S}_{i,1}, ..., \mathbf{S}_{i,k}, ..., \mathbf{S}_{i,K}]$ ($k = 1, 2, ..., K$), where $\mathbf{S}_{i,k}$ be the *k-th* joint sparsity matrix constructed for the test pixel $\mathbf{x}_i$.

In this chapter, let $\mathbf{D} = \{\mathbf{D}^1, ..., \mathbf{D}^k, ...\mathbf{D}^K\}$ be a set of dictionary which can be learnt from all the training data for *K* patches, and $\mathbf{D}^k$ is the dictionary learnt for the *k-th* level. Each dictionary $\mathbf{D}^k$ is composed of all the sub-dictionaries for each labelled class as $\mathbf{D}^k = [\mathbf{D}_1^k, ..., \mathbf{D}_c^k, ..., \mathbf{D}_C^k]$, where $\mathbf{D}_c^k$ denotes the sub-dictionary of the *c-th* labelled class.

The sparse representation of the test pixel $\mathbf{x}_i$ with its *k-th* patch can be described as:

$$\min_{\mathbf{Q}^k} \left\| \mathbf{S}_{i,k} - \mathbf{D}^k \mathbf{Q}^k \right\|_F . \tag{5.5}$$

where $\mathbf{Q}^k$ is the sparse representation coefficients for the specific patch $\mathbf{S}_{i,k}$. Equation (5.5) expresses how to sparsely represent each of the *K* patches when the sparse coefficient vector is given. Considering all the *K* patches, Equation (5.5) can be rewritten as:

$$\min_{\mathbf{Q}} \sum_{k=1}^{K} \left(\left\| \mathbf{S}_{i,k} - \mathbf{D}^k \mathbf{Q}^k \right\|_F \right). \tag{5.6}$$

where $\mathbf{Q} = [\mathbf{Q}^1, ..., \mathbf{Q}^k, ...\mathbf{Q}^K]$ is composed of K columns of the coefficient vectors. Each column of the matrix is the sparse representation coefficients corresponding to a dictionary over a specific patch.

Since the pixels belonging to the same class should have the dictionary in the same subspace spanned by the training samples, the class-specific level joint representation optimization problem can be written as:

$$\left\langle \hat{\mathbf{D}}, \hat{\mathbf{Q}} \right\rangle = \arg\min_{\mathbf{D},\mathbf{Q}} \sum_{k=1}^{K} \left(\left\| \mathbf{S}_{i,k} - \mathbf{D}^k \mathbf{Q}^k \right\|_F \right). \tag{5.7}$$

This problem can be decomposed into K sub-problems. In this chapter, the SOMP is used to solve the optimization function (5.7) and it can efficiently solve this problem in several iterations. Algorithm 5.2 introduces the implementation of the proposed framework.

After the sparsity coefficients are obtained, for a given test pixel $\mathbf{x}_i$, it would be assigned to the class which gives the smallest reconstruction residual:

$$\begin{aligned} y_i &= \arg\min_{c \in 1,2,...,C} E_c(\mathbf{x}_i). \\ E_c(\mathbf{x}_i) &= \sum_{k=1}^{K} \left\| \mathbf{S}_{i,k} - \mathbf{D}_c^k \mathbf{Q}_c^k \right\|_2^2. \end{aligned} \tag{5.8}$$

where $E_c(\mathbf{x}_i)$ is the reconstruction residual of $\mathbf{x}_i$, $\mathbf{D}_c^k$ is the dictionary for the *c-th* class over the *k-th* patch , and $\mathbf{Q}_c^k$ denotes the sparse coefficient matrix corresponding to $\mathbf{D}_c^k$.

where

Algorithm 5.2. The implementation of the proposed algorithm.

Input: training data sets belonging to the *c-th* class: $\mathbf{X}_c$, region scale: W, number of levels: K, distance threshold controlling parameter: ε, test data sets $\mathbf{X}_T$

Initialization: initialize dictionary $\mathbf{D}_c = \mathbf{X}_c$, and normalize the columns of dictionary to have unit ℓ_2

1. Compute $\mathbf{w}_l (l = 1, 2, ..., L)$ according to Equation (5.1) using the training data sets $\mathbf{X}_c$ and the corresponding labels

2. For each test pixel $\mathbf{x}_i \in \mathbf{X}_T$

   Compute adaptive weight distances between the test pixel and all the pixels in the selected neighbour region to construct $A < \mathbf{x}_i, \mathbf{x}_j >$ based on Equation (5.1)

3. Compute $\mathbf{S}_{i,k}$ based on Equation (5.2).

4. For $k = 1{:}K$

   Compute $\mathbf{Q}^k$ for each level for each class using SOMP.

5. Compute the class label y_i for test pixel based on Equations (5.8).

Output: 2- dimensional classification map.

5.3 Experimental Results and Discussion

To validate the proposed methods, three benchmark data sets are used in the experiments. They are AVIRIS Indian Pines, ROSIS University of Pavia and AVIRIS Salinas data sets.

5.3.1 Experimental Settings

In this chapter, the proposed AJSM and MLSR are compared with several benchmark classifiers: pixel-wise SVM (referred to as SVM), EMP with SVM (referred to as EMP), pixel-wise SRC (referred to as SRC), JSM with a greedy pursuit algorithm [22]. Pixel-wise SVM and pixel-wise SRC classify the images with only spectral information,

while JSM, AJSM and MLSR are sparse representation based classifiers with spatial information utilized.

During the experiments, the range of parameters is empirically determined and the optimal values are determined by cross-validation. The parameters for pixel-wise SVM are set as the default ones in [152] and implemented using the SVM library with Gaussian kernels [153]. Parameters for EMP and pixel-wise SRC are set up by following the instructions in [24] and [22], respectively. The selected regions for JSM, AJSM and MLSR are set as 3×3, 5×5, 7×7, 9×9, 11×11, 13×13 and 15×15, and the best result is described in this chapter. For AJSM, the number of pixels selected in the given region is set as: 7, 20, 40, 50, 50, 50, and 50 for the abovementioned scales, respectively. For the proposed MLSR, the number of threshold parameter ε is set as seven, and threshold values are: $\{0.1,0.2,0.3,0.4,0.5,0.7,1\}$. The predefined sparsity level is set as 3 for each data set.

Quantitative analysis metrics, OA, AA and kappa coefficient (k) are adopted to validate the proposed method. All the experiments in this chapter are repeatedly implemented ten times and the mean accuracy is presented.

5.3.2 Experimental Results

The first experiment was performed on the Indian Pines image. We randomly selected 10% samples from each class as training data and the remaining as a test data set. The optimal parameters in this experiment are set as: α =0.2, $W = 13 \times 13$. The numbers of training and test data for each class are described in Table 5.1. Classification results are listed in Table 5.2, and the classification maps are shown in Fig. 5.2. One can observe that the classification maps obtained by pixel-wise SVM and pixel-wise SRC have a more noisy appearance than other classifiers, which confirms that the contextual information is important for hyperspectral image classification. Considering the spatial information, JSM gives a smoother result; however, it still fails to classify some near-edge areas. EMP, AJSM, and the proposed MLSR deliver better results, and MLSR shows the highest classification accuracy. From Fig. 5.2, one can see that MLSR further provides a smoother classification result and preserves more useful information for HSI.

Table 5.1. Class Information for Indian Pines Data Set

Class	Class Name	Training	Test
1	Alfalfa	5	41
2	Corn-no till	143	1285
3	Corn-min till	83	747
4	Corn	24	213
5	Grass/trees	49	434
6	Grass/pasture	73	657
7	Grass/pasture-mowed	3	25
8	Hay-windrowed	48	430
9	Oats	2	18
10	Soybeans-no till	97	875
11	Soybeans-min till	246	2209
12	Soybeans-clean till	60	533
13	Wheat	21	184
14	Woods	127	1138
15	Buildings-grass-trees	39	347
16	Stone-steel towers	9	84
	Total	1029	9220

Table 5.2. Classification Accuracies (%) for Indian Pines Image.

Class	SVM	EMP	SRC	JSM	AJSM	MLSR
1	42.40	70.49	32.48	74.07	84.09	92.60
2	75.06	91.55	73.31	94.97	92.16	94.63
3	59.91	85.63	58.12	91.82	95.16	99.88
4	50.98	79.49	47.53	87.15	93.67	96.15
5	86.97	95.83	82.04	96.63	96.67	93.36
6	93.84	98.19	89.33	98.88	98.63	97.99
7	89.66	96.30	39.68	94.15	99.98	100.00
8	99.57	100.00	93.22	99.79	99.16	99.80
9	66.67	92.86	32.73	78.57	50.00	82.00
10	62.49	85.96	60.96	91.16	93.30	97.83
11	83.31	94.39	82.61	95.61	92.71	97.41
12	72.17	88.96	70.00	92.05	95.43	91.37
13	90.04	98.07	78.74	99.50	96.10	99.53
14	96.93	98.77	94.83	99.03	97.78	100.00
15	52.82	83.57	49.90	89.88	95.84	98.95
16	82.61	90.82	60.40	94.57	98.91	90.53
OA	75.41	90.77	65.82	92.52	94.74	**97.08**
AA	75.34	90.68	65.37	92.73	92.48	**95.75**
k	73.71	91.20	69.90	94.25	94.02	**96.79**

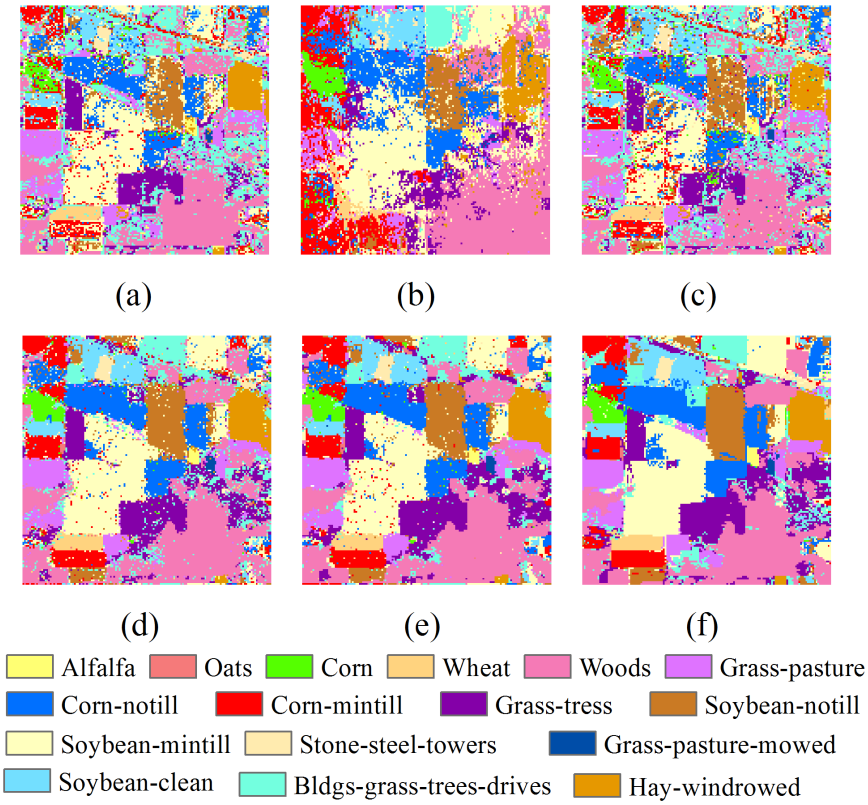


Fig. 5.2. Classification maps of Indian Pines: (a) SVM; (b) EMP; (c) SRC; (d) JSM; (e) AJSM; (f) MLSR.

The proposed AJSM improves the classification capability of JSM by exploring the different contributions of the neighbouring pixels in the selected region. This confirms the effectiveness of the adaptive weight matrix scheme. However, one can see that AJSM produces a relatively lower accuracy for Oats, which has limited training samples. The improvement of MLSR-based classification of Alfalfa and Oats which have been considered as small classes indicates that the proposed method can perform well on classes with less training samples. In addition, the adaptive local matrix imposes the local constraint on the sparsity which would improve the performance. As can be observed from the classification maps, our proposed method has a better capability to identify the near-edge areas and it benefits from the selection of most similar pixels to reconstruct the test pixel. The accuracies for MLSR are very high, which indicates that JSM can be significantly improved by multiple feature extraction approaches.

The second experiment is conducted on the Pavia University image, and Table 5.3 shows the class information. We randomly selected 250 samples as the training data, and the rest as test data. The optimal parameters in this experiment are set as: α =0.2, $W = 15 \times 15$. Classification results and maps are illustrated in Table 5.4 and Fig. 5.3, respectively. It is obvious that the multi-level information can indeed improve the results of classification of the Pavia University image compared to other SRC based methods and the popular SVMs. The improvement of MLSR compared to JSM suggests that the local adaptive matrix can preserve the most useful information and reduce the redundant information. The result is consistent with the previous experiment on the Indian Pines image where the edge pixels are predicted more precisely.

Table 5.3. Class Information for University of Pavia Image

Class No.	Class Name	Training	Test
1	Asphalt	250	6381
2	Meadows	250	18399
3	Gravel	250	1849
4	Trees	250	2814
5	Meta sheets	250	1095
6	Bare soil	250	4779
7	Bitumen	250	1080
8	Bricks	250	3432
9	Shadows	250	697
Total		2250	40526

Table 5.4. Classification Accuracies (%) for University of Pavia Image.

Class	SVM	EMP	SRC	JSM	AJSM	MLSR
1	77.42	84.38	73.46	86.84	99.41	96.32
2	96.34	97.81	95.35	98.07	92.59	99.38
3	84.03	91.44	79.53	91.47	87.09	99.90
4	72.41	81.58	68.07	83.23	98.50	97.68
5	99.92	99.93	99.92	99.11	99.11	100.00
6	82.67	88.81	78.52	90.19	90.16	100.00
7	95.07	96.65	93.64	97.27	95.19	99.92
8	92.39	95.97	89.74	96.46	84.87	99.65
9	99.89	99.54	97.65	86.54	100.00	95.04
OA	88.90	92.90	86.21	92.13	93.30	**98.85**
AA	88.92	92.95	86.47	93.73	94.10	**98.65**
k	84.38	90.22	80.76	91.51	91.21	**98.47**

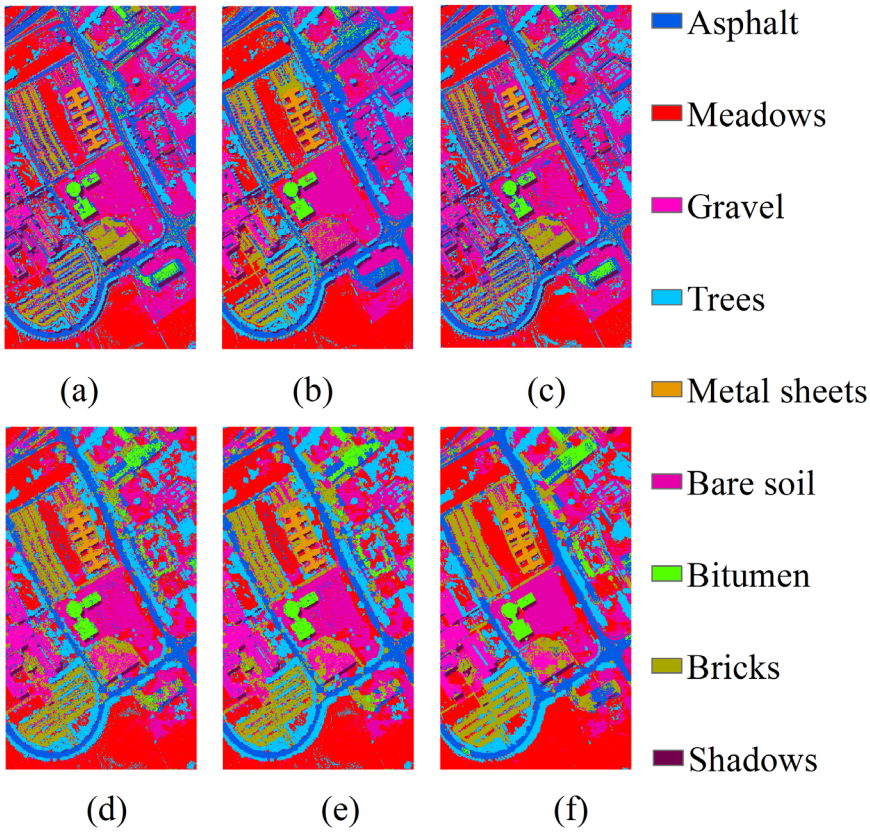


Fig. 5.3. Classification maps of University of Pavia: (a) SVM; (b) EMP; (c) SRC; (d) JSM; (e) AJSM; (f) MLSR.

The third experiment is conducted on the Salinas imagery. For each class, 1.5% samples are selected as the training data, and remaining as the test data set. The optimal parameters in this experiment are set as: α =0.2, $W = 15 \times 15$. The class information and classification results are given in Tables 5.5 and 5.6, respectively. The results are also visualized in classification maps as shown in Fig. 5.4. One can observe that the proposed MLSR yields the best accuracy for most of the classes, especially for Classes 13 and 14. Furthermore, the proposed MLSR identified the edge areas best.

Table 5.5. Class Information for Salinas Image

Class No.	Class Name	Training	Test
1	Weeds_1	30	1979
2	Weeds_2	56	3670
3	Fallow	30	1946
4	Fallow plow	21	1373
5	Fallow smooth	40	2638
6	Stubble	60	3899
7	Celery	54	3525
8	Grapes	169	11102
9	Soil	93	6110
10	Corn	49	3229
11	Lettuce 4 week	16	1052
12	Lettuce 5 week	29	1898
13	Lettuce 6 week	14	902
14	Lettuce 7 week	16	1054
15	Vineyard untrained	110	7158
16	Vineyard trellis	27	1780
	Total	814	53315

Table 5. 6. Classification Accuracies (%) for Salinas Image.

Class	SVM	EMP	SRC	JSM	AJSM	MLSR
1	98.62	99.50	98.67	99.31	99.75	97.83
2	99.65	99.76	99.65	99.70	99.58	97.97
3	95.44	97.95	95.63	97.02	98.73	100.00
4	97.25	98.43	97.39	97.94	98.92	98.85
5	97.73	98.30	97.76	98.18	98.62	99.51
6	100.00	99.90	100.00	99.92	99.77	99.92
7	98.11	99.42	98.16	99.11	99.86	99.72
8	78.87	90.74	79.91	86.79	80.18	89.77
9	99.56	99.77	99.61	99.76	98.90	99.48
10	90.60	96.03	91.13	94.12	93.72	98.96
11	89.01	95.75	89.79	93.07	99.44	100.00
12	96.05	98.45	96.40	97.55	99.07	100.00
13	94.77	96.67	94.87	95.62	97.60	99.13
14	87.11	94.25	87.56	91.44	96.17	98.13
15	59.09	79.50	60.62	71.87	82.44	97.41
16	98.28	99.17	98.39	99.16	98.78	98.56
OA	87.64	94.31	88.20	91.98	92.58	**97.25**
AA	92.51	96.47	92.85	95.04	96.33	**98.77**
k	86.29	93.68	86.91	91.09	92.00	**96.94**

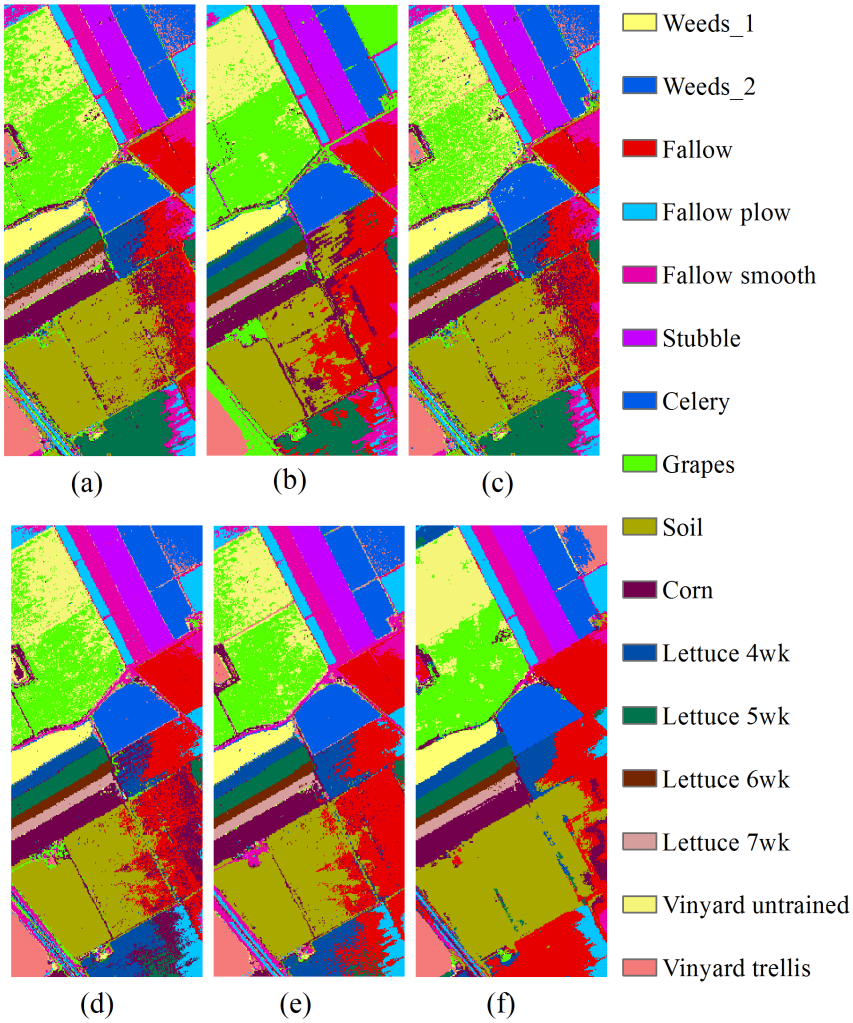


Fig. 5.4. Classification maps of Salinas Scene: (a) SVM; (b) EMP; (c) SRC; (d) JSM; (e) AJSM; (f) MLSR.

5.3.3 Parameter Analysis

This section focuses on the effects of the parameters settings on the classification performance. We first varied the value of positive parameter α that controls the influence of the ratio of the between-class and within-class distances, and the value was varied from 0 to 1 at an interval of 0.2. The experiments were conducted with AJSM on three data sets and the window sizes were fixed as the corresponding optimal values. In Fig. 5.5, the overall accuracies for three data sets fluctuate in a small range, and the best performances were obtained when α was set as 0.2 for all three data sets though the

trends for them were different. As α only controls the influence of each feature band, it is reasonable to apply the same value for MLSR in the experiments.

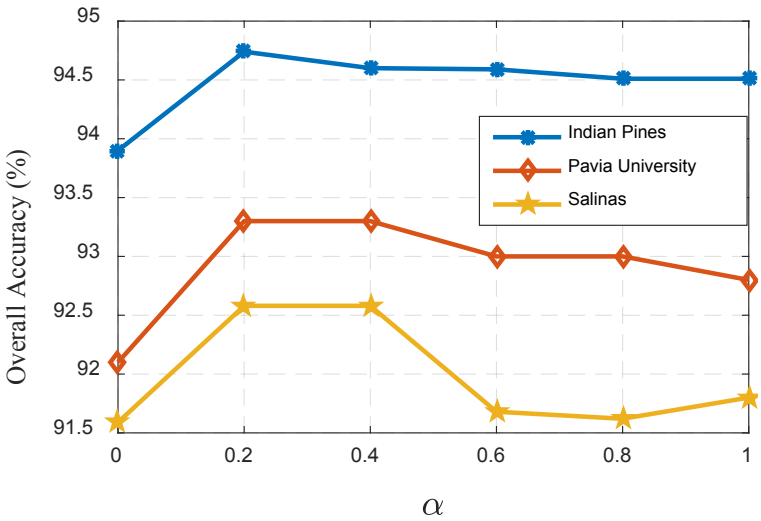
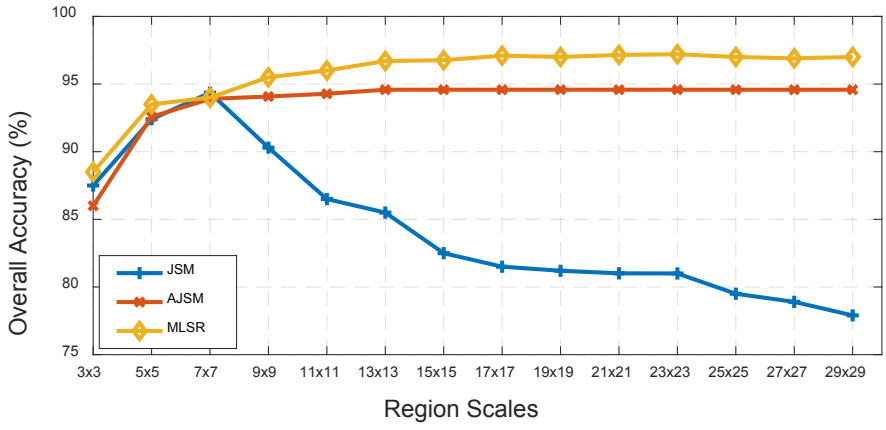
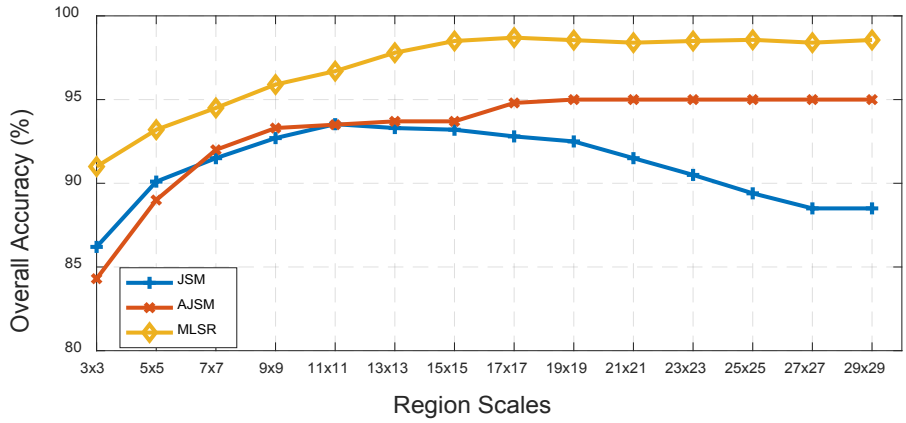


Fig. 5.5. The effect of controlling parameter α on classification resutls for three data sets.

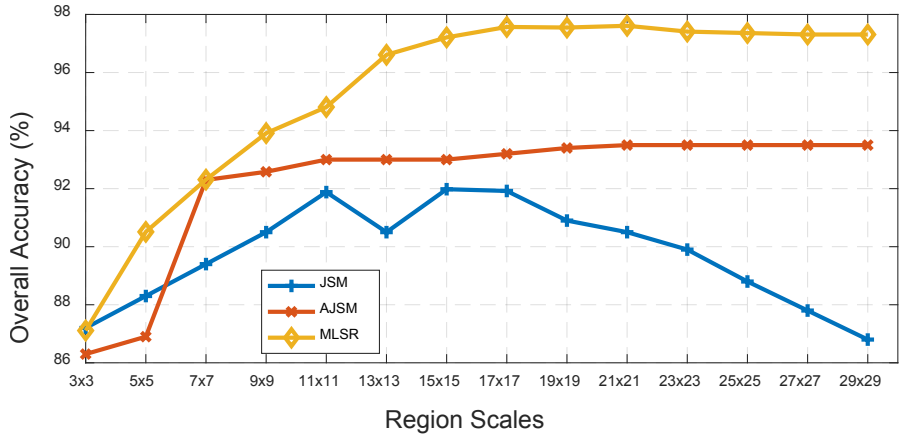
The effect of region scales for JSM, AJSM, and MLSR has also been analyzed in the experiments. In order to simply show the trends, the numbers of training and test data sets are selected to be the same as in the previous experiments. OA is shown in Fig. 5.6. For JSM, AJSM and MLSR, the region scales ranging from 3×3 to 29×29 at 2×2 intervals. As shown in Fig. 5.6, the best OA is achieved for JSM when the scale is set as 7×7, 11×11, and 15×15 for Indian Pines, Pavia University and Salinas, respectively. If the scale increases, the accuracy decreases dramatically. In most situations, AJSM performs better than JSM because the most useful information is preserved and the redundant information is rejected by the selection strategy. The accuracy for MLSR becomes stable when a larger region is selected. More specifically, the proposed MLSR performs better than other joint sparsity based models in most of regions. This result actually benefits from its mechanism of discarding outliers in the specific area, which provides a more reliable dictionary.



(a)



(b)



(c)

Fig. 5.6. The effects of region scales on JSM, AJSM and MLSR: (a) Indian Pines (b) Pavia University (c) Salinas Scene.

Another consideration is that the number of patches should be tested i.e. is having more patches better? To evaluate this, the adaptive framework is used to generate more patches. Specially, with ε set to $\{0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1\}$, we can define 11 patches. In each experiment, we randomly selected a patch subset with the number of $K \in \{1,2,3,4,5,6,7,8,9,10,11\}$ from these 11 patches and evaluated the performance of the method on three data sets. For each value of K, the experiment procedure is repeated 10 times with different subset selection. Fig. 5.7 shows the average OA result of the 10 iterations. As K increases, the performance of the framework also increases when $K \le 7$; however, it slightly decreases when $K \ge 8$. This trend shows that a certain number of patches are necessary for the improvement of the performance of the proposed method. However, too many patches can also result in a slight decrease in performance. In the experiment, we fixed five values $\{0.1,0.2,0.3,0.4,0.5\}$ and the last two values is determined from the remaining five values $\{0.6,0.7,0.8,0.9,1.0\}$ by cross-validation.

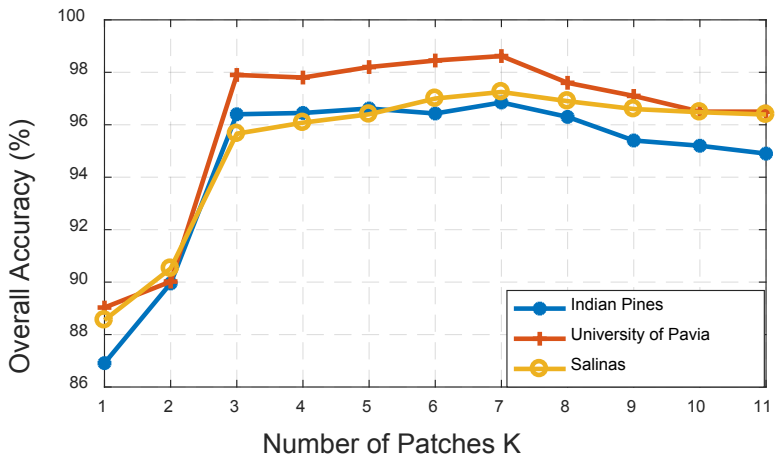
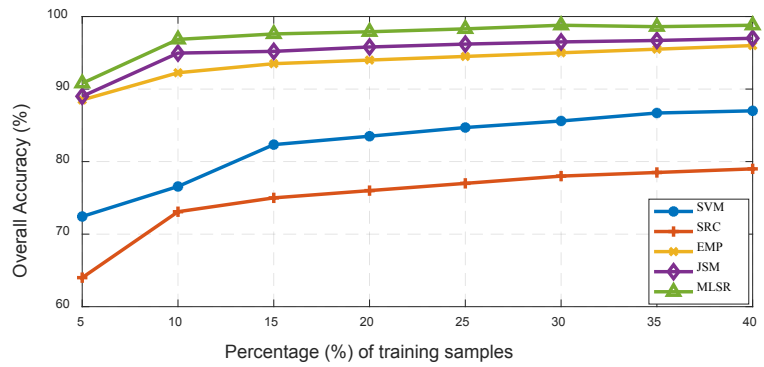


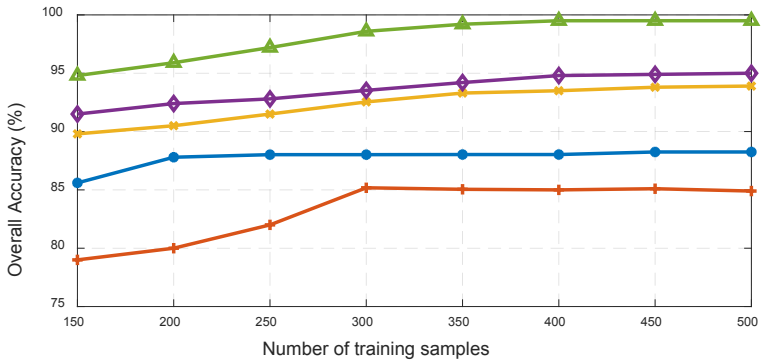
Fig. 5.7. The effect of number of patches of MLSR on three data sets.

We also conducted the experiments to evaluate the impact of the number of training samples per class for pixel-wise SVM, pixel-wise SRC, EMP with SVM, single scale JSM and the proposed MLSR. AJSM is not considered in this experiment as they exhibit a similar trend with JSM. Training samples are randomly chosen, and the rest as test samples. For the Indian Pines data sets, the number of training data ranges from 5% to 40% of the whole pixel counts at 5% intervals; For the Pavia University dataset, the

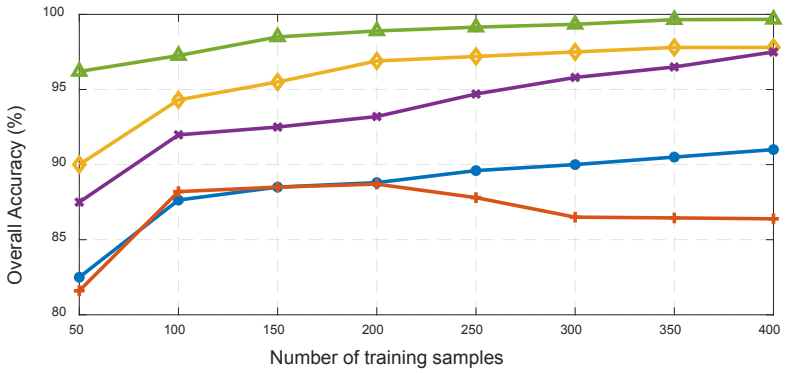
number of training samples per class ranges from 150 to 500 at 50 intervals; For the Salinas data sets, the number of training samples per class ranges from 50 to 400 at 50 intervals. Fig. 5.8 illustrates the classification results (OA) for these three data sets. As can be observed, less than 5% samples are needed for each class to obtain an OA over 90% for the Indian Pines data sets using the proposed MLSR. This is very promising because it is often difficult to collect a large training data sets in practice. For the Pavia University data sets, only 150 training samples are needed to obtain an OA of 95%. In fact, this accuracy is 3% higher than that by JSM and 4.5% higher than that by EMP with SVM. This is due to the fact that the local information included by the proposed MLSR outperforms others. The same trend can be concluded for the Salinas data set. In addition, the proposed MLSR produces very high accuracy and show the robustness with an increase of the number of training samples, and it can be observed that MLSR performs very well when training samples are limited.



(a)



(b)



(c)

Fig. 5.8. The effect of numbers of training data on five different methods: (a) Indian Pines; (b) University of Pavia; (c) Salinas Scene.

Finally, Table 5.7 shows the run time averaged over ten repeated experiments of the adopted classifiers and the proposed methods for the classification of the three data sets. As can be observed, the proposed MLSR takes more time, but its speed can be significantly accelerated by optimizing the codes with other programming languages and GPU.

Table 5.7. Run Time (Minutes) of All the Classifiers for the Classification of Three Data Sets.

	SVM	EMP	SRC	JSM	AJSM	MLSR
Indian Pines	3.3	3.7	0.7	1.5	1.8	2.5
University of Pavia	4.5	4.6	1.2	10.8	11.3	13.7
Salinas	3.4	4.2	0.8	5.6	5.7	8.9

5.4 Summary

In this chapter, we have introduced two novel sparse representation based hyperspectral classification methods. These proposed methods employ an adaptive weight matrix scheme as the neighbour selection strategy for the joint sparse matrix construction. The adaptive weight joint sparse model outperforms the traditional joint sparse models, however, it is designed for simple cases rather than complicated situations where the number of labeled training samples is not sufficient. This was overcome by introducing the second model i.e. the multi-level joint sparse model that can solve the complex

classification problem in a more effective way. The multi-level joint sparse model consists of two main parts: adaptive locality patches and a multi-level joint sparse representation model. This model is introduced to fully explore the spatial context within a given region for the test pixel. The proposed methods locally smooth the classification maps and preserve the relevant information for most labelled classes. Compared with other spatial-spectral methods and sparse representation based approaches, the proposed methods can provide a better performance on real hyperspectral scenes. This is consistent with the observation from the classification maps. Moreover, the experiments on the impact of the number of training samples also indicate that the proposed multi-level sparse approach leads to a more reliable result when only a limited number of training samples are available.

Chapter 6 A Multi-scale Conservative Smoothing Scheme and Adaptive Sparse Representation

6.1 Introduction

Chapter 5 presented a multi-level algorithm MLSR to exploit spatial information in a large neighbourhood of the pixel to be test. It preserves the most useful information and reduces redundant information based on an adaptive neighbour selection strategy; however, it may also discard some useful spatial information. In addition, MLSR does not exploit correlations among joint matrices from different levels. This chapter is an extension of Chapter 5, and this study aims to address the following questions: (1) How to exploit spatial and contextual information from multiple perspectives without discarding relevant information? (2) How to exploit the correlations among different perspectives?

In order to exploit the local relationship among the neighbouring pixels from the data with random noise, spatial smoothing has been applied to the preprocessing or postprocessing stage in HSI classification. In [154], a morphology-based filter was used for the noise reduction in the preprocessing stage prior to the classification. In [141], the authors adopted an anisotropic diffusion and a morphology algorithm to reduce the variability of the original image in both spatial and spectral dimensions. Most spatial smoothing techniques were used during the course of classification, and were applied to the probabilistic results obtained by other probabilistic classifiers. Li et al. [146] adopted a discontinuity preserving relaxation algorithm to process the probabilistic results obtained from MLR. MRFs exploit the continuity of the classification maps in a probabilistic sense [57, 66]. In [155], the authors applied a hierarchical guidance filtering which is an extension of rolling guidance filter (RGF) to generate the spectral-spatial features for HSI classification. A discriminative low-rank Gabor filtering method was used in [156] to extract suitable features based on the properties of HSIs, and spatial smoothness is achieved and class separability is enhanced. Spatial smoothing is also a technique to emphasize the main features after suppressing the undesired variation within a homogenous region. In this context, spatial structures and geometrical

features of HSIs can be enhanced and revealed after filtering, especially for the edges. However, the spatial properties can be present at various spatial scales instead of a single fixed scale, hence it is difficult to identify a single filter parameter (e.g. the region scale in spatial smoothing) suitable for capturing all of them simultaneously. Multilevel analysis can be applied to address this issue. Based on the aforementioned spatial smoothing theory, we propose a multiscale conservative smoothing algorithm to reveal the spatial characteristics at several levels.

Furthermore, inspired by the trend of multiple feature learning in the remote sensing image processing field [157, 158], various multiple feature learning-based SRC models have been proposed for the HSI classification. Zhang et al. [102] proposed a multi-feature joint sparse representation classification (MF-JSRC) for the fast classification of HSIs, and a $\ell_{\text{row},0}$-norm penalty was applied across various features. In [103], the authors improved the HSI classification performance by constructing a multi-task JSM at a super-pixel level. A shape adaptive window [98, 159] was selected for each pixel in a JSM and multi-feature SRC respectively so that the similarities and diversities of multiple features can be exploited more effectively. In this study, we propose a multiscale conservative smoothing algorithm and an adaptive sparse representation to integrate the characteristics of the series of filtered HSIs. Once different representations for a given unknown pixel are constructed by the proposed adaptive sparse representation method, the sparse coefficients of the pixel will be obtained and then be used for classification.

The proposed conservative smoothing algorithm considers adaptive weights for different neighbouring pixels around the central pixel, and the weight is measured by the spectral similarity between the neighbouring pixel and the central pixel. Therefore, it can reveal the spatial textural information and avoid oversmoothing. The multiscale-based strategy can handle complementary information carried by different HSIs from various scales used in the conservative smoothing algorithm. It should be noted that the proposed classification algorithm is also a kind of multiple feature learning-based classifiers, and the proposed conservative smoothing algorithm generates the features from different perspectives with different scales. For this proposed method, there is no

need to predefine the specific categories of features. In this chapter, the proposed HSI classification framework is named as multi-scale conservative smoothing scheme and adaptive sparse representation (MCSSR).

The rest of this chapter is organized as follows: Section 6.2 introduces the proposed conservative smoothing algorithm and the adaptive sparse representation in detail. Experimental results are presented in Section 6.3. Finally, conclusions and future works are provided in Section 6.4.

6.2 Proposed Framework

6.2.1 Conservative Smoothing

It has been acknowledged that an effective exploitation of the spectral-spatial information can improve the performance of the HSI classification [7]. In this chapter, a conservative smoothing algorithm is proposed to enforce the spatial consistency for the neighbouring pixels in the original HSI cube for noise removal and spatial structure enhancement.

Inspired by the edge preserving problem in [141, 146], for a test pixel $\mathbf{x}_i$, we propose conservative smoothing of the original image cube by solving the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{x}_i} \sum_{l=1}^{L} \sum_{j \in S_i} w_{ij} \left\| \mathbf{x}_{jl} - \mathbf{x}_{il} \right\|^2, \\ &\text{subject to} \quad \mathbf{x}_{il} > 0, \mathbf{1}^T \mathbf{x}_i = 1. \end{aligned} \tag{6.1}$$

where the pixel vector $\mathbf{x}_i$ should be normalized before the optimization. $l = 1, 2, ..., L$ denotes the spectral dimensionality of the pixel vectors, $\mathbf{1}$ denotes a vector column of L 1s. S_i denotes a local region of the neighbourhood of $\mathbf{x}_i$, and w_{ij} represents the weight controlling the influence of the neighbouring pixel $\mathbf{x}_j$ to $\mathbf{x}_i$. It should be noted that the spatial consistency is implemented within each spectral dimension in Equation (6.1). In

this chapter, the Euclidean distance is considered to evaluate the spectral similarity between $\mathbf{x}_j$ and $\mathbf{x}_i$, and the weight is computed as:

$$w_{ij} = 1 / \sqrt{\sum_{l=1}^{L} (\mathbf{x}_{il} - \mathbf{x}_{jl})^2}. \tag{6.2}$$

The weight is the inverse distance between two spectral vectors in the spectral domain, and this is consistent with the principle that a bigger weight should be assigned to the neighbouring pixel which has higher similarity with the test pixel i.e. a smaller distance in the spectral domain. Equation (6.2) is strictly convex, and the unique solution can be obtained by minimizing the objective function with respect to the variable $\mathbf{x}_{il}$ at each iteration. The implementation of the algorithm can be found in Algorithm 6.1.

Algorithm 6.1. The pseudocode of the proposed conservative smoothing algorithm.

Input: Original image cube $\mathbf{I}$, maximum iteration number t, error parameter err, convergence controlling parameter τ.

Initialization: initialize the error parameter $err^1 = \|\mathbf{I}\|$, and normalize each pixel vector to have unit ℓ_2 norm.

For each pixel $\mathbf{x}_i$ in the image cube $\mathbf{I}$

For $l = 1 : L$

iter=1

$err_l^1 = err^1$

While $err_l^{iter+1} - err_l^{iter} \le \tau$, **or** $iter \le t$

$$\mathbf{x}_{il}^{iter+1} = \sum_{j \in S_i} w_{ij} \mathbf{x}_{jl}^{iter} / \sum_{j \in S_i} w_{ij}$$

$$err_l^{iter+1} = \left\| \mathbf{X}_l^{iter+1} - \mathbf{X}_l^{iter} \right\| / \left\| \mathbf{X}_l^{iter} \right\|$$

End while

End for

End for

Output: The processed image cube $\tilde{\mathbf{I}}$

From the smoothing filter's perspective, the proposed algorithm is more reliable than the commonly used low pass (LP) filter and the medium filter for the HSI image classification in terms of reducing the undesirable intensity variability and enhancing the contrast of the edges. The LP filter replaces the value of the central pixel with the average of the values within the window, and the medium filter replaces the central pixel value with the medium value within the window. Both filters can be sensitive to the dissimilar values within the window, especially the LP filter is more sensitive. The proposed method can overcome this problem.

The proposed smoothing algorithm can be applied to correct the spatial distortions by considering the local relationship among neighbouring pixels. It can help reduce noise and enhance spatial textural information, however, the smoothing over a single local region may provide limited structures and contextual information. Therefore we apply this smoothing algorithm at several scales. With M window sizes, i.e., local regions $\mathbf{S}_i$, $i=1, \ldots M,$ it will result in M HSIs. In other words, the proposed algorithm can be used to generate multiscale spatial features. Figure 6.1 shows an example of the results obtained by the proposed conservative smoothing algorithm with different scales applied, and it can be observed that different characterizes are displayed.

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

Fig. 6.1. The false colour images of proposed conservative smoothing scheme on Indian Pines data set (band: 50, 27, 17): (a) original image; (b) Scale = 3×3; (c) Scale = 15×5; (d) Scale = 7×7.

6.2.2 Adaptive Sparse Representation

In this chapter, an adaptive sparse representation method is introduced to deal with the different properties from the series of previously obtained HSIs. Let $\mathbf{I}_m$ ($m = 1,...,M$) denote the m - th HSI, where M is the number of window sizes applied in Section 6.2.1. With M different window sizes applied, there will geneate M filtered HSIs. According to the JSRC theory, a joint matrix can be constructed for the test pixel $\mathbf{x}_i$ within a defined window, and in this chapter, a shape adaptive [98, 159] window is used instead of a fixed-size window. Let $\mathbf{X}_i^m$ denote the constructed matrix associated with $\mathbf{I}_m$ for the given test pixel $\mathbf{x}_i$, where $\mathbf{X}_i^m = [\mathbf{x}_{i1}^m, \mathbf{x}_{i2}^m, ..., \mathbf{x}_{iI^*}^m]$. The optimization function of the sparse representation model for the m - th image can be expressed as follows:

$$\begin{aligned} &\tilde{\mathbf{R}}^m = \arg\min_{\mathbf{R}^m} \left\| \mathbf{X}_i^m - \mathbf{D}^m \mathbf{R}^m \right\|_F, \\ &\text{subject to} \quad \left\| \mathbf{R}^m \right\|_{row,0} \le K. \end{aligned} \tag{6.3}$$

where $\mathbf{R}^m$ is the coefficient matrix corresponding to a dictionary over $\mathbf{X}_i^m$. Given M different HSIs, different matrix sets for the test pixel can be obtained, and the objective function of the sparse representation model can be defined as:

$$\begin{aligned} &\hat{\mathbf{R}} = \arg\min_{\mathbf{R}} \sum_{m=1}^{M} \left\| \mathbf{X}_i^m - \mathbf{D}^m \mathbf{R}^m \right\|_F, \\ &\text{subject to} \quad \left\| \mathbf{R} \right\|_{row,0} \le K. \end{aligned} \tag{6.4}$$

where $\mathbf{R} = [\mathbf{R}^1, ..., \mathbf{R}^m, ..., \mathbf{R}^M]$ is the sparse coefficient matrix. The Equation (6.4) can be solved by the SOMP algorithm jointly. In order to consider the correlation among different matrices sets, we adopt the adaptive joint sparse constraint [93] for the classification task:

$$\begin{aligned} &\hat{\mathbf{R}} = \arg\min_{\mathbf{R}} \sum_{m=1}^{M} \left\| \mathbf{X}_i^m - \mathbf{D}^m \mathbf{R}^m \right\|_F, \\ &\text{subject to} \quad \left\| \mathbf{R} \right\|_{adaptive,0} \le K. \end{aligned} \tag{6.5}$$

The implementation of the optimization of Equation (6.5) is shown in Algorithm 6.2. Similar to the multiscale adaptive sparse representation (MASR) [93] algorithm, the proposed algorithm can be iteratively optimized until the termination criterion is satisfied. The best representation atoms for different matrices sets and different classes are selected, and then the adaptive set is determined by recording the indexes for the atoms across each set and each class at each iteration.

After the sparse coefficient is obtained, the test pixel should be labeled as the class that has the minimum reconstruction residual:

$$class(\mathbf{x}) = \arg\min_{c=1,...,C} \sum_{m=1}^{M} \left\| \mathbf{X}_i^m - \mathbf{D}_c^m \mathbf{R}_c^m \right\|_F. \tag{6.6}$$

where $\mathbf{D}_c^m$ is the dictionary corresponding to the $m\text{-}th$ image for the $c\text{-}th$ class. The illustration of the adaptive sparse representation is shown in Fig. 6.2. The outline of the whole framework is displayed in Fig. 6.3.

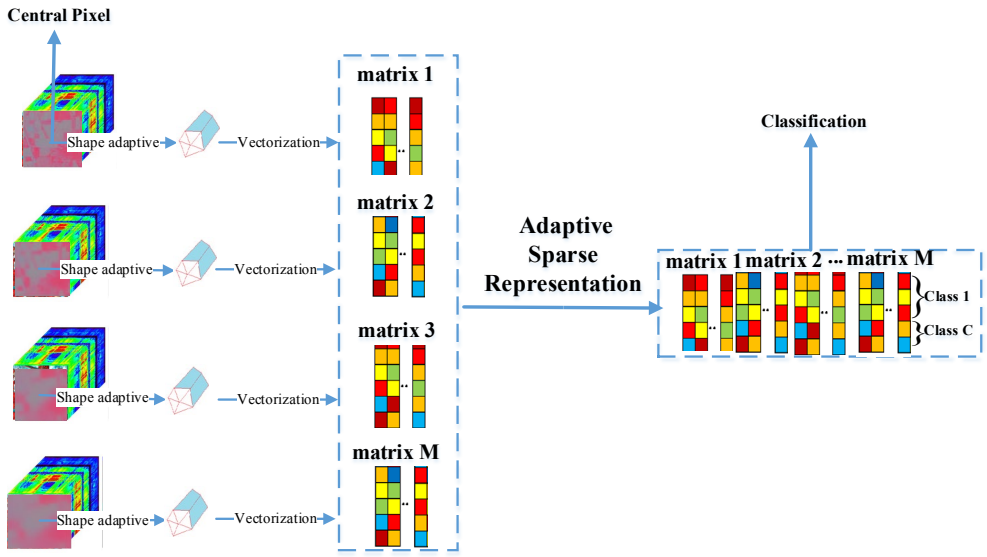


Fig. 6.2. The illustration of the adaptive sparse representation strategy.

Algorithm 6.2. The pseudocode of adaptive sparse representation.

Input: multiple adaptive sparse matrices for test pixel $\mathbf{x}_i$ associated to the processed imagery $\mathbf{I}_m$: $\mathbf{X}_i^m$ ($m = 1,\ldots,M$), dictionaries: $\mathbf{D}^m$ ($m = 1,\ldots,M$), predefined sparsity level K, number of processed imageries M, training data set for each image $\mathbf{I}_T^m (m = 1,\ldots,M)$.

Initialization: initialize dictionary $\mathbf{D}^m = \mathbf{I}_T^m$, and normalize the columns of each dictionary to have unit ℓ_2 norm, initialize the residual matrix for each HSI $\mathbf{E}_1^m = \mathbf{X}_{iSA}^m$, the iteration number $iter = 1$, and adaptive index matrix $\mathbf{I}_1 = \varnothing$.

For $iter \le K$

1. **For** $m = 1:M$

   Compute the corresponding residual correlation matrix**.**

   For $c = 1:C$

---

(1): find the corresponding residual correlation matrix for each class;

(2): find the best representation atoms' indexes and the corresponding coefficient values for each class and each matrix set.

End for

End for

2. Sum the coefficients values across M matrix sets for each class.

3. Find the maximum value's index for each class.

4. Combine the atom's indexes for each class.

5. Compute the adaptive set.

6. Update the adaptive index set.

7. **For** $m = 1 : M$

(1): Find the corresponding adaptive index set for each set;

(2): Estimate the sparse representation coefficients for each set;

(3): Update the residual matrix for each set.

End for

8. $iter = iter + 1$.

End for

Output: sparse coefficient matrix for each set $\mathbf{R}^m$ ($m = 1, \ldots, M$).

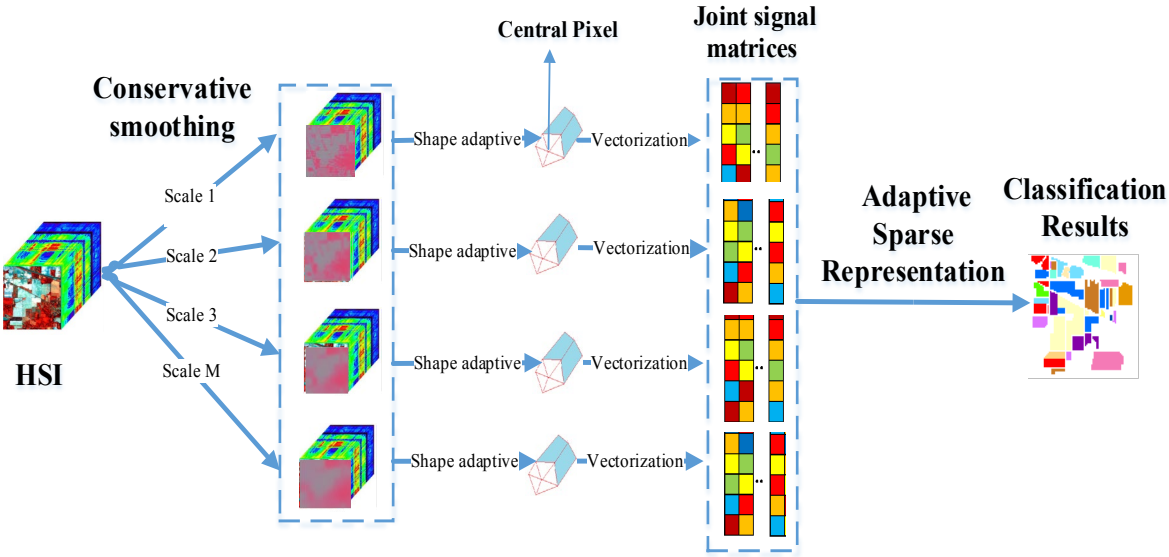


Fig. 6.3. The pipeline of the proposed framework.

Table 6.1. Classification Accuracies for Indian Pines Image Obtained by the Proposed Method with Single Scale.

Class	1×1	3×3	5×5	7×7	9×9	11×11	MCSSR
Alfalfa	92.31	97.44	94.87	97.44	100.00	94.87	100.00
Corn-no till	81.58	86.49	90.46	90.53	85.33	81.94	94.70
Corn-min till	91.71	92.98	91.58	75.77	89.67	82.27	98.08
Corn	96.74	97.28	96.74	85.87	98.91	73.37	99.15
Grass/pasture	93.96	96.20	93.29	94.63	91.50	92.62	98.59
Grass/trees	97.56	95.55	93.40	94.69	97.13	92.11	96.92
Grass/pasture-mowed	100.00	100.00	100.00	100.00	100.00	90.91	100.00
Hay-windrowed	99.54	98.63	100.00	100.00	99.54	100.00	99.59
Oats	100.00	100.00	80.00	100.00	100.00	100.00	100.00
Soybeans-no till	90.63	93.90	85.40	87.80	86.49	86.06	96.28
Soybeans-min till	80.44	81.14	87.72	90.74	79.74	85.44	96.23
Soybeans-clean till	88.83	90.07	82.98	82.27	86.35	67.38	98.21
Wheat	98.77	100.00	99.38	94.44	100.00	94.44	99.53
Woods	93.89	94.77	98.55	98.07	94.45	98.55	99.69
Buildings-grass-trees	75.76	99.09	75.45	90.61	99.09	89.09	99.74
Stone-steel towers	95.56	97.78	97.78	95.56	100.00	100.00	100.00
OA (%)	88.04	90.29	90.58	90.58	88.53	87.01	**97.38**
AA (%)	92.33	95.08	91.73	92.40	94.26	89.32	**98.54**
k	0.87	0.89	0.89	0.89	0.87	0.85	**0.97**

Table 6.2. Class Information and Classification Accuracies for Indian Pines Image Obtained by Different Classifiers.

Class	Train/Test	SVM	SRC	JSRC	MF-SRC	MF-JSRC	MNFL	MASR	MFASR	MCSSR
Alfalfa	15/32	87.04	89.13	100.00	98.15	100.00	97.83	97.83	100.00	100.00
Corn-no till	50/1379	68.13	43.49	77.45	81.24	83.12	82.49	92.79	93.14	94.70
Corn-min till	50/781	64.15	55.42	88.43	90.05	90.84	90.12	91.93	98.85	98.08
Corn	50/187	88.46	75.11	97.05	100.00	100.00	99.58	100.00	98.91	99.15
Grass/pasture	50/434	89.74	83.64	95.86	96.18	96.27	98.34	98.14	97.32	98.59
Grass/trees	50/682	91.97	87.40	98.08	98.53	100.00	100.00	99.86	96.27	96.92
Grass/pasture-mowed	15/13	96.15	92.86	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Hay-windrowed	50/429	97.96	91.42	98.74	100.00	99.58	100.00	99.79	99.32	99.59
Oats	15/5	85.00	95.00	100.00	100.00	100.00	85.00	100.00	100.00	100.00
Soybeans-no till	50/921	72.42	63.58	97.12	91.32	97.84	92.70	94.34	94.77	96.28
Soybeans-min till	50/2406	68.88	49.37	76.78	84.00	84.36	82.93	93.81	95.62	96.23
Soybeans-clean till	50/545	78.99	52.45	87.18	94.46	93.25	94.77	96.63	98.23	98.21
Wheat	50/156	100.00	94.15	100.00	99.53	100.00	100.00	100.00	99.38	99.53
Woods	50/1218	88.10	75.65	93.52	93.43	98.10	97.00	99.68	99.92	99.69
Buildings-grass-trees	50/335	74.47	56.99	95.34	97.63	97.67	93.78	100.00	98.48	99.74
Stone-steel towers	15/78	100.00	97.85	98.92	100.00	100.00	100.00	100.00	97.78	100.00
OA (%)		77.52	62.70	87.90	90.44	92.05	90.95	95.97	96.70	**97.38**
AA (%)		84.47	75.22	94.03	95.28	96.32	94.66	97.80	98.00	**98.54**
k		0.75	0.58	0.86	0.89	0.91	0.90	0.95	0.96	**0.97**

Table 6.3. Class Information and Classification Accuracies for University of Pavia Image Obtained by Different Classifiers.

Class	Train/Test	SVM	SRC	JSRC	MF-SRC	MF-JSRC	MNFL	MASR	MFASR	MCSSR
Asphalt	66/6565	95.32	74.18	58.23	90.70	95.34	99.26	80.29	98.10	98.89
Meadows	186/18463	96.31	56.82	98.69	92.09	83.48	91.59	98.92	99.99	99.68
Gravel	21/2078	55.26	66.36	82.28	93.14	87.57	85.99	75.27	93.12	98.08
Trees	31/3033	82.83	94.19	82.21	95.92	85.54	98.60	81.07	90.37	93.50
Meta sheets	13/1332	98.88	98.22	97.25	99.26	99.70	99.63	99.85	100.00	99.85
Bare soil	50/4979	77.95	92.23	81.94	83.46	98.71	88.84	85.68	99.12	99.16
Bitumen	13/1317	2.56	72.86	93.98	90.60	99.92	95.41	91.05	99.54	99.85
Bricks	37/3645	79.60	80.47	90.17	50.84	98.10	83.51	96.66	97.64	97.78
Shadows	9/938	99.47	99.68	25.13	100.00	99.89	100.00	40.55	88.26	85.81
OA (%)		86.82	71.61	85.91	87.99	90.10	92.54	90.34	98.09	**98.51**
AA (%)		76.47	81.67	78.88	88.44	94.25	93.65	83.26	96.24	**96.95**
k		0.82	0.65	0.81	0.84	0.87	0.90	0.87	0.97	**0.98**

Table 6.4. Class Information and Classification Accuracies for University of Houston Obtained by Different Classifiers.

Class	Train/Test	SVM	SRC	JSRC	MF-SRC	MF-JSRC	MNFL	MASR	MFASR	MCSSR
Grass-Healthy	99/1152	98.32	99.12	98.48	98.16	100.00	97.76	99.28	96.88	97.52
Grass-Stressed	95/1159	98.48	97.45	97.37	92.19	96.33	98.09	91.23	96.57	99.20
Grass-Synthetic	96/601	99.86	99.71	97.42	98.71	99.57	99.86	100.00	100.00	100.00
Tree	94/1150	95.90	94.86	98.79	94.61	98.47	98.31	99.52	96.95	97.27
Soil	93/1149	98.07	98.47	100.00	100.00	100.00	99.68	99.92	100.00	100.00
Water	91/234	99.38	98.77	97.85	90.46	99.38	100.00	99.08	97.85	99.38
Residential	98/1170	86.67	83.99	78.55	71.29	84.94	82.81	95.50	95.98	92.43
Commercial	95/1149	82.32	89.95	92.28	93.97	92.60	94.29	89.47	95.02	96.78
Road	96/1156	89.86	79.71	92.65	94.25	90.89	94.09	93.53	94.73	98.80
Highway	95/1132	91.85	91.04	93.64	98.21	94.38	96.66	100.00	99.02	100.00
Railway	90/1145	91.98	78.95	93.60	96.60	93.93	93.28	98.22	98.14	99.19
Parking Lot 1	96/1137	85.97	81.43	91.32	97.89	95.38	91.48	98.38	98.22	98.95
Parking Lot 2	92/387	76.33	56.72	88.27	99.57	92.11	85.29	97.65	98.08	97.01
Tennis Court	90/338	99.77	100.00	100.00	100.00	96.96	98.36	100.00	100.00	100.00
Running Track	93/567	99.09	100.00	100.00	94.09	100.00	100.00	100.00	100.00	100.00
OA (%)		92.52	89.91	94.20	94.21	95.22	95.04	97.00	97.52	**98.24**
AA (%)		92.92	90.01	94.68	94.67	95.66	95.33	97.45	97.83	**98.44**
k		0.92	0.89	0.94	0.94	0.95	0.95	0.97	0.97	**0.98**

6.3 Experimental Results and Discussion

In this section, three widely used benchmark data sets are utilized to evaluate the proposed HSI classification framework in different scenarios. Detailed analysis on the impacts of parameters influential to the experimental results is also performed. All the experiments were conducted by Matlab 2013b in an environment of Intel (R) Core (TM) i7-4790 CPU 3.6GHz and 16 GB of RAM.

6.3.1 Data Sets

For the comparison with other similar methods in the literature, the effectiveness of the proposed method is conducted on three data sets, i.e. the AVIRIS Indian Pines data set, the ROSIS University of Pavia data set and the CASI University of Houston data set *grss_dfc_2013*.

6.3.2 Experimental Setting

The proposed MCSSR method is compared with several classifiers in this chapter to validate the performance. Pixel-wise SVM (referred to as SVM hereafter) [152], pixel-wise SRC (referred to as SRC hereafter) [22], JSRC [22], multi-feature-based SRC (MF-SRC) [102], multi-feature-based JSRC (MF-JSRC) [102], the extended multi-attribute profiles-based multiple nonlinear feature learning classifier (MNFL) [158], MASR [93], multiple feature learning adaptive sparse representation (MFASR) [159] are used as benchmarks. SVM and SRC classify the images with only spectral information, while JSRC, MF-SRC, MF-JSRC, MNFL, MASR, MFASR and the proposed MCSSR are the classifiers that utilize both spectral and spatial information.

During the experiments, SVM was implemented using the SVM library with the Gaussian kernel [152]. The sparsity level is set as 3 for all sparse representation-based methods (e.g. JSRC, MASR, MF-SRC, MF-JSRC and MCSSR) as suggested in [22]. Based on trial and error, the local region was determined from 3×3 to 19×19 for JSRC, MASR, and MF-JSRC; the number of patches M is set as 6 for the proposed algorithm MCSSR; and the local region size was set as 1×1 (the original data set), 3×3, 5×5, 7×7,

9×9 and 11×11, respectively. Three quantitative metrics, OA, AA and kappa coefficient (k) are selected for the quantitative validation in this chapter.

6.3.3 Experimental Results

The first experiment is conducted on the Indian Pines data set, and in this experiment, the results obtained by the introduced adaptive sparse representation with the proposed MCSSR are compared with the ones obtained by the single-scale smoothing strategy with the adaptive sparse representation. Table 6.1 shows the quantitative results. It can be observed that the proposed framework outperforms the method that was conducted on the single scales separately, and the results demonstrate the effectiveness of the proposed multi-scale smoothing based adaptive sparse representation strategy. The performance enhancement of the proposed MCSSR compared to the single-scale based algorithm is achieved by combining the various features generated by the multi-scale filter in an adaptive strategy. For those single-scale-generated HSIs, the one processed by the proposed smoothing algorithm with local region set as 5×5 carries more distinctive classification information, and obtained the best OA.

The second experiment is also conducted on the Indian Pines data set, and the proposed method is compared with the SVM, SRC, JSRC, MF-SRC, MF-JSRC, MASR, and MFASR. The experimental results are tabulated in Table 6.2, and the classification maps are illustrated in Fig. 6.4. As can be observed from the resultant table, all the spectral-spatial classifiers performed better than spectral feature-based approaches (i.e. SVM and SRC), which demonstrates the importance of taking the contextual information into account. In addition, the proposed MCSSR achieved an OA of 97.38%, which is superior to all the other comparative classifiers, especially MCSSR gained a higher accuracy when compared with other multi-task learning based algorithms (e.g. MF-SRC, MF-JSRC, MNFL, MASR and MFASR). Compared to the single-scale JSRC, MASR achieved a higher accuracy with a relative OA of 8.07%. MASR and the proposed MCSSR both exploit the contextual information in a multiscale way. However, MCSSR preserved more discriminative information for classification than MASR, which can be observed from the results. From the classification maps, one can observe how the proposed framework produces the classification results. SVM, SRC and JSRC

produced a more “noisy” look with more scatter points shown on the maps. In contrast, the multi-task learning based methods (i.e. MASR, MF-SRC, MF-JSRC, MFASR and MCSSR) generated smoother maps, and the proposed method yielded the best performance. Moreover, for the classes (e.g. Alfalfa, and Oats) that have limited training samples, the proposed algorithm obtained the best classification results. The proposed MCSSR also shows a better capability in classifying the similar pixels, such as classes Grass/tress, Grass/pasture, and Grass/pasture-mowed.

The classification results for University of Pavia image are shown in Table 6.3, and the illustrative results are displayed in Fig. 6.5. As can be observed, the proposed method still outperformed all the comparative classifiers. Specifically, from the resultant table, it can be seen that multi-feature learning methods (i.e. MF-SRC, MF-JSRC, MNFL and MFASR) performed better than the single feature learning methods (i.e. SVM, SRC and JSRC); however, the performance is inferior to those obtained by the proposed method, which demonstrates the ability of the proposed HSI classification framework that combines different spatial contextual information in a flexible way. From the classification maps, it can be seen that the proposed algorithm generated a better appearance than the other classifiers, especially in those areas with more mixed samples. It should be also noted that the accuracies for MF-SRC, MF-JSRC, MASR, MFASR and the proposed MCSSR are higher than JSRC, which indicates that the sparse representation based methods can be improved by adopting a multi-task learning strategy.

The accuracies obtained by various classification methods are reported in Table 6.4 for the University of Houston image. As can be observed from the resultant table, the proposed multi-scale smoothing-based method shows more improvements over the spectral-spatial approach, JSRC, for this imagery in comparison with other spectral-spatial classifiers. The classification maps shown in Fig. 6.6 are consistent with previous two image sets, where the proposed method obtains the best visual quality.

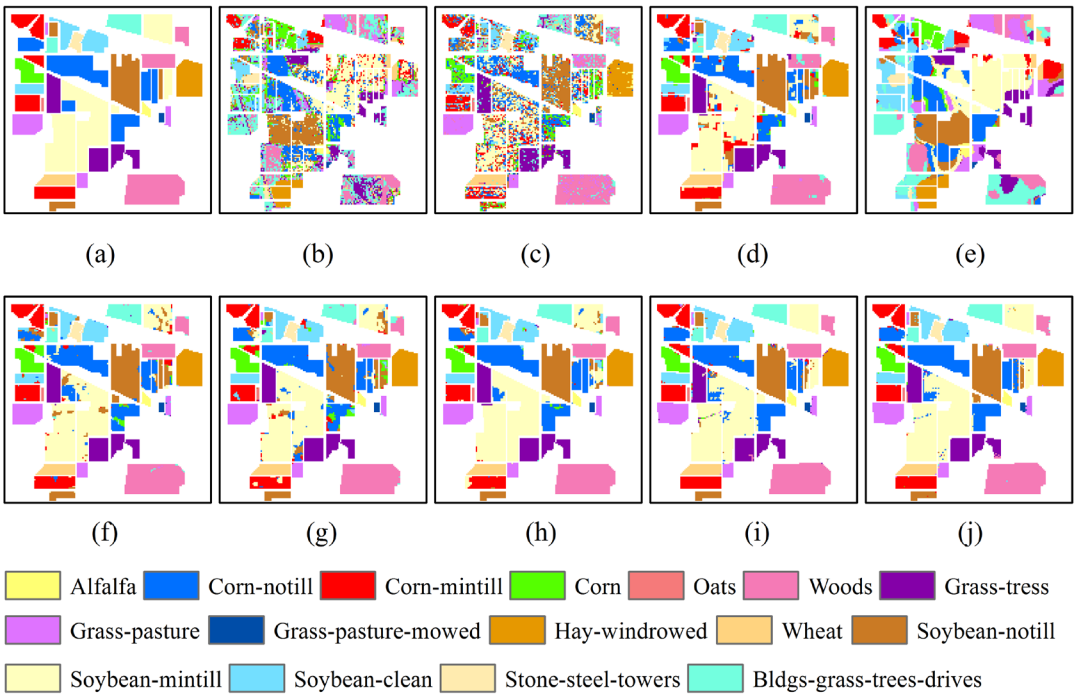


Fig. 6.4. Classification maps for the Indian Pines image. (a) Groundtruth map; (b) SVM; (C) SRC; (d) JSRC; (e) MF-SRC; (f) MF-JSRC; (g) MNFL; (h) MASR; (i) MFASR; (j) MCSSR.

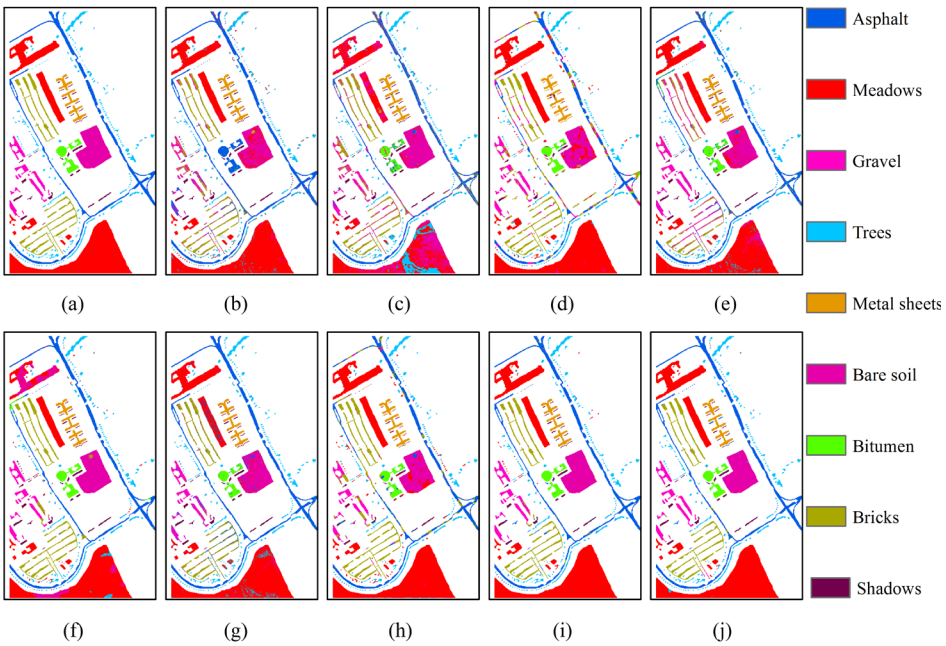


Fig. 6.5. Classification maps for University of Pavia image. (a) Groundtruth map; (b) SVM; (C) SRC; (d) JSRC; (e) MF-SRC; (f) MF-JSRC; (g) MNFL; (h) MASR; (i) MFASR; (j) MCSSR.

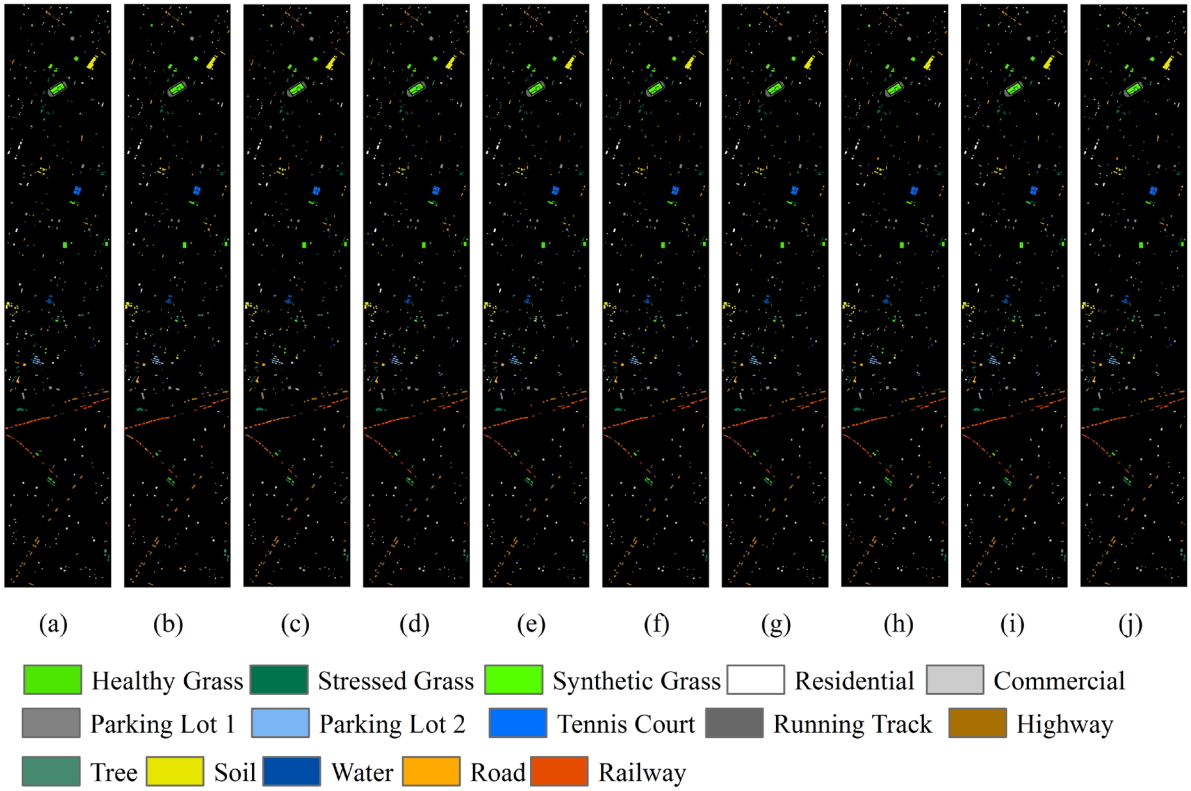
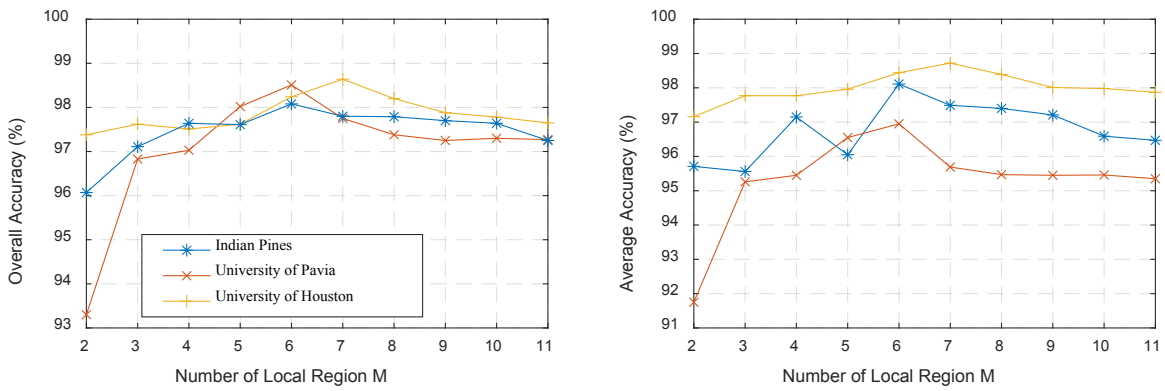
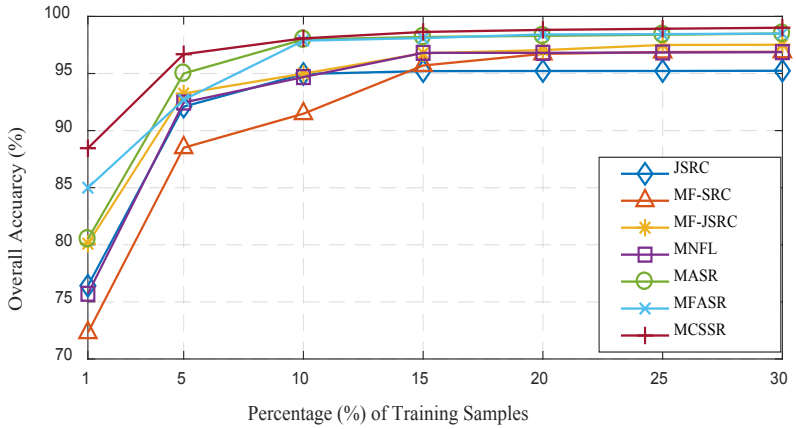


Fig. 6.6. Classification maps for University of Houston image. (a) Groundtruth map; (b) SVM; (C) SRC; (d) JSRC; (e) MF-SRC; (f) MF-JSRC; (g) MNFL; (h) MASR; (i) MFASR; (j) MCSSR.

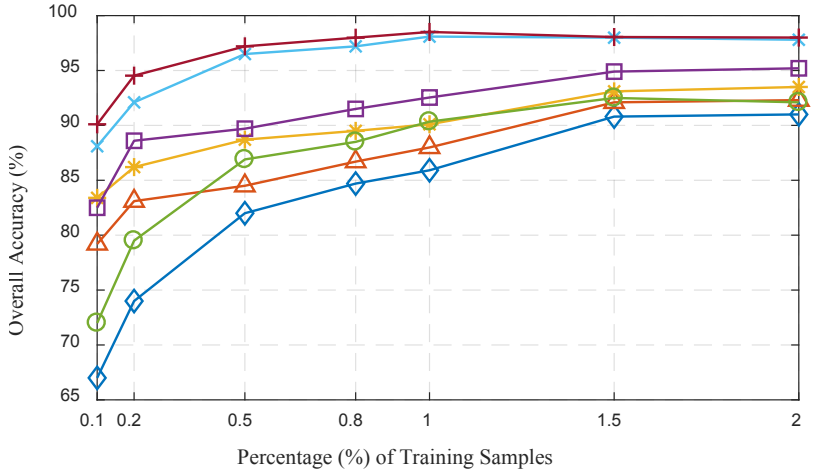


(a) (b)

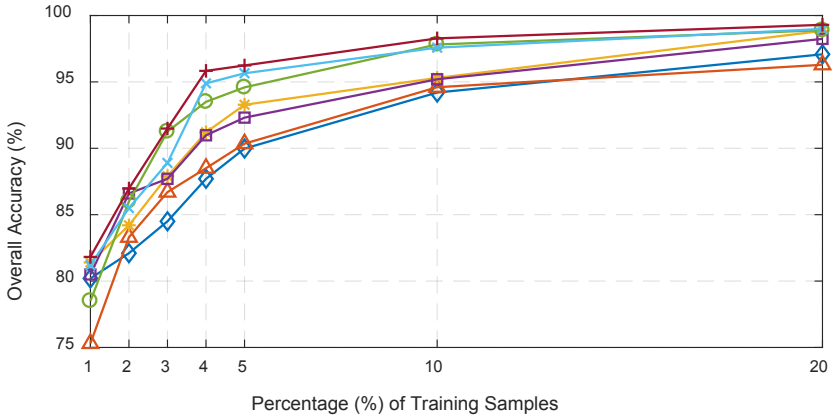
Fig. 6.7. Effect of the number of local regions adopted in the proposed conservative smoothing algorithm on the classification performance for three data sets: (a) Overall Accuracy; (b) Average Accuracy.



(a)



(b)



(c)

Fig. 6.8. Effect of the number of training samples on the accuracies for different spectral-spatial classifiers for three data sets: (a) Indian Pines; (b) University of Pavia; (c) University of Houston.

Table 6.5. Run Time (Minutes) of All the Classifiers for the Classification of Three Data Sets.

	SVM	SRC	JSRC	MF-SRC	MF-JSRC	MNFL	MASR	MFASR	MCSSR
Indian Pines	3.3	0.7	1.5	1.4	10.2	18.9	7.8	7.6	8.2
University of Pavia	4.5	1.2	10.8	9.7	30.2	38.9	40.2	34.6	40.7
University of Houston	3.4	0.9	5.6	6.5	20.2	24.2	19.8	26.4	30.5

6.3.4 Parameter Analysis

In this section, the effects of different parameters on the classification performance of the proposed framework are analyzed. The number of local regions for the proposed conservative smoothing algorithm is validated firstly in the experiments, and M ranging from 2 to 11 with the local region varying from 3×3 to 21×21. Each scale represents the smoothing algorithm applied on the current scale and its smaller scales. The OAs and AAs for the three data sets with different numbers of local regions are illustrated in Fig. 6.7. As can be observed from the figure, MCSSR achieved the best accuracies when the number of local region is set as 6 for the Indian Pines and the Pavia data sets, and 7 for the Houston image. When the number of patches increases, the accuracy will deteriorate. This is due to the fact that the redundancy among too many patches compromises the accuracy gain, and the irrelevant information would be likely chosen with a large local region size, which misleads the final classification.

The impact of the number of training samples on different spectral-spatial classifiers JSRC, MF-SRC, MF-JSRC, MNFL, MASR, MFASR and the proposed MCSSR are also evaluated in this section for the three data sets, respectively. Different percentages (from 1% to 30%) of the samples were randomly chosen as the training sets for the Indian Pines image, and the remaining as the test sets. For the University of Pavia data set, different percentages (0.1 to 2%) were randomly selected as the training samples, and the remaining as the test set. For the Houston data set, different percentages (1% to 20%) were randomly selected as training samples. Fig. 6.8 shows the overall classification accuracies for different techniques under the condition of different number of training samples. As shown in the figures, the performances of most classifiers were

improved with the increase of the number of training samples, and the accuracies tend to be stable when the number of training samples further increases. The proposed MCSSR consistently outperformed the other classifiers in most cases, especially with a small number of training samples.

The sparsity level K is also influential to the classification results. Fig. 6.9 shows the effect of the sparsity level on the OAs of the proposed MCSSR method for three data sets. The sparsity level varies from 1 to 20 in the experiment. As can be observed from the figure, the classification performance was improved for all the three data sets with the sparsity level up to 4 for the Indian Pines data set and University of Houston data set, and 5 for the University of Pavia data set, and it will deteriorate with a larger sparsity level. This may be due to the incorrect dictionary atoms from other classes when the sparsity level is large.

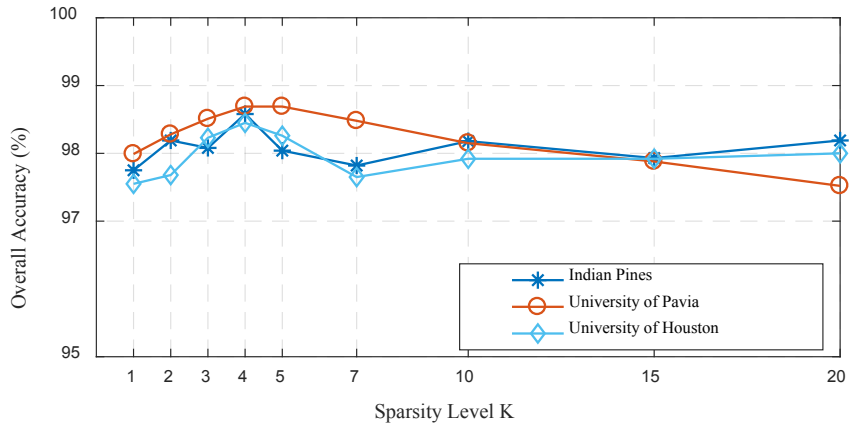


Fig. 6.9. Effect of the sparsity level on the classification accuracies of MCSSR for three data sets.

6.3.5 Computational Complexity

This section analyzes the computational complexity of the proposed conservation smoothing algorithm and the applied adaptive sparse representation algorithm. For a HSI cube $A \times B \times L$, the up-bound computational complexity of the conservative smoothing scheme is $O(MABL)$, where M is the number of local regions for the conservative algorithm. Algorithm 6.1 converges within a small number of iterations (i.e. 20). According to [98, 160], the time complexity of the shape adaptive algorithm is

$O(pkAB\log(AB))$, where p is the dimension of the candidate length vector and k represents the number of directions in the shape adaptive algorithm. For the SOMP used to optimize the JSRC algorithm, the most time-consuming step is the basic scalar multiplication. For each adaptive joint matrix, the computational complexity is $\sum_i nL(T-i)+2i^2L+i^3+iLn$, where $i=1,2,...,K$ is the sparsity level, n is the number of vectors in the joint matrix, and T denotes the number of atoms in the dictionary. The maximum time complexity for the SOMP for HSI is $O(AB(KLnT+2K^3L+K^2Ln))$, and the sparsity level is usually set small (e.g. 3 in this chapter) in the algorithm. The maximum complexity for SOMP of M HSIs is $O(MAB(KLnT+2K^3L+K^2Ln))$. Overall, the proposed MCSSR would take more computational time than JSRC, MASR and MFASR due to the computation of conservative smoothing, but the main computational cost is the inner product for the dictionary learning procedure. The computational time can be significantly reduced by optimizing the algorithm with other programming language and adopting graphics processing units (GPU).

Finally, Table 6.5 shows the run time averaged over ten repeated experiments of the adopted classifiers and the proposed MCSSR for the classification of the three data sets. As can be observed, the proposed MCSSR takes more time, but its speed can be significantly accelerated by optimizing the codes with other programming languages and GPU.

6.4 Summary

In this chapter, a conservative smoothing algorithm is proposed for the HSI classification, which utilizes the spatial consistency in the neighbouring pixels in the original image cube. The proposed method utilizes the spectral similarities between pixels to assign the weights for different neighbouring pixels in a defined local region to avoid over-smoothing automatically. In this scenario, the spatial contextual information can be revealed. Because the spatial characteristics are different when the local region is set differently, the original image is processed by the proposed smoothing algorithm

with multiscales. Then an adaptive sparse representation model is constructed, in which a shape adaptive window is adopted to fully utilize the spectral-spatial information of differently filtered HSIs. The proposed method is demonstrated to be superior to several benchmark classifiers in both quantitative and qualitative assessments on the three widely used data sets. In addition, the experiments also demonstrate that the proposed method can lead to a robust and reliable result with a limited number of training samples. All in all, the dictionary size for the multi-task learning-based method is proportional to the size of the data set, which may lead to a high computational cost for a large data set. An effective construction of the dictionary from the training samples would be useful in the future research.

Chapter 7 HSI Classification Using CNNs and Multiple Feature Learning

7.1 Introduction

Multiple feature learning aims to learn several types of features simultaneously in order to extract more representative features for image processing purposes. Multiple feature learning has been successfully applied to many computer vision-based fields, such as face detection [161], pedestrian detection [162] and multimedia search [163]. However, there is a lack of comprehensive studies on multiple feature learning for HSI classification.

Chapter 5 and Chapter 6 generated multiple HSI features using different sets of parameters, and then the multiple features were further learnt and classified by sparsity-based classifiers. However, it is difficult to find the optimal parameters for classification. Very recently, deep learning-based approaches, which can automatically extract and learn discriminative features, have been extended to HSI classification. As discussed in Chapter 2, CNN is one of the most promising branches in deep learning. Most CNN based methods consider the HSI classification as a task of extracting robust high-level deep features.

In order to extract robust and effective features for HSI classification, it is reasonable to explore CNN models which can simultaneously extract the spatial and spectral information from multiple HSI features. In this chapter, an enhanced framework that combines a CNN and a multiple feature learning method is proposed. Considering that spatial information extracted by the proposed CNN is more about the neighbouring information, other forms of geometrical information should be also investigated to boost the performance of HSI classification. Therefore, firstly, initial geometrical feature maps are extracted by four widely used attribute filters. The initial feature maps can reveal various spatial characteristics and local spatial correlations in the original image. Subsequently, the initial feature maps along with the original image are fed into a CNN which has different inputs corresponding to the different initial features. The

representative features are extracted by several groups of subsequent layers and are used as the input to a concatenating layer to form a joint feature map which represents both spectral and contextual properties of an HSI. The final labels of HSI pixels are determined by the subsequent layers with the joint feature map as input. The proposed framework does not need any post-processing step. The designed CNN consists of four key components: proper convolutional layers, a pooling layer, a concatenating layer and a rectified linear unit (ReLU) function. Since HSI suffers with a limited number of training samples, a deeper and wider network without enough training samples may result in overfitting; hence the proposed network is a relatively shallow network but is an effective one. The pooling layer can provide spatial invariance, the concatenating layer is designed to exploit the rich information, and the ReLU function will accelerate the convergence. The main contributions of this chapter include: 1) the construction of a novel CNN architecture which benefits from the multiple inputs corresponding to various image features; 2) the concurrent exploitation of both spectral and spatial contextual information; and 3) the proposed network that is robust and efficient even if a small number of training samples are available. The proposed method is referred to as "MFL_CNN" in this thesis.

The remainder of this chapter is organized as follows: Section 7.2 introduces the overall framework of the designed CNN. The proposed framework is also presented in detail in this section. The experimental results and discussions are provided in Section 7.3. The impact of several factors to the experimental results is also investigated in Section 7.3. Finally, the conclusions are drawn in Section 7.4 with some remarks. The work of this chapter has been present in Remote Sensing [126].

7.2 Proposed Framework

Fig. 7.1 illustrates the structure of the proposed framework. The first step of this framework is the extraction of multiple HSI features followed by several CNN blocks. Given T sets of features, each individual CNN block will learn the corresponding representative feature map, and all the feature maps will be joined by a concatenating layer. The weight and bias for each block are fine-tuned in this network through back propagation. The output of the network for each pixel is a vector of class membership

probability with C units, corresponding to C classes defined in the hyperspectral data set. The main principles of the proposed framework are explained in detail in the following sections.

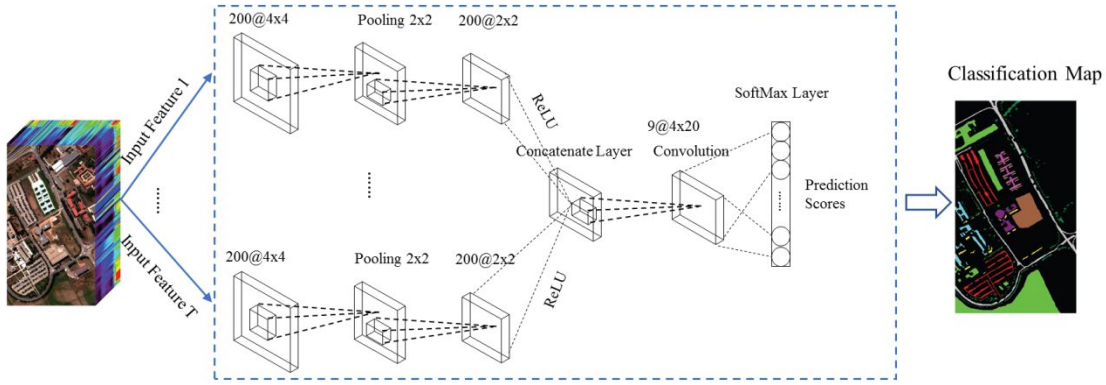


Fig. 7.1. The structure of the proposed framework.

7.2.1 Extraction of Attribute Profiles

The characterization of spatial contextual information computed by MPs can represent the variability of the structures for images [25]. However, features extracted by a specific MP cannot be modelled as other geometrical features. In order to model various geometrical characteristics simultaneously for the feature extraction in HSI classification, the application of APs is firstly introduced in the work of [35]. APs showed interesting properties in HSI processing, which can be used to generate an EAP.

APs are a generalized form of MPs, which can be obtained from an image by applying a criterion T. The construction of APs relies on the morphological AFs, and it can be obtained by applying a sequence of AFs to a scalar image [35]. AFs are defined as the connected operators which process the image by merging its connected components instead of pixels. After the operators are applied to the regions, the attribute results are compared to a pre-defined reference value. The region is determined to be preserved or removed from the image depending on whether the criterion is met or not (i.e. the attribute results are preserved if the value is larger than the pre-defined reference value). The values in the removed region will be set as the closest grayscale value of the

adjacent region. If the merged region is a lower (greater) gray level, then the thinning (thickening) operator is applied.

Subsequently, an AP can be directly constructed by using a sequence of thinning and thickening AFs which are applied to the image with a set of given criteria. By using n morphological thickening (φ^T) and n thinning (ϕ^T) operators, an AP from an image f can be constructed as:

$$AP(\mathbf{f}) = \{\varphi_n^T(\mathbf{f}), \varphi_{n-1}^T(\mathbf{f}), ..., \varphi_1^T(\mathbf{f}), \mathbf{f}, \phi_1^T(\mathbf{f}), ..., \phi_{n-1}^T(\mathbf{f}), \phi_n^T(\mathbf{f})\}. \tag{7.1}$$

Generally, there are some common criteria associated with the operators, such as area, volume, diagonal box, and standard deviation. According to the operators (thickening or thinning) used in the image processing, the image can be transformed to an extensive or anti-extensive one. In this chapter, since our goal is to measure the effectiveness of multiple feature learning by the proposed CNN, but not to achieve absolute performance maximization, only APs based on four different criterions (i.e. area, standard deviation, the moment of inertia, and length of the diagonal) are extracted as the different feature maps for classification tasks. And in this chapter, the different AP features are named by the corresponding criterions. One can find the details of various APs from [25].

7.2.2 Convolutional Neural Networks

CNNs aim to extract the representative features for different forms of data via multiple non-linear transformation architectures[122]. The features learned by a CNN are usually more reliable and effective than rules-based features. In this chapter, we consider HSI classification with the so-called *directed acyclic graphs* (DAG) where the layers are not limited to chaining one after another. For HSI classification, a neural network can realize the function of mapping the input HSI pixels to the output pixel labels. The function is composed of a sequence of simple blocks that are called layers. The basic layers in a CNN are as follows:

Mathematically, an individual neuron is computed by taking a vector of inputs x and applying an operator to it with a weight filter f and bias $\mathbf{b}$:

$$\mathbf{a} = \sigma(f\mathbf{x} + \mathbf{b}). \tag{7.2}$$

where $\sigma(\bullet)$ is a nonlinear function named as an activation function. For a convolutional layer, every neuron is related to a spatial location (i, j) with respect to the input image. The output $\mathbf{a}_{i,j}$ associated with the input can be defined as follows:

$$a_{i,j} = \sigma((F \otimes \mathbf{X})_{i,j} + \mathbf{b}). \tag{7.3}$$

where F is the kernel function with the learned weights, $\mathbf{X}$ is the input or the layer, and $\otimes$ denotes the convolution operator. Usually at least one layer of the activation function is implemented in a network. The most frequently used activation functions are the sigmoid function and the ReLU function. The ReLU function has been considered to be more efficient than the sigmoid function in the convergence of the training procedure [122]. The ReLU function is defined as follows:

$$\sigma(\mathbf{x}) = \max(0, \mathbf{x}). \tag{7.4}$$

Another important type of layers is pooling which is implemented as a down-sampling function. The most common types of pooling are the max-pooling and mean-pooling. The pooling function partitions the input feature map into a set of rectangles and outputs the max/mean value for each sub-region. Hence, the computational complexity can be reduced.

Typically, a softmax function is performed on the top layer so that a probability distribution as an output can be obtained with each unit representing a class membership probability. Based on the above principle, in this chapter, different features of the raw image are fed into each corresponding CNN block, and the network is fine-tuned through the back propagation.

7.2.3 Architecture of Convolutional Neural Network

A HSI contains several hundreds of spectral bands, and the input of a HSI classifier is usually the whole image. This is different from common classification problems. It has been acknowledged that spatial contextual information extraction is essential for HSI

classification. Based on such knowledge, we choose a three dimensional structure of the HSI pixel as input to the built CNN model. Given a HSI cube $\mathbf{X} \in \mathbb{R}^{M \times N \times L}$, $M \times N$ is the image size and L denotes the number of spectral channels. For a test pixel $\mathbf{x}_i$ (where i is the index of the test pixel), a $K \times K \times B$ format structure of this pixel will be adopted as the input with $K \times K$ being a fixed neighborhood size and B representing the dimension of the input features. For example, for the original image cube, B is equal to the number of the spectral channels L. In this chapter, after T attribute profile features (i.e. area, standard deviation, length of diagonal, and moment of inertia) are extracted, each attribute can be expressed as $\mathbf{A}_t \in \mathbb{R}^{M \times N \times B_t}$, $t = 1, 2, ...T$. A_t denotes the $t-th$ attribute of X, B_t denotes the number of spectral channels of $\mathbf{A}_t$. For each pixel in $\mathbf{A}_t$, a $K \times K \times B_t$ neighborhood region patch will be chosen as the input to the corresponding model.

Each convolutional layer has a four-dimensional convolution of $W \times W \times B \times F$, where $W \times W$ is the kernel size of the convolutional layer, B is the dimension of input variable and F denotes the number of kernels in each convolutional layer. For example, for a $2 \times 2 \times 200 \times 50$ convolutional layer with an input size of $5 \times 5 \times 200$, the output in the DAG will be a format of $4 \times 4 \times 50$ which will be the input of the next layer.

The three-dimensional format of the input in the proposed network makes the dimensionality around several hundreds ($K \times K \times B$), which may lead to an overfitting problem during the training procedure. In order to handle this situation, ReLU is applied to the proposed network. The adopted ReLU in this chapter is a simple nonlinear function that produces 0 or 1 corresponding to the positive or negative input of a neuron. It has been confirmed that ReLU can boost the performance of networks in many cases[164].

To perform the classification with the learned representative features, a softmax operator is applied to the top layer of the proposed network. Softmax is one of the probabilistic-based classification models which measure the correlation between an output value and a reference value by a probability score. It should be noted that in the

CNN construction, softmax can be applied throughout the spectral channels for all spatial locations in a convolutional manner. For the given input of three dimension ($K \times K \times B$), the probability that the input belongs to class c is computed as follows:

$$p(y=c)=\frac{e^{\mathbf{x}_{mnk}}}{\sum_{b=1}^{B} e^{\mathbf{x}^{mnb}}}. \tag{7.5}$$

In order to obtain the essential probability distribution using the softmax operator, the number of kernels of the last layer should be set as the same as the number of classes defined in the HSI data set. The whole training procedure of the network can be treated as the optimization of parameters, which can minimize a loss function between the network outputs and ground truth values for the training data set. Let $y_i = 1,...,c,...,C$ denote the target ground truth value corresponding to the text pixel $\mathbf{x}_i$, and $p(y_i)$ be the output class membership distribution with i as the index of the test pixel. The multi-class hinge loss used in this chapter is given by

$$L=\sum_{i=1}^{N}\sum_{c=1}^{C}\max(0,1-p(y_i=c)). \tag{7.6}$$

Finally, the predication label is decided by taking the argmin value of the loss function:

$$\hat{y}_i = \arg\min_{c} L. \tag{7.7}$$

7.3 Experimental Results and Discussion

Section 7.3.1 below introduces the data sets and shows the class information. Section 7.3.2 layouts the specific network architectures applied in this chapter and other relevant information regarding the experimental evaluation. Section 7.3.3 provides the experimental results for all the classifiers. Section 7.3.4 highlights some additional experiments influential to the classification results. In this thesis, the original features, as well as four attribute features extracted based on four attribute filters (i.e. area, moment of inertia, length of diagonal and standard deviation) are used as inputs to the

proposed network. The parameters for each AP criterion are set as default as the ones in [35].

In order to validate the effectiveness of the proposed mechanism, the proposed work is compared with the designed CNN with original images (referred to as O-CNN), and a CNN using all features (including the original images) stacked as input (referred to as E-CNN). As shown in Fig. 7.2, for fair comparison, these CNNs have architectures similar to the proposed network. The attribute features extracted in this chapter have the parameters set as the ones in [25]. All the programs are executed in Matlab 2015b. The test is conducted on a computer with Intel (R) Core (TM) i7-4790 CPU 3.60 GHz and 16 GB Installed Memory. All the convolutional network models are implemented based on the publicly available matconvnet [165] with some modifications, and the optimization algorithms used in this chapter are implemented by the Statistics and Machine Learning Toolbox in Matlab.

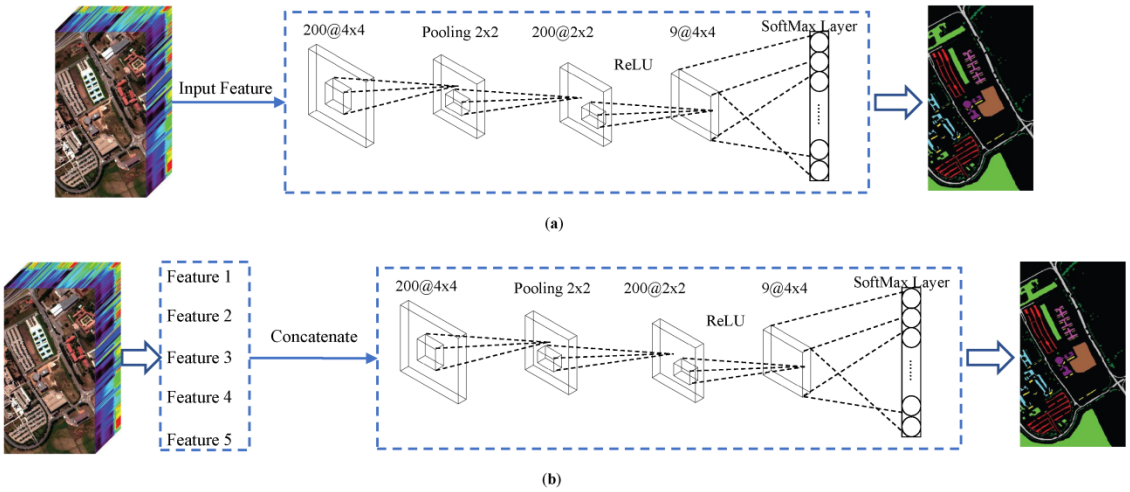


Fig. 7.2. The architecture of comparison classifiers: (a) O-CNN; (b) E-CNN.

7.3.1 Data Description

To verify the effectiveness of the proposed framework, three benchmark data sets are used in this chapter. They are AVIRIS Indian Pines data set, ROSIS University of Pavia data set and Salinas AVIRIS data set.

For each of the three data sets, the samples are split into two subsets, i.e. a training set and a test set. The details of the number of the subsets are listed in Tables 7.1-7.3. For training the architecture of each CNN block, 90% of the training pixels are used to learn the filter parameters for each CNN block and the remaining 10% are used as the validation set. The training set is used to adjust the weights on the neural network. The validation set is used to provide an unbiased evaluation of a model fit on the training data set, which means that this data set is predominately used to describe the evaluation of models when tuning hyper parameters. The test is used only to assess the performance of a fully-trained CNN model.

Table 7.1. Class Information for Indian Pines Data Set

No.	Class Name	Training	Test
1	Alfalfa	30	16
2	Corn-no till	250	1178
3	Corn-min till	250	580
4	Corn	150	87
5	Grass/trees	250	233
6	Grass/pasture	250	480
7	Grass/pasture-mowed	20	8
8	Hay-windrowed	250	228
9	Oats	15	5
10	Soybeans-no till	250	722
11	Soybeans-min till	250	2205
12	Soybeans-clean till	250	343
13	Wheat	150	55
14	Woods	250	1015
15	Buildings-grass-trees	50	336
16	Stone-steel towers	50	43
	Total	2715	7534

Table 7.2. Class Information for University of Pavia Data Set

No.	Class Name	Training	Test
1	Asphalt	250	6381
2	Meadows	250	18399
3	Gravel	250	1849
4	Trees	250	2814
5	Meta sheets	250	1095
6	Bare soil	250	4779
7	Bitumen	250	1080
8	Bricks	250	3432
9	Shadows	250	697
	Total	2250	40526

Table 7.3. Class Information for Salinas Data Set

No.	Class Name	Training	Test
1	Weeds_1	300	1709
2	Weeds_2	300	3426
3	Fallow	300	1676
4	Fallow plow	300	1094
5	Fallow smooth	300	2378
6	Stubble	300	3659
7	Celery	300	3279
8	Grapes	300	10971
9	Soil	300	5903
10	Corn	300	2978
11	Lettuce 4 week	300	768
12	Lettuce 5 week	300	1627
13	Lettuce 6 week	300	616
14	Lettuce 7 week	300	770
15	Vineyard untrained	300	6968
16	Vineyard trellis	300	1507
	Total	4800	49329

7.3.2 Network design and experimental setup

CNN blocks for different features were designed to have the same architecture. There are three convolutional layers, pooling layers, ReLU layers and concatenating layers. The details of the network structure are listed in Tables 7.4-7.6. The input images are initially normalized into [-1 1]. The number of kernels in each convolutional layer is set

as 200 empirically. The input neighbourhood of each feature is set as 5×5, 7×7 and 9×9 for the Indian Pines data set, the University of Pavia data set and the Salinas data set, respectively. The learning rate for CNN models is set as 0.01; the number of epochs is set as 100 for the Indian Pines and the University of Pavia data sets, and 150 for the Salinas data set. The batch size is set as 10. To quantitatively validate the results of the proposed framework, OA, AA and the kappa coefficient (k) are adopted as the performance metrics. Each result is shown as an average of ten times repeated experiments with the randomly chosen training samples.

Table 7.4. Network Structure for Indian Pines Data Set.

Input Features	Layer No.	Convolution	ReLU	Pooling
Original image	1	2×2×200×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Area)	1	2×2×125×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Length of diagonal)	1	2×2×175×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Moment of inertia)	1	2×2×75×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Standard deviation)	1	2×2×75×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
	Concatenating	Dim=2 (Horizontal)		
	Convolution	4×20 ×200×16		

Table 7.5. Network Structure for University of Pavia Data Set.

Input Features	Layer No.	Convolution	ReLU	Pooling
Original image	1	4×4×103×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Area)	1	4×4×20×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Length of diagonal)	1	4×4×103×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Moment of inertia)	1	4×4×12×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Standard deviation)	1	4×4×12×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
	Concatenating	Dim=2 (Horizontal)		
	Convolution	4×20 ×200×9		

Table 7.6. Network Structure for Salinas Data Set.

Input Features	Layer No.	Convolution	ReLU	Pooling
Original image	1	6×6×224×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Area)	1	6×6×15×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Length of diagonal)	1	6×6×21×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Moment of inertia)	1	6×6×9×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
AP (Standard deviation)	1	6×6×9×200	No	2×2
	2	(Transpose)2×2×200×200	Yes	No
	Concatenating	Dim=2 (Horizontal)		
	Convolution	4×20 ×200×16		

7.3.3 Experimental Results

Table 7.7 shows the classification results obtained by different classifiers for the Indian Pines data set, and the resultant maps are provided in Fig. 7.3. One can observe that all the CNN-based models achieve a good performance, and the proposed method provides the improved results on this data set. For O-CNN, the original image is set as the input for the network. In order to verify the effectiveness of the proposed mechanism, the spatial contextual features are extracted and stacked together to be fed into the network for E-CNN. E-CNN has achieved more accurate results than O-CNN, but failed to outperform the proposed method. The best performance achieved by the proposed framework is probably due to the joint exploitation of spatial-spectral information. One can conclude that the proposed method produces less "salt-and-pepper" noise on the classification maps. In comparison with O-CNN, OA, AA and kappa of the proposed method are improved by 8.43%, 3.69% and 9.5%. The same conclusion can be made when the proposed method is compared with E-CNN, especially the improvement is quite significant for the sets of similar class labels as can be observed from Table 7.7. For example, the accuracies obtained by the proposed method for the classes Soybeans-no till, Soybeans-min till and Soybeans-clean till (class no. 10, 11, and 12) are 5.76%, 7.82% and 5.74% higher than those obtained by the E-CNN. The same conclusion can be obtained when the individual class accuracies for the similar sets of Grass-tress, Grass-pasture and Grass-pasture mowed (class no. 5, 6, and 7) are inspected. The results show that the proposed algorithm has a very competitive ability in classifying the similar and mixed pixels. In addition, the proposed method has demonstrated the best performance in terms of preserving the discontinuities which can be observed from the classification maps. Moreover, CNN methods do not need predefined parameters whereas pixel-level extraction methods require them.

Table 7.7. Classification Results (%) of Indian Pines Data Set.

Class No.	O-CNN	E-CNN	MFL_CNN
1	95.65	97.83	97.83
2	87.96	95.52	94.82
3	93.86	85.66	97.23
4	98.73	100.00	99.58
5	98.14	95.24	99.59
6	97.53	95.48	99.59
7	100.00	92.86	100.00
8	98.12	100.00	100.00
9	100.00	100.00	100.00
10	90.02	88.17	93.93
11	74.95	89.41	97.23
12	91.40	93.25	98.99
13	100.00	97.07	100.00
14	94.62	96.84	99.76
15	95.34	98.70	97.93
16	100.00	94.62	98.92
OA	89.14	93.04	97.57
AA	94.77	95.04	98.46
k	87.73	92.11	97.23

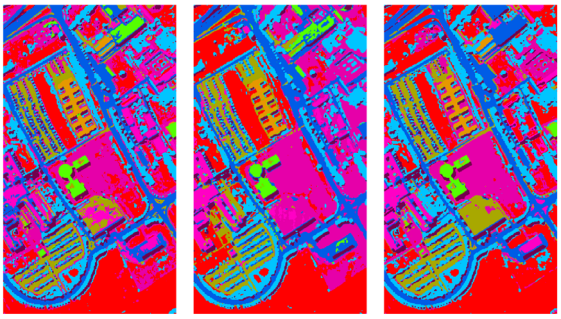
(a) (b) (c)

Fig. 7.3. Classification maps of Indian Pines data set: (a) O-CNN; (b) E-CNN; (c) MFL_CNN.

The class-specific classification accuracies for the University of Pavia image and the representative classification maps are provided in Table 7.8 and Fig. 7.4, respectively. From the results, one can see that the proposed method outperforms the other algorithms in terms of OA, AA and kappa. The proposed method significantly improves the results with a very high accuracy when tested with the University of Pavia data set. From the illustrative results in classification maps, O-CNN and E-CNN show more noisy scattered points in the images. The proposed method can remove them and lead to smoother classification results without blurring the boundaries.

Table 7.8. Classification Results (%) of University of Pavia Data Set.

Class No.	O-CNN	E-CNN	MFL_CNN
1	97.50	99.68	99.25
2	94.38	99.93	99.74
3	96.62	94.46	99.76
4	97.58	97.35	99.64
5	100.00	100.00	100.00
6	93.52	97.82	99.96
7	93.16	98.57	98.80
8	93.10	98.38	99.48
9	99.68	99.79	99.89
OA	95.25	98.99	99.64
AA	96.17	98.44	99.61
k	93.75	98.67	99.53



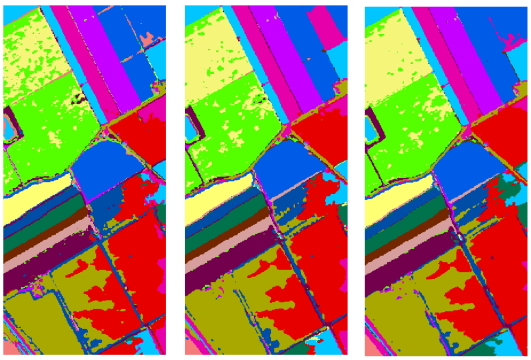
(a) (b) (c)

Fig. 7.4. Classification maps of University of Pavia data set: (a) O-CNN; (b) E-CNN; (c) MFL_CNN.

Table 7.9 shows the classification results for the Salinas data set with different classifiers, and the classification accuracies are illustrated in Fig. 7.5. The results are similar to the previous two data sets. Under the condition of the same training samples, the proposed method outperforms the other approaches in terms of OA, AA and kappa. Although E-CNN improved the classification results of O-CNN by stacking different features, the improvement is limited when compared to the proposed framework. The better performance of the proposed network proves the capacity and effectiveness of the built network for multiple feature learning.

Table 7.9. Classification Results (%) of Salinas Data Set.

Class No.	O-CNN	E-CNN	MFL_CNN
1	100.00	100.00	100.00
2	99.84	99.92	99.92
3	99.60	99.70	99.65
4	99.57	99.93	99.78
5	99.93	99.78	99.07
6	99.95	100.00	99.97
7	99.30	99.92	99.75
8	95.52	95.73	94.28
9	99.45	100.00	99.97
10	97.32	99.73	99.63
11	99.53	100.00	99.91
12	100.00	100.00	100.00
13	100.00	100.00	100.00
14	100.00	100.00	99.91
15	66.29	81.65	97.40
16	95.35	100.00	100.00
OA	94.06	96.60	98.34
AA	96.98	98.52	99.33
k	93.37	96.20	98.15



(a) (b) (c)

Fig. 7.5. Classification maps of Salinas data set: (a) O-CNN; (b) E-CNN; (c) MFL_CNN.

7.3.4 Parameter Analysis

The number of training epochs is an important parameter for the CNN-based methods. Fig. 7.6 shows that the training error varies with the number of training epochs on all three data sets. In the training process for a network, the back propagation is implemented by minimizing the training error "objective" which is computed by

$objective = -\sum_{i=1}^{N_t} \log(p_{ic})$. Here, the trend of the "error" item is computed by $error = \sum_{i=1}^{N_t} p_{ic}(\arg\max p_i \sim= c)$ where N_t denotes the number of training samples, p_{ic} denotes the $c-th$ prediction probability of the training pixel $\mathbf{x}$ which belongs to the $c-th$ class. It is helpful and useful for assessment. From Fig. 7.6, one can observe that it converges faster for the training process of the Indian Pines image and the University of Pavia image, slower for the Salinas image. ReLU is an important factor which is influential to the training procedure; ReLU can accelerate the convergence of the network and improve the training efficiency [164].

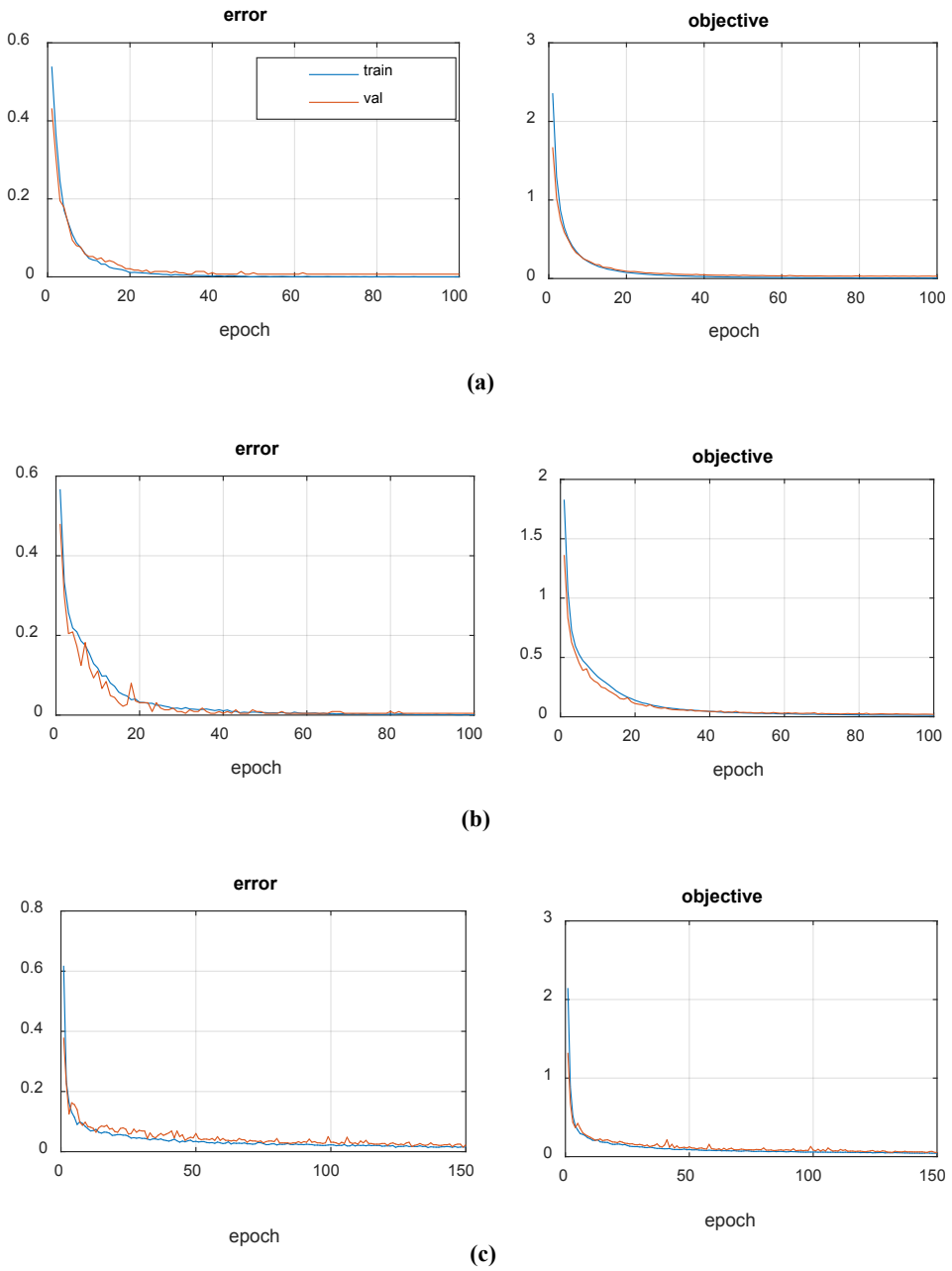
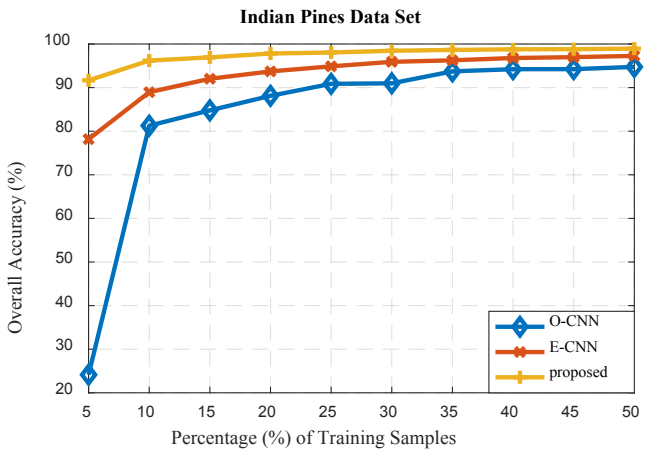


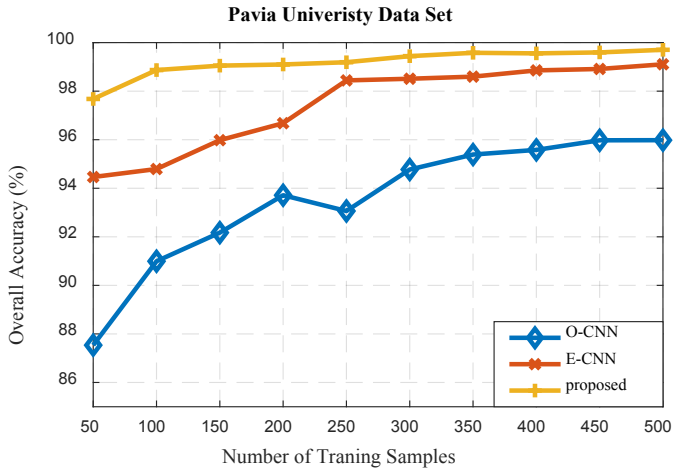
Fig. 7.6. Training error for the proposed framework of three data sets: (a) Indian Pines; (b) University of Pavia; (c) Salinas Scene.

One critical factor to the training a CNN is the number of training samples. It is widely known that a CNN may not extract effective features unless abundant training samples are available. However, it is not common for HSI to have a large number of training samples, hence it is very important to build a network that is robust and efficient for the classification task.

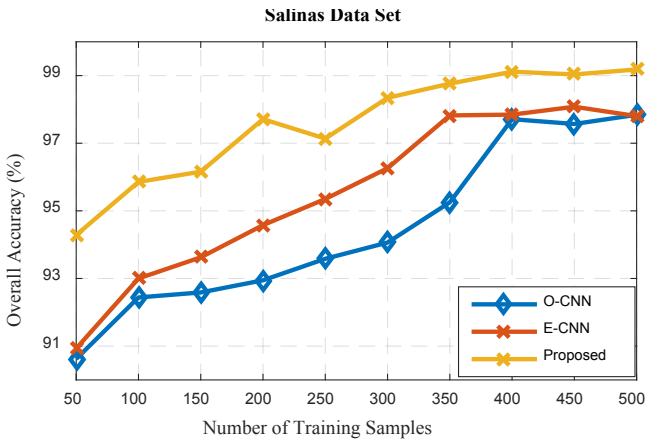
In this chapter, the impacts of the number of training samples on the accuracies of three data sets are also tested. For the Indian Pines scene, 5% to 50% of the samples are randomly selected as training pixels and the remaining pixels are used as the test set. For both the University of Pavia and the Salinas images, 50 to 500 pixels per class are chosen randomly as the training samples with the remaining as the test set. Fig. 7.7 illustrates the OA for various methods with different numbers of training pixels. From Fig. 7.7, one can see that all the methods perform better if the number of training samples increases for the Indian Pines data set, and the proposed method performs the best. Especially, the proposed method obtains an accuracy of higher than 95% with less than 10% training samples. The accuracies tend to become stabilized for these three methods if the number of training samples further increases. For the University of Pavia data set, the classification accuracies for these CNN-based methods show approximately 100% as the number of training samples further increases, especially for the proposed method which has the accuracy more than 96% with 50 samples per class. For the Salinas data set, the performances for all approaches fluctuate in a range, and the proposed method performs the best in most cases. It should be noted that for all the three data sets, the CNN-based classifiers are more sensitive to the number of training samples and the accuracy increases as the number of training samples increases. In addition, the CNN-based approaches can achieve a competitive performance with a large number of training samples, and the proposed method shows more robustness with a variety of the number of training samples.



(a)



(b)



(c)

Fig. 7.7. The effects of training samples on accuracies of three data sets: (a) Indian Pines; (b) University of Pavia; (c) Salinas.

The neighbourhood size $K \times K$ of the input image is another important factor related to the classification results. Fig. 7.8 illustrates the network architectures with inputs of different neighbourhood sizes. The only difference for the three data sets is the number of kernels in the last layer, which is 16 for the Indian Pines and the Salinas data sets, and 9 for the University of Pavia data set. It should be noted that, in order to obtain the probability scores corresponding to different classes, the number of kernels in the last layer should be the number of labeled classes for each data set. In Fig. 7.8, we take the University of Pavia data set as an example. As shown in Tables 7.10-7.12, the performances decrease with the neighbourhoods up to 7×7, 9×9 and 11×11 for three

data sets, respectively. The performance degradation may be caused by the "over-smoothing" effect across the boundaries as the neighborhood size increases. Hence, 5×5, 7×7 and 9×9 are the optimal neighborhood sizes for the three data sets in the proposed network.

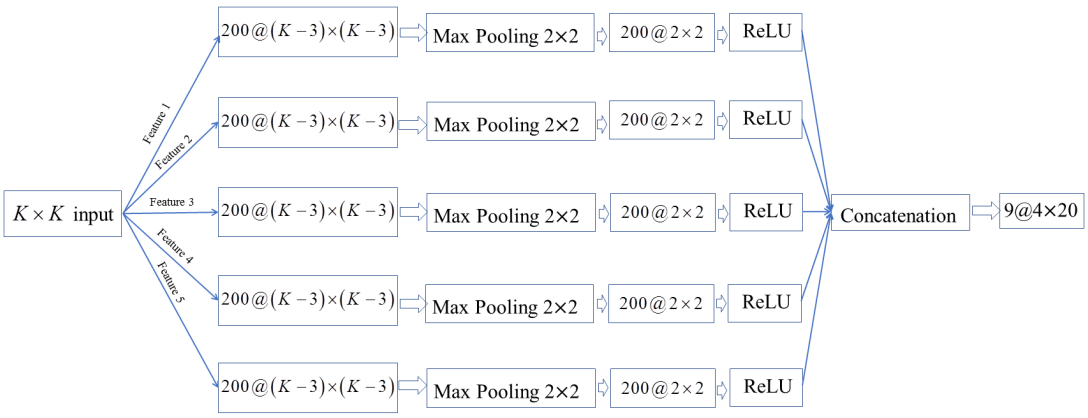


Fig. 7.8. The network architecture with different inputs of different neighbourhood sizes.

Table 7.10. Classification Results (%) of Indian Pines Data Set using Network with Inputs of Different Neighbourhood Sizes.

	5×5	7×7	9×9	11×11
OA	**97.57**	97.19	95.24	93.61
AA	**98.46**	98.05	96.12	94.28
k	**97.23**	96.80	94.58	92.71

Table 7.11. Classification Results (%) of University of Pavia Data Set using Network with Inputs of Different Neighbourhood Sizes.

	5×5	7×7	9×9	11×11
OA	99.19	**99.64**	99.60	99.49
AA	99.38	**99.74**	99.63	99.61
k	98.92	**99.53**	99.33	99.47

Table 7.12. Classification Results (%) of Salinas Data Set using Network with Inputs of Different Neighbourhood Sizes.

	5×5	7×7	9×9	11×11
OA	95.97	97.38	**98.34**	97.82
AA	98.42	98.90	**99.33**	99.16
k	95.53	97.09	**98.15**	97.58

To verify the effectiveness of the multiple feature learning, the experimental results for the designed CNN (Fig. 7.2(a)) with individual features (i.e. area, moment of inertia, length of diagonal and standard deviation) are also shown in Tables 7.13-7.15 for the validation. From these tables, one can see that the designed CNN with features of length of diagonal performs better than other networks. Compared with the results in Tables 7.7-7.9, it is obvious that E-CNN compromises the accuracy for the classification. This may be due to the data augmentation caused by the initial concatenation which is not proper for the spatial filter. The higher accuracy obtained by the proposed method benefits from the joint exploitation in the processing stage where the dimension has been cut off by the spatial filter. In addition, the concatenation of the various features at first step of E-CNN may lose the discriminative information during the training process. The various features possess different properties, learnt through the individual convolutional layers can help extract the better feature representations for the classification which leads to a superior performance. The proposed joint structure-based multi-feature learning can adaptively learning the heterogeneity of each feature, and eventually result in a better performance. It can be concluded that the comparison results with individual features reveal the effectiveness of the multiple feature learning technique of the proposed method.

Table 7.13. Classification Results (%) for Individual AP Features of Indian Pines Data Set.

Accuracy	AP (Area)	AP (Length of diagonal)	AP (Moment of inertia)	AP (Standard deviation)
OA	94.43	95.58	94.96	92.77
AA	96.85	96.60	96.83	95.66
k	93.67	94.18	94.26	91.78

Table 7.14. Classification Results (%) for Individual AP Features of University of Pavia Data Set.

Accuracy	AP (Area)	AP (Length of diagonal)	AP (Moment of inertia)	AP (Standard deviation)
OA	93.20	98.47	95.82	91.77
AA	95.93	98.52	97.35	94.28
k	91.14	98.31	94.49	89.26

Table 7.15. Classification Results (%) for Individual AP Features of Salinas Data Set.

Accuracy	AP (Area)	AP (Length of diagonal)	AP (Moment of inertia)	AP (Standard deviation)
OA	93.59	96.29	93.45	92.39
AA	96.76	97.43	96.73	96.16
k	92.85	95.88	92.68	91.50

The training and test time averaged over ten repeated experiments for the three data sets are given in Table 7.16. The training procedure for a CNN is time-consuming; however, another advantage of CNN algorithms is that they are fast for testing. In addition, the training time would take just a few seconds with GPU processing.

Table 7.16. Training/Test Time (minutes) Averaged over Ten Time Repeatedly Experiments on Three Data sets for Different Classifiers.

	O-CNN	E-CNN	MFL_CNN
	Training/Test	Training/Test	Training/Test
Indian Pines	8.7/0.74	9.7/0.8	17.1/1.5
University of Pavia	17.2/1.9	27.5/2.1	38.8/3.9
Salinas	42.8/4.5	45.6/5.2	66.1/10.2

7.4 Summary

In order to prove the potential of CNNs for HSI classification, we presented a framework consisting of a novel CNN model. The framework was designed to have several individual CNN blocks with comprehensive features as input. To enhance the learning efficiency as well as to leverage both the spatial contextual and spectral information of the HSI, the output feature maps of each block are then concatenated and fed into subsequent convolutional layers to derive the pixel label vectors. By using the proper architecture, the built network is a shallow but efficient one, and it can

concurrently exploit the interactions of different spectral and spatial contextual information by using the concatenating layer. In comparison with the CNN-based single feature learning method, the classification results are improved significantly with multiple features involved. Moreover, in contrast to the traditional rule-based classifiers, the CNN-based framework can extract the deep features automatically and in a more efficient way.

Moreover, the experiments suggest that a three-layer CNN is optimal for HSI classification, and the neighbourhood size between 2×2 to 6×6 can balance the efficiency and complexity of the network. The pooling layer with a size of 2×2 and 200 kernels in each layer can provide an enough capacity for the network. Since the training samples are very limited in HSI classification, the multiple input feature maps and ReLU in the proposed network can help alleviate the overfitting phenomenon and accelerate convergence. The tests with three benchmark data sets showed superior performances of the proposed framework. As CNNs are gaining attention due to the strong ability in extracting the relevant features for image classification, the proposed method is expected to provide various improvements for the better feature representation purpose.

Chapter 8 Discussions

In Chapters 4-7, we introduced several approaches for HSI classification. These were all evaluated under different scenarios in order to compare them with similar state-of-the-art techniques. In this chapter, we validate the performance of all proposed methods (i.e. JSDPR from Chapter 4, AJSM and MLSR from Chapter 5, MCSSR from Chapter 6 and MFL_CNN from Chapter 7) under the same conditions. Three data sets (i.e. AVIRIS Indian Pines, ROSIS University of Pavia, and AVISRIS Salinas data sets) are used. As only limited training samples are available, small subsets of the given datasets were used for training. For the Indian Pines data set, 10% of samples are randomly selected as training samples, and the remaining used as test samples. For the University of Pavia data set, 1% of samples are randomly chosen as training samples and the remaining 99% samples used as the test set. For the Salinas image, 0.05% of samples are used as training samples and the remainder used to test the classifiers. Tables 8.1-8.3 show class information as well as the numbers of training and test samples used in this chapter. Tables 8.4-8.6 illustrate the classification results in terms of OA, AA, and kappa coefficient (k).

Table 8.1. Class Information for the Indian Pines Data Set.

Class	Class Name	Train	Test
1	Alfalfa	5	41
2	Corn-no till	129	1299
3	Corn-min till	83	747
4	Corn	24	213
5	Grass/trees	48	435
6	Grass/pasture	73	657
7	Grass/pasture-mowed	5	23
8	Hay-windrowed	48	430
9	Oats	4	16
10	Soybeans-no till	97	875
11	Soybeans-min till	196	2259
12	Soybeans-clean till	59	534
13	Wheat	21	184
14	Woods	114	1151
15	Buildings-grass-trees	39	347
16	Stone-steel towers	12	81
	Total	957	9292

Table 8.2. Class Information for the University of Pavia Data Set.

Class No.	Class Name	Training	Test
1	Asphalt	66	6565
2	Meadows	186	18463
3	Gravel	21	2078
4	Trees	31	3033
5	Meta sheets	13	1332
6	Bare soil	50	4979
7	Bitumen	13	1317
8	Bricks	37	3645
9	Shadows	9	938
	Total	426	42350

Table 8.3. Class Information for the Salinas Data Set.

Class No.	Class Name	Training	Test
1	Weeds_1	30	1979
2	Weeds_2	56	3670
3	Fallow	30	1946
4	Fallow plow	21	1373
5	Fallow smooth	40	2638
6	Stubble	60	3899
7	Celery	54	3525
8	Grapes	169	11102
9	Soil	93	6110
10	Corn	49	3229
11	Lettuce 4 week	16	1052
12	Lettuce 5 week	29	1898
13	Lettuce 6 week	14	902
14	Lettuce 7 week	16	1054
15	Vineyard untrained	110	7158
16	Vineyard trellis	27	1780
	Total	814	53315

Table 8.4. Classification Results (%) of Indian Pines Data Sets Using Different Classifiers.

	JSDPR	AJSM	MLSR	MCSSR	MFL_CNN
OA	97.18	94.92	97.40	**98.35**	97.27
AA	96.88	93.98	95.93	**98.24**	97.38
k	96.79	94.37	96.85	**98.59**	97.07

Table 8.5. Classification Results (%) of University of Pavia Data Sets Using Different Classifiers.

	JSDPR	AJSM	MLSR	MCSSR	MFL_CNN
OA	88.52	87.24	95.59	**98.51**	97.21
AA	80.24	79.56	95.28	**96.95**	96.56
k	81.96	79.21	95.25	**98.27**	96.47

Table 8.6. Classification Results (%) of Salinas Data Sets Using Different Classifiers.

	JSDPR	AJSM	MLSR	MCSSR	MFL_CNN
OA	93.98	92.58	97.25	**98.02**	95.85
AA	95.65	96.33	98.77	**97.98**	95.24
k	92.47	92.00	96.94	**98.21**	95.06

From Tables 8.4-8.6, it is clear that the classification accuracies are consistent for all three data sets. The MCSSR method introduced in Chapter 6 achieved the best performance; MLSR and MFL_CNN obtained similar results, while JSDPR and AJSM produced relatively lower accuracies when using the same limited training set.

The better performance of JSDPR over AJSM for all three data sets may be due to JSDPR considering spatial information in both the probabilistic classification and refining stages, while AJSM loses some spatial information due to its top-N strategy. The better performance obtained by MLSR, MCSSR, and MFL-CNN is attributable to their common mechanism of extracting spatial information from different perspectives: MLSR discards outliers in a specific area and extracts information from different levels, MFL_CNN extracts information from various features, and MCSSR takes advantage of a multiscale conservative smoothing scheme to extract spatial and contextual information.

MCSSR achieved higher accuracies than MLSR because it exploits multiscale information without discarding any information and discovers correlations among different joint matrices by applying an adaptive norm. MLSR preserves the most useful information and reduces redundant information based on an adaptive neighbour selection strategy; however, it may also discard some relevant information, especially for higher spatial resolution data sets. In addition, MLSR does not exploit correlations

among joint matrices from different levels. The MFL_CNN is a deep learning-based method which requires adequate training data in order to produce a reliable and robust model. Therefore, MFL_CNN obtained a relatively lower accuracy given the limitation on training samples employed in this chapter.

Although MLSR and MFL_CNN produced similar results overall, MLSR performed better on the Indian Pines and Salinas data sets due to its superiority for images with considerable homogeneity. Moreover, the weight matrix for MLSR is constructed from the ratio of the between-class and within-class distances while considering a priori label information. This alleviates the negative impacts of classifying the mixed pixels and similar pixels that are present in the Indian Pines and Salinas data sets. In contrast, the Pavia image has a high spatial resolution and relatively small homogeneity, so MLSR may not perform as well.

Overall, the efficient integration of different techniques may improve classification results, and the extraction of features from different perspectives (e.g. multilevel, multiscale, and multiple numbers as used in this thesis) can significantly improve the classification of HSIs, especially under the condition of limited training samples.

Chapter 9 Conclusions and Future Research

The main aim of this work was to investigate effective spectral-spatial classification techniques for HSIs. The primary objective was to overcome the difficulties of classification, such as the insufficient extraction of spatial information and limited number of training samples. A number of recent advances for HSI classification have been reviewed in Chapter 2: mathematical morphological approaches, probabilistic graphical models, segmentation methods, sparsity representation-based techniques and deep learning-based methodology. A brief review of the data sets used in this work is presented in Chapter 3. The detailed studies of this thesis are reported in Chapters 4-8. This chapter summarizes the research conducted in this thesis and its main findings. Some remarks for future research are also given in this chapter.

9.1 Summary of the Contributions and Limitations

The methods introduced in Chapters 4-7 demonstrate that the exploitation of spatial information from multiple perspectives can boost the classification accuracy of single perspective-based methods. Chapter 4 integrates different methods, Chapter 5 constructs multi-level sparsity matrix for the test pixel, Chapter 6 applies a multiscale conservative smoothing scheme on the HSI and Chapter 7 extracts multiple features prior to the classification. As can be observed from the experimental results, the proposed approaches can obtain high accuracies given limited training samples and overcome the problems of conventional classifiers.

9.1.1 The integration of JSM and DPR

In this thesis, classification performance was addressed first by integrating two promising techniques. A JSM and a DPR algorithm were integrated in Chapter 4 for reliable and high precision HSI classification. This framework takes into account the spectral and spatial information at every step of classification. The proposed method works well for homogenous image areas without blurring the near-boundary areas.

In the study, JSM was firstly used to classify pixels in a probabilistic sense using their neighbourhood information. The sparse coefficients were used to compute the reconstruction error, leading to a probabilistic distribution for each pixel. Class-specific probabilities allow the presentation of the potential for each pixel to belong to each class. Then a DPR was applied to refine those possibilities and derive the final labels. Boundaries in the image were detected by a Sobel filter, then the homogenous areas smoothed without crossing those boundaries in order to help preserve the discontinuities in the original image. The integration of these two methods was an improvement over the results of applying them separately. Experiments conducted on two real-world data sets showed that among all the approaches compared, this method achieved the best performance, and performed very well under the condition of limited training samples. This study presents the advancement of integrating different classifiers and also provides knowledge for the assessment of methods in different areas.

The proposed method is not without limits. For example, it cannot refine the results for the edge areas detected by the Sobel filter. Once boundaries are detected, the results for the pixels in those areas are fixed by JSM. Additionally, homogeneous areas tend to be over-smoothed due to the large neighbourhood selected for JSM and DPR.

9.1.2 Multi-level Adaptive Neighbour Selection Strategy for Joint Sparse Modelling

In this study, we proposed a strategy for selecting the most representative pixels in the predefined region, thereby making the joint sparse matrix more reliable. In order to further improve classification performance, the neighbour selection strategy was used in a multi-level manner.

Theoretically, a given specific area will exhibit distinct structures and characteristics as well as some irrelevant information. If a strategy aims to find the most similar neighbouring pixels to the test pixel and reject dissimilar pixels, information concerning the correlated spatial context should be more representative for classification. Based on this principle, an adaptive neighbour selection strategy was developed that computes weights based on the distances between pixels, with the labels of training data as a

priori information. Structural similarity between the central pixel and its neighbours can be sensibly exploited by considering the different contributions of each spectral band. The adaptive neighbour selection strategy was used in two different scenarios in this study. It was firstly applied to select the most representative neighbours for the test pixels using a top *N*-nearest method, and doing so was proven to efficiently improve the classification accuracy of a JSM. Considering that features from different levels do not share the same sparsity pattern, it is reasonable to construct feature models using multiple levels in order to wholly represent the data. We therefore proposed using a multi-level weighted joint sparse model to fully integrate neighbour information as well as to avoid outliers dominating the sparse coding. Multiple local matrices were obtained using the proposed adaptive neighbor selection strategy with different distance thresholds applied.

This multi-level approach was experimentally implemented on three benchmark hyperspectral data sets, and the results showed that the proposed framework can achieve superior performance with limited training samples. Furthermore, the proposed multilevel strategy can alleviate the over-smoothing effect of JSM. This method provides information from different levels for the designation of features with high reliability, thereby improving the HSI classification.

However, the inner production computation for the multilevel sparse code learning incurs a high computational cost. Furthermore, parameter selection in this framework relies on human experience and requires manual operation. In the future, these methods will be improved by applying automated optimization approaches such as the swarm optimization.

9.1.3 Multiscale Conservative Smoothing with Adaptive Sparse Representation

Spatial information has been demonstrated to be useful for HSI classification. However, one challenge of utilizing spatial information is that spatial properties are often present at various scales rather than co-occurring on a single fixed scale. Spatial smoothing is a technique for emphasizing main features after suppressing the undesired variation within a homogenous region. Spatial structures and geometrical features in HSIs can be

enhanced and revealed by filtering, especially around the edges. Accordingly, a multi-scale conservative smoothing algorithm was proposed in Chapter 6 to reduce noise and extract spatial structure information from coarse and fine levels alike.

Firstly, a conservative smoothing algorithm was developed that considered adaptive weights for different neighbouring pixels around the central pixel; these weights was determined from the spectral similarity between the neighbouring pixel and the central pixel. This approach can reveal spatial textural information. Over-smoothing was automatically prevented by imposing a weighting scheme on the neighbouring pixels used for smoothing, where contributions from dissimilar neighbors were suppressed. The proposed smoothing scheme was then used in a multiscale way, in which a series of different HSIs were obtained by applying various neighbourhood scales. Subsequently, an adaptive sparse representation was introduced to integrate different characteristics from the series of enhanced HSIs.

Experiments were conducted on three challenging data sets, and the proposed methodology demonstrated superior classification performance when compared to several well-known classifiers. One can also conclude from the results that the spatial information extracted from multi-scale neighbourhoods provided more robust and efficient features for the classification task. Combining the various features generated by the multi-scale filter in an adaptive strategy enhanced the performance of the proposed method over a single-scale-based algorithm. Moreover, the adaptive norm applied in the sparse representation considered correlations (i.e. similarity and diversity) among different matrices sets, making the framework more robust.

However, similar to the strategy described in Chapter 5, the parameter tuning of this method needs to be improved. Moreover, its computation costs are high, and parallel computing can be used to alleviate this problem.

9.1.4 Multiple Feature Learning Using CNNs

The strengths of CNNs as applied to HSI classification are their better feature representation and high performance, while multiple feature learning has shown its effectiveness in the area of computer vision. It is reasonable to combine these methods

by applying CNN models to simultaneously extract spatial and spectral information from multiple features in order to obtain robust and effective features for HSI classification. In Chapter 7, an enhanced framework that combined a CNN and a multiple feature learning method was proposed.

We built a novel CNN architecture with various features extracted from the raw imagery as input. The network generated the corresponding relevant feature maps, and these maps were fed into a concatenating layer to form a joint feature map. The joint feature map was then input to subsequent layers to predict the final labels for each hyperspectral pixel. The proposed method not only takes advantage of the CNN capability for enhanced feature extraction, but also jointly exploits the spectral and spatial information.

It is evident in the experimental results that CNNs with multiple features learning can improve classification accuracy significantly. In addition, the parameter analysis showed the proposed CNN with its multiple feature learning outperformed those that directly classified the stacked multiple features. The proposed network was relatively shallow as the limited availability of training samples is a perennial problem for HSI analysis, where applying a deeper and wider network may result in overfitting; yet although shallow, the network was still an effective one. This work supplements existing knowledge on constructing effective CNNs for HSI classification, and is expected to provide various improvements for the purpose of better feature representation.

Since CNNs embrace a variety of architectures, how to design a CNN optimal for various HSIs is still an active subject of research. When using a CNN with multiple features, the features should be manually extracted with predefined parameters. Thus, one avenue for the future improvement of this method is implementing adaptive feature extraction for CNNs.

9.2 Future Work

In this thesis, the dictionary used for sparse learning was directly constructed from all training samples. However, this approach can lead to some redundancy. A feasible alternative is to develop discriminative algorithms to construct a more representative dictionary. In addition, since signals from different classes may share some similar characteristics while pixels from the same class may present some differences, dictionary construction in future work should consider both "globality" and "locality." For example, the adaptive neighbour selection strategy proposed in Chapter 5 can be applied to select optimal dictionary atoms, although such a pixel-wise selection scheme will result in a high computational burden.

The process of classifying each pixel with dictionary learning may lead to high computational complexity. One alternative is to integrate superpixel–based segmentation with sparse representation models, as introduced very recently in the literature. By using an efficient segmentation approach, an image can be clustered into many superpixels, thus reducing the number to be classified. However, this technique is still at a very early stage, and published segmentation methods are generally based on conventional statistical algorithms. In order to make superpixels more representative, the segmentation approach should depend on spatial structures instead of conventional histograms or higher-order statistics. Segmentation results may be refined by applying the shape adaptive method applied in Chapter 6. Subsequently, a JSM or CNN can be applied to classify each superpixel instead of each pixel.

Although multiple feature learning is a promising technique for resolving the *curse of dimensionality* and problems posed by limited numbers of training samples for the classification of hyperspectral data, its performance is influenced by the type and number of features. At present, there is a need to manually extract the features prior to classification. For example, in Chapter 5, the number of levels needed to be predefined empirically. In Chapter 6, the number of scales for the conservative smoothing scheme also needed to be tuned manually. Furthermore, the features in Chapter 7 were handcrafted features extracted prior to classification. Therefore, another direction for future research is to develop strategies for automatic feature selection. Such a strategy

should be adaptive for different objects in terms of shape and size. Another means for improving hyperspectral image classification is the nonlinear combination of multiple features, as hyperspectral pixels may not be linearly separable [166].

In order to exploit similarity and diversity among different features, Chapter 7 used the adaptive norm to allow pixels of each scale to be represented by an appropriate term. Chapter 8 built a concatenation layer to explore the correlations of various features. A future direction is suggested here, that the feature maps can initially be learnt by multiple nonlinear functions, then combined together as the input of an efficient classifier (e.g. a CNN) to discover more discriminative information for classification.

The training procedure for any model is reliant on the training samples used, and a main concern for HSI classification is that sufficient training samples are not available in most cases. The utilization of other data sources (e.g. multi-temporal images) can be a potential direction for improving classification performance.

Although deep learning-based methods have been applied to HSI classification, they are still in the early stage of development. Deep learning can be incorporated in tandem with other approaches, such as graphical models and segmentation methods, to achieve better classification performance.

References

[1] J. A. Richard, and X. Jia, *Remote Sensing Digital Imge Analysis: An Introduction*, 4th ed.: Springer-Verlag, 2006.

[2] R. Loudon, *The quantum theory of light*: OUP Oxford, 2000.

[3] J. A. Benediktsson, J. Chanussot, and W. M. Moon, "Very high-resolution remote sensing: Challenges and opportunities [point of view]," *Proceedings of the IEEE,* vol. 100, no. 6, pp. 1907-1910, 2012.

[4] X. Jia, "Classification tecniques for hyperspectral remote sensing image data. (Unpublished doctoral dissertation)," University of New South Wales, Australia, 1996.

[5] J. B. Campbell, and R. H. Wynne, *Introduction to remote sensing*: Guilford Press, 2011.

[6] G. Camps-Valls, D. Tuia, L. Bruzzone *et al.*, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Processing Magazine,* vol. 31, no. 1, pp. 45-54, 2014.

[7] M. Fauvel, Y. Tarabalka, J. A. Benediktsson *et al.*, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE,* vol. 101, no. 3, pp. 652-675, 2013.

[8] P. M. Baggenstoss, "Class-specific classifier: avoiding the curse of dimensionality," *IEEE Aerospace and Electronic Systems Magazine,* vol. 19, no. 1, pp. 37-52, 2004.

[9] D. L. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," *AMS Math Challenges Lecture,* vol. 1, no. 32, pp. 375, 2000.

[10] A. Agarwal, T. El-Ghazawi, H. El-Askary *et al.*, "Efficient hierarchical-PCA dimension reduction for hyperspectral imagery," in *IEEE International Symposium on Signal Processing and Information Technology*, 2007, pp. 353-356.

[11] 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.

[12] M. Pal, and G. M. Foody, "Feature selection for classification of hyperspectral data by SVM," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 48, no. 5, pp. 2297-2307, 2010.

[13] G. Camps-Valls, and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 43, no. 6, pp. 1351-1362, 2005.

[14] S. Kuching, "The performance of maximum likelihood, spectral angle mapper, neural network and decision tree classifiers in hyperspectral image analysis," *Journal of Computer Science,* vol. 3, no. 6, pp. 419-423, 2007.

[15] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression," *IEEE Geoscience and Remote Sensing Letters,* vol. 10, no. 2, pp. 318-322, 2013.

[16] 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, 2010.

[17] Q. Du, "Modified Fisher's linear discriminant analysis for hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters,* vol. 4, no. 4, pp. 503-507, 2007.

[18] Q. Gao, S. Lim, and X. Jia, "Improved Joint Sparse Models for Hyperspectral Image Classification Based on a Novel Neighbour Selection Strategy," *Remote Sensing,* vol. 10, no. 6, pp. 905, 2018.

[19] P. Ghamisi, E. Maggiori, S. Li *et al.*, "New Frontiers in Spectral-Spatial Hyperspectral Image Classification: The Latest Advances Based on Mathematical Morphology, Markov Random Fields, Segmentation, Sparse Representation, and Deep Learning," *IEEE Geoscience and Remote Sensing Magazine,* vol. 6, no. 3, pp. 10-43, 2018.

[20] X. Cao, L. Xu, D. Meng *et al.*, "Integration of 3-dimensional discrete wavelet transform and Markov random field for hyperspectral image classification," *Neurocomputing,* vol. 226, pp. 90-100, 2017.

[21] F. I. Alam, J. Zhou, A. W.-C. Liew *et al.*, "Conditional Random Field and Deep Feature Learning for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2018.

[22] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 49, no. 10, pp. 3973-3985, 2011.

[23] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing,* vol. 219, pp. 88-98, 2017.

[24] M. Fauvel, J. A. Benediktsson, J. Chanussot *et al.*, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 46, no. 11, pp. 3804-3814, 2008.

[25] M. Dalla Mura, A. Villa, J. A. Benediktsson *et al.*, "Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis," *IEEE Geoscience and Remote Sensing Letters,* vol. 8, no. 3, pp. 542-546, 2011.

[26] P. Ghamisi, R. Souza, J. A. Benediktsson *et al.*, "Hyperspectral data classification using extended extinction profiles," *IEEE Geoscience and Remote Sensing Letters,* vol. 13, no. 11, pp. 1641-1645, 2016.

[27] P. Ghamisi, M. S. Couceiro, F. M. Martins *et al.*, "Multilevel image segmentation based on fractional-order Darwinian particle swarm optimization," *IEEE Transactions on Geoscience and Remote sensing,* vol. 52, no. 5, pp. 2382-2394, 2014.

[28] M. Pesaresi, and J. A. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 39, no. 2, pp. 309-320, 2001.

[29] D. Lu, and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote sensing,* vol. 28, no. 5, pp. 823-870, 2007.

[30] G. M. Foody, "Status of land cover classification accuracy assessment," *Remote Sensing of Environment,* vol. 80, no. 1, pp. 185-201, 2002.

[31] P. Smits, S. Dellepiane, and R. Schowengerdt, "Quality assessment of image classification algorithms for land-cover mapping: a review and a proposal for a cost-based approach," *International Journal of Remote Sensing,* vol. 20, no. 8, pp. 1461-1486, 1999.

[32] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 43, no. 3, pp. 480-491, 2005.

[33] J. Plaza, A. J. Plaza, and C. Barra, "Multi-channel morphological profiles for classification of hyperspectral images using support vector machines," *Sensors,* vol. 9, no. 1, pp. 196-218, 2009.

[34] Y. Gu, T. Liu, X. Jia *et al.*, "Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 54, no. 6, pp. 3235-3247, 2016.

[35] M. Dalla Mura, J. A. Benediktsson, B. Waske *et al.*, "Morphological attribute profiles for the analysis of very high resolution images," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 48, no. 10, pp. 3747-3762, 2010.

[36] M. Dalla Mura, J. Atli Benediktsson, B. Waske *et al.*, "Extended profiles with morphological attribute filters for the analysis of hyperspectral data," *International Journal of Remote Sensing,* vol. 31, no. 22, pp. 5975-5991, 2010.

[37] N. Falco, J. A. Benediktsson, and L. Bruzzone, "Spectral and spatial classification of hyperspectral images based on ICA and reduced morphological attribute profiles," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 53, no. 11, pp. 6223-6240, 2015.

[38] M. Pedergnana, P. R. Marpu, M. Dalla Mura *et al.*, "A novel technique for optimal feature selection in attribute profiles based on genetic algorithms," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 51, no. 6, pp. 3514-3528, 2013.

[39] P. R. Marpu, M. Pedergnana, M. Dalla Mura *et al.*, "Automatic generation of standard deviation attribute profiles for spectral–spatial classification of remote sensing data," *IEEE Geoscience and Remote Sensing Letters,* vol. 10, no. 2, pp. 293-297, 2013.

[40] P. R. Marpu, M. Pedergnana, M. D. Mura *et al.*, "Classification of hyperspectral data using extended attribute profiles based on supervised and unsupervised feature

extraction techniques," *International Journal of Image and Data Fusion,* vol. 3, no. 3, pp. 269-298, 2012.

[41] J. Li, P. R. Marpu, A. Plaza *et al.*, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 51, no. 9, pp. 4816-4829, 2013.

[42] J. Xia, M. Dalla Mura, J. Chanussot *et al.*, "Random subspace ensembles for hyperspectral image classification with extended morphological attribute profiles," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 53, no. 9, pp. 4768-4786, 2015.

[43] Z. Ye, Y. Yan, L. Bai *et al.*, "Feature extraction based on morphological attribute profiles for classification of hyperspectral image," in *International Conference on Digital Image Processing*, 2018, pp. 108060C.

[44] M.-T. Pham, S. Lefèvre, E. Aptoula *et al.*, "Recent developments from attribute profiles for remote sensing image classification," *arXiv preprint arXiv:1803.10036*, 2018.

[45] P. Ghamisi, M. Dalla Mura, and J. A. Benediktsson, "A survey on spectral–spatial classification techniques based on attribute profiles," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 53, no. 5, pp. 2335-2353, 2015.

[46] J. A. Benediktsson, and P. Ghamisi, Spectral-spatial classification of hyperspectral remote sensing images: Artech House, 2015.

[47] P. Ghamisi, R. Souza, J. A. Benediktsson *et al.*, "Extinction profiles for the classification of remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 54, no. 10, pp. 5631-5645, 2016.

[48] P. Du, J. Xia, P. Ghamisi *et al.*, "Multiple composite kernel learning for hyperspectral image classification," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Fort Worth, Texas, USA, 2017, pp. 2223-2226.

[49] L. Fang, N. He, S. Li *et al.*, "Extinction profiles fusion for hyperspectral images classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 56, no. 3, pp. 1803-1815, 2018.

[50] P. Ghamisi, B. Höfle, and X. X. Zhu, "Hyperspectral and LiDAR data fusion using extinction profiles and deep convolutional neural network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 10, no. 6, pp. 3011-3024, 2017.

[51] M. Zhang, P. Ghamisi, and W. Li, "Classification of hyperspectral and LIDAR data using extinction profiles with feature fusion," *Remote Sensing Letters,* vol. 8, no. 10, pp. 957-966, 2017.

[52] J. Xia, P. Ghamisi, N. Yokoya *et al.*, "Random forest ensembles and extended multiextinction profiles for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 56, no. 1, pp. 202-216, 2018.

[53] S. Geman, and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *Readings in computer vision*, pp. 564-584: Elsevier, 1987.

[54] V. Kolmogorov, and R. Zabin, "What energy functions can be minimized via graph cuts?," *IEEE transactions on pattern analysis and machine intelligence,* vol. 26, no. 2, pp. 147-159, 2004.

[55] P. Chen, J. D. Nelson, and J.-Y. Tourneret, "Toward a sparse Bayesian Markov random field approach to hyperspectral unmixing and classification," *IEEE Transactions on Image Processing,* vol. 26, no. 1, pp. 426-438, 2017.

[56] L. Xu, and J. Li, "Bayesian classification of hyperspectral imagery based on probabilistic sparse representation and Markov random field," *IEEE Geoscience and Remote Sensing Letters,* vol. 11, no. 4, pp. 823-827, 2014.

[57] B. Zhang, S. Li, X. Jia *et al.*, "Adaptive Markov random field approach for classification of hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters,* vol. 8, no. 5, pp. 973-977, 2011.

[58] G. Moser, and S. B. Serpico, "Combining support vector machines and Markov random fields in an integrated framework for contextual image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 51, no. 5, pp. 2734-2752, 2013.

[59] P. Ghamisi, J. A. Benediktsson, and M. O. Ulfarsson, "Spectral–spatial classification of hyperspectral images based on hidden Markov random fields," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 52, no. 5, pp. 2565-2574, 2014.

[60] H. Aghighi, J. Trinder, K. Wang *et al.*, "Smoothing parameter estimation for Markov random field classification of non-Gaussian distribution image," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 2, no. 7, pp. 1, 2014.

[61] W. Li, S. Prasad, and J. E. Fowler, "Hyperspectral image classification using Gaussian mixture models and Markov random fields," *IEEE Geoscience and Remote Sensing Letters,* vol. 11, no. 1, pp. 153-157, 2014.

[62] J. Xia, J. Chanussot, P. Du *et al.*, "Spectral–spatial classification for hyperspectral data using rotation forests with local feature extraction and Markov random fields," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 53, no. 5, pp. 2532-2546, 2015.

[63] S. Sun, P. Zhong, H. Xiao *et al.*, "An MRF model-based active learning framework for the spectral-spatial classification of hyperspectral imagery," *IEEE Journal of Selected Topics in Signal Processing,* vol. 9, no. 6, pp. 1074-1088, 2015.

[64] M. Mignotte, "A multiresolution markovian fusion model for the color visualization of hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 48, no. 12, pp. 4236-4247, 2010.

[65] M. Golipour, H. Ghassemian, and F. Mirzapour, "Integrating hierarchical segmentation maps with MRF prior for classification of hyperspectral images in a Bayesian

framework," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 54, no. 2, pp. 805-816, 2016.

[66] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 50, no. 3, pp. 809-823, 2012.

[67] U. Srinivas, Y. Chen, V. Monga *et al.*, "Exploiting sparsity in hyperspectral image classification via graphical models," *IEEE Geoscience and Remote Sensing Letters,* vol. 10, no. 3, pp. 505-509, 2013.

[68] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 11, pp. 1222-1239, 2001.

[69] A. T. Ihler, W. F. John III, and A. S. Willsky, "Loopy belief propagation: Convergence and effects of message errors," *Journal of Machine Learning Research,* vol. 6, no. May, pp. 905-936, 2005.

[70] M. F. Tappen, and W. T. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters," in *IEEE International Conference on Computer Vision*, Nice, France, France, 2003, pp. 900.

[71] P. Zhong, and R. Wang, "Learning conditional random fields for classification of hyperspectral images," *IEEE Transactions on Image Processing,* vol. 19, no. 7, pp. 1890-1907, 2010.

[72] F. Li, L. Xu, P. Siva *et al.*, "Hyperspectral image classification with limited labeled training samples using enhanced ensemble learning and conditional random fields," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 8, no. 6, pp. 2427-2438, 2015.

[73] R. Roscher, and B. Waske, "Superpixel-based classification of hyperspectral data using sparse representation and conditional random fields," in *IEEE Geoscience and Remote Sensing Symposium*, Quebec City, QC, Canada, 2014, pp. 3674-3677.

[74] F. Yao, Y. Qian, Z. Hu *et al.*, "A novel hyperspectral remote sensing images classification using Gaussian Processes with conditional random fields," in *IEEE International Conference on Intelligent Systems and Knowledge Engineering*, Hangzhou, China, 2010, pp. 197-202.

[75] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proceedings of the IEEE,* vol. 78, no. 4, pp. 678-689, 1990.

[76] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," *Pattern Recognition,* vol. 43, no. 7, pp. 2367-2379, 2010.

[77] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, "Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 47, no. 8, pp. 2973-2987, 2009.

[78] N. L. Kazanskiy, P. G. Serafimovich, and E. A. Zimichev, "Spectral-spatial classification of hyperspectral images with k-means++ partitional clustering," in *Optical Technologies for Telecommunications* Kazan, Russian Federation, 2015, pp. 95330M.

[79] Y. Tarabalka, J. A. Benediktsson, J. Chanussot *et al.*, "Classification of hyperspectral data using support vector machines and adaptive neighborhoods," in *Proc. 6th EARSeL SIG IS Workshop*, 2009, pp. 1-6.

[80] D. Akbari, A. Safari, and S. Homayouni, "Rule-based Classification of a Hyperspectral Image using MSSC Hierarchical Segmentation," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 1, pp. W3, 2013.

[81] Z. Zhang, E. Pasolli, M. M. Crawford *et al.*, "An active learning framework for hyperspectral image classification using hierarchical segmentation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 9, no. 2, pp. 640-654, 2016.

[82] Y. Tarabalka, J. C. Tilton, J. A. Benediktsson *et al.*, "A marker-based approach for the automated selection of a single segmentation from a hierarchical set of image segmentations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 5, no. 1, pp. 262-272, 2012.

[83] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using minimum spanning forest grown from automatically selected markers," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 40, no. 5, pp. 1267-1279, 2010.

[84] S. Valero, P. Salembier, and J. Chanussot, "Hyperspectral image representation and processing with binary partition trees," *IEEE Transactions on image processing,* vol. 22, no. 4, pp. 1430-1443, 2013.

[85] M. A. Veganzones, G. Tochon, M. Dalla-Mura *et al.*, "Hyperspectral image segmentation using a new spectral unmixing-based binary partition tree representation," *IEEE Transactions on Image Processing,* vol. 23, no. 8, pp. 3574-3589, 2014.

[86] E. Maggiori, Y. Tarabalka, and G. Charpiat, "Optimizing partition trees for multi-object segmentation with shape prior," in *British Machine Vision Conference*, Swansea, United Kingdom, 2015.

[87] E. Maggiori, Y. Tarabalka, and G. Charpiat, "Improved partition trees for multi-class segmentation of remote sensing images," in *International Geoscience and Remote Sensing Symposium (IGARSS)*, Milan, Italy, 2015, pp. 1016-1019.

[88] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Hyperspectral image segmentation using a new Bayesian approach with active learning," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 49, no. 10, pp. 3947-3960, 2011.

[89] M. Elad, and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, NY, USA, USA, 2006, pp. 895-900.

[90] Y. Wu, E. Blasch, G. Chen *et al.*, "Multiple source data fusion via sparse representation for robust visual tracking," in *International Conference on Information Fusion*, Chicago, IL, USA, 2011, pp. 1-8.

[91] W. Dong, L. Zhang, G. Shi *et al.*, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing,* vol. 22, no. 4, pp. 1620-1630, 2013.

[92] K. Huang, and S. Aviyente, "Sparse representation for signal classification," in *Advances in neural information processing systems*, Vancouver, B.C., Canada., 2007, pp. 609-616.

[93] L. Fang, S. Li, X. Kang *et al.*, "Spectral–spatial hyperspectral image classification via multiscale adaptive sparse representation," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 52, no. 12, pp. 7738-7749, 2014.

[94] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal processing,* vol. 86, no. 3, pp. 572-588, 2006.

[95] B. Tu, S. Huang, L. Fang *et al.*, "Hyperspectral Image Classification via Weighted Joint Nearest Neighbor and Sparse Representation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2018.

[96] J. Zou, W. Li, and Q. Du, "Sparse representation-based nearest neighbor classifiers for hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters,* vol. 12, no. 12, pp. 2418-2422, 2015.

[97] H. Zhang, J. Li, Y. Huang *et al.*, "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, 2014.

[98] W. Fu, S. Li, L. Fang *et al.*, "Hyperspectral image classification via shape-adaptive joint sparse representation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 9, no. 2, pp. 556-567, 2016.

[99] Y. Chen, N. M. Nasrabadi, and T. D. Tran., "Hyperspectral Image Classification via Kernel Sparse Representation," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 51, no. 1, pp. 217-231, Jan, 2013.

[100] B. Song, J. Li, M. Dalla Mura *et al.*, "Remotely sensed image classification using sparse representations of morphological attribute profiles," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 52, no. 8, pp. 5122-5136, 2014.

[101] L. Gan, J. Xia, P. Du *et al.*, "Multiple feature kernel sparse representation classifier for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 56, no. 9, pp. 5343-5356, 2018.

[102] E. Zhang, X. Zhang, H. Liu *et al.*, "Fast multifeature joint sparse representation for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters,* vol. 12, no. 7, pp. 1397-1401, 2015.

[103] J. Li, H. Zhang, and L. Zhang, "Efficient superpixel-level multitask joint sparse representation for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 53, no. 10, pp. 5338-5351, 2015.

[104] L. Fang, S. Li, X. Kang *et al.*, "Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 53, no. 8, pp. 4186-4201, 2015.

[105] W. Fu, S. Li, L. Fang *et al.*, "Adaptive spectral–spatial compression of hyperspectral image with sparse representation," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 55, no. 2, pp. 671-682, 2017.

[106] J. Cao, K. Zhang, M. Luo *et al.*, "Extreme learning machine and adaptive sparse representation for image classification," *Neural networks,* vol. 81, pp. 91-102, 2016.

[107] B. Tu, X. Zhang, X. Kang *et al.*, "Hyperspectral Image Classification via Fusing Correlation Coefficient and Joint Sparse Representation," *IEEE Geoscience and Remote Sensing Letters,* vol. 15, no. 3, pp. 340-344, 2018.

[108] J. Feng, L. Liu, X. Cao *et al.*, "Marginal Stacked Autoencoder With Adaptively-Spatial Regularization for Hyperspectral Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 11, no. 9, pp. 3297-3311, 2018.

[109] J. Zabalza, J. Ren, J. Zheng *et al.*, "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging," *Neurocomputing,* vol. 185, pp. 1-10, 2016.

[110] C. Xing, L. Ma, and X. Yang, "Stacked denoise autoencoder based feature extraction and classification for hyperspectral images," *Journal of Sensors,* vol. 2016, 2015.

[111] Y. Chen, Z. Lin, X. Zhao *et al.*, "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, 2014.

[112] C. Tao, H. Pan, Y. Li *et al.*, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geoscience and Remote Sensing Letters,* vol. 12, no. 12, pp. 2438-2442, 2015.

[113] X. Ma, H. Wang, and J. Geng, "Spectral–spatial classification of hyperspectral image based on deep auto-encoder," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 9, no. 9, pp. 4073-4085, 2016.

[114] Z. Lin, Y. Chen, X. Zhao *et al.*, "Spectral-spatial classification of hyperspectral image using autoencoders," in *International Conference on Information, Communications & Signal Processing*, Tainan, Taiwan, 2013, pp. 1-5.

[115] C. Li, Y. Wang, X. Zhang *et al.*, "Deep Belief Network for Spectral–Spatial Classification of Hyperspectral Remote Sensor Data," *Sensors,* vol. 19, no. 1, pp. 204, 2019.

[116] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 8, no. 6, pp. 2381-2392, 2015.

[117] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *International Conference on Image Processing (ICIP)*, Paris, France, 2014, pp. 5132-5136.

[118] P. Zhong, Z. Gong, S. Li *et al.*, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 55, no. 6, pp. 3516-3530, 2017.

[119] I. Sutskever, and G. E. Hinton, "Deep, narrow sigmoid belief networks are universal approximators," *Neural computation,* vol. 20, no. 11, pp. 2629-2636, 2008.

[120] B. Ayhan, and C. Kwan, "Application of deep belief network to land cover classification using hyperspectral images," in *International Symposium on Neural Networks*, Hokkaido, Japan, 2017, pp. 269-276.

[121] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 55, no. 7, pp. 3639-3655, 2017.

[122] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, Lake Tahoe, NV, USA, 2012, pp. 1097-1105.

[123] W. Hu, Y. Huang, L. Wei *et al.*, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors,* vol. 2015, 2015.

[124] H. Lee, and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Transactions on Image Processing,* vol. 26, no. 10, pp. 4843-4855, 2017.

[125] Y. Chen, H. Jiang, C. Li *et al.*, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 54, no. 10, pp. 6232-6251, 2016.

[126] Q. Gao, S. Lim, and X. Jia, "Hyperspectral Image Classification Using Convolutional Neural Networks and Multiple Feature Learning," *Remote Sensing,* vol. 10, no. 2, pp. 299, 2018.

[127] J. Zhu, L. Fang, and P. Ghamisi, "Deformable Convolutional Neural Networks for Hyperspectral Image Classification," *IEEE Geoscience and Remote Sensing Letters,* vol. 15, no. 8, pp. 1254-1258, 2018.

[128] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral–spatial feature learning via deep residual conv–deconv network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 56, no. 1, pp. 391-406, 2018.

[129] L. Shu, K. McIsaac, and G. R. Osinski, "Hyperspectral image classification with stacking spectral patches and convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, no. 99, pp. 1-10, 2018.

[130] M. Paoletti, J. Haut, J. Plaza *et al.*, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 145, pp. 120-147, 2018.

[131] M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Transactions on Image Processing,* vol. 27, no. 6, pp. 2623-2634, 2018.

[132] G. Cheng, Z. Li, J. Han *et al.*, "Exploring hierarchical convolutional features for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, no. 99, pp. 1-11, 2018.

[133] Y. Chen, L. Zhu, P. Ghamisi *et al.*, "Hyperspectral images classification with Gabor filtering and convolutional neural network," *IEEE Geoscience and Remote Sensing Letters,* vol. 14, no. 12, pp. 2355-2359, 2017.

[134] H. Liang, and Q. Li, "Hyperspectral imagery classification using sparse representations of convolutional neural network features," *Remote Sensing,* vol. 8, no. 2, pp. 99, 2016.

[135] J. Cao, Z. Chen, and B. Wang, "Deep Convolutional networks with superpixel segmentation for hyperspectral image classification," in *International Geoscience and Remote Sensing Symposium (IGARSS)*, Beijing, China, 2016, pp. 3310-3313.

[136] F. Hu, G.-S. Xia, J. Hu *et al.*, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing,* vol. 7, no. 11, pp. 14680-14707, 2015.

[137] J. M. Haut, M. E. Paoletti, J. Plaza *et al.*, "Active learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach," *IEEE Transactions on Geoscience and Remote Sensing*, no. 99, pp. 1-22, 2018.

[138] IEEE GRSS. (2013). Image analysis and data fusion. [Online]. Available: http://www.grss-ieee-org/community/technical-committess/data-fusion/.

[139] D. A. Landgrebe, Signal theory methods in multispectral remote sensing: John Wiley & Sons, 2005.

[140] P. Meneses, and T. Almeida, Remote Sensing Digital Image Analysis: An Introduction, 1999.

[141] Y. Wang, R. Niu, and X. Yu, "Anisotropic diffusion for hyperspectral imagery enhancement," *IEEE Sensors Journal,* vol. 10, no. 3, pp. 469-477, 2010.

[142] J. Richards, D. Landgrebe, and P. Swain, "Pixel labeling by supervised probabilistic relaxation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 188-191, 1981.

[143] L. Sun, Z. Wu, J. Liu *et al.*, "Supervised hyperspectral image classification using sparse logistic regression and spatial-tv regularization," in *International Geoscience and Remote Sensing Symposium (IGARSS)*, Melbourne, VIC, Australia, 2013, pp. 1019-1022.

[144] R. Dubes, A. Jain, S. Nadabar *et al.*, "MRF model-based algorithms for image segmentation," in *International Conference on Pattern Recognition*, Atlantic City, NJ, USA, 1990, pp. 808-814.

[145] L. Sun, Z. Wu, J. Liu *et al.*, "Supervised spectral–spatial hyperspectral image classification with weighted Markov random fields," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 53, no. 3, pp. 1490-1503, 2015.

[146] J. Li, M. Khodadadzadeh, A. Plaza *et al.*, "A discontinuity preserving relaxation scheme for spectral–spatial hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 9, no. 2, pp. 625-639, 2016.

[147] Q. Gao, S. Lim, and X. Jia, "Hyperspectral Image Classification Using Joint Sparse Model and Discontinuity Preserving Relaxation," *IEEE Geoscience and Remote Sensing Letters,* vol. 15, no. 1, pp. 78-82, 2018.

[148] S. G. Mallat, and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing,* vol. 41, no. 12, pp. 3397-3415, 1993.

[149] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, 2010, pp. 3517-3524.

[150] J. Li, H. Zhang, and L. Zhang, "Supervised segmentation of very high resolution images by the use of extended morphological attribute profiles and a sparse transform," *IEEE Geoscience and Remote Sensing Letters,* vol. 11, no. 8, pp. 1409-1413, 2014.

[151] R. Roscher, and B. Waske, "Shapelet-based sparse representation for landcover classification of hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 54, no. 3, pp. 1623-1634, 2015.

[152] F. Melgani, and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on geoscience and remote sensing,* vol. 42, no. 8, pp. 1778-1790, 2004.

[153] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí *et al.*, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters,* vol. 3, no. 1, pp. 93-97, 2006.

[154] I. Yıldırım, O. K. Ersoy, and B. Yazgan, "Improvement of classification accuracy in remote sensing using morphological filter," *Advances in Space Research,* vol. 36, no. 5, pp. 1003-1006, 2005.

[155] B. Pan, Z. Shi, and X. Xu, "Hierarchical guidance filtering-based ensemble classification for hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 55, no. 7, pp. 4177-4189, 2017.

[156] L. He, J. Li, A. Plaza *et al.*, "Discriminative low-rank Gabor filtering for spectral–spatial hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 55, no. 3, pp. 1381-1395, 2017.

[157] L. Zhang, L. Zhang, D. Tao *et al.*, "On combining multiple features for hyperspectral remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 50, no. 3, pp. 879-893, 2012.

[158] J. Li, X. Huang, P. Gamba *et al.*, "Multiple feature learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 53, no. 3, pp. 1592-1606, 2015.

[159] L. Fang, C. Wang, S. Li *et al.*, "Hyperspectral image classification via multiple-feature-based adaptive sparse representation," *IEEE Transactions on Instrumentation and Measurement,* vol. 66, no. 7, pp. 1646-1657, 2017.

[160] V. Katkovnik, A. Foi, K. Egiazarian *et al.*, "Directional varying scale approximations for anisotropic signal processing," in *European Signal Processing Conference* Vienna, Austria, 2004, pp. 101-104.

[161] C. Ding, C. Xu, and D. Tao, "Multi-task pose-invariant face recognition," *IEEE Transactions on Image Processing,* vol. 24, no. 3, pp. 980-993, 2015.

[162] C. Zhu, and Y. Peng, "A boosted multi-task model for pedestrian detection with occlusion handling," *IEEE Transactions on Image Processing,* vol. 24, no. 12, pp. 5619-5629, 2015.

[163] W. Liu, T. Mei, Y. Zhang *et al.*, "Multi-task deep visual-semantic embedding for video thumbnail selection," in *Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 3707-3715.

[164] V. Nair, and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning* Haifa, Israel, 2010, pp. 807-814.

[165] A. Vedaldi, and K. Lenc, "MatConvNet - Convolutional Neural Networks for MATLAB," in *International Conference on Multimedia*, Brisbane, Australia, 2015, pp. 689-692

[166] J. Li, H. Zhang, and L. Zhang, "Column-generation kernel nonlocal joint collaborative representation for hyperspectral image classification," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 94, pp. 25-36, 2014.