Hyperspectral Image Denoising

Using Low-Rank-Based Methods

Bin Zhao

Dissertation submitted in partial fulfillment of a *Philosophiae Doctor* degree
in
Electrical and Computer Engineering

Advisors
Professor Jóhannes R. Sveinsson
Professor Magnús Örn Úlfarsson

PhD Committee
Professor Jóhannes R. Sveinsson
Professor Magnús Örn Úlfarsson
Professor Jocelyn Chanussot

Opponents
Professor Javier Plaza Miguel
Professor Björn Waske

Faculty of Electrical and Computer Engineering
School of Engineering and Natural Sciences
University of Iceland
Reykjavik, December 2021

Hyperspectral Image Denoising Using Low-Rank-Based Methods
Dissertation submitted in partial fulfillment of a *Philosophiae Doctor* degree in
Electrical and Computer Engineering

Copyright © Bin Zhao 2021
All rights reserved

Faculty of Electrical and Computer Engineering
School of Engineering and Natural Sciences
University of Iceland
Hjarðarhagi 2-6
107, Reykjavik
Iceland

Telephone: 525-4000

Bibliographic information:
Bin Zhao, 2021, *Hyperspectral Image Denoising Using Low-Rank-Based Methods*,
PhD dissertation, Faculty of Electrical and Computer Engineering, University of Iceland

ISBN 978-9935-9647-0-0

Printing: Háskólaprent
Reykjavik, Iceland, December 2021

Abstract

Hyperspectral images (HSIs) acquired by hyperspectral imaging sensors contain hundreds of spectral bands. The abundant spectral information provided by an HSI makes it possible to discriminate different materials in a scene. Therefore, HSIs have been widely used in various fields, such as agricultural production, mineral exploration, and environmental protection. However, HSIs are usually corrupted by different noises such as quantization noise, thermal noise, and shot noise, due to the influence of photon effects, atmospheric absorption, and sensor disturbance. Therefore, denoising is an important preprocessing step for HSI analysis and applications. HSI has the low-rank characteristic, which is exploited by representing the spectral vectors of HSI in a low-dimensional subspace. Thus, it is an effective way to use the low-rank characteristic of HSI for HSI denoising. This thesis proposes several low-rank-based denoising methods for HSI to improve HSI applications.

The main contributions of the thesis are the following.

- Two new low-rank based HSI denoising methods. One is the non-local means low-rank approximation (NLMLRA) method, which uses a Slanted Butterworth function to construct a low-rank approximation for non-local means (NLM) operator. The Slanted Butterworth function is a filter function, that removes the noise in the NLM operator and retains the eigenvalues above a given threshold value to preserve the low-rank charactoristic of HSI. The Chebyshev polynomials is used for NLMRA to improve the practicability and reduce the computational cost. The other is the wavelet-based block low-rank representations (WBBLRR) denoising method. WBBLRR uses 3-D wavelet transform to decompose HSI into different sub-images (approximation ciefficients sub-image and detail coefficients sub-images). Each sub-image can be considered as a block, where each block utilizes a low-rank representations model with different regularization parameters to obtain the denoised block.

- A new sparse and low-rank based HSI denoising method. This is the spectral-spatial transform-based sparse and low-rank representations (SSWSLRR) method. The proposed method uses both ℓ_1 penalty and weighted nuclear norm penalty in the transform domain to exploit the sparse and low-rank characteristics of HSI to remove noise.

- Several mixtures of factor analyzers (MFA) low-rank based methods are proposed for HSI denoising and feature extraction.

- Four new MFA based HSI denoising methods. MFA is a low-rank probabilistic method. HSIs usually have different classes with different characteristics and cannot be assumed to obey a Gaussian distribution. Thus, the Gaussian mixture model can be used to solve the non-normal distribution problem. MFA uses the Gaussian

mixture model to allow a low-rank representation of the Gaussians for different samples of HSI to remove noise. Furthermore, MFA is applied in the wavelet domain to remove noise is also analyzed. In addition, mixed noise is discussed using MFA with local spatial-spectral correlation of HSI.

- Six new unsupervised, supervised, and semi-supervised feature extraction methods. All the methods are proposed based MFA to exploit the low-rank representation for HSI. And the methods set different number of layers of MFA to construct the algorithm frameworks and use different samples to train the models. Unsupervised methods only use unlabeled samples to extract features of HSI. Supervised methods only utilize labeled samples to extract features of HSI. Semi-supervised methods use both unlabeled and labeled samples to extract features of HSI.

Ágrip

Hyperspectral myndir (HSI) sem fengnar eru af skynjun skynjara geisla innihalda hundruð litrómsbanda. Nægar litrófsupplýsingar sem gefnar eru af HSI gera það mögulegt að mismuna mismunandi efnum í senu. Þess vegna hafa HSI verið mikið notaðar á ýmsum sviðum, svo sem landbúnaðarframleiðsla, steinefnaleit og umhverfisvernd. HSI eru þó venjulega skemmdir af mismunandi hávaða eins og magnstuðli, hitauppstreymi og skothávaða, vegna áhrifa ljósáhrifa, frásogs andrúmslofts og truflunar skynjara. Þess vegna er afneitun mikilvægt forvinnsluskref fyrir HSI greiningu og forrit. Þessi ritgerð leggur til nokkrar afmörkunaraðferðir fyrir HSI til að bæta árangur fyrrnefndra umsókna. Helstu framlög ritgerðarinnar eru eftirfarandi.

- Tvær nýjar lágstéttar HSI afbótaaðferðir. Önnur er NLMLRA aðferðin sem er ekki staðbundin og notar slantaða Butterworth aðgerð til að búa til lágmarks nálgun fyrir rekstraraðila sem ekki er staðbundinn (NLM) og er á skilvirkan hátt útfærð á grundvelli Chebyshev margliða. Hitt er aðferðin sem byggir á bylgjulögun (low-rank representations) (WBBLRR) afneitunaraðferð. WBBLRR notar 3-D bylgjubreytingu til að brjóta niður HSI í mismunandi blokkir, þar sem hver blokk notar lág-stig framsetningarmódel til fá denoised blokkina, og notaðu síðan andhverfa 3-D bylgjubreytingu fyrir alla denoised blokkina til að fá denoised HSI.

- Ný dreifðar og lágstéttar HSI afneitunaraðferðir. Þetta er litrófs-staðbundna bylgjubundna dreifðar og lágstiga framsetning (SSWSLRR) aðferð. Fyrirhuguð aðferð notar 3-D stakan wavelet umbreytingu (3-D DWT) til að brjóta niður hávær HSI til dreifðra bylgjustuðla. Hávaðinn í HSI er niðurbrot í bylgjustuðla með dreifðri dreifingu lítilla amplituda. Til að fjarlægja hávaðann á áhrifaríkan hátt í wavelet léninu er lagt til veginn kjarnorku norm reglugerð, sem veitir góða nálgun fyrir raunverulega stöðu Og lögð er til ný strjál og lítil refsiaðferð sem byggð er á vegnu kjarnorkuviðmiði og ströngum viðurlögum, sem geta dregið saman aðlögunarhæfilega hvern bylgjustuðul, þ.e. stærri stuðlarnir skreppa minna saman en þeir minni munu minnka meira, þar sem stærri stuðlar bera meiri upplýsingar en minni.

- Lagðar eru til nokkrar blöndur af þáttargreiningartækjum (MFA) með lágu stigi aðferðir til að afmarka HSI og draga út eiginleika.

- Fjórar nýjar MFA byggðar HSI afbótaaðferðir. MFA, 2-D bylgjumiðað MFA (WMFA-2D) og 3-D bylgjumyndað MFA (WMFA-3D) er lagt til að fjarlægja Gaussískan hávaða. WMFA-2D og WMFA-3D nota MFA í wavelet domian til að fjarlægja hávaða er mælt með staðbundnum staðbundnum og litróf fylgni byggðri MFA (LSSC-MFA) til að fjarlægja blandaðan hávaða.

- Sex nýjar aðferðir til útdráttar án eftirlits, eftirlits og hálfgerðar eftirlits. MFA, djúpt MFA (DMFA), MFA (SMFA), DMFA (SDMFA), MFA (S^2MFA) , og hálfstýrð DMFA (S^2DMFA) lögun útdráttaraðferða er fyrirhuguð HSI flokkun. The MFA lengir

Gaussian blöndu líkanið til að leyfa lágvíddar framsetningu Gaussians. DMFA er djúp útgáfa af MFA og samanstendur af tveggja laga MFA, þ.e. sýnum úr aftari dreifingu við fyrsta lagið er innflutt í MFA líkan í öðru laginu. SMFA samanstendur af eins lags MFA og SDMFA er tveggja laga SMFA. SMFA og SDMFA nýta merktar upplýsingar til að draga úr eiginleikum HSI á áhrifaríkan hátt. S^2MFA og S^2DMFA nýta bæði merktar og ómerktar upplýsingar til að draga út eiginleika HSI.

Acknowledgments

I am so very grateful to my supervisors Prof. Johannes R. Sveinsson and Prof. Magnus O. Ulfarsson for their guidance, support, and assistance for my research and life. It is my great honor and luck to work with them. I am also grateful to my co-supervisor Prof. Jocelyn Chanussot for recommending me the PhD position at the University of Iceland and being part of my committee. His guidance and encouragement gave me great help during PhD study.

I am also deeply thank Prof. Javier Plaza Miguel and Prof. Björn Waske to be my PhD opponents. Thank you very much for your excellent comments and observations.

I would like to express my gratitude to my friends and colleagues: Jakob Sigurdsson, Burkni Palsson, Han V. Nguyen, Hans Atlason, Frosti Palsson, Lotta Ellingsen, Weiwei Song, Fanghong Xiao, Magnus Magnusson, Hilda Deborah, Sveinn Eirikur Armannsson for their friendship, support, company, and help during these years.

I am also very thankful to the Doctoral Grants of the University of Iceland Research Fund and Icelandic Research Fund under Grant 174075-05 for supporting fund for my research work.

Last but not least, I would like to thank my family for their constant support and believe me all the time.

Contents

List of Figures

List of Tables

Abbreviations

AA	Average Accuracy
ADMM	Alternating Direction Method of Multipliers
CA	Class Accuracy
DMFA	Deep MFA
DR	Dimensionality Reduction
DWT	Discrete Wavelet Transform
FA	Factor Analysis
FE	Feature Extraction
HSI	Hyperspectral Image
KC	Kappa Coefficient
MFA	Mixtures of Factor Analyzer
MFSIM	Mean Feature Similarity
MSSIM	Mean Structural Similarity
OA	Overall Accuracy
PSNR	Peak Signal-to-Noise Ratio
S^{2}DMFA	Semi-Supervised DMFA
S^{2}FE	Semi-Supervised Feature Extraction
S^{2}MFA	Semi-Supervised MFA
SAM	Spectral Angle Mapper
SDMFA	Supervised DMFA
SFE	Supervised Feature Extraction
SMFA	Supervised MFA
SVM	Support Vector Machine
UFE	Unsupervised Feature Extraction

Notations

$\mathbf{X}$	Matrix
$\mathbf{X}^\mathsf{T}$	Transpose of matrix $\mathbf{X}$
$\mathbf{x}_{(j)}$	jth column of matrix $\mathbf{X}$
$\mathbf{x}_i^\mathsf{T}$	ith row of matrix $\mathbf{X}$
x_{ij}	ith row and jth column element of matrix $\mathbf{X}$
x_i	ith element of vector $\mathbf{x}$
$\mathrm{tr}(\mathbf{X})$	Trace of $\mathbf{X}$
$\mathrm{diag}(\mathbf{X})$	Diagonal matrix with $\mathbf{X}$ on its main diagonal, other values are zero
$\mathrm{vec}(\mathbf{X})$	vectorization of $\mathbf{X}$ where the columns are stacked on top of one another
$\|\cdot\|_F^2$	Frobenius norm
$\|\cdot\|_*$	Nuclear norm
$\|\cdot\|_{\omega,*}$	Weighted nuclear norm

Chapter 1

Introduction

This chapter begins with introducing hyperspectral images (HSI). The concept of hyperspectral denoising is explained and main methods used in denoising are reviewed. Feature extraction and its relationship to hyperspectral denoising is also discussed. The chapter concludes with the goals and novelties of the work presented here, along with an overview of the thesis.

1.1 Hyperspectral Images

Remote sensing is the process of detecting and monitoring the material characteristics of a scene by measuring its reflected and emitted radiation from satellite or aircraft sensors.

Hyperspectral remote sensing image is collected by hyperspectral sensors. Hyperspectral sensors can measure reflected light within a scene in hundreds of narrow spectral bands from across the electromagnetic spectrum. Thus, Hyperspectral image (HSI) contains contiguous hundreds of spectral bands and usually covers the visible to the near infrared spectral region. The abundant spectral information provided by an HSI makes it possible to discriminate different materials in a scene. Therefore, HSIs have been widely used in various fields, such as agricultural production, mineral exploration, and environmental protection.

Hyperspectral dataset is often presented as a 3-D data cube in which the first and the second dimension are the spatial dimensions and the third dimension is the spectral dimension. One such data cube is shown in Fig. 1.1. The data is Indian Pines hyperspectral dataset and is collected by the Airborne Visible/Infrared Imaging Spectrometer at Indian Pines. The image contains 145 × 145 pixels with spatial resolution of 20m, and 220 spectral bands from 400 nm to 2500 nm with spectral resolution of 9.15 nm.

HSIs are usually corrupted by different noises such as quantization noise, thermal noise, and shot noise, due to the influence of photon effects, atmospheric absorption, and sensor disturbance. Quantization and thermal noise are typically modeled as signal independent Gaussian additive noise, while shot noise can be modeled as variance dependent additive noise [1–3].

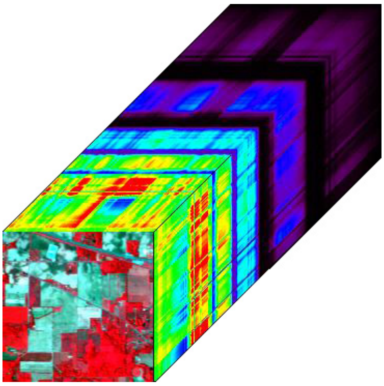


Figure 1.1: Indian Pines dataset

1.2 Hyperspectral image Denoising

HSI denoising has considerably evolved in recent years [4–11] and can be divided into model-based and deep learning-based denoising methods. This dissertation only focus on the model-based denoising methods.

1.2.1 2-D Band-Wise Based Methods

By treating each spectral band of an HSI as a separate grayscale image 2-D denoising methods can be extended to HSI denoising. Examples of those methods are the non-local means (NLM) filter [12], the 2-D wavelet denoising [13], the block-matching and 3-D filter (BM3D) [14], and the weighted nuclear norm minimization (WNNM) [15]. Single-band methods utilize only spatial information leading to spectral distortion.

1.2.2 Spectral-Spatial Based Methods

HSIs are spectrally correlated. Therefore, denoising methods should jointly exploit spatial and spectral information [16]. Several spectral-spatial denoising methods have been proposed [17]. Yuan *et al.* [18] suggested a spectral-spatial adaptive total variation denoising method for removing Gaussian noise. Chen *et al.* [19] proposed a denoising method based on spectral-spatial domain mixing prior. Huang *et al.* [20] used a spatial-spectral weighted nuclear minimization denoising method for removing mixed types of noise in HSI. Besides all the methods above, there are other methods based on transforming HSI to the wavelet domain and utilize wavelet-based spatial and spectral information to remove the noise [21]. Chen *et al.* [22] used principal component analysis (PCA) of the HSI and then transformed the low-energy PCA output channels to wavelet domain to remove the noise using 2-D spatial and 1-D spectral wavelet shrinkage denoising. Othman *et al.* [23] suggested a hybrid of spatial and spectral wavelet shrinkage by using a spectral derivative to elevate noise and then utilizes wavelet-based spatial and spectral denoising. Rasti *et al.* [24] presented a wavelet domain-based spectral-spatial penalty method for HSI denoising. These wavelet-based spectral-spatial methods use spatial and spectral information of HSI to remove noise and utilize wavelet transform to provide a sparse representation for clean HSI to further remove noise in the wavelet domain.

1.2.3 Low-Rank Based Methods

Recently, many low-rank model-based methods have been proposed for HSI denoising [25–31]. Zhang *et al.* [32] presented a low-rank matrix recovery (LRMR) method for removing mixed types of noise in HSI. An extension of the LRMR method [33] was proposed using an adaptive iteration regularization framework for HSI denoising. Chen *et al.* [34] presented a nonconvex low-rank matrix approximation method using a nonconvex regularizer. These low-rank model-based methods exploit the low-dimensional structures in the HSIs. HSIs are usually self-similarity and have sparse characteristics in the spatial domain and hence can be denoised using sparse representation. Lu *et al.* [35], and Li *et al.* [36] suggested a HSI denoising method based on sparse representation using the high similar spatial information of HSI. More recently, several denoising methods based on a combination of low-rank and sparse representation have been presented [37]. Zhuang *et al.* [38] utilized similar patches in the low-dimensional subspace to exploit the sparse and low-rank characteristics of HSI. Rasti *et al.* [39] used sparse low-rank model, which utilizes sparse penalized least squares for estimating the unknown signal for denoising. Tensor-based denoising methods given in [40–45] exploited the low-rank and the sparse characteristic of similar tensors to remove noise.

1.3 Feature Extraction

HSIs provide abundant spectral information about a scene. In general, an HSI contains hundreds of spectral bands with high spectral resolution. However, the high dimensionality of hyperspectral images (HSIs) makes the processing computationally and memory costly, and the reliability of supervised classifiers and classification accuracy can not be ensured without enough number of training samples [46]. As a result, reducing the spectral dimensionality is of great interest in HSI analysis [47]. Dimensionality reduction (DR) is a pre-processing method that decreases the dimensionality of HSI extracts information for HSI post-processing and can be divided into feature selection (FS) and feature extraction (FE) [17]. In FS, a subset of spectral bands is selected from the original set of spectral bands. FE transforms the HSI into a new feature space and in that space extracts features. This dissertation only focus on the FE methods.

1.3.1 Unsupervised Feature Extraction

FE has been widely used in HSI processing [48]. FE can be divided into three parts, unsupervised FE (UFE), supervised FE (SFE), and semi-supervised FE (S^2FE). The UFE methods use unlabeled samples, while SFE methods require labeled samples to extract features. S^2FE methods use both labeled and unlabeled samples to extraction features. A classical UFE method is principal component analysis (PCA), which searches for a projection to maximize the signal variance [49]. Recently, some probabilistic methods were proposed to reduce the dimensionality of HSI, such as factor analysis (FA) [50]. FA is a linear Gaussian latent variable model, in which all of the marginal and conditional distributions are assumed to be Gaussian. Probabilistic principal component analysis (PPCA) [49], and mixtures of factor analyzers (MFA) [51, 52]. In addition, there are several UFE methods in the literature, such as multiscale quaternion Weber local descriptor histogram (MQWLDH) [53], orthogonal total variation component analysis (OTVCA) [54], sparse and smooth low-rank analysis (SSLRA) [47], and segmented stacked autoencoder (S-SAE) [55].

1.3.2 Supervised Feature Extraction

SFE methods use labeled samples to improve class separability. Linear discriminant analysis (LDA) [56] and nonparametric weighted feature extraction (NWFE) [57] are examples of widely used SFE methods for HSI processing. LDA maximizes the separation between different classes to extract features. NWFE uses the nonparametric extension of scatter matrices to extract features. Extension to these methods are, for example, modified Fishers LDA (MFLDA) [58], regularized LDA [59], and kernel NWFE [60]. There are other SFE methods used for HSI processing that use, for example, manifold-learning based HSI feature extraction [61], and low-rank representation with the ability to preserve the local pairwise constraints information (LRLPC) [62].

1.3.3 Semi-supervised Feature Extraction

It is labor-intensive and time-consuming to collect a large number of labeled samples, so in real applications, the number of labeled samples is usually limited [63, 64]. In contrast, it is easier to obtain unlabeled samples. Therefore, S^2FE methods which utilize both labeled and unlabeled samples are widely employed for HSI processing [65]. Examples of S^2FE methods are semi-supervised local Fisher discriminant analysis (SELF) [66], semi-supervised local discriminant analysis (SELD) [67], and Graph-based semi-supervised FE methods [68].

1.4 Thesis Contributions and Organization

In this thesis, there are four major contributions. First, we propose two new low-rank based denoising methods for HSI. One is the non-local means low-rank approximation (NLMLRA) method. The other is the wavelet-based block low-rank representations (WBBLRR) denoising method. NLMLRA uses low-rank approximation for non-local means operator in the original domain. WBBLRR utilizes a low-rank representations model for each block in the wavelet domain.

Another contribution of the thesis is that we propose a denoising method based on sparse spectral-spatial and low-rank representations (SSSLRR) using 3-D orthogonal transform (3-DOT). SSSLRR can be effectively used to remove Gaussian and mixed noise. SSSLRR uses 3-DOT to decompose noisy HSI to sparse transform coefficients. 3-D discrete orthogonal wavelet transform (3-D DWT) is a representative 3-DOT suitable for denoising since it concentrates the signal in few transform coefficients, the 3-D discrete orthogonal cosine transform (3-D DCT) is another example. An SSSLRR using 3-D DWT will be called SSSLRR-DWT. SSSLRR-DWT is an iterative algorithm based on the alternating direction method multipliers (ADMM) that uses sparse and nuclear norm penalties. We use an ablation study to show the effectiveness of the penalties we employ in the method. Both simulated and real hyperspectral datasets demonstrate that SSSLRR outperforms other comparative methods in quantitative and visual assessments to remove Gaussian and mixed noise.

The third major contributions of the thesis are that we propose several mixtures of factor analyzers (MFA) low-rank based methods for HSI denoising. MFA uses a Gaussian mixture model to segment the original HSI into different parts, where each part follows Gaussian distribution and then utilizes a factor analyzer to get a low-rank factor loading matrix, and finally uses the inverse transformation of the matrix to get the denoised hyperspectral dataset. Also, we investigate the performance of MFA in the wavelet domain for HSI denoising and analyze the denoising result of MFA combining with local spatial-spectral correlation model for removing mixed noise.

The fourth major contributions of the thesis are that we propose six new unsupervised, supervised, and semi-supervised MFA-based feature extraction (FE) methods for HSI classification. There are MFA, deep MFA (DMFA), supervised MFA (SMFA), supervised DMFA (SDMFA), semi-supervised MFA (S^{2}MFA), and semi-supervised DMFA (S^{2}DMFA) FE methods for HSI. We also propose an image segmentation method based on the Gaussian mixture model for these FE methods to solve the problem of a non-normal distribution. MFA assumes a low-dimensionality representation of the Gaussians in the Gaussian mixture model. DMFA consists of a two-layer MFA, which inputs the samples from the posterior distribution at the first layer to an MFA model at the second layer. SMFA and SDMFA are supervised FE methods that use labeled samples to extract features of HSI. S^{2}MFA and S^{2}DMFA exploit both labeled and unlabeled information to extract features of HSI. Based on these three FE methods, a framework for HSI classification is also proposed. While the dimensionality of the desired features needs to be selected by the user in conventional DR methods, the proposed framework automatically determines the dimensionality of features according to classification accuracy without prior supervision by the user.

In Chapter 2, two low-rank based methods is proposed for HSI denoising. There are NLMLRA and WBBLRR denoising methods. LNLLRA is an extension of the idea of [69], which focuses on grayscale images. The distinction of our proposed method is that we use full-band pixel-based patches to calculate the weighting function to take full advantage of the high spatial and spectral correlation and constructs a low-rank approximation function for denoising operator to improve the NLM denoised performance rather than only using spatial information alone as in [69]. To improve the practicability and reduce computational cost, Chebyshev polynomials are used in the NLMLRA. WBBLRR mainly includes three steps for HSI denoising. First, the noisy HSI is decomposed to different sub-images blocks in wavelet domain using 3-D wavelet decomposition. Second, each block uses low-rank representations model to obtain the denoised block. Third, the denoising HSI is obtained using inverse 3-D wavelet transformation for all the denoised blocks.

In Chapter 3, a sparse and low-rank based method (SSSLRR) is proposed using 3-D orthogonal transform (3-DOT) for HSI denoising to remove Gaussian and mixed noise. The thesis used both 3-D discrete orthogonal wavelet transform (3-D DWT) and 3-D discrete orthogonal cosine transform (3-D DCT) as representative of 3-DOT. The 3-D DWT is exploited in SSSLRR-DWT to provide a sparse representation for HSI by jointly using the correlated spatial and spectral information. The idea is that the wavelet transform concentrates the signal energy in few coefficients while the noise has energy spread over all coefficients [70, 71]. A new sparse and low-rank penalized model is proposed for removing noise in the wavelet domian. The method uses the ℓ_1 penalty and the weighted nuclear norm low-rank penalty to adaptively shrink the wavelet coefficients and penalize the singular values to remove the noise.

An algorithm based on the alternating direction method of multipliers (ADMM) is developed to get the optimal parameters and denoised results for SSSLRR-DWT. To analyze the effectiveness of different penalties for SSSLRR-DWT both in signal and wavelet domain, four different methods were analyzed. The analysis shows that SSSLRR-DWT uses both ℓ_1 penalty and weighted nuclear norm low-rank penalty in the wavelet domain yields better denoising results than the other methods. Moreover, a new SSSLRR-DCT denoising method was also proposed, which uses the 3-D discrete orthogonal cosine transform (3-D DCT) for SSSLRR. The analysis shows that both SSSLRR-DWT and SSSLRR-DCT have good denoising results. The experimental results indicate that our proposed method improves the denoising performance for both simulated and real noisy HSI datasets.

In Chapter 4, several MFA-based low-rank methods are proposed for HSI denoising. Three new MFA, 2-D wavelet based MFA (WMFA-2D), and 3-D wavelet based MFA (WMFA-3D) are proposed for removing Gaussian noise. MFA utilizes Gaussian mixture model to segment the original HSI to different parts, where each part follows a Gaussian distribution and then utilizes a factor analyzer to get a low-rank factor loading matrix, and finally uses the inverse transformation of the low-rank factor loading matrix to get the hyperspectral data without noise. WMFA-2D and WMFA-3D use the MFA in the wavelet domain. MFA is particularly suitable for denoising of HSI with a high level of noise. While WMFA-2D and WMFA-3D can be effectively used to denoising of HSI with a low level of noise. One new local spatial-spectral correlation based automatic MFA (LSSC-AMFA) is proposed for removing mixed noise. LSSC-

AMFA utilizes a novel local spatial-spectral correlation (LSSC) method to remove the missing lines noise. Then LSSC-AMFA uses MFA to remove the missing pixels, salt and pepper noise, and Gaussian noises.

In Chapter 5, six new unsupervised, supervised, and semi-supervised MFA-based methods are proposed for HSI feature extraction and are then used for classification of them. There are MFA, deep MFA (DMFA), supervised MFA (SMFA), supervised DMFA (SDMFA), semi-supervised MFA (S^{2}MFA), and semi-supervised DMFA (S^{2}DMFA) feature extraction methods. All of them are probabilistic dimensionality reduction (DR) methods, instead of assuming that a whole HSI obeys a Gaussian distribution, the methods use a Gaussian mixture model to extract more effective information for DR. The Gaussian mixture model is used for MFA to allow a low-dimensionality representation of the Gaussian. A two-layer MFA, DMFA, utilizes the samples from the posterior at the first layer to an MFA model at the second layer. MFA and DMFA are two unsupervised DR method. The methods are particularly suitable for DR of HSI with a non-normal distribution and unlabeled samples. SMFA and SDMFA are two supervised DR methods and use labeled samples to extract features. SDMFA is a deep version of SMFA and consists of a two-layer SMFA. SMFA and SDMFA can be effectively used to DR of HSI with a non-normal distribution and labeled samples. S^{2}MFA and S^{2}DMFA are two semi-supervised DR methods, which simultaneously consider labeled and unlabeled samples to extract features. S^{2}MFA uses a Gaussian mixture model to segment image to different parts, each part follows a Gaussian distribution and contains many labeled and unlabeled samples and uses a factor analyzer to get a factor loading matrix. This matrix uses labeled samples to improve the class discrimination and employs both labeled and unlabeled samples to preserve the local spatial features of the data and then is used for transforming the original HSI to an optimal low-dimensional subspace to achieve DR. S^{2}DMFA is a two-layer S^{2}MFA. S^{2}MFA and S^{2}DMFA are particularly suitable for DR of HSI with a complicated probability distribution and labeled and unlabeled samples. Based on the six DR methods, we also proposed a framework for HSI classification, the overall accuracy of a classifier on validation samples is used to automatically determine the optimal number of features of DR for HSI classification. This framework can automatically extract the most effective feature for HSI classification. To validate the performance of DR, we conduct experiments in terms of SVM classification based on real HSIs. The experimental results show that our proposed methods can give better results than statistical DR comparison methods.

Finally, in Chapter 6, conclusions are summarized and future work is detailed.

1.5 Publications

Chapter 2 is based on the following publications:

[a] Bin Zhao, Johannes R. Sveinsson, Magnus O. Ulfarsson, Jocelyn Chanussot, "Non-local means low-rank approximation for hyperspectral denoising", accepted

by IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 4147-4150, 2021.

[b] Bin Zhao, Johannes R. Sveinsson, Magnus O. Ulfarsson, Jocelyn Chanussot, "Wavelet-based block low-rank representations for hyperspectral denoising", accepted by IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 2484-2487, 2021.

Chapter 3 is based on the following publications:

[c] Bin Zhao, Johannes R. Sveinsson, Magnus O. Ulfarsson, Jocelyn Chanussot, "Hyperspectral image denoising using spectral-spatial transform-based sparse and low-rank representations", submitted to IEEE Transactions on Geoscience and Remote Sensing, 2021.

Chapter 4 is based on the following publications:

[d] Bin Zhao, Johannes R. Sveinsson, Magnus O. Ulfarsson, Jocelyn Chanussot, "Local spatial-spectral correlation based mixtures of factor analyzers for hyperspectral denoising", IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1488-1491, 2020.

[e] Bin Zhao, Johannes R. Sveinsson, Magnus O. Ulfarsson, Jocelyn Chanussot, "Hyperspectral images denoising based on mixtures of factor analyzers", IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1516-1519, 2020.

Chapter 5 is based on the following publications:

[f] Bin Zhao, Johannes R. Sveinsson, Magnus O. Ulfarsson, Jocelyn Chanussot, "Semi-supervised mixtures of factor analyzers feature extraction for hyperspectral images", IEEE Geoscience and Remote Sensing Letters, pp. 1–5, 2020.

[g] Bin Zhao, Johannes R. Sveinsson, Magnus O. Ulfarsson, Jocelyn Chanussot, "Unsupervised and supervised feature extraction methods for hyperspectral images based on mixtures of factor analyzers", Remote Sensing, vol. 12, no. 7, pp. 1179, 2020.

[h] Bin Zhao, Johannes R. Sveinsson, Magnus O. Ulfarsson, Jocelyn Chanussot, "Mixtures of factor analyzers and deep mixtures of factor analyzers dimensionality reduction algorithms for hyperspectral images classification", IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 891-894, 2019.

[i] Bin Zhao, Johannes R. Sveinsson, Magnus O. Ulfarsson, Jocelyn Chanussot, "(Semi-) supervised mixtures of factor analyzers and deep mixtures of factor analyzers dimensionality reduction algorithms for hyperspectral images classification", IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 887-890, 2019.

Chapter 2

Low-Rank Based Methods

In this chapter, two low-rank based methods are proposed for HSI denoising. The two methods are non-local means low-rank approximation (NLMLRA) and wavelet-based block low-rank representations(WBBLRR) denoising methods. NLMLRA uses the patch-based similarity weighting function to construct the non-local means (NLM) denoising operator and utilizes Chebyshev polynomials-based low-rank approximation to improve the denoising performance. WBBLRR uses 3-D wavelet transformation to decompose HSI into different blocks, where each block utilizes a low-rank representations model to obtain the denoised block, and then uses inverse 3-D wavelet transformation for all the denoised blocks to obtain the denoised HSI.

2.1 Non-Local Means Low-Rank Approximation

2.1.1 Introduction

NLM is a single-band image denoising method, which uses the spatial self-similarity of image to denoise. NLM utilizes the patch-based weighted function to get the denoised results. The denoising operator of NLM is constructed from patches of the corrupted noisy image. Thus, the denoising operator is affected by noise. NLMLRA is an extension of the idea of [69], which focuses on grayscale images. The distinction of our proposed method is that we use full-band pixel-based patches to calculate the weighting function to take full advantage of the high spatial and spectral correlation and constructs a low-rank approximation function for denoising operator to improve the NLM denoised performance rather than only using spatial information alone as in [69]. To improve the practicability and reduce computational cost, Chebyshev polynomials are used in the NLMLRA.

2.1.2 Problem Formulation

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^T \in \mathbb{R}^{n \times B}$ denote a clean HSI with n pixels and B spectral bands. Here, $\mathbf{x}_i \in \mathbb{R}^B$ is the ith pixel. The additive noise is assumed in hyperspectral denoising problem, the observational model can be described as

$$\mathbf{Y} = \mathbf{X} + \mathbf{N}, \tag{2.1}$$

where $\mathbf{Y}$, $\mathbf{N} \in \mathbb{R}^{n \times B}$ represent the observed noisy HSI and noise, respectively. Therefore, HSI denoising problem is to find an estimate $\widehat{\mathbf{X}} \in \mathbb{R}^{n \times B}$ of $\mathbf{X}$ from $\mathbf{Y}$.

Non-local means (NLM) is a classical image denoising method and can be expressed as

$$\widehat{\mathbf{x}}_i = \frac{\sum_{j=1}^{m} w_{i,j}\mathbf{y}_j}{\sum_{j=1}^{m} w_{i,j}}, \tag{2.2}$$

where m is the number of similar patches, $w_{i,j}$ is a patch-based weighting function, i.e.,

$$w_{i,j} = \exp(-\frac{\parallel \mathcal{Y}_j - \mathcal{Y}_i \parallel_2^2}{2h^2}), \tag{2.3}$$

where $\mathcal{Y}_i \in \mathbb{R}^{p\times B}$ is a patch and is centered around $\mathbf{y}_i$, p is the size of the patch, $h > 0$ is filtering parameter. In matrix notation NLM is

$$\widehat{\mathbf{X}} = \mathbf{AY}, \tag{2.4}$$

where $\mathbf{A} = \mathbf{D}^{-1}\mathbf{W}$ is an NLM operator of $\mathbf{Y}$, $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix given by $\mathbf{D}_{ii} = \sum_{j=1}^{m} w_{i,j}$, and $w_{i,j}$ are the elements of $\mathbf{W}$.

Image $\mathbf{Y}$ is noisy, $\mathbf{A}$ is constructed from noisy data, and is thus noisy as well [72]. We would like to replace $\mathbf{A}$ with an operator which is less noisy. In particular, we look for replacing $\mathbf{A}$ with its low-rank approximation (LRA). A simple way to construct the LRA is to use truncated singular-value decomposition (SVD). However, directly calculating the SVD of $\mathbf{A}$ is computationally expensive and often impractical. The Slanted Butterworth (SB) function [69] is a filter function, that can suppress eigenvalues with small magnitude when applied to a matrix and accurately approximate the remaining low-rank operator, while preserving the fundamental properties of the original operator. Therefore, we use SB function to construct an LRA for NLM operator.

Constructing LRA for the NLM operator based on Chebyshev polynomials

The SB function can be expressed as

$$f_{\omega,d}^{sb}(x) = x(1 + (\frac{1-x}{1-\omega})^{2d})^{-\frac{1}{2}}, \quad x \in [0,1] \tag{2.5}$$

where $0 \leq \omega \leq 1$ is filter cutoff, $d \in \mathbb{N}$ is the order of the filter. The SB function based on NLM operator $\mathbf{A}$ becomes

$$f_{\omega,d}^{sb}(\mathbf{A}) = \mathbf{A}(\mathbf{I}_n + (\frac{\mathbf{I}_n - \mathbf{A}}{1-\omega})^{2d})^{-\frac{1}{2}}, \tag{2.6}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix. The direct evaluation of the matrix function $f_{\omega,d}^{sb}(\mathbf{A})$ for a large matrix $\mathbf{A}$ is infeasible [69]. Here, we use Chebyshev polynomials to evaluate the $f_{\omega,d}^{sb}(\mathbf{A})$ function.

The Chebyshev polynomials of the first kind of degree k are given by

$$T_k(x) = \cos(k \arccos(x)), \; x \in [-1,1], \; k = 0,1,2,... \tag{2.7}$$

The Chebyshev expansion is

$$f(x) = \sum_{j=0}^{\infty} \alpha_j T_j(x) \tag{2.8}$$

$$\alpha_0 = \frac{1}{2}\langle f, T_0\rangle_T,\ \alpha_n = \langle f, T_n\rangle_T,\ n \in \mathbb{N}, \tag{2.9}$$

where α_i is Chebychev coefficient for $f(x)$, $\langle f, T_j\rangle_T = \frac{1}{N+1}\sum_{k=1}^{N+1} f(x_k)T_j(x_k),\ j = 0,...,N$, and $x_k = \cos(\frac{\pi(k-\frac{1}{2})}{N})$.

We evaluate $f^{sb}_{\omega,d} : [0,1] \to \mathbb{R}$ by truncating (2.8), i.e.,

$$f^{sb}_{\omega,d}(z) \approx \sum_{j=0}^{N} \alpha_j T_j(y), \tag{2.10}$$

where N is the length of the truncation and is set to 50 as a default value, and $y = 2z - 1$ maps $f^{sb}_{\omega,d}$ from [0,1] to [-1,1]. Thus $f^{sb}_{\omega,d}(\mathbf{A})$ is approximated by

$$f^{sb}_{\omega,d}(\mathbf{A}) \approx S_N(f^{sb}_{\omega,d}, \mathbf{A}) = \sum_{j=0}^{N} \alpha_j T_j(2\mathbf{A} - \mathbf{I}_n). \tag{2.11}$$

Finally, the clean image of $\mathbf{Y}$ is estimated as

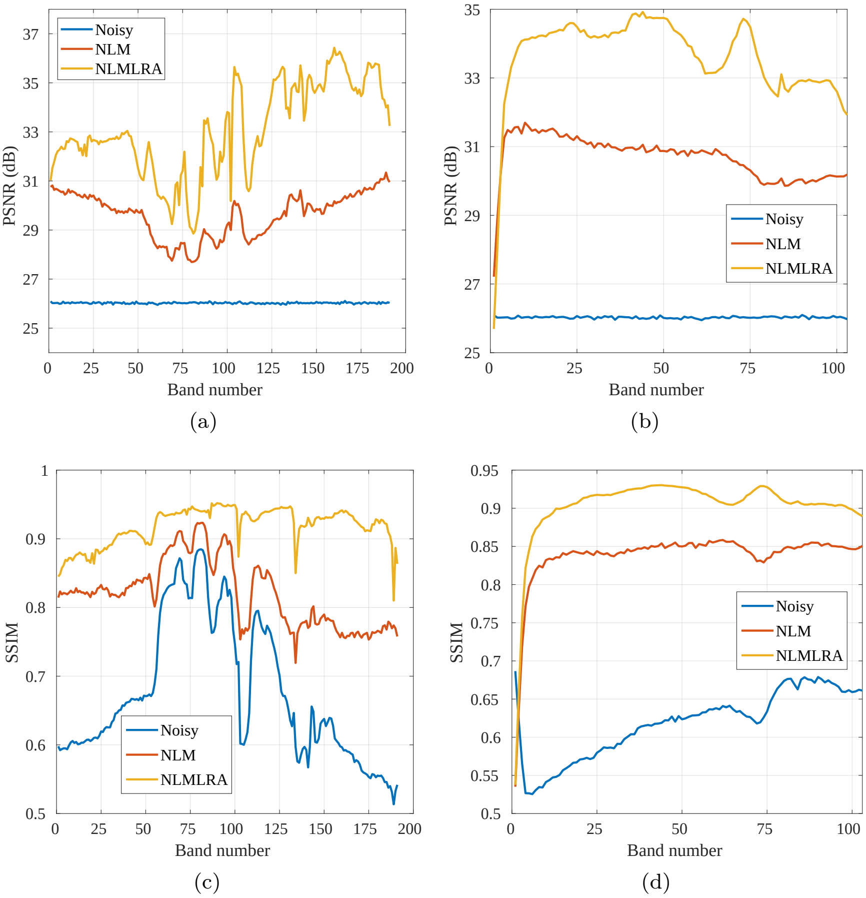
$$\widehat{\mathbf{X}} = S_N(f^{sb}_{\omega,d}, \mathbf{A})\mathbf{Y}. \tag{2.12}$$

2.1.3 Experiments and Results

In the following experiments, we use two simulated and one real noisy HSI datasets. The simulated datasets are Washington DC Mall (WDCM) dataset (Appendix B.1) and University of Pavia (UP) dataset (Appendix B.2). In the experiments, two small parts of the WDCM and UP datasets are used. Their sizes are 200 × 200 × 191 for the WDCM dataset and 200 × 200 × 103 for the UP dataset. Both datasets are assumed to be noise-free HSIs. The datasets are normalized band by band between [0,1] before adding noise. The Gaussian noise with zero-mean and standard deviation σ was added to the clean WDCM and UP datasets. In the experiments, we set σ to two different values 0.05, and 0.1.

To evaluate the denoised results of the simulated datasets, the quantitative metrics are peak signal-to-noise ratio (PSNR) (Appendix A.1) and mean structural similarity (MSSIM) (Appendix A.2) are used.

Here, we evaluate the performance of NLMLRA and NLM for both quantitative metrics and visualization of the denoised WDCM and UP HSIs. The PSNR and SSIM of each band are calculated and depicted in Fig. 2.1 for quantitative assessment. Table 4.1



(a)

(b)

(c)

(d)

Figure 2.1: PSNR (a)-(b), and SSIM (c)-(d) values of each band of denoised WDCM and UP datasets ($\sigma = 0.05$).

Table 2.1: The quantitative evaluation results (PSNR(dB), MSSIM, and Time (second)) for the different denoising methods for the Washington DC Mall (WDCM) and the University of Pavia (UP) datasets. The best results are in bold typeface.

Dataset	Noisy level	Metric	Noisy	NLM	NLMLRA
WDCM	$\sigma = 0.05$	PSNR	26.02	29.52	**32.83**
		MSSIM	0.6750	0.8198	**0.9069**
		Time	-	11.24	**9.07**
	$\sigma = 0.1$	PSNR	20.01	25.79	**28.56**
		MSSIM	0.4081	0.6549	**0.7793**
		Time	-	11.96	**9.16**
UP	$\sigma = 0.05$	PSNR	26.03	30.65	**33.69**
		MSSIM	0.6184	0.8381	**0.9060**
		Time	-	5.59	**4.60**
	$\sigma = 0.1$	PSNR	20.01	26.69	**29.56**
		MSSIM	0.3427	0.6828	**0.8036**
		Time	-	5.61	**4.61**

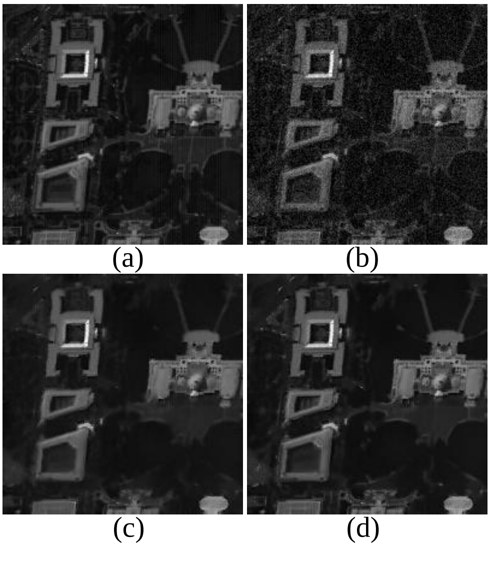


Figure 2.2: The results for the different denoising methods for the WDCM dataset. (a) Original band 1, (b) Noisy band 1, (c) NLM, and (d) NLMLRA.

gives the denoising results in terms of PSNR, MSSIM, and Time for both the WDCM and UP datasets. From the figure and table, it can be seen that the obtained PSNR, SSIM, and MSSIM of NLMLRA are higher than for NLM method. The denoising methods are implemented using MATLAB R2019a on a computer having Intel Core i7-67000 processor (3.40 GHz), 8.00 GB of memory and 64-bit Windows 10 Operating System. The running times for NLMLRA are faster than NLM.

We choose $\sigma = 0.05$ for both the WDCM and UP datasets to demonstrate visual quality. Figs. 2.2 and 2.3 show the denoised results for band 1 of the WDCM dataset and band 32 of the UP dataset, respectively. NLM cannot completely remove the noise as there is still noise in the denoised images. NLMLRA works well in filtering out the noise and provides cleaner denoised images.

For the experiments based on the real dataset, the Indian Pines dataset (Appendix B.3) is used. Fig. 2.4 presents the denoised results for bands 2, and 105. It can be clearly observed that NLM gives oversmooth denoised images. The proposed NLMLRA method gives better denoised images.

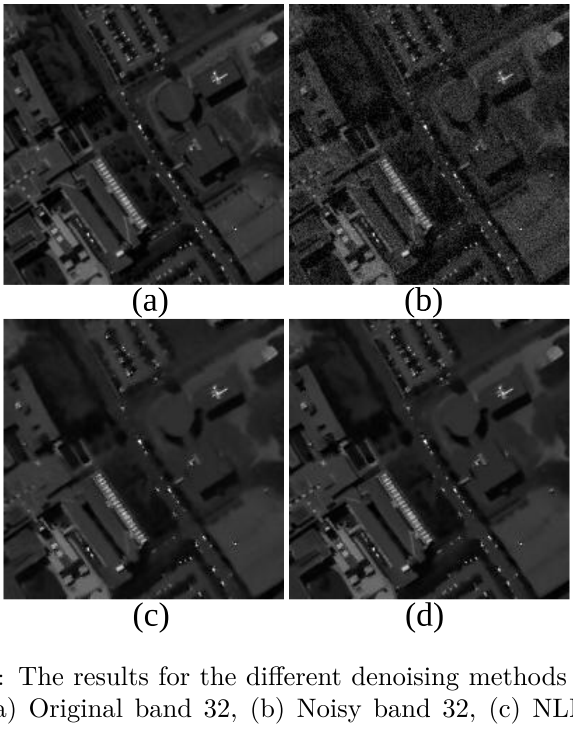


Figure 2.3: The results for the different denoising methods for the UP dataset. (a) Original band 32, (b) Noisy band 32, (c) NLM, and (d) NLMLRA.

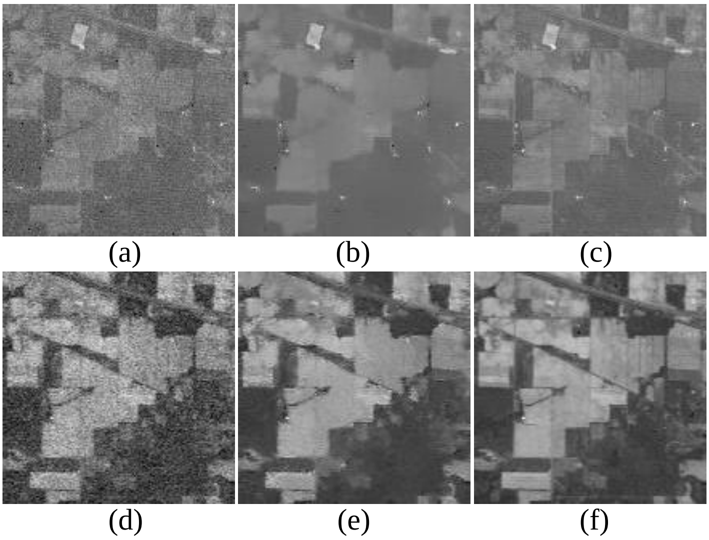


Figure 2.4: The results for the different denoising methods for the Indian Pines dataset. (a) Original band 2, (b) NLM, (c) NLMLRA, (d) Original band 105, (e) NLM, (f) NLMLRA.

2.2 Wavelet-Based Block Low-Rank Representations

2.2.1 Introduction

HSI can be represented by low-rank approximation owing to the high correlation in the spectral domain. Using this characteristic, singular value thresholding (SVT) and block SVT (BSVT) denoising methods were proposed in [73]. SVT uses singular value decomposition (SVD) for noisy HSI and then utilizes soft thresholding for singular value and finally uses inverse SVD for a low-rank matrix to obtain the denoised HSI. BSVT is an extended denoising method of SVT that segments HSI to different blocks and then uses SVT for each block. In this chapter, we propose the WBBLRR denoising method for HSIs. WBBLRR mainly includes three steps for HSI denoising. First, the noisy HSI is decomposed to different sub-images blocks in wavelet domain using 3-D wavelet decomposition. Second, each block uses low-rank representations model to obtain the denoised block. Third, the denoising HSI is obtained using inverse 3-D wavelet transformation for all the denoised blocks.

2.2.2 Problem Formulation

Let $\mathbf{X} \in \mathbb{R}^{n \times B}$ denote a clean HSI with n pixels and B spectral bands. The additive noise is assumed in hyperspectral denoising problem, the observational model can be described as

$$\mathbf{Y} = \mathbf{X} + \mathbf{N}, \tag{2.13}$$

where $\mathbf{Y}$, $\mathbf{N} \in \mathbb{R}^{n \times B}$ represent the observed noisy HSI and noise, respectively. The HSI denoising problem is to find an estimate $\widehat{\mathbf{X}} \in \mathbb{R}^{n \times B}$ of $\mathbf{X}$ from $\mathbf{Y}$. Using 3-D discrete wavelet transform (3-D DWT) decomposition, the model can be expressed as

$$\mathbf{Y} = \mathbf{D}\mathbf{W} + \mathbf{N}, \tag{2.14}$$

where $\mathbf{D} \in \mathbb{R}^{n \times n}$ is the 3-D orthogonal wavelet basis, $\mathbf{W} \in \mathbb{R}^{n \times B}$ is a wavelet coefficients matrix. To enforce the low-rank, nuclear norm penalty is used on the wavelet coefficients $\mathbf{W}$. Here, the wavelet-based low-rank denoising problem can be expressed as

$$\widehat{\mathbf{W}} = \arg\min_{\mathbf{W}} \quad \|\mathbf{H} - \mathbf{W}\|_F^2 + \lambda \parallel \mathbf{W} \parallel_*, \tag{2.15}$$

where $\mathbf{H} = \mathbf{D}^T\mathbf{Y}$, $\| \cdot \|_F^2$ is the Frobenius norm, and $\parallel \mathbf{W} \parallel_*$ is the nuclear norm of $\mathbf{W}$. The solution of the abovementioned optimization (2.15) is

$$\widehat{\mathbf{W}} = \mathbf{U}\mathbf{S}_{\mathbf{w}}(\mathbf{\Sigma})\mathbf{V}^T \tag{2.16}$$

where $\mathbf{W} = \mathbf{U}(\mathbf{\Sigma})\mathbf{V}^T$ is the singular value decomposition (SVD) of $\mathbf{W}$, and for each diagonal element $\mathbf{\Sigma}_{ii}$ of $\mathbf{\Sigma}$, $\mathbf{S}_{\mathbf{w}}(\mathbf{\Sigma})$ is the generalized soft-thresholding operator with λ.

$$\mathbf{S}_{\mathbf{w}}(\mathbf{\Sigma})_{ii} = \max((\mathbf{\Sigma})_{ii} - \lambda, 0). \tag{2.17}$$

HSI can be decomposed into different sub-images (approximation coefficients sub-image and detail coefficients sub-images) using 3-D DWT decomposition. Each sub-image can be considered as a block. Here, the main idea of WBBLRR is to apply different regularization parameters for different blocks. Therefore, the WBBLRR problem can be expressed as

$$\widehat{\mathbf{W}} = \arg\min_{\mathbf{W}} \quad \sum_{k=1}^{7L+1} \|\mathbf{H}_k - \mathbf{W}_k\|_F^2 + \lambda_k \parallel \mathbf{W}_k \parallel_*, \tag{2.18}$$

where $\mathbf{H} = [\mathbf{H}_1; \mathbf{H}_2; ...; \mathbf{H}_{7L+1}]$ (the notation ';' shows vertical concatenation) and L is the level of the 3-D DWT decomposition, and $\mathbf{W} = [\mathbf{W}_1; \mathbf{W}_2; ...; \mathbf{W}_{7L+1}]$. Therefore, the solution to each block optimization problem is given by

$$\widehat{\mathbf{W}}_k = \mathbf{U}_k\mathbf{S}_{\mathbf{w}}(\mathbf{\Sigma}_k)\mathbf{V}_k^T, \tag{2.19}$$

where $\mathbf{W}_k = \mathbf{U}_k(\mathbf{\Sigma}_k)\mathbf{V}_k{}^T$ is the SVD of $\mathbf{W}_k$, $k = 1, 2, ..., 7L + 1$, and $\mathbf{S}_{\mathbf{w}}(\mathbf{\Sigma}_k) = \max((\mathbf{\Sigma}_k)_{ii} - \lambda_k, 0)$.

Finally, the clean image of $\mathbf{Y}$ is estimated as

$$\widehat{\mathbf{X}} = \mathbf{D}\widehat{\mathbf{W}}. \tag{2.20}$$

2.2.3 Experiments and Results

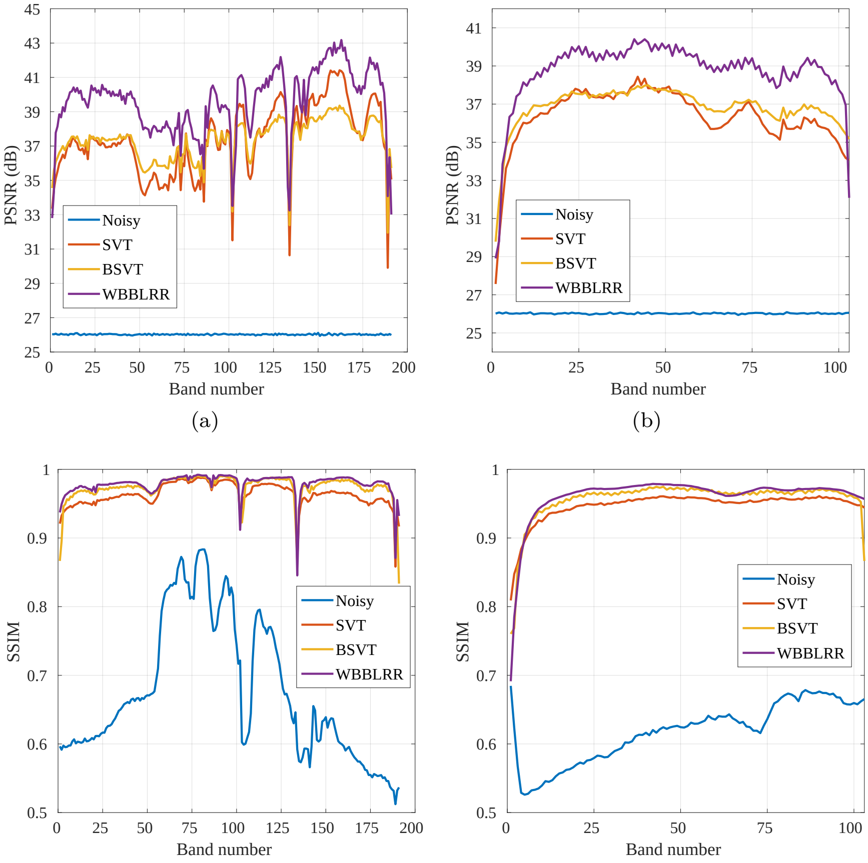
In the following experiments, SVT, BSVT, and WBBLRR denoising methods are compared on two simulated and one real noisy HSI datasets. The simulated datasets are Washington DC Mall (WDCM) dataset (Appendix B.1) and University of Pavia (UP) dataset (Appendix B.2). In the experiments, two small parts of the WDCM and UP datasets are used. Their sizes are 200 × 200 × 191 for the WDCM dataset and 200 × 200 × 103 for the UP dataset. Both datasets are assumed to be noise-free HSIs. The datasets are normalized band by band between [0,1] before adding noise. The Gaussian noise with zero-mean and standard deviation σ was added to the clean WDCM and UP datasets. In the experiments, we set σ to two different values 0.05, and 0.1.

To evaluate the denoised results of the simulated datasets, the quantitative metrics are peak signal-to-noise ratio (PSNR) (Appendix A.1) and mean structural similarity (MSSIM) (Appendix A.2) are used.

Table 2.2: The quantitative evaluation results (PSNR(dB), MSSIM, and Time (second)) for the different denoising methods for the Washington DC Mall (WDCM) and the University of Pavia (UP) datasets. The best results are in bold typeface.

Dataset	Noisy level	Metric	Noisy	SVT	BSVT	WBBLRR
WDCM	$\sigma = 0.05$	PSNR	26.02	36.84	37.14	**39.11**
		MSSIM	0.6750	0.9688	0.9705	**0.9731**
		Time	-	25.19	71.24	**16.13**
	$\sigma = 0.1$	PSNR	20.01	28.94	31.34	**35.33**
		MSSIM	0.4081	0.9293	0.9388	**0.9425**
		Time	-	24.74	71.36	**16.13**
UP	$\sigma = 0.05$	PSNR	26.03	35.94	36.62	**38.07**
		MSSIM	0.6184	0.9406	0.9469	**0.9562**
		Time	-	11.86	34.28	**8.84**
	$\sigma = 0.1$	PSNR	20.01	27.74	30.30	**34.30**
		MSSIM	0.3427	0.8744	0.8859	**0.9070**
		Time	-	11.68	34.21	**9.19**

Here, we evaluate the performance of SVT, BSVT, and WBBLRR for both quantitative metrics and visualization of the denoised WDCM and UP HSIs. The PSNR and SSIM of each band are calculated and depicted in Fig. 2.5 for quantitative assessment.



(a)

(b)

(c)

(d)

Figure 2.5: PSNR (a)-(b), and SSIM (c)-(d) values of each band of denoised WDCM and UP datasets ($\sigma = 0.05$).

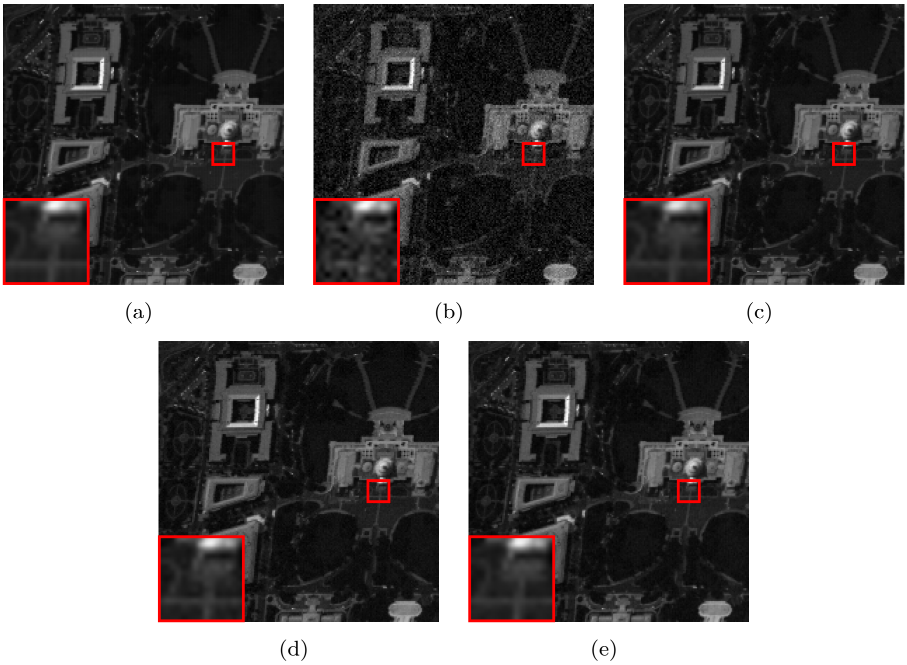


Figure 2.6: The results for the different denoising methods for the WDCM dataset. (a) Original band 5, (b) Noisy band 5, (c) SVT, (d) BSVT, and (e) WBBLRR.

Table 2.2 gives the denoising results in terms of PSNR, MSSIM, and Time for both the WDCM and UP datasets. From the figure and table, it can be seen that the obtained PSNR, SSIM, and MSSIM of WBBLRR are higher than for BSVT and SVT methods. The denoising methods are implemented using MATLAB R2019a on a computer having Intel Core i7-67000 processor (3.40 GHz), 8.00 GB of memory and 64-bit Windows 10 Operating System. The running times for WBBLRR are faster than BSVT and SVT.

We choose $\sigma = 0.05$ for both the WDCM and UP datasets to demonstrate visual quality. Figs. 2.6 and 2.7 show the denoised results for band 5 of the WDCM dataset and the UP dataset, respectively. WBBLRR works well in filtering out the noise and provides cleaner denoised images.

For the experiments based on the real dataset, the Indian Pines dataset (Appendix B.3) is used. Fig. 2.8 presents the denoised results for bands 4, 5, and 200, respectively. It can be clearly observed that the noise is suppressed by all the denoising methods and the proposed WBBLRR method gives good denoised images.

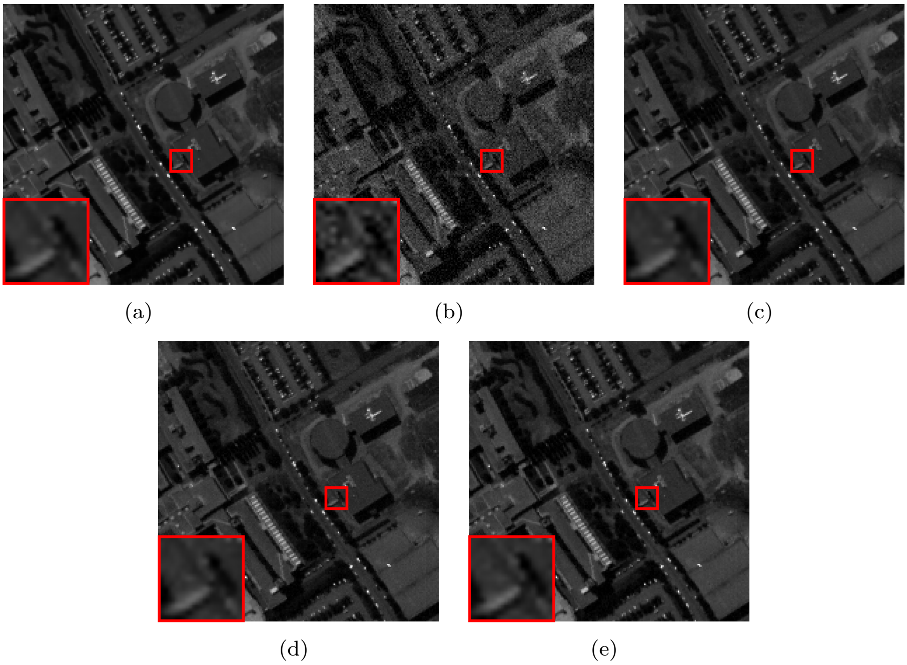
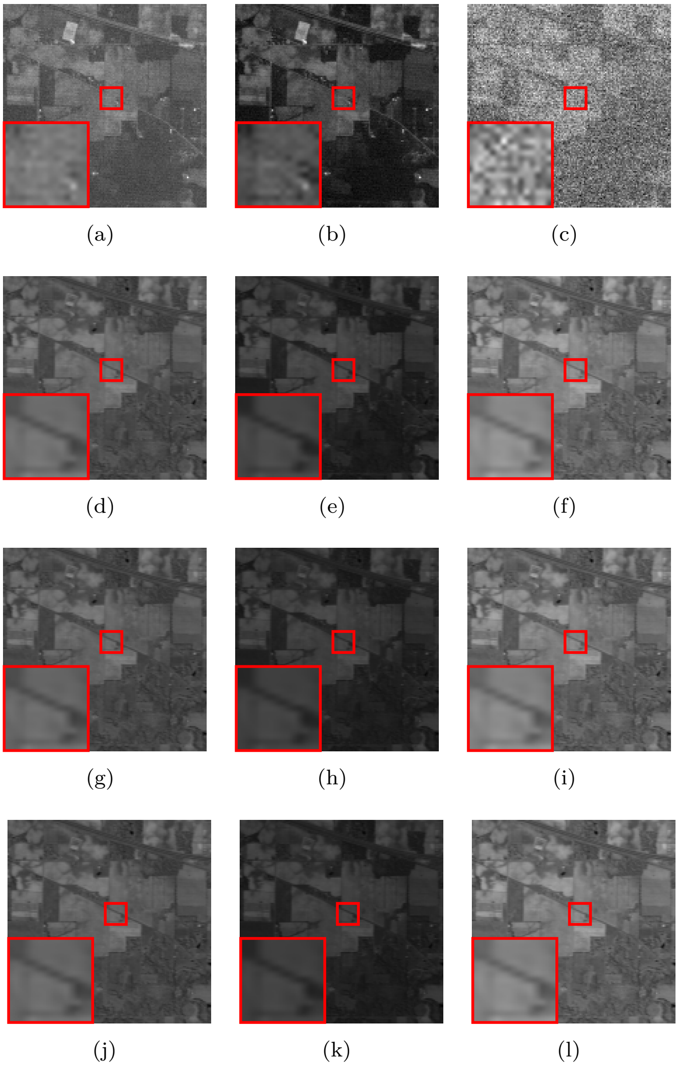


Figure 2.7: The results for the different denoising methods for the UP dataset. (a) Original band 5, (b) Noisy band 5, (c) SVT, (d) BSVT, and (e) WBBLRR.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 2.8: The results for the different denoising methods for the Indian Pines dataset, from left to right bands 4, 5, and 200, respectively, and from top to bottom: Original band, SVT, BSVT, and WBBLRR.

2.3 Conclusions

In this chapter, we propose two new low-rank based denoising methods for HSI. There are NLMLRA and WBBLRR denoising method. NLMLRA uses the patch-based similarity weighting function to construct the NLM denoising operator. To improve the practicability and reduce computational cost, Chebyshev polynomials are used in the NLMLRA. WBBLRR is a block low-rank representations method that uses 3-D wavelet transformation to segment HSI to different blocks and utilizes low-rank representations to obtain denoised blocks. To validate the performance of denoising methods, two simulated datasets and one real HSI dataset are used in the experiments. The experiment results show that the NLMLRA and WBBLRR give good denoised results.

Chapter 3

Sparse and Low-Rank Based Method

In this chapter, a denoising method based on sparse spectral-spatial and low-rank representations (SSSLRR) using 3-D orthogonal transform (3-DOT). SSSLRR can be effectively used to remove Gaussian and mixed noise. SSSLRR uses 3-DOT to decompose noisy HSI to sparse transform coefficients. 3-D discrete orthogonal wavelet transform (3-D DWT) is a representative 3-DOT suitable for denoising since it concentrates the signal in few transform coefficients, the 3-D discrete orthogonal cosine transform (3-D DCT) is another example. An SSSLRR using 3-D DWT will be called SSSLRR-DWT. SSSLRR-DWT is an iterative algorithm based on the alternating direction method multipliers (ADMM) that uses sparse and nuclear norm penalties. We use an ablation study to show the effectiveness of the penalties we employ in the method. Both simulated and real hyperspectral datasets demonstrate that SSSLRR outperforms other comparative methods in quantitative and visual assessments to remove Gaussian and mixed noise.

3.1 Introduction

Recently, many low-rank model-based methods have been proposed for HSI denoising [25–31]. Zhang *et al.* [32] presented a low-rank matrix recovery (LRMR) method for removing mixed types of noise in HSI. An extension of the LRMR method [33] was proposed using an adaptive iteration regularization framework for HSI denoising. Chen *et al.* [34] presented a nonconvex low-rank matrix approximation method using a nonconvex regularizer. These low-rank model-based methods exploit the low-dimensional structures in the HSIs. HSIs are usually self-similarity and have sparse characteristics in the spatial domain and hence can be denoised using sparse representation. Lu *et al.* [35], and Li *et al.* [36] suggested a HSI denoising method based on sparse representation using the high similar spatial information of HSI. More recently, several denoising methods based on a combination of low-rank and sparse representation have been presented [37]. Liu *et al.* [37] uses group sparse and nuclear norm low-rank penalties for removing stripe noise. Zhuang *et al.* [38] utilized similar patches in the low-dimensional subspace to exploit the sparse and low-rank characteristics of HSI. Rasti *et al.* [39] used sparse low-rank model, which utilizes sparse penalized least squares for estimating the unknown signal for denoising. Tensor-based denoising methods given in [40–45] exploited the low-rank and the sparse characteristic of similar tensors to remove noise.

In this chapter, we propose a denoising method based on sparse spectral-spatial and low-rank representations (SSSLRR) using 3-D orthogonal transform (3-DOT) for HSI denoising to remove Gaussian and mixed noise. The paper used both 3-D discrete orthogonal wavelet transform (3-D DWT) and 3-D discrete orthogonal cosine transform (3-D DCT) as representative of 3-DOT. The 3-D DWT is exploited in SSSLRR-DWT to provide a sparse representation for HSI by jointly using the correlated spatial and spectral information. The idea is that the wavelet transform concentrates the signal energy in few coefficients while the noise has energy spread over all coefficients [70, 71]. A new sparse and low-rank penalized model is proposed for removing noise in the wavelet domian. The method uses the ℓ_1 penalty and the weighted nuclear norm low-rank penalty to adaptively shrink the wavelet coefficients and penalize the singular values to remove the noise.

3.2 Problem Formulation and Proposed Denoiser

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_B] \in \mathbb{R}^{n\times B}$ denote a clean HSI with n pixels and B spectral bands. Here, $\mathbf{x}_i \in \mathbb{R}^n$ is the ith vectorized spectral band. We assume the following observational model

$$\mathbf{Y} = \mathbf{X} + \mathbf{N}, \tag{3.1}$$

where $\mathbf{Y}$, $\mathbf{N} \in \mathbb{R}^{n\times B}$ represent the observed noisy HSI and noise, respectively. Therefore, estimating $\widehat{\mathbf{X}} \in \mathbb{R}^{n\times B}$ of $\mathbf{X}$ from $\mathbf{Y}$ is the HSI denoising problem.

3.2.1 SSSLRR

Using the 3-DOT decomposition, the observational model is formulated as

$$\mathbf{Y} = \mathbf{D}_2\mathbf{W}\mathbf{D}_1^T + \mathbf{N}, \tag{3.2}$$

where $\mathbf{D}_2 \in \mathbb{R}^{n\times n}$ represents a spatial part of the 3-DOT, $\mathbf{D}_1 \in \mathbb{R}^{B\times B}$ represents a spectral part of the 3-DOT, and $\mathbf{W} \in \mathbb{R}^{n\times B}$ is a coefficients matrix.

To enforce sparsity and low-rank a cost function using an ℓ_1 and a weighted nuclear norm low-rank penalties is defined as

$$\widehat{\mathbf{W}} = \arg\min_{\mathbf{W}} \ \|\mathbf{H} - \mathbf{W}\|_F^2 + \lambda_1\|\mathbf{W}\|_{1,1} + \|\mathbf{W}\|_{\omega,*}, \tag{3.3}$$

where $\mathbf{H} = \mathbf{D}_2^T\mathbf{Y}\mathbf{D}_1$, $\|\cdot\|_F^2$ is the Frobenius norm, $\|\mathbf{W}\|_{1,1} := \sum_{i=1}^{B} \|\mathbf{w}_i\|_1$ ($\mathbf{w}_i$ denotes the ith column of $\mathbf{W}$), and $\|\mathbf{W}\|_{\omega,*}$ is the weighted nuclear norm of $\mathbf{W}$, which is defined as

$$\|\mathbf{W}\|_{\omega,*} = \Sigma_i|\omega_i\sigma_i(\mathbf{W})|, \tag{3.4}$$

where ω_i is the weight assigned to the ith singular value $\sigma_i(\mathbf{W})$ of $\mathbf{W}$.

We adapt alternating direction method of multipliers (ADMM) [74] for solving (3.3). The optimization problem (3.3) is transformed into the equivalent problem

$$\min_{\mathbf{W},\mathbf{Z}} \ \|\mathbf{H} - \mathbf{W}\|_F^2 + \lambda_1\|\mathbf{W}\|_{1,1} + \|\mathbf{Z}\|_{\omega,*},$$

$$\text{s.t.} \quad \mathbf{W} - \mathbf{Z} = \mathbf{0}. \tag{3.5}$$

The augmented Lagrangian function of the abovementioned optimization is

$$\begin{aligned} L_\mu(\mathbf{W}, \mathbf{Z}, \mathbf{U}) =& \|\mathbf{H} - \mathbf{W}\|_F^2 + \lambda_1 \|\mathbf{W}\|_{1,1} + \|\mathbf{Z}\|_{\omega,*} \\ &+ \frac{\mu}{2} \|\mathbf{W} - \mathbf{Z} + \mathbf{U}\|_F^2, \end{aligned} \tag{3.6}$$

where $\mu > 0$ is the ADMM penalty parameter. The ADMM consists of the iterations:

W-Step: Fix $\mathbf{Z}^k$, $\mathbf{U}^k$ and estimate $\mathbf{W}^{k+1}$ by solving

$$\mathbf{W}^{k+1} := \arg\min_{\mathbf{W}} \; L_\mu(\mathbf{W}, \mathbf{Z}^k, \mathbf{U}^k). \tag{3.7}$$

Z-Step: Fix $\mathbf{W}^{k+1}$, $\mathbf{U}^k$ and estimate $\mathbf{Z}^{k+1}$ by solving

$$\mathbf{Z}^{k+1} := \arg\min_{\mathbf{Z}} \; L_\mu(\mathbf{W}^{k+1}, \mathbf{Z}, \mathbf{U}^k). \tag{3.8}$$

U-Step: Fix $\mathbf{W}^{k+1}$, $\mathbf{Z}^{k+1}$ and update $\mathbf{U}^{k+1}$ with

$$\mathbf{U}^{k+1} := \mathbf{U}^k + \mathbf{W}^{k+1} - \mathbf{Z}^{k+1}. \tag{3.9}$$

The minimization problem (3.7) is equivalent to

$$\arg\min_{\mathbf{W}} \; \|\mathbf{H} - \mathbf{W}\|_F^2 + \lambda_1 \|\mathbf{W}\|_{1,1} + \frac{\mu}{2} \|\mathbf{W} - \mathbf{Z}^k + \mathbf{U}^k\|_F^2. \tag{3.10}$$

By ignoring irrelevant terms and using trace properties, the minimization problem (3.10) is equivalent to

$$\arg\min_{\mathbf{W}} \; \frac{1}{2} \|\mathbf{W} - \mathbf{B}\|_F^2 + \lambda \|\mathbf{W}\|_{1,1}, \tag{3.11}$$

where $\mathbf{B} = \frac{2\mathbf{H} + \mu \mathbf{Z}^k - \mu \mathbf{U}^k}{2+\mu}$, and $\lambda = \frac{\lambda_1}{2+\mu}$. This problem (3.11) can be solved by using soft-thresholding function [75]

$$w_{i,j}^{k+1} = \operatorname{sign}(b_{i,j}) \max\{|b_{i,j}| - \lambda, 0\}, \tag{3.12}$$

where $w_{i,j}^{k+1}$ and $b_{i,j}$ are the entries in the ith row and jth column of $\mathbf{W}^{k+1}$ and $\mathbf{B}$, respectively.

The minimization problem (3.8) is equivalent to

$$\mathbf{Z}^{k+1} = \arg\min_{\mathbf{Z}} \ \frac{\mu}{2}\|\mathbf{W}^{k+1} - \mathbf{Z} + \mathbf{U}^k\|_F^2 + \|\mathbf{Z}\|_{\omega,*} \tag{3.13}$$

has the closed-form solution [15]

$$\mathbf{Z}^{k+1} = \mathbf{P}\mathbf{S}_\omega(\Sigma)\mathbf{Q}^T, \tag{3.14}$$

where $\mathbf{W}^{k+1} + \mathbf{U}^k = \mathbf{P}\Sigma\mathbf{Q}^T$ is the singular value decomposition (SVD) of $\mathbf{W}^{k+1} + \mathbf{U}^k$ with singular values ordered by nonincreasing magnitude. $\mathbf{S}_\omega(\Sigma)$ is a generalized soft-thresholding function on the diagonal matrix Σ. For the ith singular value Σ_{ii},

$$\mathbf{S}_\omega(\Sigma)_{ii} = \max(\Sigma_{ii} - \omega_i, 0). \tag{3.15}$$

The weight ω_i is

$$\omega_i = \frac{\lambda_\omega}{\mu(\Sigma_{ii} + \varepsilon)}, \tag{3.16}$$

where ε is a nonzero constant and is used to avoid dividing by zero and λ_ω is a constant. This means that larger singular values will be penalized less than smaller singular values. Thus, the weighted nuclear norm low-rank penalty provides better estimation for the rank compared with the nuclear norm which treats all the singular values equally, since larger singular values implying major information that are more important than smaller ones [15].

Finally, the estimate is obtained as

$$\widehat{\mathbf{X}} = \mathbf{D}_2\widehat{\mathbf{W}}\mathbf{D}_1^T. \tag{3.17}$$

The SSSLRR method is summarized in Algorithm 1 and illustrated in Fig. 3.1.

Algorithm 1 SSSLRR HSI Denoising Algorithm

```
Input: $\mathbf{Y}$, $\lambda_1$, $\lambda_\omega$, and $\mu$
Initialization: $\mathbf{Z}^0 = \mathbf{U}^0 = 0$, and $k = 0$
repeat
    $\mathbf{B} = \frac{2\mathbf{D}_2^T\mathbf{Y}\mathbf{D}_1 + \mu\mathbf{Z}^k - \mu\mathbf{U}^k}{2+\mu}$
    $\mathbf{W}^{k+1} = \arg\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{W} - \mathbf{B}\|_F^2 + \frac{\lambda_1}{2+\mu}\|\mathbf{W}\|_{1,1}$
    $\mathbf{Z}^{k+1} = \arg\min_{\mathbf{Z}} \frac{\mu}{2}\|\mathbf{W}^{k+1} - \mathbf{Z} + \mathbf{U}^k\|_F^2 + \|\mathbf{Z}\|_{\omega,*}$
    $\mathbf{U}^{k+1} = \mathbf{U}^k + \mathbf{W}^{k+1} - \mathbf{Z}^{k+1}$
    $k \leftarrow k + 1$
until stopping criterion is satisfied, i.e., $\|\mathbf{W}^{k+1} - \mathbf{W}^k\|_F^2/\|\mathbf{W}^k\|_F^2 \leq \epsilon$
Output: $\widehat{\mathbf{X}} = \mathbf{D}_2\widehat{\mathbf{W}}\mathbf{D}_1^T$
```

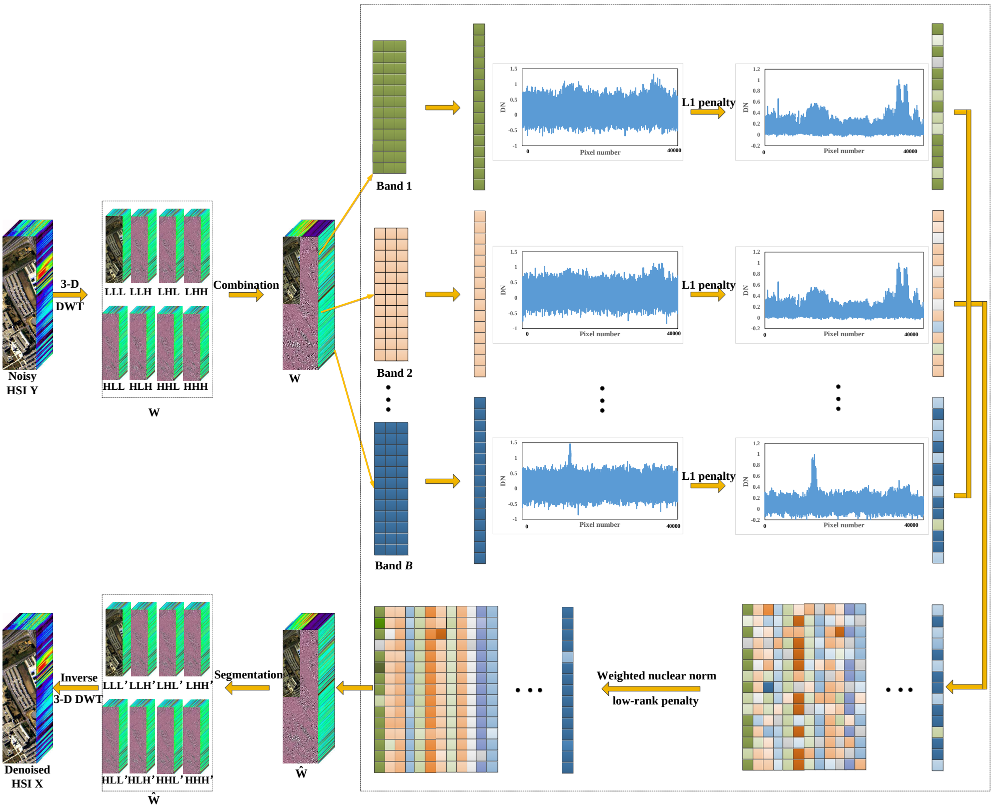


Figure 3.1: The schematic of the SSSLRR, which uses the 3-D DWT for the 3-DOT.

3.2.2 SSSLRR-DWT

The SSSLRR-DWT denoising method uses the 3-D discrete orthogonal wavelet transform (3-D DWT) to be a representative of the 3-DOT. In SSSLRR-DWT, the 2-D orthogonal (spatial) wavelet basis $\mathbf{D}_2 \in \mathbb{R}^{n\times n}$ and the 1-D orthogonal (spectral) wavelet basis $\mathbf{D}_1 \in \mathbb{R}^{B\times B}$ are used for 3-D DWT to decompose the 3-D HSI to sparse wavelet coefficients $\mathbf{W} \in \mathbb{R}^{n\times B}$. SSSLRR-DWT uses the sparse and low-rank penalized model for the wavelet coefficients to remove noise.

3.2.3 SSSLRR-DCT

3-D discrete orthogonal cosine transform (3-D DCT) is also a classical 3-DOT. SSSLRR-DCT denoising method uses 3-D DCT as the 3-DOT in SSSLRR. For SSSLRR-DCT, the 2-D orthogonal (spatial) cosine basis $\mathbf{D}_2 \in \mathbb{R}^{n\times n}$ and the 1-D orthogonal (spectral) cosine basis $\mathbf{D}_1 \in \mathbb{R}^{B\times B}$ are used for 3-D DCT to decompose the noisy HSI to sparse cosine coefficients $\mathbf{W} \in \mathbb{R}^{n\times B}$. The sparse and low-rank penalized model is utilized for SSSLRR-DCT to shrink the cosine coefficients to remove noise.

The 3-D DWT or 3-D DCT can concentrate the signal energy in few coefficients while the noise spreads over all coefficients. Both SSSLRR-DWT or SSSLRR-DCT are iterative algorithms and can adaptively shrink either wavelet coefficients or cosine transform coefficients and penalize the singular values of the coefficient matrix. The larger (signal) singular values will be shrunk less while the smaller ones (noise) will be shrunk more leading to denoising. Both SSSLRR-DWT and SSSLRR-3DCT yield good denoising performance. However, the focus of this paper is on SSSLRR-DWT.

3.2.4 Ablation Study

SSSLRR-DWT uses the sparse and low-rank penalized model based on the ℓ_1 and weighted nuclear norm low-rank penalties in the wavelet domain for HSI denoising. To further analyze the effectiveness of the penalties for the SSSLRR-DWT denoising, an ablation study of SSSLRR-DWT is performed. SSSLRR-DWT is compared with the following denoising methods:

1. *SSWSR*: Only the ℓ_1 penalty is used, i.e., the cost function is

$$\widehat{\mathbf{W}} = \arg\min_{\mathbf{W}} \ \|\mathbf{H} - \mathbf{W}\|_F^2 + \lambda_1 \|\mathbf{W}\|_{1,1}. \tag{3.18}$$

2. *SSWLRR1*: Only the nuclear norm penalty is used, i.e., the cost function is

$$\widehat{\mathbf{W}} = \arg\min_{\mathbf{W}} \ \|\mathbf{H} - \mathbf{W}\|_F^2 + \lambda_* \|\mathbf{W}\|_*, \tag{3.19}$$

   where $\|\cdot\|_*$ is the nuclear norm.

3. *SSWLRR2*: The weighted nuclear norm low-rank penalty is used in the signal domain instead of using it in the wavelet domain, i.e., the cost function is

$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \ \|\mathbf{Y} - \mathbf{X}\|_F^2 + \|\mathbf{X}\|_{\omega,*}, \tag{3.20}$$

4. *SLRR*: The ℓ_1 penalty and the weighted nuclear norm low-rank penalty are used in the signal domain, i.e., the cost function is

$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \ \|\mathbf{Y} - \mathbf{X}\|_F^2 + \lambda_1 \|\mathbf{X}\|_{1,1} + \|\mathbf{X}\|_{\omega,*}, \tag{3.21}$$

The ADMM algorithm is adapted to get the solutions for these optimization problems.

The simulated WDCM dataset for Cases 1-3 and UP dataset for Case 7 (detailed experimental setting is given in Section III) is used for this experiment. Tables 3.1 and 3.2 and Figs. 3.2-3.8 present the denoising results. SSWSR can remove most of the Gaussian and the salt and pepper noise but can not remove the stripe noise. The difference in results between SSWLRR1 and SSWLRR2 shows that the extra flexibility provided by the weighted nuclear norm compared to the traditional nuclear norm leads to better denoising results. Both SLRR and SSSLRR-DWT can effectively

Table 3.1: The quantitative evaluation results (PSNR(dB), MSSIM, MFSIM, and SAM(degrees)) for the different denoising methods using the Washington DC Mall (WDCM) dataset. The best results are in bold typeface.

Noise level	Metric	Noisy	SSWSR	SSWLRR1	SSWLRR2	SLRR	SSSLRR-DWT
$\sigma = 0.1$	PSNR	20.004	32.171	33.304	35.627	35.906	**37.968**
	MSSIM	0.4528	0.9263	0.9449	0.9582	0.9593	**0.9794**
	MFIM	0.7348	0.9598	0.9707	0.9745	0.9790	**0.9867**
	SAM	30.350	8.5714	7.4659	6.0733	6.0960	**4.0887**
$\sigma = 1, \eta = 10$	PSNR	22.820	32.429	31.771	33.965	38.267	**44.165**
	MSSIM	0.8674	0.9633	0.9553	0.9691	0.9857	**0.9968**
	MFIM	0.9306	0.9777	0.9701	0.9781	0.9910	**0.9977**
	SAM	23.863	8.9575	7.7902	6.0998	4.8411	**1.9880**
$\sigma \sim U(0.1, 0.2)$	PSNR	16.371	28.816	30.774	32.777	33.250	**35.544**
	MSSIM	0.3083	0.8686	0.9199	0.9274	0.9326	**0.9661**
	MFIM	0.6423	0.9289	0.9565	0.9573	0.9655	**0.9782**
	SAM	39.689	12.054	9.1468	7.9920	8.0173	**5.2188**

Table 3.2: The quantitative evaluation results (PSNR(dB), MSSIM, MFSIM, and SAM(degrees)) for the different denoising methods using the University of Pavia (UP) dataset. The best results are in bold typeface.

Noise level	Metric	Noisy	SSWSR	SSWLRR1	SSWLRR2	SLRR	SSSLRR-DWT
Case 7	PSNR	12.891	23.738	24.054	26.119	28.432	**30.933**
	MSSIM	0.1465	0.6590	0.6876	0.7489	0.7883	**0.9066**
	MFIM	0.5321	0.8601	0.8735	0.8902	0.9169	**0.9553**
	SAM	41.441	12.021	12.124	11.309	8.8532	**7.2942**

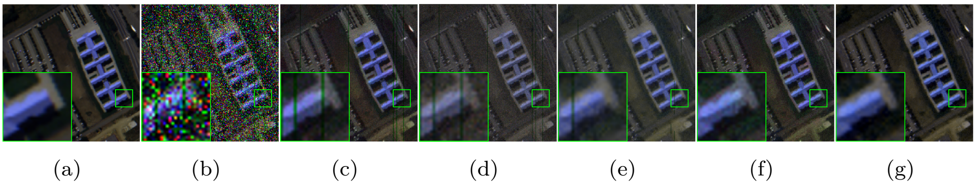


Figure 3.2: Denoising for the UP dataset, Case 7 using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 70, G: 120, B: 170), (b) Noisy image, (c) SSWSR, (d) SSWLRR1, (e) SSWLRR2, (f) SLRR, and (g) SSSLRR-DWT.

remove mixed noise. The benefit of using the ℓ_1 and weighted nuclear norm low-rank penalties can clearly be seen by noticing that both SLRR and SSSLRR-DWT

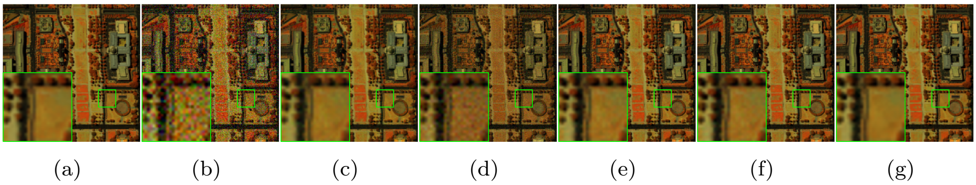


Figure 3.3: Denoising for the WDCM dataset, Case 1 with $\sigma = 0.1$, using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 70, G: 120, B: 170), (b) Noisy image, (c) SSWSR, (d) SSWLRR1, (e) SSWLRR2, (f) SLRR, and (g) SSSLRR-DWT.

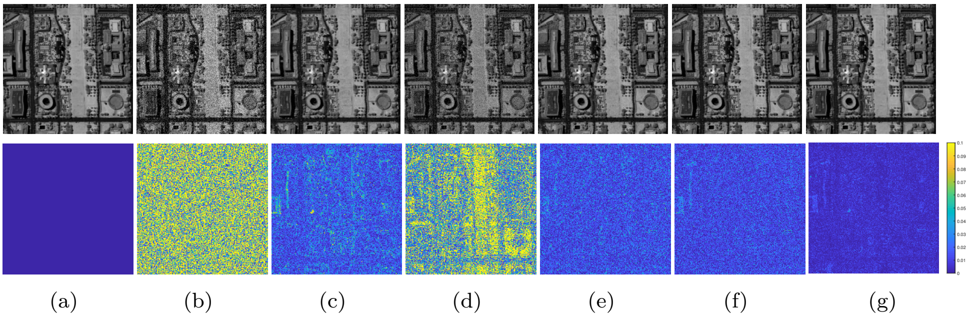


Figure 3.4: Denoised images (the first row) and residual images (the second row) for the WDCM dataset, Case 1 with $\sigma = 0.1$, using different methods. (a) Original band 95, (b) Noisy band 95, (c) SSWSR, (d) SSWLRR1, (e) SSWLRR2, (f) SLRR, and (g) SSSLRR-DWT.

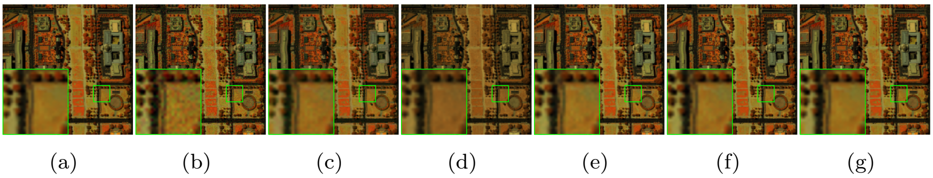


Figure 3.5: Denoising for the WDCM dataset, Case 2 with $\sigma = 1$, $\eta = 10$, using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 70, G: 120, B: 170), (b) Noisy image, (c) SSWSR, (d) SSWLRR1, (e) SSWLRR2, (f) SLRR, and (g) SSSLRR-DWT.

outperform SSWSR, SSWLRR1, and SSWLRR2. Finally, SSSLRR-DWT outperforms SLRR and yields the best results of all the methods, highlighting that working in the wavelet domain yields some gains.

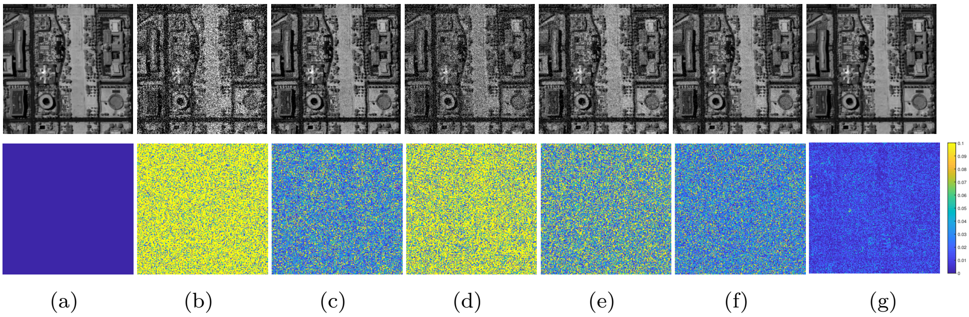


Figure 3.6: Denoised images (the first row) and residual images (the second row) for the WDCM dataset, Case 2 with $\sigma = 1$, $\eta = 10$, using different methods. (a) Original band 95, (b) Noisy band 95, (c) SSWSR, (d) SSWLRR1, (e) SSWLRR2, (f) SLRR, and (g) SSSLRR-DWT.

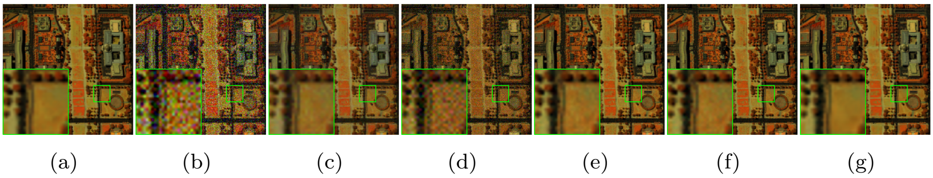


Figure 3.7: Denoising for the WDCM dataset, Case 3 with $\sigma \sim U(0.1, 0.2)$, $\eta = 10$, using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 70, G: 120, B: 170), (b) Noisy image, (c) SSWSR, (d) SSWLRR1, (e) SSWLRR2, (f) SLRR, and (g) SSSLRR-DWT.

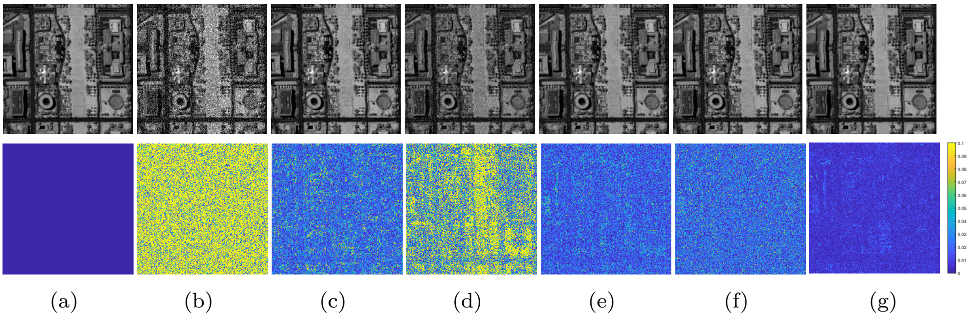


Figure 3.8: Denoised images (the first row) and residual images (the second row) for the WDCM dataset, Case 3 with $\sigma \sim U(0.1, 0.2)$, using different methods. (a) Original band 95, (b) Noisy band 95, (c) SSWSR, (d) SSWLRR1, (e) SSWLRR2, (f) SLRR, and (g) SSSLRR-DWT.

3.3 Experimental Results and Discussion

In the experiments both simulated and real datasets are used to demonstrate the effectiveness of the proposed method. Experimental results for SSSLRR-DWT are compared with the following denoising methods: two patch based methods, i.e., BM3D[1] [14], and nonlocal transform-domain filter for volumetric data (BM4D[2]) [76], two low-rank based methods, i.e., low-rank tensor approximation (LRTA[3]) [77], and noise-adjusted iterative low-rank matrix approximation (NAILRMA[4]) [33], four sparse and low-rank based methods, i.e., tensor learning dictionary (TDL[5]) [78], Kronecker-basis-representation based tensor sparsity measure (KBR[6]) [79], sparse and low-rank model HSI restoration (HyRes[7]) [39], and fast hyperspectral denoising based on the low-rank and sparse representation (FastHyDe[8]) [38]. Each datasets are normalized between [0,1] band by band before adding the noise and are returned to the original level after denoising.

3.3.1 Simulated Datasets Experiments

Data Description

We use two public HSI datasets to simulate noisy HSI. The simulated datasets are Washington DC Mall (WDCM) dataset (Appendix B.1) and University of Pavia (UP) dataset (Appendix B.2). In the experiments, two small parts of the WDCM and UP datasets are used. Their sizes are 200 × 200 × 191 for the WDCM dataset and 200 × 200 × 103 for the UP dataset. Both datasets are assumed to be noise-free HSIs. The noisy simulated datasets are created as follows:

1. *Case 1*: A zero-mean isotropic Gaussian noise with standard deviation $\sigma \in \{0.05, 0.1, 0.2\}$, is added to each band.
2. *Case 2*: A zero-mean bandwise Gaussian noise with varied variance, σ_i, is added to each band. The bell shape variance varies according to

$$\sigma_i^2 = \sigma^2 \frac{e^{-\frac{(i-B/2)^2}{2\eta^2}}}{\sum_{j=1}^{B} e^{-\frac{(i-B/2)^2}{2\eta^2}}}, \tag{3.22}$$

   where σ, and η control the noise intensity and the bell width, respectively [80]. We set σ, η to two different values 0.5 and 1, 10, and 20, respectively.

[1] http://www.cs.tut.fi/~foi/GCF-BM3D/

[2] http://www.cs.tut.fi/~foi/GCF-BM3D/

[3] https://www.sandia.gov/tgkolda/TensorToolbox/

[4] https://sites.google.com/site/rshewei/home

[5] http://gr.xjtu.edu.cn/web/dymeng/3

[6] http://gr.xjtu.edu.cn/web/dymeng/3

[7] https://www.researchgate.net/profile/Behnood-Rasti

[8] https://github.com/LinaZhuang/FastHyDe_FastHyIn

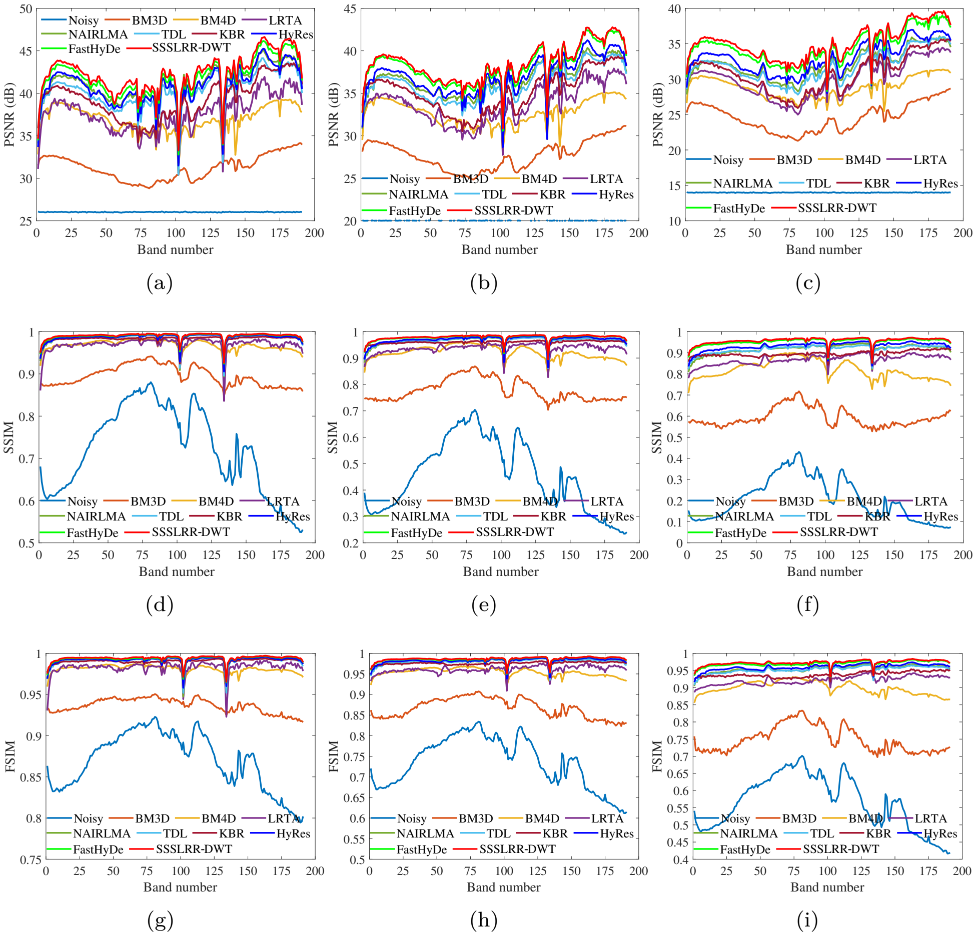
3. *Case 3*: A zero mean Gaussian noise is added to each band with standard deviation σ is drawn from a uniform distribution between 0.1 and 0.2 [$\sigma \sim U(0.1, 0.2)$].

Evaluation Metric

To evaluate the denoised results of the simulated datasets, the quantitative metrics are the peak signal-to-noise ratio (PSNR) (Appendix A.1), mean structural similarity (MSSIM) (Appendix A.2), the mean feature similarity (MFSIM) (Appendix A.2), and the spectral angle mapper (SAM) (Appendix A.3) are used.

Table 3.3: The quantitative evaluation results (PSNR(dB), MSSIM, MFSIM, SAM(degrees), and Time (second)) for the different denoising methods using the Washington DC Mall (WDCM) dataset. The best results are in bold typeface.

WDCM	Noise level	Metric	Noisy	BM3D	BM4D	LRTA	NAIRLMA	TDL	KBR	HyRes	FastHyDe	SSSLRR-DWT
Case 1	$\sigma = 0.05$	PSNR	26.025	31.031	36.624	37.035	40.288	39.615	38.617	40.297	41.219	**41.681**
		MSSIM	0.7191	0.8948	0.9677	0.9699	0.9862	0.9855	0.9830	0.9870	0.9898	**0.9908**
		MFSIM	0.8691	0.9365	0.9808	0.9829	0.9919	0.9910	0.9903	0.9921	0.9935	**0.9942**
		SAM	18.048	8.6547	5.1559	5.1145	3.6577	3.6349	3.0349	3.4451	2.9787	**2.8475**
		Time	-	118.12	462.12	1.5192	58.323	8.0064	1253.3	1.5588	**0.3309**	11.754
	$\sigma = 0.1$	PSNR	20.004	27.429	32.468	33.181	36.109	35.581	34.582	36.419	37.671	**37.968**
		MSSIM	0.4528	0.7778	0.9209	0.9370	0.9661	0.9657	0.9596	0.9701	0.9779	**0.9794**
		MFSIM	0.7348	0.8675	0.9541	0.9612	0.9806	0.9792	0.9758	0.9821	0.9858	**0.9867**
		SAM	30.350	10.742	7.5690	6.6810	5.6988	5.3089	4.4250	5.2171	4.1810	**4.0887**
		Time	-	137.68	461.66	1.4653	48.741	7.8660	1337.1	1.2664	**0.3454**	12.593
	$\sigma = 0.2$	PSNR	13.983	24.199	28.540	29.300	31.653	31.280	30.145	32.442	33.948	**34.518**
		MSSIM	0.2061	0.5929	0.8197	0.8675	0.9177	0.9178	0.8956	0.9324	0.9528	**0.9582**
		MFSIM	0.5644	0.7482	0.8994	0.9215	0.9548	0.9524	0.9383	0.9599	0.9700	**0.9734**
		SAM	46.309	12.615	10.651	8.3784	8.4539	7.1401	6.2672	7.0750	5.9433	**5.6292**
		Time	-	120.97	463.11	1.7205	42.016	7.5671	2644.2	1.3871	**0.3403**	13.466
Case 2	$\sigma = 0.5$, $\eta = 10$	PSNR	28.840	34.056	39.026	29.008	43.879	29.006	35.239	46.498	42.381	**46.750**
		MSSIM	0.9321	0.9707	0.9890	0.9330	0.9955	0.9336	0.9713	0.9978	0.9935	**0.9979**
		MFSIM	0.9641	0.9808	0.9926	0.9643	0.9969	0.9649	0.9832	0.9984	0.9955	**0.9985**
		SAM	13.704	6.7747	3.9992	13.476	2.3161	13.471	6.2553	1.5969	2.5904	**1.5664**
		Time	-	118.91	464.55	1.3040	100.21	9.8575	1223.6	1.6714	**0.2571**	44.097
	$\sigma = 1$, $\eta = 10$	PSNR	22.820	30.645	35.060	23.096	40.088	22.987	29.944	43.602	39.003	**44.165**
		MSSIM	0.8674	0.9401	0.9754	0.8680	0.9912	0.8694	0.9329	0.9965	0.9864	**0.9968**
		MFSIM	0.9306	0.9636	0.9845	0.9307	0.9937	0.9316	0.9634	0.9975	0.9906	**0.9977**
		SAM	23.863	8.9355	5.9647	23.266	3.3176	23.530	11.356	2.1116	3.5566	**1.9880**
		Time	-	116.93	463.54	2.0748	100.27	10.306	1233.6	1.5828	**0.3015**	44.075
	$\sigma = 1$, $\eta = 20$	PSNR	22.815	29.410	34.324	24.515	38.917	24.686	34.515	39.877	38.928	**40.436**
		MSSIM	0.7853	0.9087	0.9668	0.8188	0.9873	0.8314	0.9629	0.9902	0.9859	**0.9913**
		MFSIM	0.8927	0.9454	0.9790	0.9034	0.9915	0.9148	0.9773	0.9931	0.9903	**0.9939**
		SAM	23.972	9.4715	6.2886	20.424	3.7875	20.236	6.0661	3.1943	3.5651	**3.0618**
		Time	-	115.41	465.22	4.6855	99.264	9.5967	1233.0	1.4268	**0.3048**	36.186
Case 3	$\sigma \sim U$ (0.1,0.2)	PSNR	16.371	25.161	30.069	23.942	33.646	32.977	31.668	34.509	35.331	**35.544**
		MSSIM	0.3083	0.6624	0.8694	0.6447	0.9455	0.9384	0.9256	0.9546	0.9645	**0.9661**
		MFSIM	0.6423	0.8008	0.9258	0.8075	0.9691	0.9651	0.9562	0.9732	0.9773	**0.9782**
		SAM	39.689	12.821	9.3562	20.305	7.0918	6.8392	5.5020	6.1270	5.3139	**5.2188**
		Time	-	119.62	461.44	4.5692	40.548	8.1913	2276.4	1.3455	**0.3276**	12.620



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 3.9: PSNR (a)-(c), SSIM (d)-(f), and FSIM (g)-(i) values of each band of the denoised WDCM dataset. The first to third column correspond to the Case 1 with $\sigma = 0.05$, $\sigma = 0.1$, and $\sigma = 0.2$.

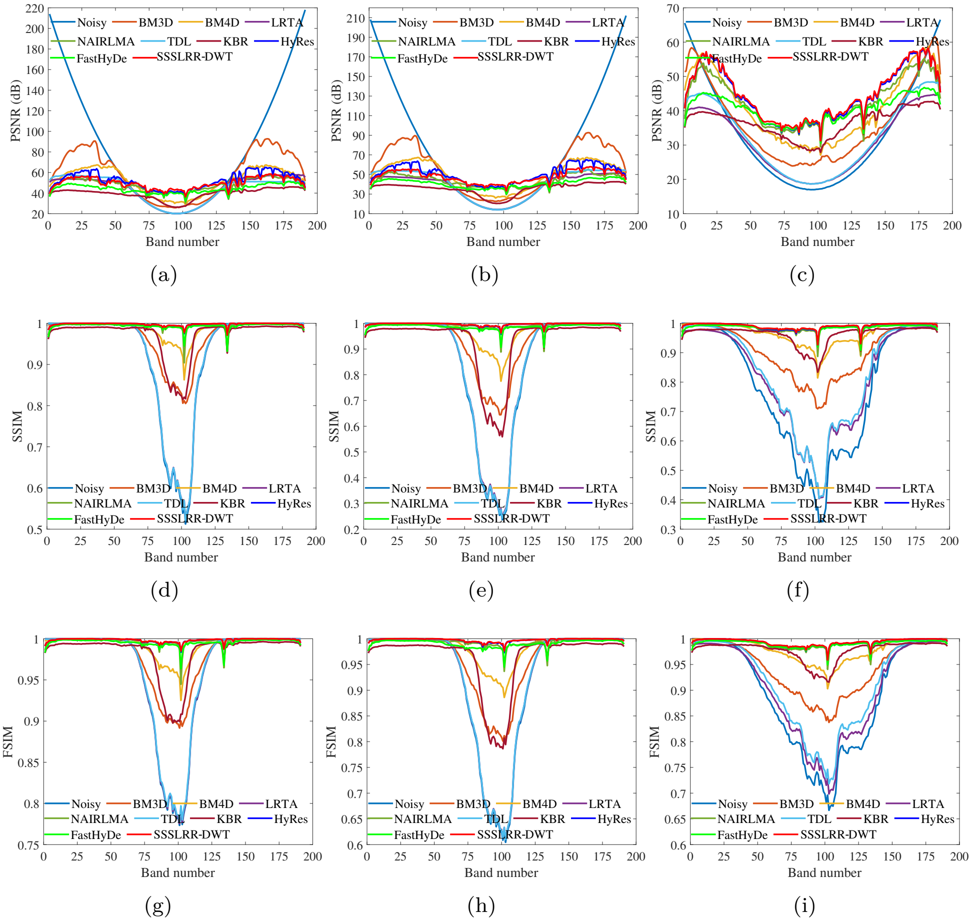
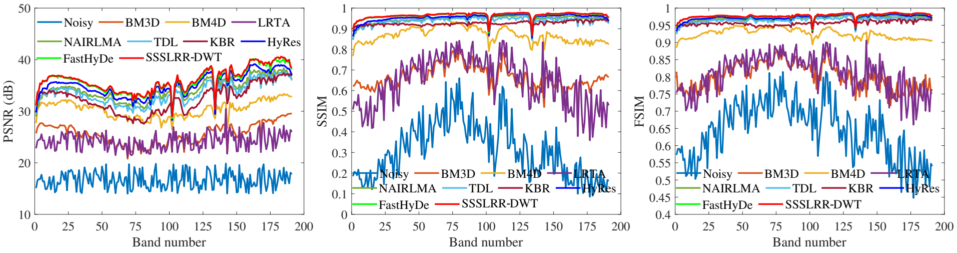
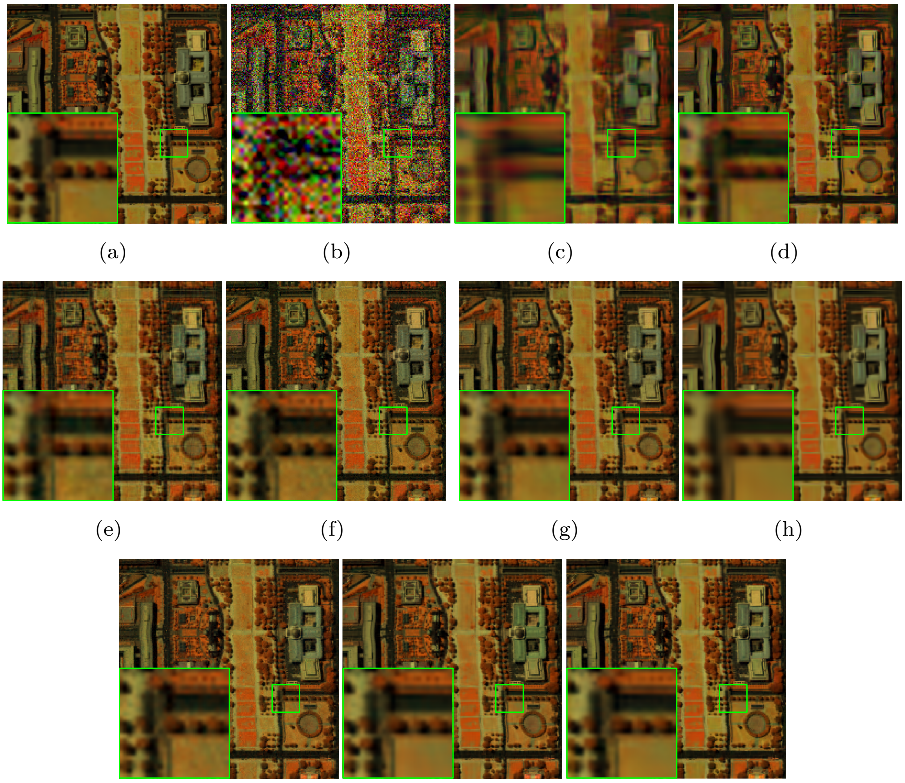


Figure 3.10: PSNR (a)-(c), SSIM (d)-(f), and FSIM (g)-(i) values of each band of the denoised WDCM dataset. The first to third column correspond to the Case 2 with $\sigma = 0.5$, $\eta = 10$, $\sigma = 1$, $\eta = 10$, and $\sigma = 1$, $\eta = 20$.



(a) (b) (c)

Figure 3.11: PSNR (a), SSIM (b), and FSIM (c) values of each band of denoised the WDCM dataset in Case 3 with $\sigma \sim U(0.1, 0.2)$.



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

(e) (f) (g) (h)

(i) (j) (k)

Figure 3.12: Denoising for the WDCM dataset, Case 1 with $\sigma = 0.2$, using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 70, G: 120, B: 170), (b) Noisy image, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.

Table 3.4: The quantitative evaluation results (PSNR(dB), MSSIM, MFSIM, SAM(degrees), and Time (second)) for the different denoising methods using the University of Pavia (UP) dataset. The best results are in bold typeface.

UP	Noise level	Metric	Noisy	BM3D	BM4D	LRTA	NAIRLMA	TDL	KBR	HyRes	FastHyDe	SSSLRR-DWT
Case 1	$\sigma = 0.05$	PSNR	26.026	31.713	37.118	35.031	37.410	38.001	37.499	37.860	38.379	**38.807**
		MSSIM	0.6512	0.8931	0.9607	0.9304	0.9623	0.9680	0.9658	0.9646	0.9706	**0.9728**
		MFSIM	0.8549	0.9330	0.9784	0.9714	0.9837	0.9857	0.9831	0.9849	0.9862	**0.9875**
		SAM	13.900	6.1373	3.6802	4.6048	3.5091	3.1956	3.1937	3.3337	3.1292	**3.0070**
		Time	-	60.969	245.00	2.5617	27.388	4.9310	647.12	0.7363	**0.3052**	4.2892
	$\sigma = 0.1$	PSNR	20.002	28.063	33.603	31.461	33.812	34.526	34.803	34.332	35.545	**35.704**
		MSSIM	0.3880	0.7875	0.9234	0.8657	0.9165	0.9411	0.9457	0.9290	0.9507	**0.9520**
		MFSIM	0.7053	0.8659	0.9550	0.9342	0.9658	0.9706	0.9702	0.9681	0.9741	**0.9751**
		SAM	25.525	7.5487	5.0964	6.1013	5.0108	4.3931	3.9783	4.6130	4.0236	**3.9670**
		Time	-	59.788	243.39	0.7045	20.610	4.4749	681.14	0.6542	**0.3430**	4.5737
	$\sigma = 0.2$	PSNR	13.980	25.117	29.652	27.924	29.540	30.917	30.940	30.696	32.544	**32.997**
		MSSIM	0.1700	0.6401	0.8354	0.7505	0.8136	0.8832	0.8878	0.8612	0.9166	**0.9232**
		MFSIM	0.5242	0.7623	0.9061	0.8719	0.9236	0.9378	0.9333	0.9336	0.9520	**0.9567**
		SAM	42.357	9.3165	7.2347	7.6180	7.3585	5.7785	5.2794	6.4479	5.1509	**4.9689**
		Time	-	60.677	242.48	0.7807	21.781	4.1931	1257.4	0.6370	**0.3394**	4.8983
Case 2	$\sigma = 0.5$, $\eta = 10$	PSNR	26.144	34.154	39.424	27.257	38.215	27.306	34.766	41.055	40.213	**41.202**
		MSSIM	0.7731	0.9375	0.9758	0.7879	0.9706	0.8010	0.9329	0.9804	0.9783	**0.9822**
		MFSIM	0.8928	0.9591	0.9852	0.8970	0.9848	0.9097	0.9671	0.9909	0.9891	**0.9914**
		SAM	13.663	4.4674	2.7893	12.075	3.1835	12.019	4.8530	2.1801	2.5442	**2.3102**
		Time	-	57.901	244.17	1.2375	49.585	5.2778	642.83	0.8239	**0.2925**	13.858
	$\sigma = 1$, $\eta = 10$	PSNR	20.135	31.272	35.481	22.042	35.150	21.472	29.689	37.044	36.854	**37.721**
		MSSIM	0.6385	0.8920	0.9475	0.6513	0.9484	0.6718	0.8530	0.9531	0.9588	**0.9648**
		MFSIM	0.8060	0.9295	0.9682	0.8160	0.9729	0.8305	0.9283	0.9796	0.9778	**0.9813**
		SAM	25.097	5.8888	4.1183	20.728	4.3844	22.023	8.7923	3.2496	3.4898	**3.2060**
		Time	-	56.629	244.10	3.6193	49.192	5.3541	639.44	0.7383	**0.3294**	13.436
	$\sigma = 1$, $\eta = 20$	PSNR	20.124	29.355	34.217	24.728	34.460	27.264	34.465	35.703	36.191	**36.484**
		MSSIM	0.4599	0.8331	0.9291	0.6189	0.9299	0.7401	0.9358	0.9411	0.9542	**0.9565**
		MFSIM	0.7313	0.8936	0.9575	0.8000	0.9682	0.8866	0.9656	0.9729	0.9756	**0.9773**
		SAM	25.205	7.0915	4.8462	15.492	4.7633	11.523	4.4364	3.9780	3.7887	**3.6863**
		Time	-	59.374	243.54	2.4560	29.093	5.8211	644.80	0.6525	**0.3370**	4.6615
Case 3	$\sigma \sim U$ (0.1,0.2)	PSNR	16.191	26.003	31.098	23.846	31.275	30.760	32.209	32.537	33.630	**34.270**
		MSSIM	0.2536	0.6879	0.8756	0.5575	0.8676	0.8516	0.9134	0.8990	0.9310	**0.9379**
		MFSIM	0.6008	0.8049	0.9275	0.7719	0.9456	0.9418	0.9496	0.9529	0.9614	**0.9660**
		SAM	35.723	9.1650	6.3804	16.364	6.2691	6.7892	4.8065	5.3489	4.7269	**4.4690**
		Time	-	62.464	243.46	2.0414	18.791	4.8703	1173.4	0.6298	**0.3412**	4.9486

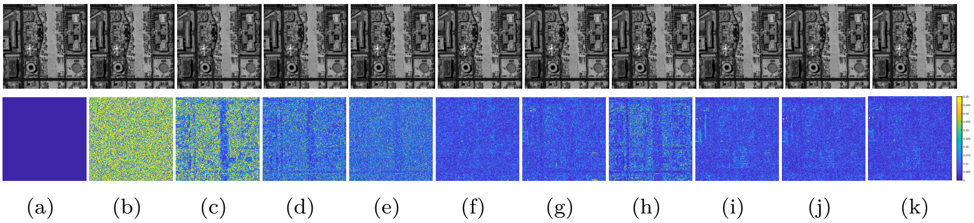


Figure 3.13: Denoised images (the first row) and residual images (the second row) for the WDCM dataset, Case 1 with $\sigma = 0.05$, using different methods. (a) Original band 95, (b) Noisy band 95, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.

Parameters Setting

All the parameters used for the comparison methods follow either the authors' source code settings or are experimentally evaluated according to the reference recommendations. SSSLRR-DWT uses three parameters, and they are set as follows for all the simulated and real datasets experiments, the regularization parameters λ_1, λ_ω, and μ are set to 0.07, 0.3, and 0.5, respectively. We give the reason for this decision in the discussion part.

Results on WDCM Dataset

Table 3.3 presents the PSNR, MSSIM, MFSIM, SAM, and running time of SSSLRR-DWT and all comparison methods for the WDCM dataset. The highest values of PSNR, MSSIM, and MFSIM and the lowest values of SAM and running time are shown in bold. SSSLRR-DWT outperforms other methods in terms of all metrics. Also, SSSLRR-DWT yields the best performance for all noise settings. Specifically, SSSLRR-DWT achieves an approximately 0.6 dB (Case 1 and 3) and 5 dB (Case 2), 0.5 dB (Case 2) and 2 dB (Case 1 and 3) improvement in PSNR, and a few percent of MSSIM and MFSIM in comparison with the second-best methods, FastHyDe and HyRes. Figs.3.9-3.11 show the PSNR, SSIM, and FSIM for each band in Cases 1-3 for the WDCM dataset. SSSLRR-DWT (red curve) has the best PSNR, SSIM, and FSIM for almost all the bands. This also shows the effectiveness of SSSLRR-DWT for HSI denoising. Figs. 3.18-3.20 show the spectral signatures of pixel (173, 150) for the WDCM dataset with different noise settings. From Figs. 3.18-3.20 it can be clearly observed that the proposed SSSLRR-DWT yields higher similarity of the spectral signatures with the clean pixel than the other denoising methods.

In terms of visual quality, Figs. 3.12, 3.14, and 3.16 show the pseudocolor images of the WDCM dataset denoised by nine different methods under Case 1 with $\sigma = 0.2$, Case 2 with $\sigma = 1$, $\eta = 20$, and Case 3 with $\sigma \sim U(0.1, 0.2)$, respectively. The same area of each subfigure marked with a green box was enlarged for a better comparison.

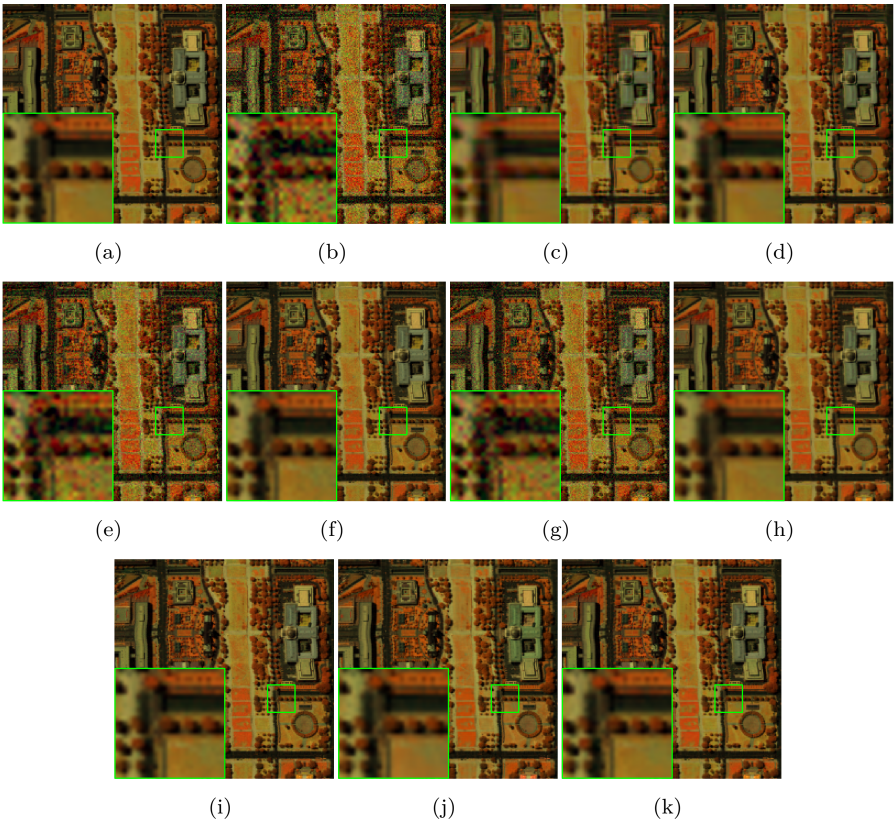


Figure 3.14: Denoising for the WDCM dataset, Case 2 with $\sigma = 1$, $\eta = 20$ using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 70, G: 120, B: 170), (b) Noisy image, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.

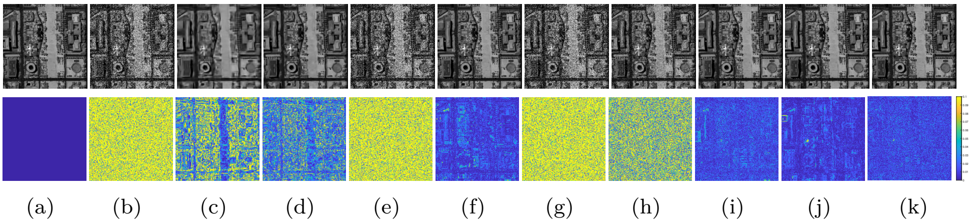


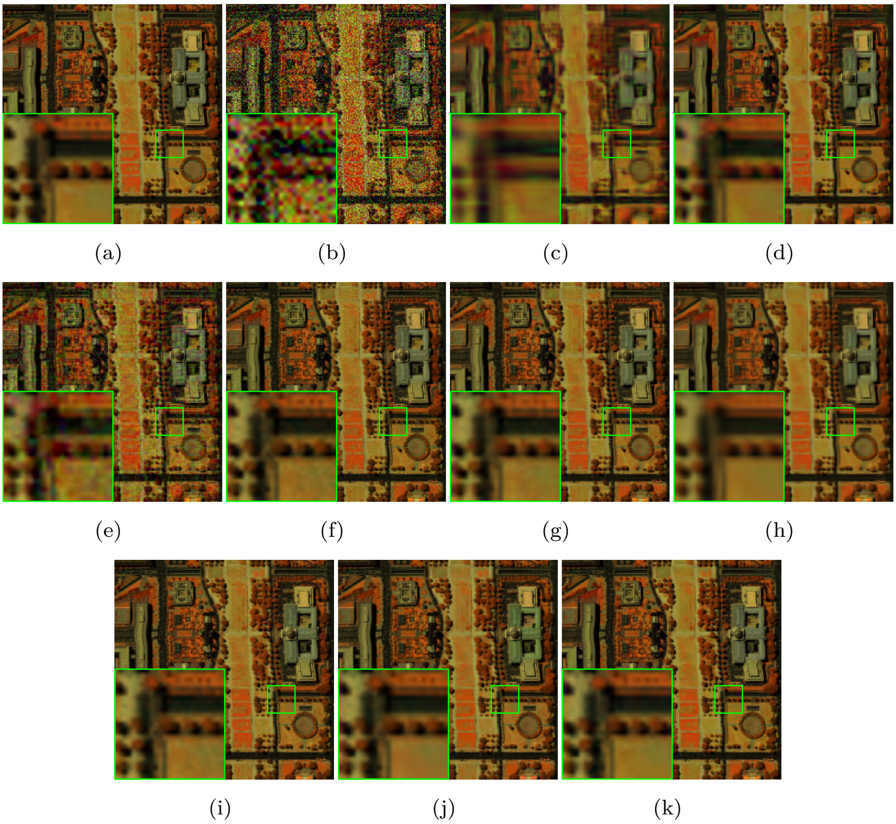
Figure 3.15: Denoised images (the first row) and residual images (the second row) for the WDCM dataset, Case 2 with $\sigma = 1$, $\eta = 10$, using different methods. (a) Original band 95, (b) Noisy band 95, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.

For LRTA there is still noise in the denoised images as can be seen in Figs. 3.12, 3.14, and 3.16. BM3D removes the noise but gives oversmooth denoised images. All the remaining methods effectively remove the noise. It is visually difficult to compare the comparison denoising methods. To show the denoising results, the difference between each original band and the denoised band is used, i.e. residual image. Figs. 3.13, 3.15, and 3.17 show the band 95 of the denoising results and its residual images under Case 1 with $\sigma = 0.05$, Case 2 with $\sigma = 1$, $\eta = 10$, and Case 3 with $\sigma \sim U(0.1, 0.2)$, respectively. By comparing the residual images, SSSLRR-DWT achieves the smallest residual in all cases. NAIRLMA HyRes, and FastHyDe also do good job in denoising in most cases and achieve small residual. However, for the denoising WDCM dataset in Case 2 (see Fig. 3.15), NAIRLMA and FastHyDe are slightly worse than HyRes and SSSLRR-DWT.

UP Dataset

Table 3.4 presents the PSNR, MSSIM, MFSIM, SAM, and running time of SSSLRR-DWT and all comparison denoising methods for the UP dataset. The highest values of PSNR, MSSIM, and MFSIM and the lowest values of SAM and running time are highlighted in bold. It can be seen that the SSSLRR-DWT gets the best results in terms of all metrics. SSSLRR-DWT achieves an approximately 0.5-2 dB gain in PSNR. This further validates that our proposed SSSLRR-DWT method can remove noise effectively. Figs. 3.21-3.23 show the PSNR, SSIM, and FSIM of each band in Cases 1-3 for the UP dataset. Compared with all the other methods, SSSLRR-DWT achieves higher PSNR, SSIM, and FSIM (red curve) for almost all the bands, showing the robustness of SSSLRR-DWT for HSI denoising.

Figs. 3.24, 3.26, and 3.28 show the pseudocolor images of the UP dataset denoised by nine different methods under Case 1 with $\sigma = 0.2$, Case 2 with $\sigma = 1$, $\eta = 20$, and Case 3 with $\sigma \sim U(0.1, 0.2)$, respectively. The same area of each subfigure marked with a green box is enlarged for a better comparison. As shown in Figs. 3.24, 3.26, and 3.28, BM3D smooths as before some details information of the original image, and


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

(e) (f) (g) (h)

(i) (j) (k)

Figure 3.16: Denoising for the WDCM dataset, Case 3 with $\sigma \sim U(0.1, 0.2)$, using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 70, G: 120, B: 170), (b) Noisy image, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.

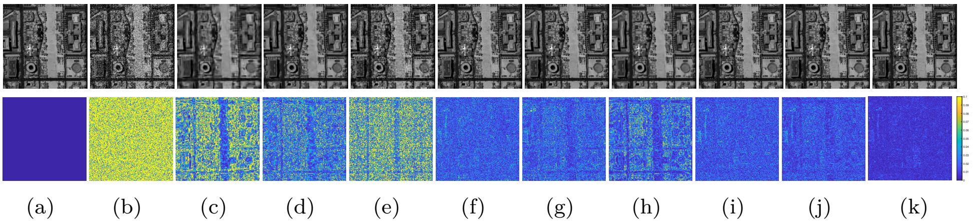


Figure 3.17: Denoised images (the first row) and residual images (the second row) for the WDCM dataset, Case 3 with $\sigma \sim U(0.1, 0.2)$, using different methods. (a) Original band 95, (b) Noisy band 95, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.

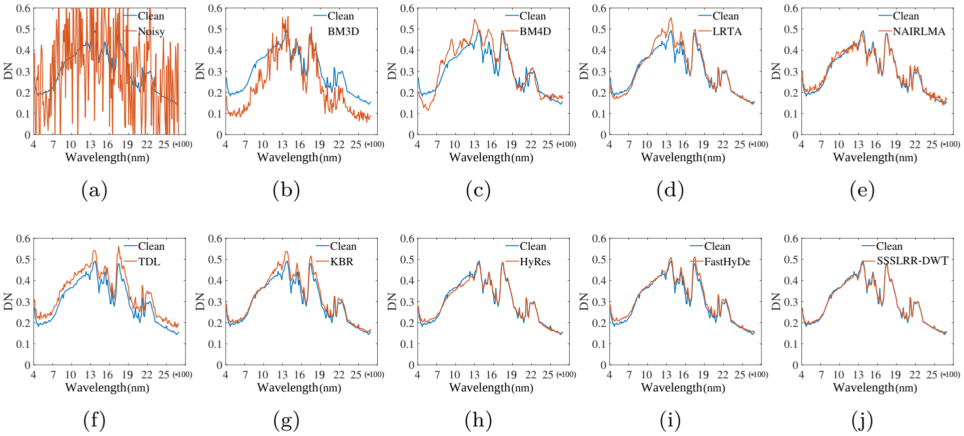
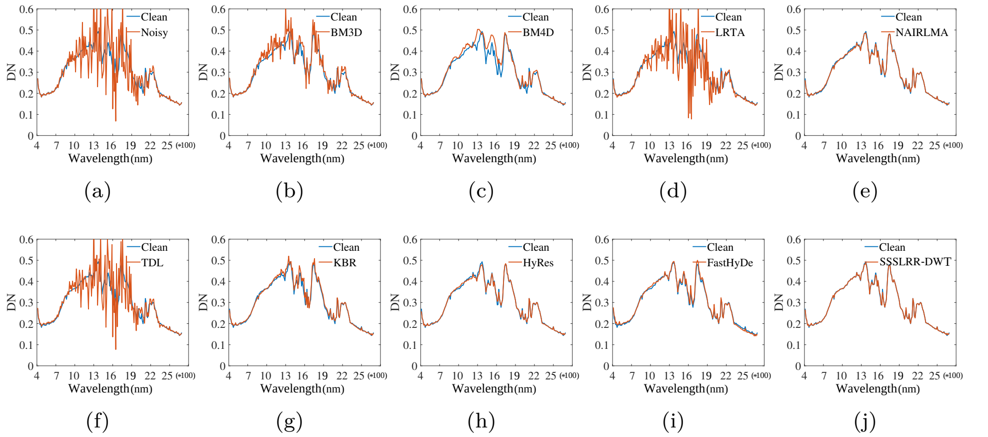


Figure 3.18: Spectrum of pixel (173, 150) for the WDCM dataset, Case 1 with $\sigma = 0.2$, using different denoising methods. (a) Noisy image, (b) BM3D, (c) BM4D, (d) LRTA, (e) NAIRLMA (f), TDL, (g), KBR, (h) HyRes, (i) FastHyDe, and (j) SSSLRR-DWT.

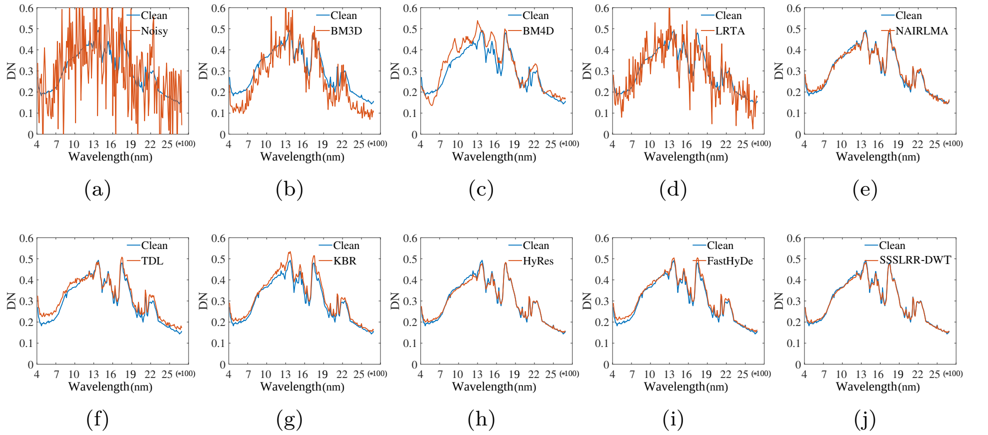
LRTA still cannot completely remove the noise. All the other methods get satisfactory denoising results in most cases. Figs. 3.25, 3.27, and 3.29 show the band 51 of the denoising results and its residual images under Case 1 with $\sigma = 0.05$, Case 2 with $\sigma = 1$, $\eta = 10$, and Case 3 with $\sigma \sim U(0.1, 0.2)$, respectively. As observed, the proposed SSSLRR-DWT method achieves good denoising result.



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

(f) (g) (h) (i) (j)

Figure 3.19: Spectrum of pixel (173, 150) for the WDCM dataset, Case 2 with $\sigma = 1$, $\eta = 20$, using different denoising methods. (a) Noisy image, (b) BM3D, (c) BM4D, (d) LRTA, (e) NAIRLMA (f), TDL, (g), KBR, (h) HyRes, (i) FastHyDe, and (j) SSSLRR-DWT.



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

(f) (g) (h) (i) (j)

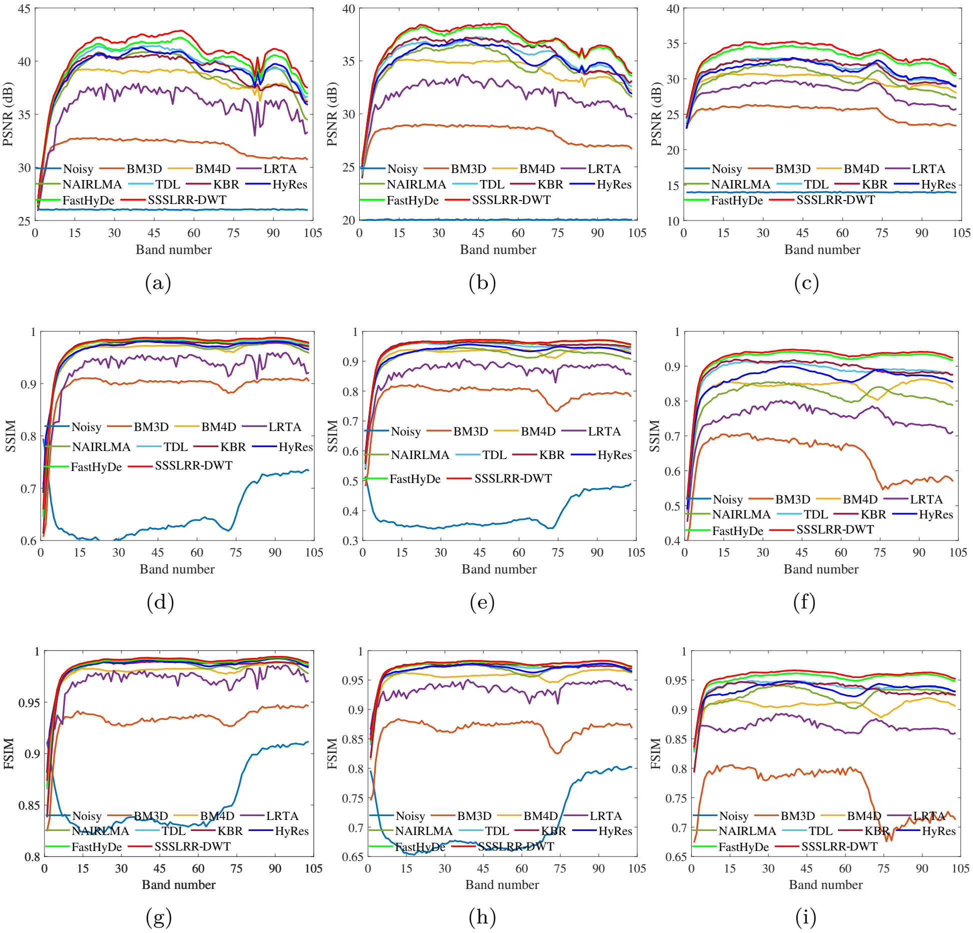
Figure 3.20: Spectrum of pixel (173, 150) for the WDCM dataset, Case 3 with $\sigma \sim U(0.1, 0.2)$, using different denoising methods. (a) Noisy image, (b) BM3D, (c) BM4D, (d) LRTA, (e) NAIRLMA (f), TDL, (g), KBR, (h) HyRes, (i) FastHyDe, and (j) SSSLRR-DWT.

3.3.2 Real Datasets Experiments

Data Description

The Indian Pines and the Urban datasets are used in the real datasets experiments. The Indian Pines dataset[9] was collected by the Airborne Visible/Infrared Imaging

[9] http://lesun.weebly.com/hyperspectral-data-set.html

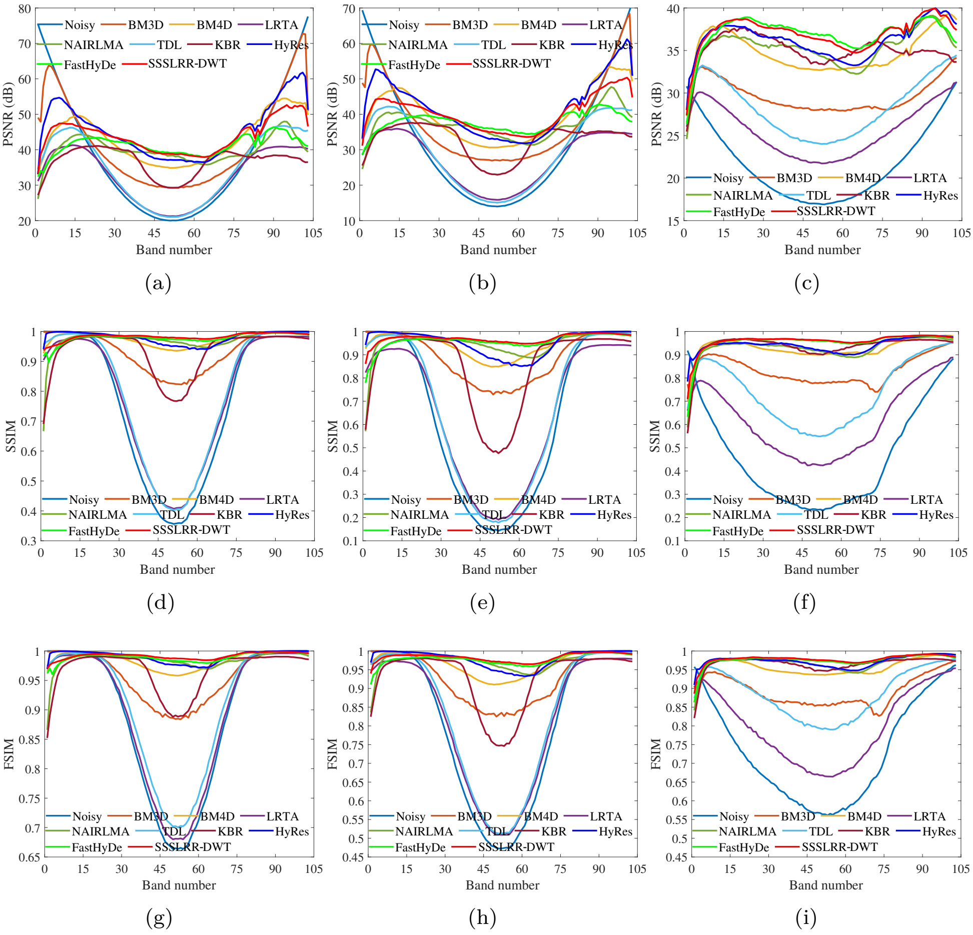


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.21: PSNR (a)-(c), SSIM (d)-(f), and FSIM (g)-(i) values of each band of the denoised UP dataset. The first to third column correspond to the Case 1 with $\sigma = 0.05$, $\sigma = 0.1$, and $\sigma = 0.2$.

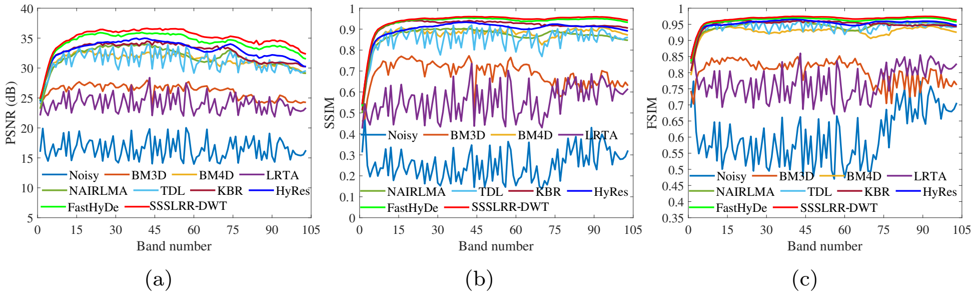


(a) (b) (c)

(d) (e) (f)

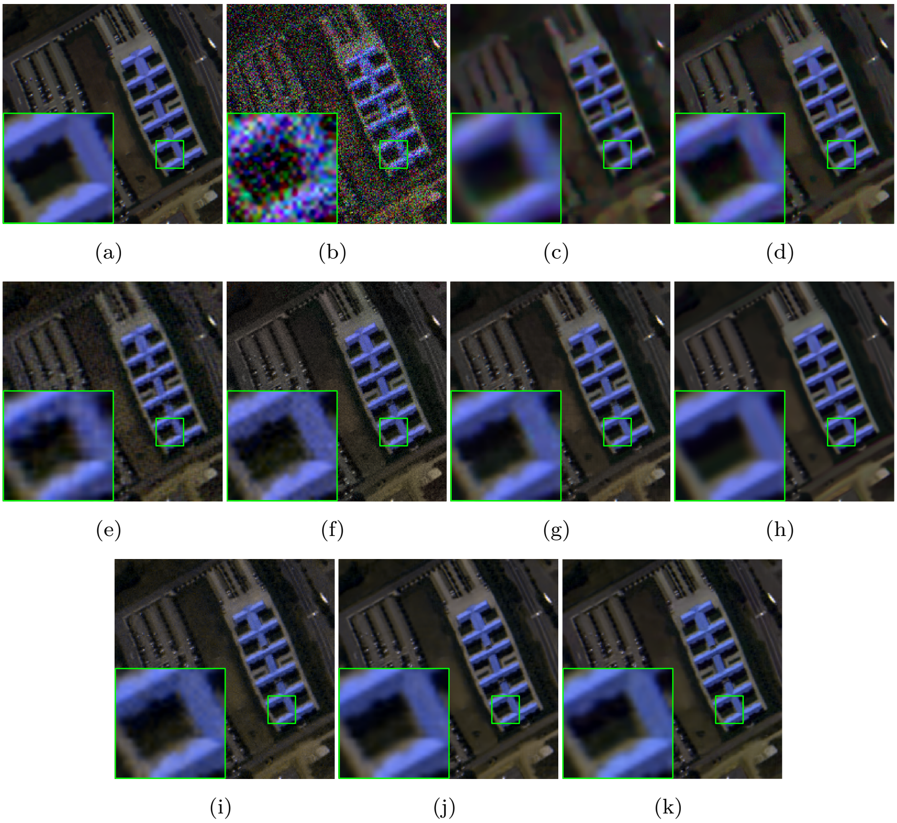
(g) (h) (i)

Figure 3.22: PSNR (a)-(c), SSIM (d)-(f), and FSIM (g)-(i) values of each band of the denoised UP dataset. The first to third column correspond to the Case 2 with $\sigma = 0.5$, $\eta = 10$, $\sigma = 1$, $\eta = 10$, and $\sigma = 1$, $\eta = 20$.



(a) (b) (c)

Figure 3.23: PSNR (a), SSIM (b), and FSIM (c) values of each band of the denoised UP dataset in Case 3 with $\sigma \sim U(0.1, 0.2)$.



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

(e) (f) (g) (h)

(i) (j) (k)

Figure 3.24: Denoising for the UP dataset, Case 1 with $\sigma = 0.2$, using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 64, G: 45, B: 10), (b) Noisy image, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.

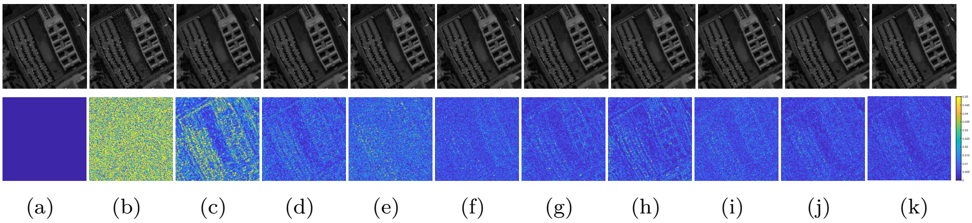


Figure 3.25: Denoised images (the first row) and residual images (the second row) for the UP dataset, Case 1 with $\sigma = 0.05$, using different methods. (a) Original band 51, (b) Noisy band 51, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.

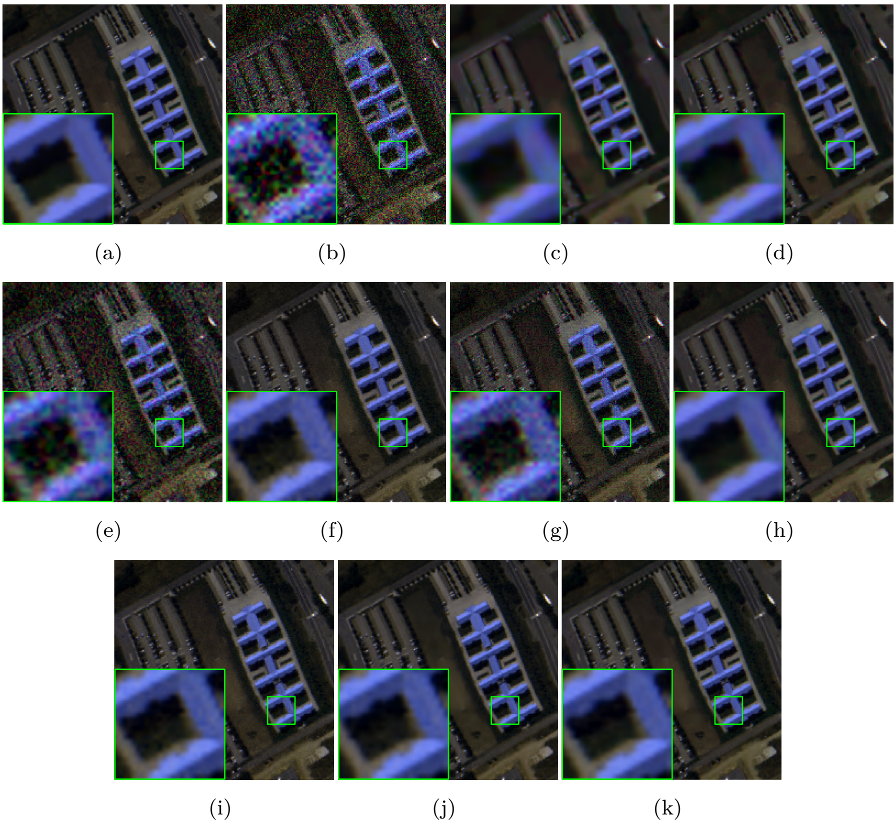
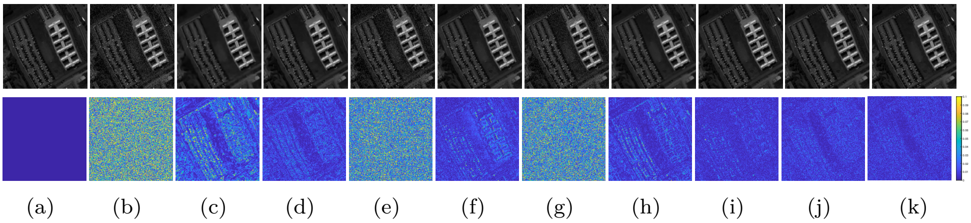
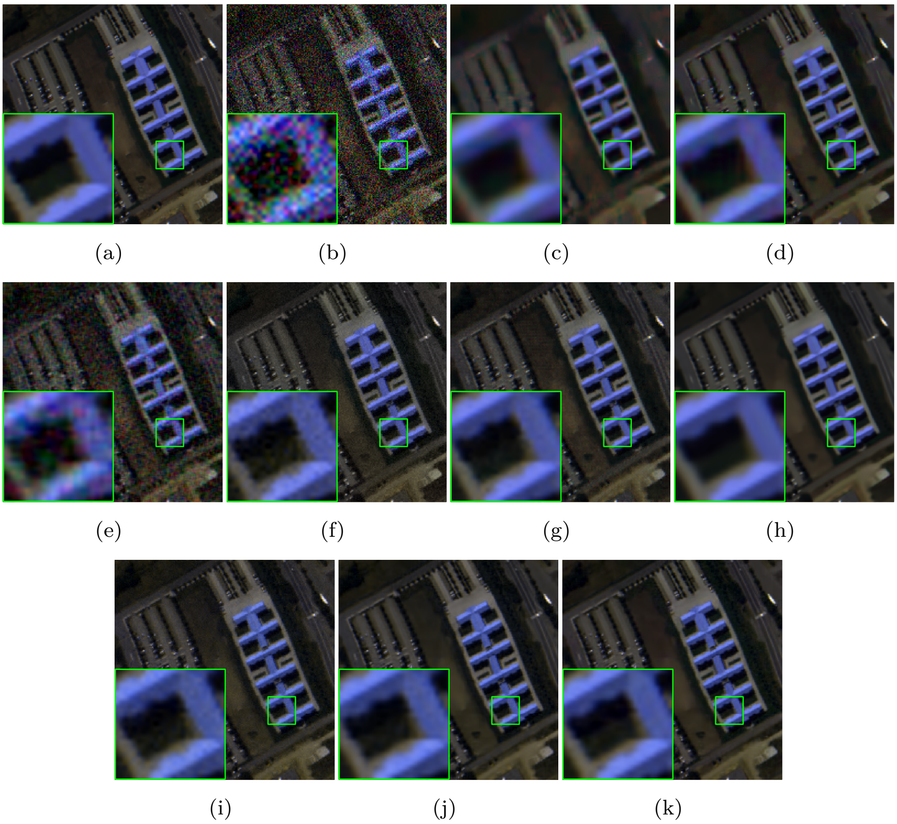


Figure 3.26: Denoising for the UP dataset, Case 2 with $\sigma = 1$, $\eta = 20$ using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 64, G: 45, B: 10), (b) Noisy image, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k)

Figure 3.27: Denoised images (the first row) and residual images (the second row) for the UP dataset, Case 2 with $\sigma = 1$, $\eta = 10$, using different methods. (a) Original band 30, (b) Noisy band 30, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.



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

(e) (f) (g) (h)

(i) (j) (k)

Figure 3.28: Denoising for the UP dataset, Case 3 with $\sigma \sim U(0.1, 0.2)$, using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 64, G: 45, B: 10), (b) Noisy image, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.

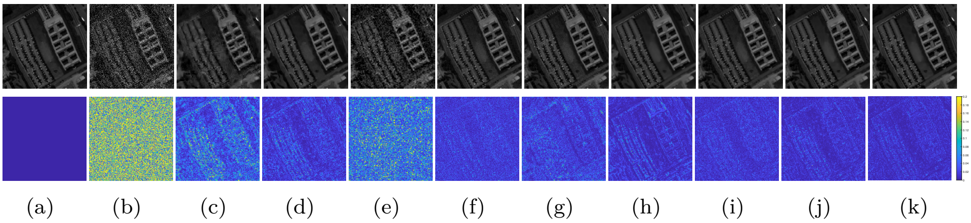


Figure 3.29: Denoised images (the first row) and residual images (the second row) for the UP dataset, Case 3 with $\sigma \sim U(0.1, 0.2)$, using different methods. (a) Original band 51, (b) Noisy band 51, (c) BM3D, (d) BM4D, (e) LRTA, (f) NAIRLMA (g), TDL, (h), KBR, (i) HyRes, (j) FastHyDe, and (k) SSSLRR-DWT.

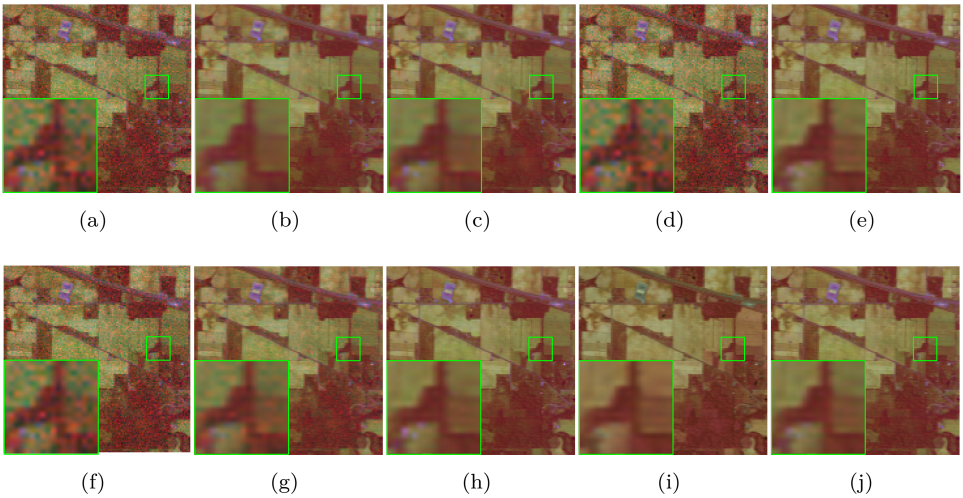
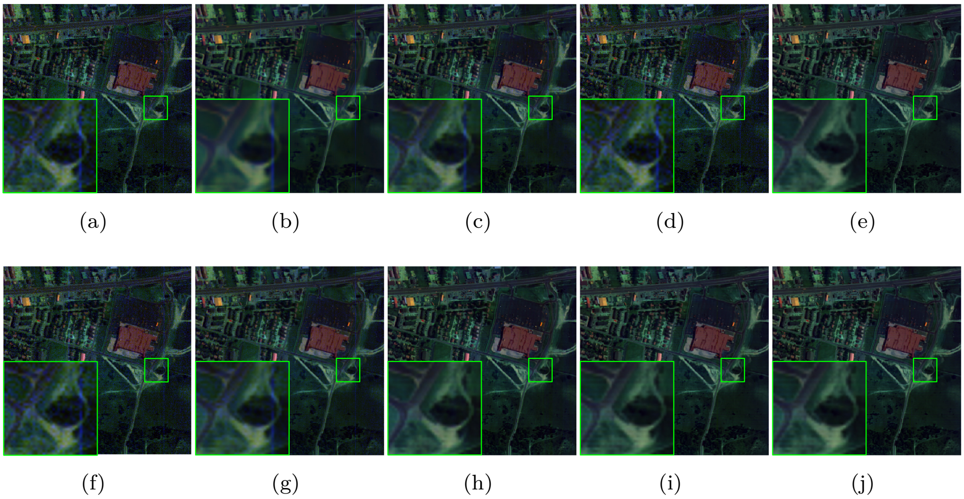


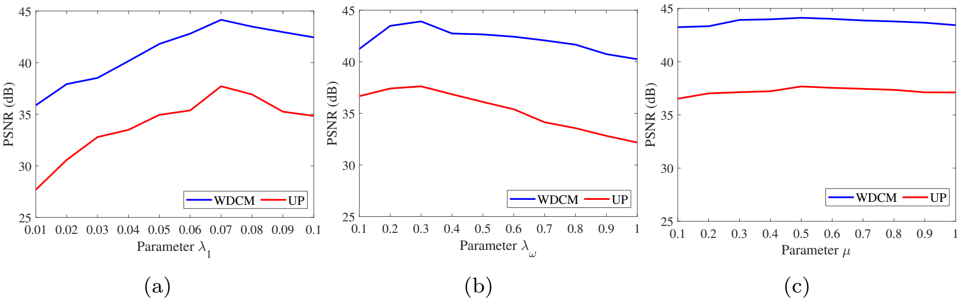
Figure 3.30: Denoising for the Indian Pines dataset. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 220, G: 144, B: 3), (b) BM3D, (c) BM4D, (d) LRTA, (e) NAIRLMA (f), TDL, (g), KBR, (h) HyRes, (i) FastHyDe, and (j) SSSLRR-DWT.



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

(f) (g) (h) (i) (j)

Figure 3.31: Denoising for the Urban dataset. The small green square is a zoomed-in area shown in the big square. (a) Original pseudo-color image (R: 30, G: 110, B: 207), (b) BM3D, (c) BM4D, (d) LRTA, (e) NAIRLMA (f), TDL, (g), KBR, (h) HyRes, (i) FastHyDe, and (j) SSSLRR-DWT.



(a) (b) (c)

Figure 3.32: PSNR as a function of the parameters (a) λ_1, (b) λ_ω (is a constant and is related to the weight ω_i), and (c) μ for SSSLRR-DWT denoising method for the WDCM and the UP datasets.

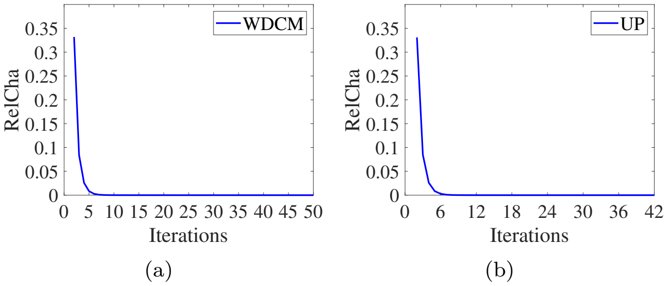


Figure 3.33: RelCha values with respect to the number of iterations. (a) WDCM dataset, (b) UP dataset.

Spectrometer (AVIRIS) sensor and the image contains 145×145 pixels and 220 spectral bands. The Urban dataset[10] was collected by HYDICE and contains 307 × 307 pixels and 210 spectral bands. Both Indian Pines and Urban datasets are contaminated by various types of noise, such as the Gaussian, Poisson, missing pixels, salt and pepper, and stripped noises.

Results on Indian Pines Dataset

Fig. 3.30 depicts the Pseudocolor images for both original and denoised Indian Pines datasets . As shown in Fig. 3.30, BM3D, LRTA, TDL, and KBR can remove some noise, and BM4D smooths some details. NAIRLMA, HyRes, FastHyDe, and SSSLRR-DWT achieve satisfactory denoising results. The area marked by a green box reveals that BM3D, BM4D, LRTA, TDL, and KBR have edge distortion or blurring, and NAIRLMA, HyRes, FastHyDe, and SSSLRR-DWT preserve details like sharp and clear edges.

Urban Dataset

Pseudocolor images for both original corrupted by noise and strips and the denoised Urban dataset are shown in Fig. 3.31 for each method. The denoised images of BM3D, BM4D, LRTA, TDL, and KBR still have blue stripes. NAIRLMA, HyRes, FastHyDe, and SSSLRR-DWT remove Gaussian noise and blue strip and provide good denoised results. This also demonstrates that the proposed SSSLRR-DWT method can remove the Gaussian noise and strips in real HSI. To further illustrate the denoising effect, Fig. 3.34 shows the horizontal mean profiles of band 206 before and after denoising. The horizontal mean profile of a band is obtained by calculating the mean of each row. Fig. 3.34(a) has rapid fluctuations that are due to the mixed noise, and all the denoising methods diminish them. BM3D, NAIRLMA, HyRes, and SSSLRR-DWT achieve smoother curves.

[10] http://lesun.weebly.com/hyperspectral-data-set.html

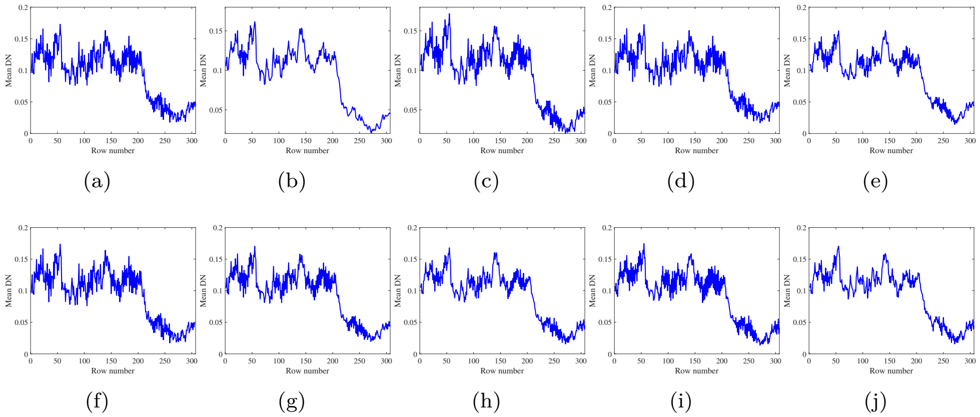


Figure 3.34: Horizontal mean profiles of band 206 of the Urban dataset. (a) Original data, denoising results of (b) BM3D, (c) BM4D, (d) LRTA, (e) NAIRLMA (f), TDL, (g), KBR, (h) HyRes, (i) FastHyDe, and (j) SSSLRR-DWT.

3.3.3 Discussion

Convergence of the Proposed Method

The convergence is studied on the WDCM and UP datasets. Fig. (3.33) displays the RelCha of two denoised results for the WDCM and UP datasets for the Case 2 with $\sigma = 1$, $\eta = 10$. RelCha is the relative change between iterations, i.e., $\|\mathbf{W}^{k+1} - \mathbf{W}^k\|_F^2 / \|\mathbf{W}^k\|_F^2$, where $\mathbf{W}^k$ are the wavelet coefficient at the kth iteration. The RelCha values for SSSLRR-DWT monotonically decrease with iterations until stopping criterion is satisfied (RelCha $\leq \epsilon$).

Parameter Analysis

For SSSLRR-DWT, three parameters have to be obtained or chosen for the HSI denoising. Here we analyze parameter selection for the simulated WDCM and UP datasets in Case 2 with $\sigma = 1$, $\eta = 10$.

λ_1 is related to the strength of sparse wavelet coefficients. SSSLRR-DWT is tested using different values of λ_1 over [0.01, 0.1] and shown in Fig. 3.32(a). SSSLRR-DWT achieves the best performance when λ_1 is set as 0.07.

λ_ω is a constant and is related to the weight ω_i and is correlated with the low-rank characteristic of HSI. Fig. 3.32(b) presents the denoised results based on the different λ_ω. SSSLRR-DWT achieves the highest PSNR when λ_ω is 0.3.

Parameter μ is used for the ADMM multipliers. Fig. 3.32(c) shows the robustness of the proposed SSSLRR-DWT method to the changes in μ. When μ lies in the range of [0.5, 0.6], SSSLRR-DWT achieves a high PSNR value.

Further Discussion

In this section, we further compare SSSLRR-DWT with four denoising methods for removing Gaussian noise with high noise level and mixed noise. The compared denoising methods are low-rank matrix recovery (LRMR[11]) [32], non-local low-rank tensor approximation (NGmeet[12]) [81], tensor low-rank constraint and ℓ_0 total variation (TLR-ℓ_0TV[13]) [82], and SSSLRR using 3-D discrete orthogonal cosine transform (SSSLRR-DCT) instead of using 3-D discrete orthogonal wavelet transform. The noisy simulated datasets are created as follows:

1. *Case 4*: A zero-mean Gaussian noise is added to each band with standard deviation σ drawn from a uniform distribution between 0.1 and 0.4 [$\sigma \sim U(0.1, 0.4)$].
2. *Case 5* (Gaussian noise + salt and pepper noise): The Gaussian noise is added in the same way as Case 3. The salt and pepper noise is added to all the bands with noise density 10%.
3. *Case 6* (Gaussian noise + stripe): The Gaussian noise is added in the same way as Case 3. 30% of the bands are randomly selected to add the stripes. The number of stripes for the selected band is randomly sampled from 1 to 10.
4. *Case 7* (Gaussian noise + salt and pepper noise+ stripe): The Gaussian noise and salt and pepper noise are added in the same way as Case 5. The stripes are added in the same way as in Case 6.

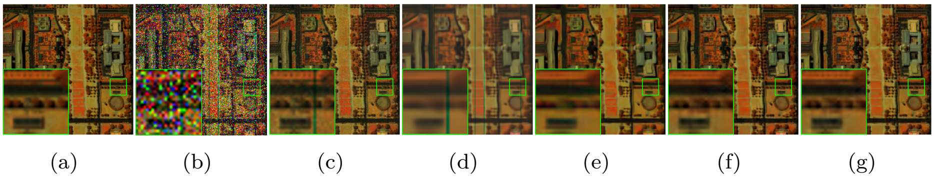


Figure 3.35: Denoising for the WDCM dataset, Case 7 using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 70, G: 120, B: 170), (b) Noisy image, (c) LRMR, (d) NGmeet, (e) TLR-ℓ_0TV, (f) SSSLRR-DCT, and (g) SSSLRR-DWT.

The simulated WDCM and UP datasets for Cases 1-7 and the real Indian Pines and Urban datasets are used in the experiments. Tables 3.5, 3.6, 3.7, and 3.8 present the

[11] https://sites.google.com/site/rshewei/home
[12] https://sites.google.com/site/rshewei/home
[13] https://openremotesensing.net/kb/codes/denoising/

Table 3.5: The quantitative evaluation results (PSNR(dB), MSSIM, MFSIM, SAM(degrees), and Time (second)) for the different denoising methods using the Washington DC Mall (WDCM) dataset. The best results are in bold typeface.

WDCM	Noise level	Metric	Noisy	LRMR	Ngmeet	TLR-ℓ_0TV	SSSLRR-DCT	SSSLRR-DWT
Case 1	$\sigma = 0.05$	PSNR	26.025	38.529	41.453	39.925	41.546	**41.681**
		MSSIM	0.7191	0.9802	**0.9910**	0.9858	0.9907	0.9908
		MFIM	0.8691	0.9879	**0.9942**	0.9913	0.9940	**0.9942**
		SAM	18.048	4.6330	2.9266	3.6415	2.9036	**2.8475**
		Time	-	24.506	38.838	1011.4	11.836	**11.754**
	$\sigma = 0.1$	PSNR	20.004	34.082	**38.089**	35.572	37.739	37.968
		MSSIM	0.4528	0.9473	**0.9805**	0.9655	0.9785	0.9794
		MFIM	0.7348	0.9700	**0.9878**	0.9789	0.9861	0.9867
		SAM	30.350	7.3018	**3.8188**	5.4473	4.1484	4.0887
		Time	-	25.062	39.104	1031.1	12.882	**12.593**
	$\sigma = 0.2$	PSNR	13.983	29.687	33.990	33.277	34.015	**34.518**
		MSSIM	0.2061	0.8734	0.9544	0.9516	0.9547	**0.9582**
		MFIM	0.5644	0.9335	0.9717	0.9697	0.9721	**0.9734**
		SAM	46.309	11.069	5.8062	5.9847	5.8048	**5.6292**
		Time	-	24.571	48.150	1035.6	13.659	**13.466**
Case 2	$\sigma = 0.5$, $\eta = 10$	PSNR	28.840	38.094	39.511	43.957	45.966	**46.750**
		MSSIM	0.9321	0.9825	0.9874	0.9962	0.9943	**0.9979**
		MFIM	0.9641	0.9886	0.9923	0.9971	0.9975	**0.9985**
		SAM	13.704	5.0619	3.6387	2.3002	1.9318	**1.5664**
		Time	-	**21.054**	35.629	1009.3	44.387	44.097
	$\sigma = 1$, $\eta = 10$	PSNR	22.820	34.332	33.472	41.545	43.727	**44.165**
		MSSIM	0.8674	0.9642	0.9655	0.9932	0.9961	**0.9968**
		MFIM	0.9306	0.9778	0.9777	0.9942	0.9975	**0.9977**
		SAM	23.863	7.4990	7.0825	2.8441	2.1047	**1.9880**
		Time	-	**23.543**	37.961	1012.8	44.229	44.075
	$\sigma = 1$, $\eta = 20$	PSNR	22.815	35.041	34.315	39.135	40.198	**40.436**
		MSSIM	0.7853	0.9670	0.9691	0.9885	0.9910	**0.9913**
		MFIM	0.8927	0.9789	0.9809	0.9914	0.9934	**0.9939**
		SAM	23.972	6.7136	6.9198	3.4272	3.1831	**3.0618**
		Time	-	**25.728**	39.083	1016.6	37.125	36.186
Case 3	$\sigma \sim U(0.1, 0.2)$	PSNR	16.371	31.519	34.364	34.672	35.435	**35.544**
		MSSIM	0.3083	0.9112	0.9573	0.9549	0.9659	**0.9661**
		MFIM	0.6423	0.9518	0.9734	0.9738	0.9781	**0.9782**
		SAM	39.689	9.3786	6.2681	6.1109	5.2643	**5.2188**
		Time	-	24.746	37.458	1014.6	13.418	**12.620**

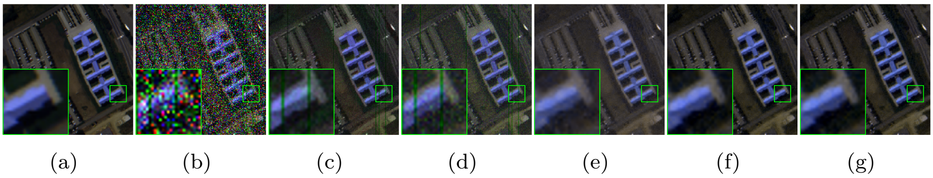


Figure 3.36: Denoising for the UP dataset, Case 7 using different methods. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 70, G: 120, B: 170), (b) Noisy image, (c) LRMR, (d) NGmeet, (e) TLR-ℓ_0TV, (f) SSSLRR-DCT, and (g) SSSLRR-DWT.

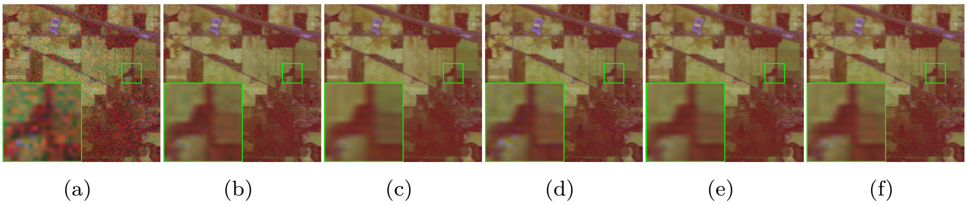


Figure 3.37: Denoising for the Indian Pines dataset. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 220, G: 144, B: 3), (b) LRMR, (c) NGmeet, (d) TLR-ℓ_0TV, (e) SSSLRR-DCT, and (f) SSSLRR-DWT.

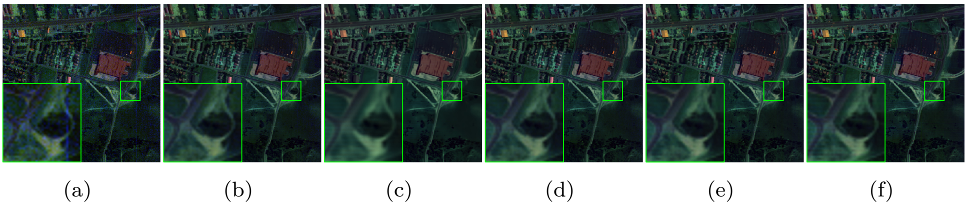


Figure 3.38: Denoising for the Urban dataset. The small green square is a zoomed-in area shown in the big square. (a) Original pseudocolor image (R: 30, G: 110, B: 207), (b) LRMR, (c) NGmeet, (d) TLR-ℓ_0TV, (e) SSSLRR-DCT, and (f) SSSLRR-DWT.

PSNR, MSSIM, MFSIM, SAM, and running time of SSSLRR-DWT and all comparison methods for the WDCM and UP datasets. The highest values of PSNR, MSSIM, and MFSIM and the lowest values of SAM and running time are represented in bold. It can be seen that SSSLRR-DWT yields the best denoising results in most cases. NGmeet yields good denoising results for low Gaussian noise cases. SSSLRR-DWT is slightly better than SSSLRR-DCT. Both of them obtain good denoising results. This means that SSSLRR using both 3-D discrete orthogonal wavelet transform and 3-D discrete orthogonal cosine transform can achieve good denoising performance. Figs.

Table 3.6: The quantitative evaluation results (PSNR(dB), MSSIM, MFSIM, and SAM(degrees)) for the different denoising methods using the Washington DC Mall (WDCM) dataset. The best results are in bold typeface.

Noise level	Metric	Noisy	LRMR	NGmeet	TLR-ℓ_0TV	SSSLRR-DCT	SSSLRR-DWT
Case 4	PSNR	11.376	28.233	29.194	29.613	32.381	**32.490**
	MSSIM	0.1831	0.8357	0.8809	0.8826	0.9321	**0.9345**
	MFIM	0.5228	0.9162	0.9317	0.9338	0.9591	**0.9599**
	SAM	53.557	12.914	7.5161	8.8449	7.1005	**6.7911**
	Time	-	30.622	44.436	1135.7	12.793	**12.586**
Case 5	PSNR	12.829	29.356	28.571	29.787	31.633	**32.053**
	MSSIM	0.1792	0.8766	0.8633	0.8886	0.9241	**0.9251**
	MFIM	0.5766	0.9432	0.9206	0.9350	0.9532	**0.9562**
	SAM	43.896	9.4171	9.3268	8.6681	7.4483	**7.2766**
	Time	-	30.191	44.701	1143.3	12.873	**12.191**
Case 6	PSNR	16.370	29.419	26.798	30.266	33.502	**33.851**
	MSSIM	0.3104	0.8935	0.796	0.8988	0.9433	**0.9515**
	MFIM	0.6432	0.9453	0.8828	0.9416	0.9665	**0.9692**
	SAM	39.813	9.8694	8.0446	8.4550	6.5269	**6.0191**
	Time	-	29.913	43.974	1133.5	12.491	**12.448**
Case 7	PSNR	12.698	27.659	24.205	28.341	30.636	**31.309**
	MSSIM	0.1714	0.8483	0.8208	0.8563	0.9077	**0.9139**
	MFIM	0.5694	0.9318	0.9388	0.9175	0.9439	**0.9501**
	SAM	44.373	10.215	11.504	9.7621	7.8932	**7.8694**
	Time	-	30.145	44.338	1138.2	12.752	**12.381**

3.35, 3.36, 3.37, and 3.38 show the pseudocolor images of the simulated WDCM and UP datasets for Case 7 and the real Indian Pines and Urban datasets denoised by five different methods. The same area of each subfigure marked with a green box was enlarged for a better comparison. From Figs. 3.35 and 3.36 we can see that LRMR can remove most stripes, but there is still Gaussian noise and salt and pepper noise in the denoised image. NGmeet removes Gaussian noise and salt and pepper noise effectively for the WDCM dataset and removes most Gaussian noise and salt and pepper noise for the UP dataset and can not remove stripes and gives an over smoothed denoised image. TLR-ℓ_0TV, SSSLRR-DCT, and SSSLRR-DWT can remove Gaussian noise, salt and pepper noise, and stripe efficiently and give good denoising results. From Figs. 3.37 and 3.38 we can see that all the five denoising methods achieve satisfactory denoising results for the real Indian Pines and Urban datasets.

Running Time

The running times (in seconds) for SSSLRR-DWT and the competitive methods applied to the WDCM and the UP datasets are evaluated in this section. All the exper-

Table 3.7: The quantitative evaluation results (PSNR(dB), MSSIM, MFSIM, SAM(degrees), and Time (second)) for the different denoising methods using the University of Pavia (UP) dataset. The best results are in bold typeface.

UP	Noise level	Metric	Noisy	LRMR	Ngmeet	TLR-ℓ_0TV	SSSLRR-DCT	SSSLRR-DWT
Case 1	$\sigma = 0.05$	PSNR	26.026	36.588	**39.090**	37.1927	38.143	38.807
		MSSIM	0.6512	0.9538	**0.9772**	0.9611	0.9682	0.9728
		MFIM	0.8549	0.9802	**0.9891**	0.9834	0.9852	0.9875
		SAM	13.900	3.9784	**2.7183**	3.5889	3.2753	3.0070
		Time	-	22.326	35.553	966.89	4.3200	**4.2892**
	$\sigma = 0.1$	PSNR	20.002	32.279	**36.173**	34.131	35.356	35.704
		MSSIM	0.3880	0.8862	**0.9611**	0.9288	0.9514	0.9520
		MFIM	0.7053	0.9535	**0.9804**	0.9678	0.9745	0.9751
		SAM	25.525	6.2864	**3.4468**	4.7002	4.2018	3.9670
		Time	-	23.732	37.725	971.42	4.8847	**4.5737**
	$\sigma = 0.2$	PSNR	13.980	27.753	32.484	31.681	32.605	**32.997**
		MSSIM	0.1700	0.7473	0.9152	0.9088	0.9164	**0.9232**
		MFIM	0.5242	0.8965	0.9509	0.9428	0.9514	**0.9567**
		SAM	42.357	9.8416	5.2406	5.7382	5.2012	**4.9689**
		Time	-	24.493	43.302	968.23	4.9211	**4.8983**
Case 2	$\sigma = 0.5$, $\eta = 10$	PSNR	26.144	35.085	35.367	39.110	40.472	**41.202**
		MSSIM	0.7731	0.9348	0.9532	0.9746	0.9788	**0.9822**
		MFIM	0.8928	0.9712	0.9770	0.9842	0.9894	**0.9914**
		SAM	13.663	4.9504	4.6545	2.8983	2.5342	**2.3102**
		Time	-	23.433	35.457	971.22	14.682	**13.858**
	$\sigma = 1$, $\eta = 10$	PSNR	20.135	31.435	29.585	34.948	37.141	**37.721**
		MSSIM	0.6385	0.8750	0.9011	0.9566	0.9567	**0.9648**
		MFIM	0.8060	0.9452	0.9511	0.9687	0.9803	**0.9813**
		SAM	25.097	7.3800	9.0351	4.0144	3.4374	**3.2060**
		Time	-	24.753	35.889	967.97	13.491	**13.436**
	$\sigma = 1$, $\eta = 20$	PSNR	20.124	32.279	32.304	35.073	36.320	**36.484**
		MSSIM	0.4599	0.8821	0.9091	0.9387	0.9557	**0.9565**
		MFIM	0.7313	0.9505	0.9578	0.9675	0.9761	**0.9773**
		SAM	25.205	6.4972	6.6344	4.2612	3.7267	**3.6863**
		Time	-	24.339	36.044	968.80	8.7625	**4.6615**
Case 3	$\sigma \sim U(0.1, 0.2)$	PSNR	16.191	29.575	33.357	33.416	34.164	**34.270**
		MSSIM	0.2536	0.8147	0.9301	0.9208	0.9372	**0.9379**
		MFIM	0.6008	0.9246	0.9637	0.9624	0.9656	**0.9660**
		SAM	35.723	8.2789	5.2755	4.8358	4.4698	**4.4690**
		Time	-	23.236	33.619	968.21	6.5452	**4.9486**

Table 3.8: The quantitative evaluation results (PSNR(dB), MSSIM, MFSIM, and SAM(degrees)) for the different denoising methods using the University of Pavia (UP) dataset. The best results are in bold typeface.

Noise level	Metric	Noisy	LRMR	NGmeet	TLR-ℓ_0TV	SSSLRR-3DCT	SSSLRR-DWT
Case 4	PSNR	11.645	26.204	26.089	29.261	32.477	**32.567**
	MSSIM	0.1530	0.6893	0.7831	0.8032	0.9201	**0.9216**
	MFIM	0.4891	0.8707	0.8999	0.9199	0.9587	**0.9607**
	SAM	49.501	11.907	12.345	8.5770	5.8926	**5.8172**
	Time	-	30.341	44.308	971.14	6.8324	**5.7550**
Case 5	PSNR	12.892	28.348	23.947	30.218	31.983	**32.098**
	MSSIM	0.1468	0.7852	0.6937	0.8327	0.8910	**0.9143**
	MFIM	0.5326	0.9156	0.8878	0.9319	0.9483	**0.9561**
	SAM	41.225	8.9158	11.786	7.7761	6.3496	**5.5627**
	Time	-	30.237	34.630	970.26	5.3745	**4.5971**
Case 6	PSNR	16.239	28.755	31.594	30.830	32.212	**32.843**
	MSSIM	0.2529	0.7969	0.9136	0.8925	0.9201	**0.9242**
	MFIM	0.6014	0.9175	0.9590	0.9228	0.9644	**0.9651**
	SAM	35.805	8.7441	5.4931	7.3799	5.4581	**5.2077**
	Time	-	29.833	34.250	969.71	5.7278	**5.3020**
Case 7	PSNR	12.891	27.691	23.759	28.871	30.103	**30.933**
	MSSIM	0.1465	0.7654	0.6623	0.7899	0.8839	**0.9066**
	MFIM	0.5321	0.9069	0.8744	0.9142	0.9453	**0.9553**
	SAM	41.441	9.4280	11.9194	8.9173	8.1103	**7.2942**
	Time	-	30.293	34.020	972.06	5.8311	**5.1261**

iments are run on a Linux computer using MATLAB R2019b. The computer has an eight-core Intel CPU 3.2 GHz and 64-GB RAM. Tables 3.3, 3.4, 3.5, 3.6, 3.7, and 3.8 show the running times. FastHyDe is the fastest method using less than 1 s for denoising the WDCM and the UP datasets, while KBR is the slowest method. SSSLRR-DWT has an average running time in the middle between the slow group methods (BM4D and KBR) and the fast group methods (FastHyDe, HyRes, and LRTA) and is 24.97 s for the WDCM dataset and 7.24 s for the UP dataset.

3.4 Conclusions

In this chapter, a denoising method based on sparse spectral-spatial and low-rank representations (SSSLRR) using 3-D orthogonal transform (3-DOT) was proposed. SSSLRR can be effectively used to remove Gaussian and mixed noise. 3-DOT is used for SSSLRR to decompose noisy HSI to sparse transform coefficients. 3-D discrete orthogonal wavelet transform (3-D DWT) and 3-D discrete orthogonal cosine transform (3-D DCT) are two examples of 3-DOT appropriate for denoising since they concentrate the signal in few transform coefficients. SSSLRR-DWT uses 3-D DWT to be the 3-DOT. SSSLRR-DWT transforms the original HSI to the wavelet domain. A weighted nuclear norm low-rank regularizer was proposed for the global image, which gives more flexibility than the nuclear norm. The weighted nuclear norm low-rank regularizer uses different weights to shrink the singular values of the wavelet coefficient matrix.

HSI has sparse and low-rank characteristics and can be preserved using the ℓ_1 penalty and the weighted nuclear norm low-rank penalty, respectively. Thus, a novel sparse and low-rank penalized model was proposed using both the ℓ_1 penalty and the weighted nuclear norm low-rank penalty. SSSLRR-DWT uses the sparse and low-rank penalized model can adaptively shrink wavelet coefficients and penalize the singular values of the wavelet coefficient matrix. Thus, SSSLRR-DWT uses the sparse and low-rank penalized model to remove noise effectively and recover the denoised image. The algorithm for SSSLRR-DWT was developed using ADMM. To analyze the effectiveness of different penalties for SSSLRR-DWT both in signal and wavelet domain, four different methods were analyzed. The analysis shows that SSSLRR-DWT uses both ℓ_1 penalty and weighted nuclear norm low-rank penalty in the wavelet domain yields better denoising results than the other methods. Moreover, a new SSSLRR-DCT denoising method was also proposed, which uses the 3-D discrete orthogonal cosine transform (3-D DCT) for SSSLRR. The analysis shows that both SSSLRR-DWT and SSSLRR-DCT have good denoising results. The experimental results indicate that our proposed method improves the denoising performance for both simulated and real noisy HSI datasets.

Chapter 4

MFA Low-Rank Based Methods for HSI Denoising

This chapter proposes four mixtures of factor analyzers (MFA) low-rank based methods for HSI denoising. Three new MFA, 2-D wavelet based MFA (WMFA-2D), and 3-D wavelet based MFA (WMFA-3D) are proposed for removing Gaussian noise. MFA uses a Gaussian mixture model to segment the original HSI into different parts, where each part follows Gaussian distribution and then utilizes a factor analyzer to get a low-rank factor loading matrix, and finally uses the inverse transformation of the matrix to get the denoised hyperspectral dataset. WMFA-2D and WMFA-3D use the MFA in the wavelet domain to remove the noise in HSI. HSIs are usually degraded by different noise types such as missing lines (ML), missing pixels (MP), salt and pepper noise (SP), and Gaussian noise. One new local spatial-spectral correlation based automatic MFA (LSSC-AMFA) is proposed for removing mixed noise. The proposed method, hierarchically, removes the mixed noise. Firstly, we develop a novel local spatial-spectral correlation (LSSC) method to remove the ML noise. Then LSSC-MFA uses the mixtures of factor analyzers (MFA) method to remove the MP, SP, and Gaussian noises. The proposed methods are evaluated by using both simulated and real hyperspectral datasets.

4.1 MFA and WMFA HSI Denoising

4.1.1 Introduction

Low-rank model based methods describe the problem of finding and exploiting low-dimensional structures in high-dimensional data have been proposed and successfully used in HSI denoising [32]. An example is hyperspectral restoration (HyRes) [39], which is a low-rank SSTV-based method. Another examples are low-rank matrix recovery (LRMR) [32], and noise-adjusted iterative low-rank matrix approximation (NAILRMA) [33]. LRMR explores the low-rank property of the HSI by lexicographically ordering a patch of the HSI into a 2-D matrix, while NAILRMA utilizes patch-wise low-rank matrix approximation and adaptive iteration factor selection to remove noise.

In this paper, we propose two low-rank model-based denoising methods for HSI, i.e., mixtures of factor analyzers (MFA) and wavelet-based MFA (WMFA). MFA, as it is used to HSI denoising, utilizes Gaussian mixture model to segment the original HSI to different parts, where each part follows a Gaussian distribution and then utilizes a factor analyzer to get a low-rank factor loading matrix, and finally uses the inverse transformation of the low-rank factor loading matrix to get the hyperspectral data without noise. WMFA is a MFA in the wavelet domain. In WMFA, the whole HSI is firstly decomposed into sub-images (approximation coefficients (AC_1) sub-image and detail coefficients sub-images) using wavelet decomposition. Secondly, using all the sub-images to compose a new hyperspectral dataset with the same size as the original HSI. Thirdly, this new dataset is put in the MFA algorithm to get a denoised dataset. Fourthly, the denoised dataset is split to different sub-images (new approximation coefficients (AC_2) sub-image and new detail coefficients sub-images) using the inverse way as the process of composition in the second step. Finally, using wavelet reconstruction transformation based on the approximation coefficients (AC_1) sub-image and new detail coefficients sub-images to reconstruct a new denoised HSI with the same size as the original HSI.

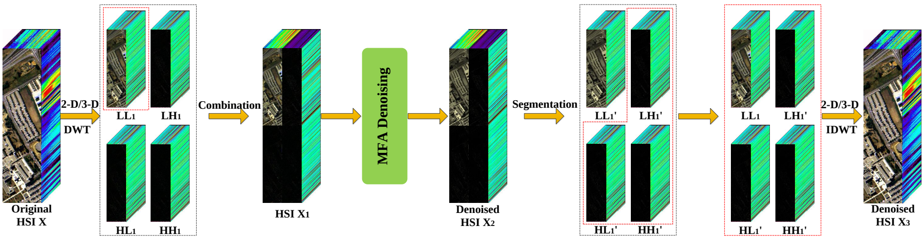


Figure 4.1: The schematic of the WMFA, which uses 1-level 2-D DWT as an input.

4.1.2 MFA Denoiser

Let $\mathbf{x}$ denote a D-dimensional spectral vector, $\mathbf{z}$ denote a d-dimensional latent vector, and $c \in \{1, ..., C\}$ denote the component indicator variable of the C factor analyzers in MFA. The standard multivariate normal prior on the latent factors is

$$p(\mathbf{z}|c) = p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}), \tag{4.1}$$

where $p(c) = \pi_c$, $\sum_{c=1}^{C} \pi_c = 1$, $\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ means that $\mathbf{z}$ is Gaussian vector with zero mean and $d \times d$ identity matrix $\mathbf{I}$ as the covariance matrix.

The MFA model can be defined as

$$\mathbf{x} = \mathbf{W}_c\mathbf{z} + \boldsymbol{\mu}_c + \boldsymbol{\epsilon}, \tag{4.2}$$

where $\boldsymbol{\epsilon}$ is noise which is independently distributed as $\mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$, $\boldsymbol{\Psi}$ is diagonal matrix which represents the independent noise variances for each of the observed variables. The parameters of the c-th factor analyzer include a mixing proportion π_c, a factor loading matrix $\mathbf{W}_c$, and mean $\boldsymbol{\mu}_c$.

By integrating out the latent variables $\mathbf{z}$, the MFA model becomes a mixture of Gaussians [83]. The parameters of MFA are estimated by using an expectation maximization (EM) algorithm [83].

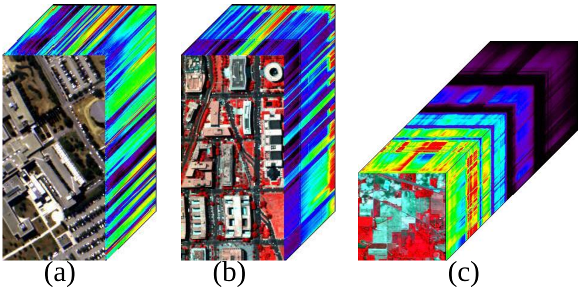


Figure 4.2: The experimental datasets. (a) University of Pavia (R: 64, G: 45, and B: 10), (b) Washington DC Mall (R: 60, G: 27, and B: 17), and (d) Indian Pines (R: 50, G: 27, and B: 17).

Table 4.1: The quantitative evaluation results (PSNR(dB)) for the different denoising methods for the University of Pavia (UP) and the Washington DC Mall (WDCM) datasets. The best results are in bold typeface.

Datasets	Noise Cases	Noise Parameters	Noisy	SSTV	LRMR	NAILRMA	HyRes	MFA	WMFA-3D	WMFA-2D
UP	Case 1	$\sigma = 0.1$, $\eta = 10$	38.05	43.85	44.34	46.33	46.69	46.50	46.62	**46.92**
	Case 2	$\sigma = 0.5$, $\eta = 10$	24.08	34.56	33.17	36.75	39.31	**39.50**	39.29	39.37
	Case 3	$\sigma = 1$, $\eta = 12$	18.07	29.10	29.49	33.86	35.14	**35.20**	34.96	35.02
WDCM	Case 1	$\sigma = 0.1$, $\eta = 10$	38.05	43.85	44.34	46.13	46.51	46.44	**46.83**	46.62
	Case 2	$\sigma = 0.5$, $\eta = 10$	24.09	34.63	33.55	38.22	38.73	**40.04**	38.50	38.62
	Case 3	$\sigma = 1$, $\eta = 12$	18.06	29.09	29.59	33.34	34.92	**35.06**	33.91	33.99

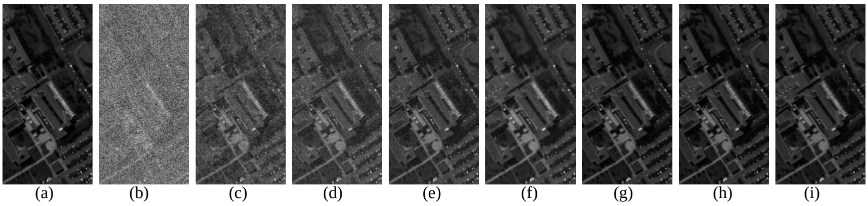


Figure 4.3: The results for the different denoising methods for the simulated UP dataset. (a) Original band 32, (b) Noisy band 32, (c) SSTV, (d) LRMR, (e) NAILRMA, (f) HyRes, (g) MFA, (h) WMFA-3D, and (i) WMFA-2D.

Algorithm 2 WMFA algorithm

```
1: **Input** HSI $\mathbf{X}$.
2: Use 2-D (or 3-D) DWT to decompose $\mathbf{X}$ to approximation sub-image and details sub-images.
3: Compose all the sub-images to a new HSI $\mathbf{X}_1$.
4: Use MFA to remove the noise of $\mathbf{X}_1$, get denoised dataset $\mathbf{X}_2$.
5: Split $\mathbf{X}_2$ to different sub-images by using the inverse way as the process of composition in the third step.
6: Perform IDWT based on the original approximation sub-image and denoised details sub-images to get the denoised HSI $\mathbf{X}_3$.
7: **Output** the denoised HSI $\mathbf{X}_3$.
```

4.1.3 WMFA Denoiser

WMFA uses the MFA in the wavelet domain to remove the noise in an HSI. Firstly, using 2-D discrete wavelet transform (DWT) to decompose the original noisy HSI $\mathbf{X}$ with $m \times n$ pixels and b spectral bands to a sequence of different spatial resolution sub-images, i.e., approximation and details. In case of using 2-D DWT, a K level decomposition can be performed resulting in $3K + 1$ different sub-images namely, LL, LH, HL, and HH. The next level of DWT is applied to the low-frequency sub-image LL only. Secondly, putting all the sub-images together to constitute a new HSI $\mathbf{X}_1$ with the same size as $\mathbf{X}$ and then using MFA to remove the noise from $\mathbf{X}_1$ in the wavelet domain. The Gaussian noise will be averaged out in low-frequency wavelet coefficients, therefore only the wavelet coefficients in the high-frequency levels, i.e., the details (LH, HL, and HH) sub-images are needed to denoise [70]. Thirdly, splitting the denoised image to different sub-images by using the inverse way as the process of composition in the second step. Finally, performing 2-D inverse DWT (IDWT) based on the original approximation sub-image and denoised details sub-images to reconstruct the denoised HSI.

In the decomposition and reconstruction of WMFA, we can also use the 3-D DWT to decompose and the 3-D IDWT to reconstruct. In this paper, the WMFA using 2-D WDT is called WMFA-2D, and WMFA-3D when the 3-D DWT is used. The WMFA model is summarized in Algorithm 2, and the schematic of the WMFA is shown in Fig. 4.1.

4.1.4 Experiments and Results

In this part, the experiments are conducted on two simulated noisy datasets and one real noisy hyperspectral dataset. The following denoising methods SSTV, LRMR, NAILRMA, HyRes, MFA, and WMFA (WMFA-2D, and WMFA-3D) are compared.

Two HSI datasets, the University of Pavia (UP) dataset (Appendix B.2), and the Washington DC Mall (WDCM) dataset (Appendix B.1), are used in the simulated experiments. In the experiments, only a sub-image of size 256× 128 ×64 for UP dataset is used, and is given in Fig. 4.2(a). The size of sub-image for WDCM dataset used in experiments is selected as 256 × 128 × 64, and is given in Fig. 4.2(b). Before the simulated datasets are denoised, the gray values of each band in the HSI are normalized between [0, 1]. After denoising, the gray values of each band are stretched to the original level. The Gaussian noise was added in such a way that the variance (σ_i^2) varies along the spectral axis according to

$$\sigma_i^2 = \sigma^2 \frac{e^{-\frac{(i-p/2)^2}{2\eta^2}}}{\sum_{j=1}^{p} e^{-\frac{(i-p/2)^2}{2\eta^2}}}. \tag{4.3}$$

Note that the power of the noise is controlled by σ, and η behaves like the standard deviation of a Gaussian bell curve [80]. In the experiments, we set σ to three different values 0.1, 0.5, and 1, and η to two different values 10, and 12.

To evaluate the denoised results of the simulated datasets, the quantitative metric is peak signal-to-noise ratio (PSNR) (Appendix A.1) is used.

Table 4.1 shows the quantitative evaluation results for the different denoising methods for both the UP dataset and the WDCM dataset. From the table, it can be seen that the proposed MFA and WMFA (WMFA-2D, and WMFA-3D) methods outperform the other methods. When the level of noise is high such as in Case 2 and 3, MFA outperforms WMFA-3D and WMFA-2D and achieves the highest PSNR values. When the level of noise is low such as in Case 1, WMFA-3D and WMFA-2D are similar and slightly better than MFA. Fig. 4.3 shows the results for the different denoising methods for a selected band 32 of the simulated UP dataset. The visual comparison reveals that the proposed methods successfully remove noise from the corrupted band.

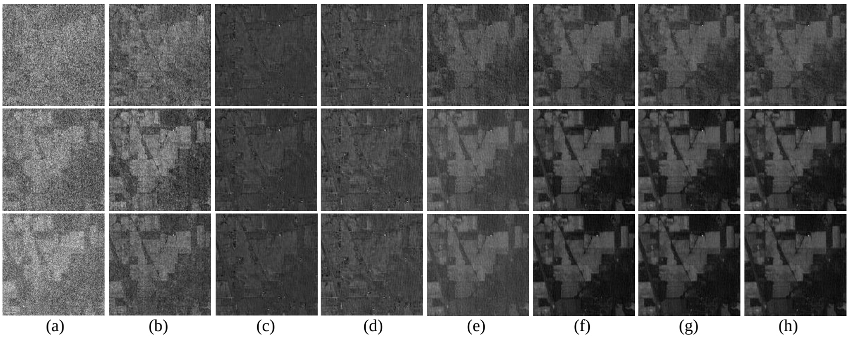


Figure 4.4: The results for the different denoising methods for the Indian Pines dataset, from top to bottom bands 150, 163, and 220, respectively. (a) Original band, (b) SSTV, (c) LRMR, (d) NAILRMA, (e) HyRes, (f) MFA, (g) WMFA-3D, and (h) WMFA-2D.

For the experiments based on the real dataset, the Indian Pines dataset (Appendix B.3) is used. The dataset is given in Fig. 4.2(c). Fig. 4.4 presents bands 150, 163, and 220 for different denoising methods. It can be clearly observed that the denoised images obtained by SSTV, LRMR, NAILRMA, and HyRes still have some visible noises. By contrast, the proposed MFA, WMFA (WMFA-3D, and WMFA-2D) methods give better denoised images, all of them have smoother appearance and preserve more details on edges.

4.2 LSSC-AMFA HSI Denoising

4.2.1 Introduction

MFA and WMFA can only remove Gaussian noise. HSIs are often corrupted by different noises such as missing vertical lines (MVL) noise, missing horizontal lines (MHL) noise, salt and pepper (SP) noise, missing pixels (MP) noise, and Gaussian noise, etc. This is due to the influence of dark current, fluctuations in power supply, and atmospheric change. Therefore, HSI mixed noise denoising is an important preprocessing task since it improves HSI quality and enhances the performance of HSI applications. In this chapter, we propose a local spectral-spatial correlation based automatic mixtures of factor analyzers (LSSC-AMFA) denoising method for HSIs. The method can simultaneously remove MVL, MHL, SP, MP, and Gaussian noises. LSSC-AMFA firstly uses a local spectral-spatial correlation (LSSC) method to remove the MVL and MHL noises, and then using an automatic mixtures of factor analyzers (AMFA) method to remove SP, MP, and Gaussian noises.

4.2.2 LSSC-AMFA Denoiser

Let us consider an HSI dataset $\mathbf{X}$ that is a $M \times N \times B$ 3-D matrix and has $M \times N$ pixels and B spectral bands. The HSIs are received by sensors, the line stripping problem mostly happens when sensors go out of radiometric calibration [84]. The dataset might be corrupted by MVL and MHL noises. Therefore, the HSI dataset $\mathbf{X}$ can be modeled by

$$\mathbf{X} = \mathbf{X}_1 + \mathbf{N}_1, \tag{4.4}$$

where $\mathbf{X}_1$ is a missing-lines-noise-free signal, and $\mathbf{N}_1$ represents the MVL and MHL noises. To estimate $\mathbf{X}_1$, a LSSC method is proposed to remove the MVL and MHL noises. For the LSSC method, it is important to find the positions of MVL and MHL noises, precisely. We propose the following method to find the positions of MVL and MHL noises. We detect MVL noise if the following condition is true

$$v_{i,b} = \left\| \mathbf{v}_{i,b} ./ \overline{\mathbf{v}}_{:,b} \right\|_1, \tag{4.5}$$

$$|v_{i,b} - \mu_v| > 3\sigma_v, \tag{4.6}$$

where $\mathbf{v}_{i,b}$ is the i-th column vector in the b-th spectral band of $\mathbf{X}$, $b \in \{1,...,B\}$, $\overline{\mathbf{v}}_{:,b}$ is the mean vector of all the column vectors in the b-th spectral band of $\mathbf{X}$, $\mu_v = \frac{1}{N}\sum_{i=1}^{N} v_{i,b}$ is the mean, and $\sigma_v = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_{i,b}-\mu_v)^2}$ is the standard deviation. We detect MHL noise if the following condition is true

$$h_{i,b} = \left\| \mathbf{h}_{i,b}./\overline{\mathbf{h}}_{:,b} \right\|_1, \tag{4.7}$$

$$|h_{i,b} - \mu_h| > 3\sigma_h, \tag{4.8}$$

where $\mathbf{h}_{i,b}$ and $\overline{\mathbf{h}}_{:,b}$ are the i-th row vector and the mean vector of all the row vectors in the b-th spectral band of $\mathbf{X}$, respectively. $\mu_h = \frac{1}{M}\sum_{i=1}^{M} h_{i,b}$ is the mean, and $\sigma_h = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(h_{i,b}-\mu_h)^2}$ is the standard deviation.

When $v_{i,b}$ and $h_{i,b}$ satisfy (4.6) and (4.8), respectively, $\mathbf{v}_{i,b}$ and $\mathbf{h}_{i,b}$ have MVL and MHL noises, respectively. Then, the estimation $\widehat{\mathbf{X}}_1$ of $\mathbf{X}_1$ is obtained by using the following model to remove MVL and MHL noises,

$$\widehat{\mathbf{v}}_{i,b} = \frac{1}{k_1}(\mathbf{v}_{i-1,b} + \mathbf{v}_{i+1,b} + \mathbf{v}_{i,b-1} + \mathbf{v}_{i,b+1}), \tag{4.9}$$

$$\widehat{\mathbf{h}}_{i,b} = \frac{1}{k_2}(\mathbf{h}_{i-1,b} + \mathbf{h}_{i+1,b} + \mathbf{h}_{i,b-1} + \mathbf{h}_{i,b+1}), \tag{4.10}$$

where $\widehat{\mathbf{v}}_{i,b}$ and $\widehat{\mathbf{h}}_{i,b}$ are the denoised version of $\mathbf{v}_{i,b}$ and $\mathbf{h}_{i,b}$, respectively. k_1 and k_2 are positive integers constant. Suppose $\mathbf{V} = \{v_{i-1,b}, v_{i+1,b}, v_{i,b-1}, v_{i,b+1}\}$, $k_1 = 4$ when all the elements in the $\mathbf{V}$ do not satisfy (4.6). When k elements in the $\mathbf{V}$ satisfy (4.6), using the renewed (4.9) by removing these k vectors from (4.9) to estimate $\mathbf{v}_{i,b}$, and here $k_1 = 4-k$, $k \in \{1,2,3\}$. When all the elements in the $\mathbf{V}$ satisfy (4.6), i.e., $k=4$, $\widehat{\mathbf{v}}_{i,b} = \frac{1}{B-3}(\mathbf{v}_{i,1} + \mathbf{v}_{i,2} + ... + \mathbf{v}_{i,b-2} + \mathbf{v}_{i,b+2} + ... + \mathbf{v}_{i,B})$ is used to estimate $\mathbf{v}_{i,b}$. $\widehat{\mathbf{h}}_{i,b}$ and k_2 are similarly set. $\widehat{\mathbf{X}}_1$ might contain SP, MP, and Gaussian noises. Therefore, the $\widehat{\mathbf{X}}_1$ can be modeled by

$$\widehat{\mathbf{X}}_1 = \mathbf{X}_2 + \mathbf{N}_2, \tag{4.11}$$

where $\mathbf{X}_2$ is the unknown noise-free image and $\mathbf{N}_2$ is the mixture of SP, MP and Gaussian noises. In order to estimate $\mathbf{X}_2$, we propose an automatic mixtures of factor analyzers (AMFA) method to remove $\mathbf{N}_2$. The AMFA model can be defined as

$$\widehat{\mathbf{X}}_1 = \mathbf{W}_c\mathbf{Z} + \boldsymbol{\mu}_c + \mathbf{N}_2, \tag{4.12}$$

where $\widehat{\mathbf{X}}_1$ is reshaped to be a $B \times P$ matrix, B is the number of spectral bands, $P = M \times N$ is the number of pixels, $\mathbf{Z}$ denotes the $r \times P$ latent factor matrix, r is subspace dimension, and $r \leq \min(B,P)$, the subscript $c \in \{1,...,C\}$ denotes the component indicator variable of the C factor analyzers in the AMFA. The parameters of the c-th factor analyzer include a $B \times r$ factor loading matrix $\mathbf{W}_c$, and a $B \times P$ mean matrix $\boldsymbol{\mu}_c$.

The parameter r is estimated by using a minimum error based hyperspectral subspace identification (HySime) method [80], and the parameters $\mathbf{Z}$, $\mathbf{W}_c$, $\boldsymbol{\mu}_c$, and $\mathbf{N}_2$ of AMFA are estimated by using an expectation-maximization (EM) algorithm [85].

Table 4.2: The parameters of the mixed noises for different cases.

Cases	Gaussian Noise	MVL	MHL	SP	MP	NB
Case 1	$\sigma = 1$, $\eta = 15$	N-MVL = 10	N-MHL = 10	P-SP = 10%	N-MP = 10	NB = 10
Case 2	$\sigma = 1$, $\eta = 15$	N-MVL = 20	N-MHL = 20	P-SP = 20%	N-MP = 20	NB = 20
Case 3	$\sigma = 0.5$, $\eta = 15$	N-MVL = 10	N-MHL = 10	P-SP = 10%	N-MP = 10	NB = 10
Case 4	$\sigma = 0.5$, $\eta = 15$	N-MVL = 20	N-MHL = 20	P-SP = 20%	N-MP = 20	NB = 20

Finally, the clear image is obtained by

$$\widehat{\mathbf{X}}_2 = \widehat{\mathbf{W}}_c \widehat{\mathbf{Z}} + \widehat{\boldsymbol{\mu}}_c, \tag{4.13}$$

where $\widehat{\mathbf{X}}_2$, $\widehat{\mathbf{W}}_c$, $\widehat{\mathbf{Z}}$, and $\widehat{\boldsymbol{\mu}}_c$ denote the estimations of $\mathbf{X}_2$, $\mathbf{W}_c$, $\mathbf{Z}$, and $\boldsymbol{\mu}_c$, respectively.

The LSSC-AMFA denoising method is summarized in Algorithm 3.

Algorithm 3 LSSC-AMFA algorithm

```
Input HSI $\mathbf{X}$.
Find the positions of MVL and MHL noises.
Remove the MVL and MHL noises using the LSSC method.
Use the AMFA method to further remove the SP, MP, and Gaussian noises.
Output the denoised HSI $\widehat{\mathbf{X}}_2$.
```

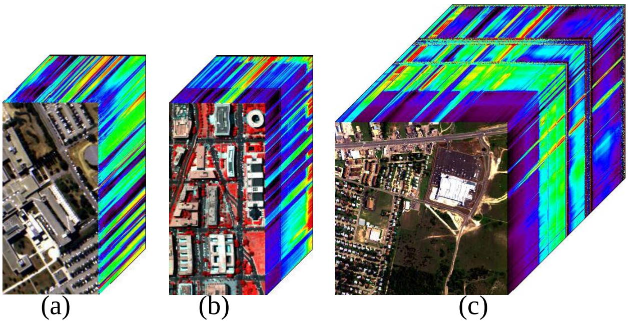


Figure 4.5: The experimental datasets. (a) University of Pavia (R: 64, G: 45, and B: 10), (b) Washington DC Mall (R: 60, G: 27, and B: 17), and (c) Urban (R: 53, G: 35, and B: 10).

4.2.3 Experiments and Results

The experiments are conducted on two noisy simulated datasets and one real noisy hyperspectral dataset. The following denoising methods GSP, GAP, BM3D, BM4D, SSTV, LRMR, NAILRMA, and HyRes are used as comparison to the proposed LSSC-AMFA denoising method.

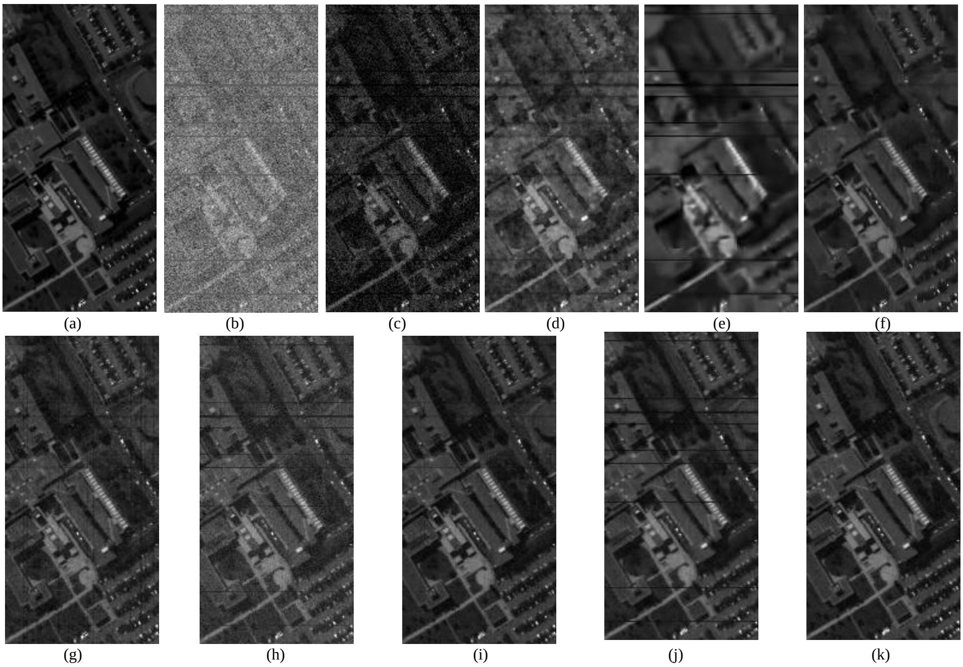


Figure 4.6: The results for the different denoising methods for the University of Pavia (UP) dataset. (a) Original band 31, (b) Noisy band 31, (c) GSP, (d) GAP, (e) BM3D, (f) BM4D, (g) SSTV, (h) LRMR, (i) NAILRMA, (j) HyRes and (k) LSSC-AMFA.

Table 4.3: The quantitative evaluation results (PSNR (dB)) for the different denoising methods for the University of Pavia (UP) and the Washington DC Mall (WDCM) datasets. The best results are in bold typeface.

Data	Cases	Noisy	GSP	GAP	BM3D	BM4D	SSTV	LRMR	NAILRMA	HyRes	LSSC-AMFA
UP	Case 1	16.74	24.28	27.35	24.87	29.94	28.17	28.65	29.47	29.45	**30.83**
	Case 2	13.69	22.78	23.72	20.30	23.74	24.72	24.75	23.89	22.92	**25.31**
	Case 3	19.99	27.56	29.59	24.47	27.57	33.10	30.72	31.08	30.98	**33.81**
	Case 4	14.75	24.84	24.30	20.01	21.54	26.59	24.23	24.93	23.67	**27.11**
WDCM	Case 1	16.69	22.67	26.60	23.11	27.30	27.77	26.47	26.53	27.29	**31.57**
	Case 2	13.67	19.96	21.68	18.88	21.49	22.99	21.31	20.92	21.26	**24.95**
	Case 3	19.81	25.11	28.84	23.64	25.94	32.38	27.33	26.98	27.62	**34.65**
	Case 4	14.52	21.29	22.72	18.22	20.49	24.60	21.09	20.77	20.71	**26.88**

The University of Pavia (UP) dataset (Appendix B.2), and the Washington DC Mall (WDCM) dataset (Appendix B.1) are used in the simulated experiments. In the experiments, only a sub-image of size 256× 128 ×64 is used for UP dataset and is given in Fig. 4.5(a). The size of the sub-image for WDCM dataset used in experiments is 256 × 128 × 64, and it is given in Fig. 4.5(b). Before the simulated datasets are denoised, the gray values of each band in the HSI are normalized between [0, 1]. After denoising, the gray values of each band are stretched to the original level. The added Gaussian noise has variance (σ_i^2) that varies along the spectral axis and is given by

$$\sigma_i^2 = \sigma^2 \frac{e^{-\frac{(i-p/2)^2}{2\eta^2}}}{\sum_{j=1}^{p} e^{-\frac{(i-p/2)^2}{2\eta^2}}}. \tag{4.14}$$

In (4.14), σ controls the power of the noise and η behaves like the standard deviation of a Gaussian bell curve [80]. In the experiments, σ has two different values 1, and 0.5, and we set $\eta = 15$.

The parameters for the MVL, MHL, SP, MP, and Gaussian noises added to the simulated datasets are given in Table 4.2. The number of MVL, MHL, and MP are called N-MVL, N-MHL, and N-MP, respectively. The percentage of the SP noise is denoted by P-SP (equal percentages for salt and pepper). The number of bands selected to add each type of MVL, MHL, SP, and MP is denoted by NB.

To evaluate the denoised results of the simulated datasets, the quantitative metric is peak signal-to-noise ratio (PSNR) (Appendix A.1) is used.

Table 4.3 shows the quantitative evaluation results for the different denoising methods with different noise cases for both the UP dataset and the WDCM dataset. From the table, it can be seen that the performance of the proposed LSSC-AMFA method is better than the other methods used in the experiments. LSSC-AMFA achieves the highest PSNR for both datasets for all different noise cases. For the UP dataset, LSSC-AMFA improves the PSNR by 6.55 dB in Case 1, 5.01 dB in Case 2, 9.34 dB in Case 3, and 7.10 dB in Case 4 compared to other methods used in the experiment. For the WDCM dataset, LSSC-AMFA improves the PSNR by 8.90 dB in Case 1, 4.99 dB in Case 2, 11.01 dB in Case 3, and 8.66 dB in Case 4 compared to other methods used in the experiment. Fig. 4.6 and 4.7 show the results for the different denoising methods for band 31 of the simulated UP dataset and band 64 of the simulated WDCM dataset, respectively. The visual comparison shows that GSP, GAP, BM3D, BM4D, SSTV, LRMR, NAILRMA, and HyRes have a hard time removing the mixed noises as were expected. But LSSC-AMFA removes the mixed noises from the corrupted band.

The Urban dataset (Appendix B.4) is used in experiments for the real dataset. The dataset is given in Fig. 4.5(c). It should be noted that, the GSP and GAP methods do not work for the Urban dataset since one necessary running condition for GSP is that the number of pixels for experimental dataset is even, while the Urban dataset is odd (307 × 307 = 94249). And the size of the Urban dataset is too big to be run

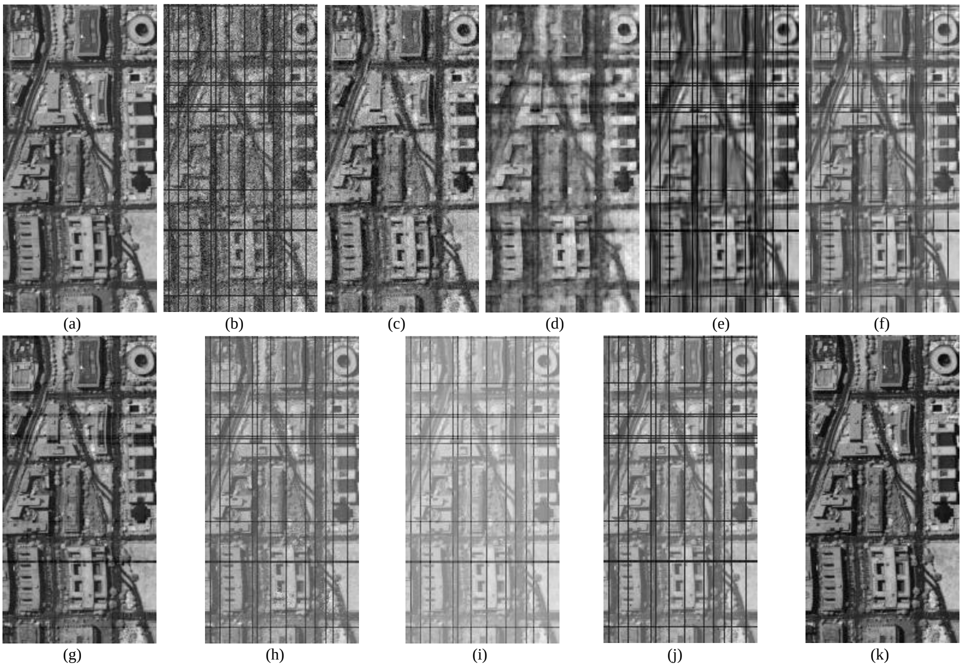


Figure 4.7: The results for the different denoising methods for the Washington DC Mall (WDCM) dataset. (a) Original band 64, (b) Noisy band 64, (c) GSP, (d) GAP, (e) BM3D, (f) BM4D, (g) SSTV, (h) LRMR, (i) NAILRMA, (j) HyRes and (k) LSSC-AMFA.

for GAP. Therefore, there are no experimental results for GSP and GAP in this part of the experiments. Fig. 4.8 shows bands 150, 208, and 210 for the different denoising methods. It can be clearly observed that the denoised images obtained by GSP, GAP, BM3D, BM4D, SSTV, LRMR, NAILRMA, and HyRes still have some visible noises. By contrast, LSSC-AMFA gives much better denoised images and preserves more details on edges.

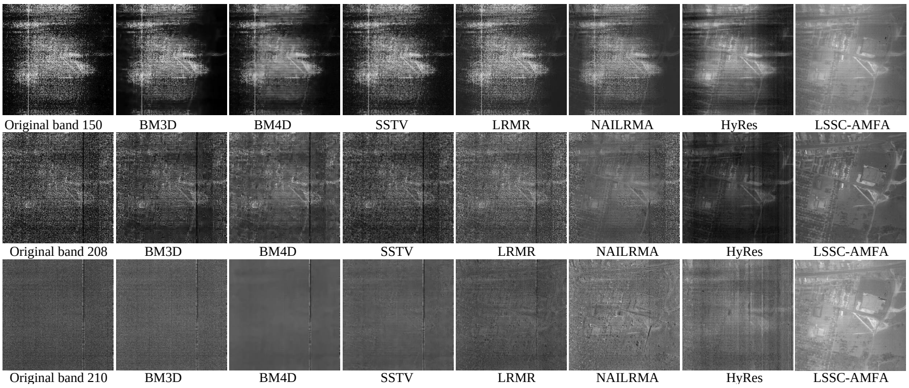


Figure 4.8: The results for the different denoising methods for the Urban dataset.

The CPU processing time (in seconds) is given in Table 4.4 for different denoising methods applied to the UP dataset. The results given are mean values over ten experiments. All the methods are implemented in MATLAB R2019a on a computer having Intel(R) Core(TM) i7-6700 processor (3.40GHz), 8.00 GB of memory and 64-bit Windows 10 Operating System. It can be seen that the running time for LSSC-AMFA is 0.77 s in Case 1, 0.73 s in Case 2, 0.65 s in Case 3, and 1.01 s in Case 4. LSSC-AMFA is a fast denoising method and mostly less computational expensive than the other methods used in the experiments.

Table 4.4: CPU processing times in seconds for different denoising methods applied on the University of Pavia dataset.

Noise Cases	GSP	GAP	BM3D	BM4D	SSTV	LRMR	NAILRMA	HyRes	LSSC-AMFA
Case 1	90.43	7339.94	33.68	107.57	127.59	16.78	30.37	**0.74**	0.77
Case 2	89.67	7430.61	33.80	107.15	127.68	16.63	31.65	0.74	**0.73**
Case 3	92.05	7202.84	34.31	107.65	127.38	16.54	40.64	0.76	**0.65**
Case 4	94.86	7333.70	33.72	107.44	127.24	16.84	41.49	**0.79**	1.01

4.3 Conclusions

In this chapter, we propose three new, MFA, WMFA-2D, and WMFA-3D, denoising methods for removing Gaussian noise of HSI. MFA uses the inverse transformation of the loading matrix to get the denoised HSI. WMFA-2D and WMFA-3D use the MFA in the wavelet domain. MFA is particularly suitable for denoising of HSI with a high level of noise. While WMFA-2D and WMFA-3D can be effectively used to denoising of HSI with a low level of noise. To validate the performance of denoising, two simulated datasets and a real dataset are used in the experiments. The experimental results show that MFA, WMFA-2D, and WMFA-3D can give good denoised results.

LSSC-AMFA denoising method was proposed for removing mixed noise of HSI. In the LSSC-AMFA, the MVL and MHL noises were first removed using the proposed LSSC method and then the mixture of SP, MP, and Gaussian noises was removed using the proposed AMFA method. To validate the performance of denoising methods, two simulated datasets and one real HSI dataset was used in the experiments. These experimental results show that LSSC-AMFA is a fast denoising method and gives better results compared to state-of-the-art denoising methods and can effectively and simultaneously remove the mixture of MVL, MHL, SP, MP, and Gaussian noises.

Chapter 5

MFA Low-Rank Based Methods for HSI Feature Extraction

This chapter proposes six new unsupervised, supervised, and semi-supervised MFA-based methods are proposed for HSI feature extraction and are then used for classification of them. There are MFA, deep MFA (DMFA), supervised MFA (SMFA), supervised DMFA (SDMFA), semi-supervised MFA (S^2MFA), and semi-supervised DMFA (S^2DMFA) feature extraction methods. All of them are probabilistic dimensionality reduction (DR) methods, instead of assuming that a whole HSI obeys a Gaussian distribution, the methods use a Gaussian mixture model to extract more effective information for DR. The Gaussian mixture model is used for MFA to allow a low-dimensionality representation of the Gaussian. A two-layer MFA, DMFA, utilizes the samples from the posterior at the first layer to an MFA model at the second layer. MFA and DMFA are two unsupervised DR method. The methods are particularly suitable for DR of HSI with a non-normal distribution and unlabeled samples. SMFA and SDMFA are two supervised DR methods and use labeled samples to extract features. SDMFA is a deep version of SMFA and consists of a two-layer SMFA. SMFA and SDMFA can be effectively used to DR of HSI with a non-normal distribution and labeled samples. S^2MFA and S^2DMFA are two semi-supervised DR methods, which simultaneously consider labeled and unlabeled samples to extract features. S^2MFA uses a Gaussian mixture model to segment image to different parts, each part follows a Gaussian distribution and contains many labeled and unlabeled samples and uses a factor analyzer to get a factor loading matrix. This matrix uses labeled samples to improve the class discrimination and employs both labeled and unlabeled samples to preserve the local spatial features of the data and then is used for transforming the original HSI to an optimal low-dimensional subspace to achieve DR. S^2DMFA is a two-layer S^2MFA. S^2MFA and S^2DMFA are particularly suitable for DR of HSI with a complicated probability distribution and labeled and unlabeled samples. Based on the six DR methods, we also proposed a framework for HSI classification, the overall accuracy of a classifier on validation samples is used to automatically determine the optimal number of features of DR for HSI classification. This framework can automatically extract the most effective feature for HSI classification. To validate the performance of DR, we conduct experiments in terms of SVM classification based on real HSIs. The experimental results show that our proposed methods can give better results than statistical DR comparison methods.

5.1 Unsupervised and Supervised MFA-Based Feature Extraction Methods

5.1.1 Introduction

Hyperspectral images (HSIs) provide abundant spectral information about a scene [86]. In general, an HSI contains hundreds of spectral bands with high spectral resolution [87–89]. Having sufficient spectral information makes it possible to discriminate different materials within a scene by using a classifier [90–93]. However, the high dimensionality of HSIs makes the processing computationally and memory costly. To achieve an acceptable classification accuracy for an image of high dimensionality many conventional HSI processing require many training samples [94–96]. This is known as the Hughes phenomenon or the curse of dimensionality [97]. Thus when we have a limited number of training samples, we have a trade-off between classification accuracy and the number of spectral bands [48, 98–105]. Dimensionality reduction (DR) is a very effective way to solve this problem [17, 106–115]. Dimensionality reduced data should be a good representation of the original data. In addition, both the computing time and the number of training samples required will become less when the data dimensionality is lower. Therefore, DR is a very important pre-processing step for HSI classification [116–122]. In general, DR can be divided into feature selection (FS) and feature extraction (FE). In this paper, we focus on FE. There exist several classical and novel statistical FE methods in the literature that have been used in HSI processing. FE methods are either unsupervised or supervised. Principal component analysis (PCA) [123] is a classical unsupervised FE method. PCA projects the original data onto a lower dimensional linear subspace of the original data space and can also be expressed as the maximum likelihood solution of a probabilistic latent variable model [50]. This reformulation of PCA is called probabilistic principal component analysis (PPCA) [49] and is an example of a linear-Gaussian framework, in which all of the marginal and conditional distributions are assumed to be Gaussian. Factor analysis (FA) [50, 124] is also a linear Gaussian latent variable model closely related to PPCA. For FA, the conditional distribution of the observed variables given the latent variable have diagonal rather than an isotropic covariance matrix. In addition to these classical unsupervised FE methods, there are several novel unsupervised FE methods in the literature, such as orthogonal total variation component analysis (OTVCA) [54], edge-preserving filtering [125], Gaussian pyramid based multi-scale feature extraction (MSFE) [126], sparse and smooth low-rank analysis (SSLRA) [47], etc. For supervised FE methods, the new features should contain most discriminative information based on the labeled samples. There exist several supervised FE methods, such as linear discriminant analysis (LDA) [56], nonparametric weighted feature extraction (NWFE) [57], manifold-learning based HSI feature extraction [61], low-rank representation with the ability to preserve the local pairwise constraints information (LRLPC) [62], etc. Supervised methods are usually better than unsupervised methods for HSI classification [127–129], since they have access to labeled data. However, the effectiveness depends on how well the labeled dataset represents the whole original dataset.

For both PPCA and FA, all the marginal and conditional distributions of the HSI are assumed to be Gaussian. However, in practice, most HSIs cannot be assumed to obey a Gaussian distribution. To overcome this problem, we propose mixtures of factor analyzers (MFA), deep MFA (DMFA), supervised MFA (SMFA), and supervised DMFA (SDMFA) FE methods for HSI. We also propose an image segmentation method based on the Gaussian mixture model for MFA, DMFA, SMFA, and SDMFA to solve the problem of a non-normal distribution. MFA, is the same as it is used to HSI denoising, assumes a low-dimensionality representation of the Gaussians in the Gaussian mixture model. DMFA consists of a two-layer MFA, which inputs the samples from the posterior distribution at the first layer to an MFA model at the second layer. SMFA and SDMFA are supervised FE method that uses labeled samples to extract features of HSI. Based on these four FE methods, a framework for HSI classification is also proposed in this paper. While the dimensionality of the desired features needs to be selected by the user in conventional DR methods, the proposed framework automatically determines the dimensionality of features according to classification accuracy without prior supervision by the user.

5.1.2 Proposed FE Methods and Framework

MFA

Let $\mathbf{x}$ denote a D-dimensional spectral vector, $\mathbf{z}$ denote a d-dimensional latent vector, and $m \in \{1, ..., M\}$ denote the component indicator variable of the M factor analyzers in MFA. The MFA model can be defined as

$$p(m) = \pi_m, \sum_{m=1}^{M} \pi_m = 1, \tag{5.1}$$

$$p(\mathbf{z}|m) = p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}), \tag{5.2}$$

$$p(\mathbf{x}|\mathbf{z}, m) = \mathcal{N}(\mathbf{x}; \mathbf{W}_m\mathbf{z} + \boldsymbol{\mu}_m, \boldsymbol{\Psi}), \tag{5.3}$$

where $\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ means that $\mathbf{z}$ is Gaussian vector with zero mean and $d \times d$ identity matrix $\mathbf{I}$ as the covariance matrix. The parameters of the m-th factor analyzer include a mixing proportion π_m, mean $\boldsymbol{\mu}_m$, a $D \times d$ factor loading matrix $\mathbf{W}_m$, and a $D \times D$ diagonal matrix $\boldsymbol{\Psi}$ which represents the independent noise variances for each band.

The parameters $\mathbf{z}$, $\boldsymbol{\mu}_m$, $\mathbf{W}_m$, and $\boldsymbol{\Psi}$ of MFA are estimated (trained) by using an expectation maximization (EM) algorithm [85]. An example demonstrating how MFA works is shown in Figure 5.1(a) and (b). The schematic of the MFA is shown in Figure 5.2.

The performance of MFA for classification can be improved by increasing the dimensionality d of the latent factors per component of the mixture of factor analyzers or the number M of mixture components. However, for high dimensionality data, this approach quickly leads to overfitting. Below we discuss a cross-validation scheme to select d while avoiding overfitting.

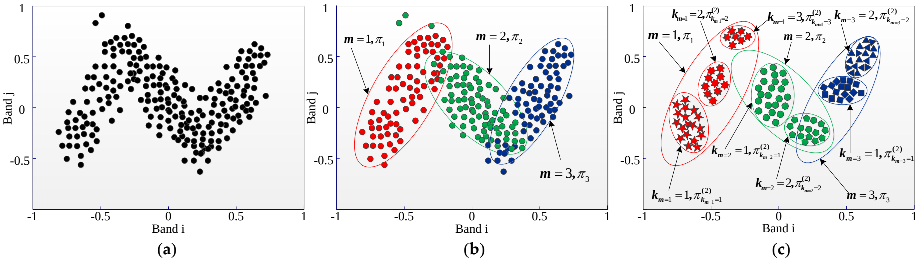


Figure 5.1: (**a**) A scatterplot of HSI samples over two spectral bands. (**b**) Illustration of the MFA model with each ellipse representing a Gaussian component. MFA has three components colored red ($m = 1$), green ($m = 2$) and blue ($m = 3$). Their mixing proportions are given by π_m. (**c**) Illustration of DMFA model with each ellipse representing a Gaussian component. The number of components and mixing proportions of the first layer of DMFA are the same as MFA. For the red component, we further learn a second layer of DMFA with three components. For the green and blue components, both of them are learned a second layer of DMFA with two components, respectively. We also introduce the second layer component indicator variable $k_m = 1, 2, ..., K_m$ and mixing proportions $\pi_m^{(2)}$, where K_m is the total number of the second layer components associated with the first layer component m. K_m is specific to the first layer component and need not be the same for all m. In this example, $K_1 = 3$, $K_2 = 2$ and $K_3 = 2$.

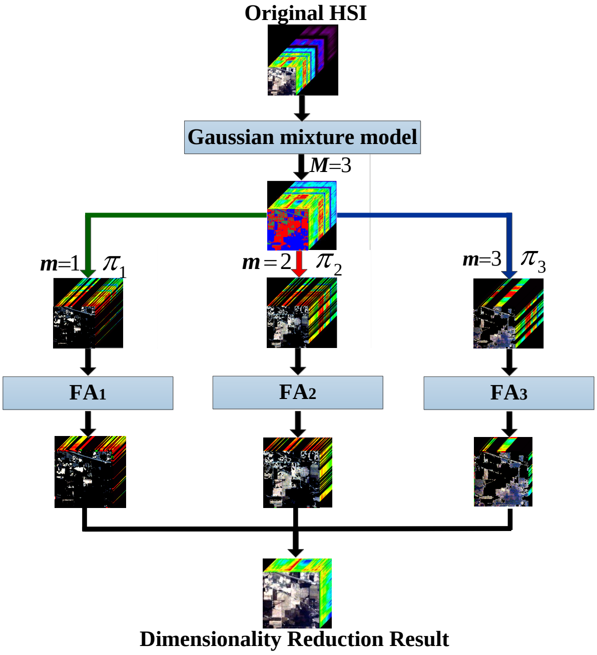


Figure 5.2: The schematic of the MFA corresponding to Figure 5.14**b**.

DMFA

Figure 5.1(c) shows a case where the posteriors have non-normal distribution, to solve this problem the DMFA model was proposed. Instead of a simple standard normal prior, the DMFA model uses a more powerful MFA prior:

$$p(\mathbf{z}|m) = \mathrm{MFA}(\theta_m^{(2)}), \tag{5.4}$$

where $\theta_m^{(2)}$ is the model parameter in the second layer and it emphasizes that the new MFA's parameters are at the second layer and specific to component m of the first layer MFA, while holding the first layer parameters fixed. Thus, the DMFA is equivalent to fitting component-specific second layer MFAs with vectors drawn from $p(\mathbf{z}, m, |\mathbf{x}; \theta_m^{(1)})$ as data, where $\theta_m^{(1)}$ is the model parameter in the first layer.

Using $p(k_m|m) = \pi_{k_m}^{(2)}$ to denote the second layer mixing proportion of mixture component k_m, and K_m denote the total number of factor analyzers in the second layer for specific m of the first layer, so

$$\forall m : \sum_{k_m=1}^{K_m} \pi_{k_m}^{(2)} = 1, \tag{5.5}$$

$$p_{DMFA}(\mathbf{z}, m) = p(m)p(\mathbf{z}|m) = p(m)p(k_m|m)p(\mathbf{z}|k_m). \tag{5.6}$$

For convenience denote all possible second layer mixture components with $s = 1, ..., S$, where $S = \sum_{m=1}^{M} K_m$. The mixing proportions are $\pi_s^{(2)} = p(m(s))p(k_m(s)|m(s))$, where $m(s)$ and $k_m(s)$ are the first and second layer mixture components m and k_m to which s corresponds.

Therefore, the DMFA model is

$$p(s) = \pi_s^{(2)}, \tag{5.7}$$

$$p(\mathbf{z}^{(2)}|s) = \mathcal{N}(\mathbf{z}^{(2)}; 0, \mathbf{I}), \tag{5.8}$$

$$p(\mathbf{z}^{(1)}|\mathbf{z}^{(2)}, s) = \mathcal{N}(\mathbf{z}^{(1)}; \mathbf{W}_s^{(2)}\mathbf{z}^{(2)} + \boldsymbol{\mu}_s^{(2)}, \mathbf{\Psi}^{(2)}), \tag{5.9}$$

$$m \leftarrow m(s), (\text{deterministic}), \tag{5.10}$$

$$p(\mathbf{x}|\mathbf{z}^{(1)}, m) = \mathcal{N}(\mathbf{x}; \mathbf{W}_m^{(1)}\mathbf{z}^{(1)} + \boldsymbol{\mu}_m^{(1)}, \mathbf{\Psi}^{(1)}), \tag{5.11}$$

where (5.10) is fully deterministic as each s belongs to one and only one m. $\mathbf{z}^{(1)} \in \mathrm{R}^{d^{(1)}}$, $\mathbf{z}^{(2)} \in \mathrm{R}^{d^{(2)}}$, $\mathbf{W}_m^{(1)} \in \mathrm{R}^{D \times d^{(1)}}$, $\mathbf{W}_s^{(2)} \in \mathrm{R}^{d^{(1)} \times d^{(2)}}$, $\boldsymbol{\mu}_m^{(1)} \in \mathrm{R}^{d^{(1)}}$, $\boldsymbol{\mu}_s^{(2)} \in \mathrm{R}^{d^{(2)}}$, $\mathbf{\Psi}^{(1)}$ and $\mathbf{\Psi}^{(2)}$ are $D \times D$ and $d^{(1)} \times d^{(1)}$ diagonal matrices of the first and second layers, respectively.

For the DMFA algorithm, the same scheme can be extended to training third-layer MFAs, but in this paper, we only consider the two-layer DMFA model.

The DMFA model can be trained by using a greedy layers-wise algorithm. The first layer of DMFA is trained as described above in Section 5.1.2, when training the second layer of DMFA, freezing the first layer parameters and treating the sampled first layer factor values for each mixture component $\{\mathbf{z}_n^{(1)}\}_m$ as training data for the second layer of DMFA. The DMFA model is summarized in Algorithm 4, and an illustration of the DMFA are shown in Figure 5.1 (c) and Figure 5.3, respectively.

Algorithm 4 DMFA algorithm

```
Step 1: Input HSI $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, the maximum number of EM iteration (default=1000).
Step 2: Train the first layer of DMFA on $\mathbf{X}$ with $M$ mixture components and $d^{(1)}$ dimensional latent factors using the EM algorithm.
Step 3: Use the first layer latent factor dataset $Y_m = \{\mathbf{z}_n^{(1)}\}_m$ for each of the $M$ mixture components as training data for the second layer of DMFA.
Step 4: Train the second layer of DMFA on $Y_m$ with $d^{(2)}$ dimensional latent factors and $K_m$ mixture components using the EM algorithm.
Step 5: Output DR results $\mathbf{Z} = \{\mathbf{z}_1^{(2)}, \mathbf{z}_2^{(2)}, ..., \mathbf{z}_N^{(2)}\}$.
```

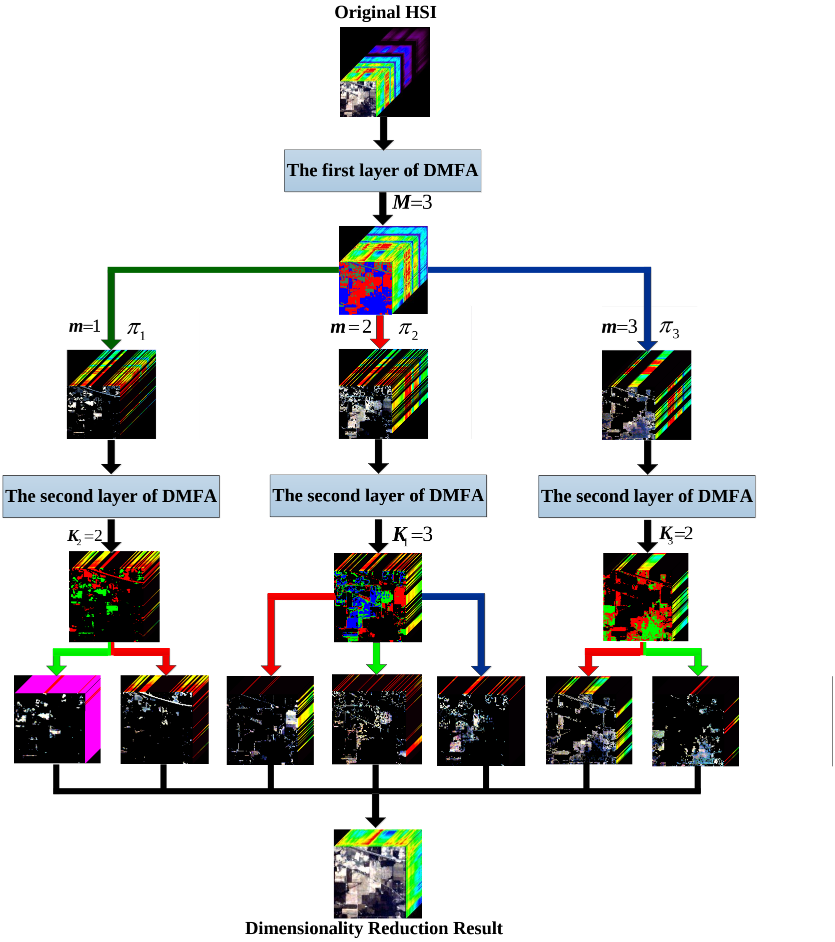


Figure 5.3: The schematic of the DMFA corresponding to Figure 5.14**c**.

SMFA

SMFA is a supervised FE method, let y denote an output value (label) for each D-dimensional labeled spectral vector $\mathbf{x}$. The SMFA model can be defined as

$$p(\mathbf{x}|\mathbf{z}, m) = \mathcal{N}(\mathbf{x}; \mathbf{W}_m\mathbf{z} + \boldsymbol{\mu}_m, \boldsymbol{\Psi}), \tag{5.12}$$

$$p(y|\mathbf{z}, m) = \mathcal{N}(y; \mathbf{W}_{ym}\mathbf{z} + \boldsymbol{\mu}_{ym}, \boldsymbol{\Psi}_y), \tag{5.13}$$

where the parameters of the m-th factor analyzer include mean $\boldsymbol{\mu}_m$, a $D \times d$ factor loading matrix $\mathbf{W}_m$, and a $D \times D$ diagonal matrix $\boldsymbol{\Psi}$ which represents the independent noise variances for each band. $\mathbf{W}_{ym}$, $\boldsymbol{\mu}_{ym}$, and $\boldsymbol{\Psi}_y$ are similar defined.

The parameters $\mathbf{z}$, $\boldsymbol{\mu}_m$, $\mathbf{W}_m$, $\boldsymbol{\Psi}$, $\mathbf{W}_{ym}$, $\boldsymbol{\mu}_{ym}$, and $\boldsymbol{\Psi}_y$ of SMFA are also estimated by using the EM algorithm [85]. The schematic of the SMFA is shown in Figure 5.4.

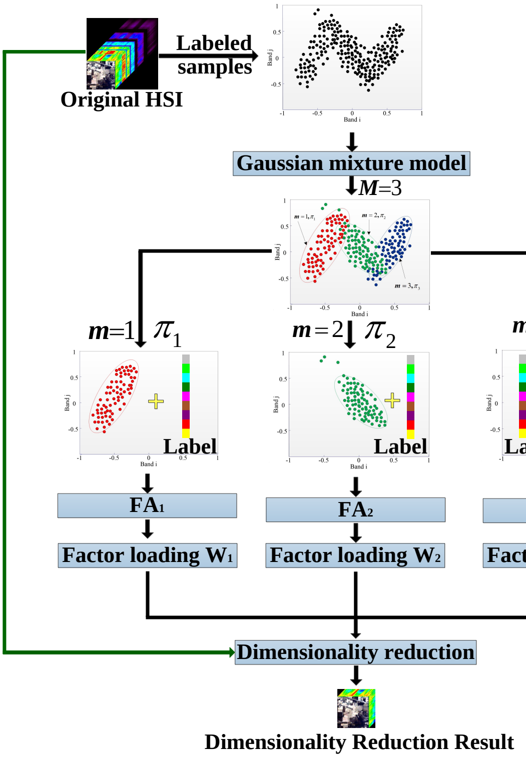


Figure 5.4: The schematic of the SMFA corresponding to Figure 5.14(**b**).

SDMFA

SDMFA consists of two layers SMFA. In the second layer, instead of a simple standard normal prior, the SDMFA model uses a more powerful SMFA prior [130]:

$$p(\mathbf{z}|m) = \text{SMFA}(\theta_m^{(2)}), \tag{5.14}$$

where $\theta_m^{(2)}$ is the model parameter in the second layer and it emphasizes that the new SMFA's parameters are at the second layer and specific to component m of the first layer SMFA, while holding the first layer parameters fixed. Thus, the SDMFA is equivalent to fitting component-specific second layer SMFAs with vectors drawn from $p(\mathbf{z}, m, |\mathbf{x}; \theta)$ as data [130], where θ is the model parameter in the first layer. The same scheme can be extended to training third-layer SMFAs, but in this paper, we only consider two-layer SDMFA model. The SDMFA is summarized in Algorithm 5.

Algorithm 5 SDMFA algorithm

```
1: **Input:** Assume $n$ labeled samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$.
2: Train the first layer of SDMFA by using (5.12) and (5.13) on $\mathbf{x}_i$ with $M$ mixture components and $d^{(1)}$ dimensionality factors using the EM algorithm.
3: Use the first layer factor dataset $Z_m = \{\mathbf{z}_i^{(1)}\}_m$ for each of the $M$ mixture components as training data for the second layer of SDMFA.
4: Train the second layer of SDMFA on $Z_m$ with $d^{(2)}$ dimensionality factors and $K_m$ mixture components using the EM algorithm, where $K_m$ is the total number of the second layer components associated with the first layer component $m$.
5: **Output:** Dimensionality reduction results by calculating the second layer latent factors $\mathbf{z}_i^{(2)}$. For HSI $\{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{z}_i^{(2)} = \sum_{m=1}^M \pi_m^{(1)} \{\sum_{k_m=1}^{K_m} \pi_{k_m}^{(2)} [(\mathbf{W}_{k_m}^T \mathbf{W}_{k_m} + \boldsymbol{\Psi}_{k_m})^{-1} \mathbf{W}_{k_m}^T (\mathbf{z}_i^{(1)} - \boldsymbol{\mu}_{k_m})]\}$, $\mathbf{z}_i^{(1)} = (\mathbf{W}_m^T \mathbf{W}_m + \boldsymbol{\Psi})^{-1} \mathbf{W}_m^T (\mathbf{x}_i - \boldsymbol{\mu}_m)$, where $N$ is the total number of samples, the parameters of the $m$-th factor analyzer mixing proportion in the first layer include a mixing proportion $\pi_m^{(1)}$, a factor loading matrix $\mathbf{W}_m$, mean $\boldsymbol{\mu}_m$, and a diagonal noise variances matrix $\boldsymbol{\Psi}$, the parameters of the $k_m$-th factor analyzer in the second layer include a mixing proportion $\pi_{k_m}^{(2)}$, a factor loading matrix $\mathbf{W}_{k_m}$, mean $\boldsymbol{\mu}_{k_m}$, and a diagonal noise variances matrix $\boldsymbol{\Psi}_{k_m}$ associated with the first layer component $m$.
```

Framework

Traditionally, in DR, the dimensionality of desired features has to be initialized by the user. In this chapter, we propose a framework that automatically selects the optimal dimensionality of desired features for HSI. We use the classification accuracy of a classifier on validation samples to automatically determine the dimensionality of the features. Different classifiers such as maximum likelihood (ML), support vector machine (SVM), and random forest (RF), can be used in this framework. The framework based on MFA, SMFA, and DMFA for HSI classification are summarized in Algorithms 6 and 7, respectively.

Algorithm 6 Framework based on MFA and SMFA

```
Step 1: Input HSI $\mathbf{X}$, training samples;
Step 2: Automatically select the optimal number of features $d$ and mixture components $M$:
        for $M = 2 : Mc$
            for $d = 3 : \frac{D}{2}$
                Run MFA (or SMFA);
                Five-fold cross-validation of SVM (ML, or RF) on training
                samples; Save the cross-validation (CV) score $\mathrm{CV}_{M,d}$;
            end
        end
        Return $\widehat{M}$ and $\widehat{d}$ corresponding to the best CV;
Step 3: Run MFA (or SMFA) with $\widehat{M}$ and $\widehat{d}$;
Step 4: Run SVM (ML or RF) classification;
Step 5: Output Classification results.
```

5.1.3 Experiments and Results

The experiments were done using ML, RF, and SVM classifiers, but since ML and RF gave inferior or slightly inferior results compared to SVM results, only the results of the SVM classifier are reported.

Experimental Datasets

The first dataset is the Indian Pines dataset (Appendix B.3). In the experiments, noisy bands and atmospheric vapor absorption bands are excluded leaving 200 spectral bands. Figure 5.5 shows the false-color composite of the Indian Pines image and the corresponding ground reference map, respectively. The nine largest classes are considered for classification [17,46]. For the Indian Pines, the University of Pavia, and the Salinas datasets, 10% of the labeled samples for each class are randomly selected as training samples, and the remaining 90% are used as the test set, respectively. Tables 5.1, 5.3 and 5.4 provide information on the number of training and test samples for each class of interest, respectively.

The second dataset is the Houston dataset (Appendix B.5). This HSI contains fifteen classes of interest. Figure 5.6 shows the false-color composite of the Houston image and the corresponding ground reference map, respectively. The training and test samples were given according to the IEEE GRSS Data Fusion Contest in 2013. The spatial positions and the number of training and test samples for each class of interest is fixed by the IEEE GRSS Data Fusion Contest. Table 5.2 provides information on the number of training and test samples for each class of interest. It is important to note that the standard sets of training and test samples were used for the dataset to make

Algorithm 7 Framework based on DMFA

```
Step 1: Input HSI $\mathbf{X}$, training samples;
Step 2: Automatically select the optimal number of features in the first layer $d^{(1)}$,
in the second layer $d^{(2)}$, mixture components in the first layer $M$, and in the second
layer $K_m$:
        for $M = 2 : Mc$
            for $d_1 = 3 : \frac{D}{2}$
                for $K_m = 2 : M$
                    for $d_2 = 3 : d_1$
                        Run DMFA;
                        Five-fold cross-validation of SVM (ML, or RF) on
                        training samples; Save the cross-validation accuracy
                        (CVA) $\mathrm{CVA}_{M,d_1,K_m,d_2}$;
                    end
                end
            end
        end
        Return $\widehat{M}$, $\widehat{d_1}$, $\widehat{K_m}$, and $\widehat{d_2}$ according to the best CVA;
Step 3: Run DMFA with $M = \widehat{M}$, $d^{(1)} = \widehat{d_1}$, $K_m = \widehat{K_m}$, and $d^{(2)} = \widehat{d_2}$;
Step 4: Run SVM (ML or RF) classification;
Step 5: Output Classification results.
```

the results entirely comparable with most of the methods available in the literature.

The third dataset is the University of Pavia dataset (Appendix B.2). In the experiments, this data contains nine classes of interest and has 103 spectral bands after removing 12 noisy bands. Figure 5.7 shows the false-color composite of the University of Pavia image and the corresponding ground reference map.

The last dataset is the Salinas dataset (Appendix B.6). This image contains sixteen classes of interest. Figure 5.8 shows the false-color composite of the Salinas image and the corresponding ground reference map.

Experimental Setup

An SVM classifier is used to evaluate the performance of the proposed methods. The SVM classifier is a supervised classification method that uses a kernel method to map the data with a non-linear transformation to a higher dimensional space and in that space tries to find a linear separating hyperplane from different classes. In the experiments, for SVM, the LibSVM Toolbox for MATLAB was applied with a radial basis function (RBF) kernel [131,132]. The five-fold cross-validation is used to find the best parameters, i.e., the kernel parameter and regularization parameter, in SVM. The evaluation metrics used are overall accuracy (OA) (Appendix A.4), average accuracy (AA) (Appendix A.6), and Kappa coefficient (KC) (Appendix A.7), as well as standard deviation (STD). To further evaluate the performance of the proposed algorithms, the following statistical and DR methods: PCA, PPCA, FA, LDA, NWFE, MFA, DMFA, and SMFA are used for comparison. Each experiment is run ten times, and the average of these ten experiments is reported.

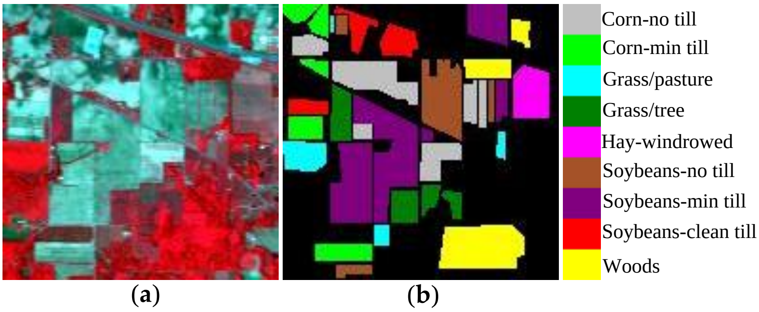


Figure 5.5: Indian Pines dataset. (**a**) Three-band false-color image. (**b**) Ground truth-map containing nine land-cover classes.

Table 5.1: Indian Pines HSI: Number of training and test samples.

Class number	Class name	Training samples	Test samples
1	Corn-no till	143	1291
2	Corn-min till	83	751
3	Grass/Pasture	50	447
4	Grass/Trees	75	672
5	Hay-windrowed	49	440
6	Soybean-no till	97	871
7	Soybean-min till	247	2221
8	Soybean-clean till	61	553
9	Woods	129	1165
	Total	934	8411

Tuning Parameter Estimation and Assessment

For the MFA and SMFA algorithms, we need to estimate the number of mixture components M, and the dimensionality of latent factors d. In the experiments, $M \in \{2, 3, ..., Mc\}$, where Mc is the maximal number of classes considered, $d \in \{3, 4, ..., D/2\}$, where D is the input dimensionality of original datasets, we use five-fold cross-validation to obtain the optimal parameters M and d.

The assessment of the effect of the tuning parameters (M and d) on the performance of the proposed methods was of interest. Since we were interested in the classification accuracy, we investigated the effect of the number of mixture components M and the dimensionality of latent factors d on OA. Figure 5.9(a)-(b) show the 3-dimensional surface of the OA of MFA and SMFA with respect to the values of parameters M and d for the SVM classifier, respectively. It can be seen that the OA gradually increases in the beginning as the d increases, and then keeps stable with slight fluctuation as

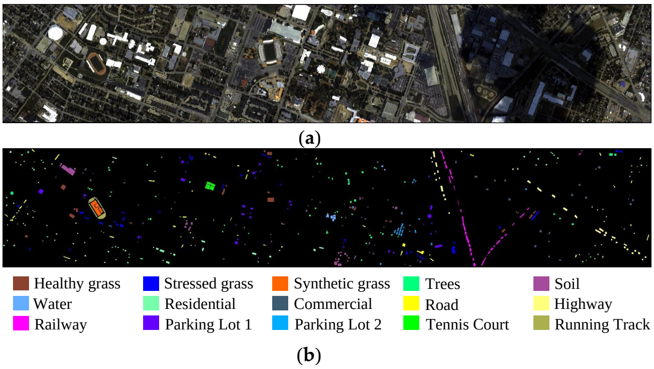


Figure 5.6: Houston dataset. (**a**) Three-band false-color image. (**b**) Ground truth-map reference.

Table 5.2: Houston dataset: Number of training and test samples.

Class number	Class name	Training samples	Test samples
1	Healthy grass	198	1053
2	Stressed grass	190	1064
3	Synthetic grass	192	505
4	Trees	188	1056
5	Soil	186	1056
6	Water	182	143
7	Residential	196	1072
8	Commercial	191	1053
9	Road	193	1059
10	Highway	191	1036
11	Railway	181	1054
12	Parking Lot 1	192	1041
13	Parking Lot 2	184	285
14	Tennis Court	181	247
15	Running Track	187	473
	Total	2832	12197

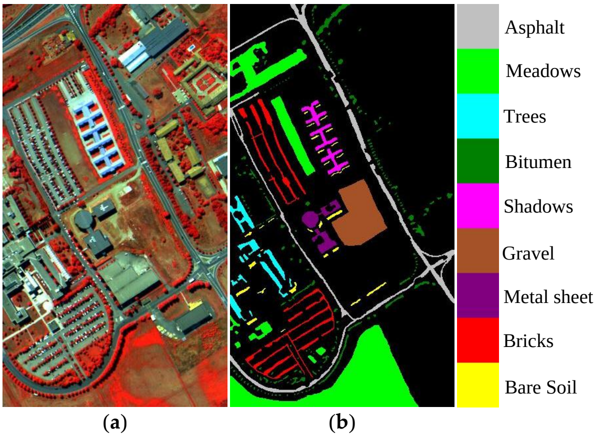


Figure 5.7: University of Pavia dataset. (**a**) Three-band false-color image. (**b**) Ground truth-map reference.

Table 5.3: University of Pavia dataset: Number of training and test samples.

Class number	Class name	Training samples	Test samples
1	Asphalt	663	5968
2	Meadows	1865	16784
3	Gravel	210	1889
4	Trees	306	2758
5	Painted metal sheets	135	1210
6	Bare Soil	503	4526
7	Bitumen	133	1197
8	Self-Blocking Bricks	368	3314
9	Shadows	95	852
	Total	4278	38498

Table 5.4: Salinas dataset: Number of training and test samples.

Class number	Class name	Training samples	Test samples
1	Brocoli_green_weeds_1	201	1808
2	Brocoli_green_weeds_2	373	3353
3	Fallow	198	1778
4	Fallow_rough_plow	139	1255
5	Fallow_smooth	268	2410
6	Stubble	396	3563
7	Celery	358	3221
8	Grapes_untrained	1127	10144
9	Soil_vinyard_develop	620	5583
10	Corn_senesced_green_weeds	328	2950
11	Lettuce_romaine_4wk	107	961
12	Lettuce_romaine_5wk	193	1734
13	Lettuce_romaine_6wk	92	824
14	Lettuce_romaine_7wk	107	963
15	Vinyard_untrained	727	6541
16	Vinyard_vertical_trellis	181	1626
	Total	5415	48714

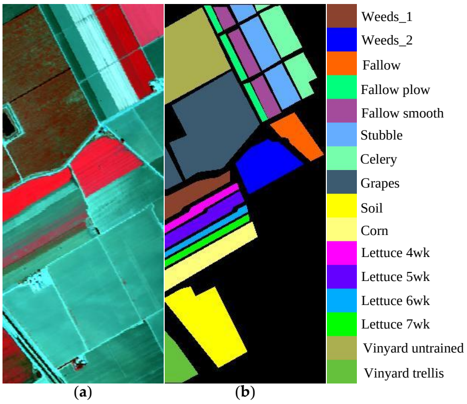


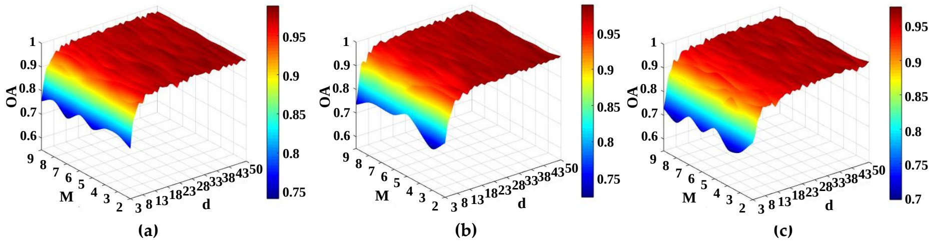
Figure 5.8: Salinas dataset. (**a**) Three-band false-color image. (**b**) Ground truth-map reference.

Table 5.5: CPU processing times in seconds by different DR methods applied to the Indian Pines (INPS), Houston (HSN), University of Pavia (UPA), and Salinas (SAS) datasets (the number of features=20).

Datasets	PCA	PPCA	FA	LDA	NWFE	MFA	DMFA	SMFA
INPS	0.11	63.66	70.89	0.12	1.05	3.85	18.07	0.12
HSN	2.91	1335.09	21.45	0.89	5.02	103.77	151.91	4.54
UPA	0.62	285.01	4.01	0.21	0.91	28.10	79.30	1.55
SAS	0.56	356.44	22.15	0.38	–	18.97	87.85	1.62

the d increases, but decreases slightly when the d reaches some value. It can also be observed that the OA is insensitive to M.

For the DMFA algorithm, we need to estimate the number of mixture components M and K_m, and the dimensionality of latent factors $d^{(1)}$ and $d^{(2)}$, in the first and second layer of DMFA, respectively. In the experiments, M is the same as in the MFA, $K_m \in \{2, ..., M\}$, we set $K_m = 2$ for all m in our experiments. $d^{(1)}$ is the same as d, $d^{(2)} \in \{3, 4, ..., d^{(1)}\}$. Five-fold cross-validation can be used to obtain the optimal parameters. To analyze the impact of the number of mixture components M and the dimensionality of latent factors $d^{(1)}$ in the first layer on the performance of DMFA, the output results of the first layer of DMFA were used for HSI classification. Figure 5.9(c) shows the OAs with respect to the values of parameters M and $d^{(1)}$. Figure 5.9(a) and (b) show similar things for MFA and SMFA, respectively.



(a) (b) (c)

Figure 5.9: OAs versus the reduced dimensionality d and the number of mixture components M in the proposed methods with SVM classifier on the Indian Pines dataset. (**a**) MFA (**b**) SMFA, and (**c**) DMFA in the first layer.

Classification

The first experiment was performed on the Indian Pines dataset (Appendix B.3). Figure 5.10 shows the classification maps obtained by different methods. From the figures, it can be seen that the classification maps obtained by PCA, PPCA, FA, LDA, and NWFE are not very satisfactory since they have lots of visible noise. By contrast, MFA, DMFA, and SMFA give much better classification maps, all of them have a smoother appearance and preserve more details on edges. Besides visual comparison, Table 5.6 presents the quantitative classification results for all the methods, and there it can be observed that the MFA, DMFA, and SMFA achieve much higher classification accuracies than PCA, PPCA, and FA, respectively. These results imply that the performance of DR could be improved by considering the Gaussian mixture model. Moreover, DMFA and SMFA clearly outperform MFA, and the performance of DMFA and SMFA is similar, both of them present the highest OA, AA, and KC and achieves most of the top classification accuracy values for individual classes. This indicates that MFA, DMFA, and SMFA could extract more useful information for classification from a complicated HSI. Moreover, SMFA is better than LDA and NWFE, this means that SMFA is an effective supervised DR method. MFA, DMFA, and SMFA improve the OA by 11.11%, 13.38%, and 13.42% using SVM compared to other methods in the experiment, respectively. It is interesting to note that all DR methods based on FA (SMFA, DMFA, MFA, and FA) gave a better performance than PCA and PPCA. The reason for this is that noise could be distributed inconsistently for different components in real HSI. Table 5.6 also gives the STDs of classification results for different DR methods for the Indian Pines dataset. It can be seen that all the methods give similar and stable classification results. Table 5.5 compares the CPU processing time (in seconds) used by different DR methods for the Indian Pines dataset. All methods were implemented in Matlab R2019a on a computer having Intel(R) Core(TM) i7-6700 processor (3.40GHz), 8.00 GB of memory, and 64-bit Windows 10 Operating System. It can be seen that the running times for MFA, DMFA, and SMFA were 3.85, 18.07, and 0.12 s, respectively. It is worth noting that the running time for the supervised methods (LDA, NWFE, and SMFA) is affected considerably by the number of labeled (training) samples used, and the unsupervised methods are affected by the total size of the dataset.

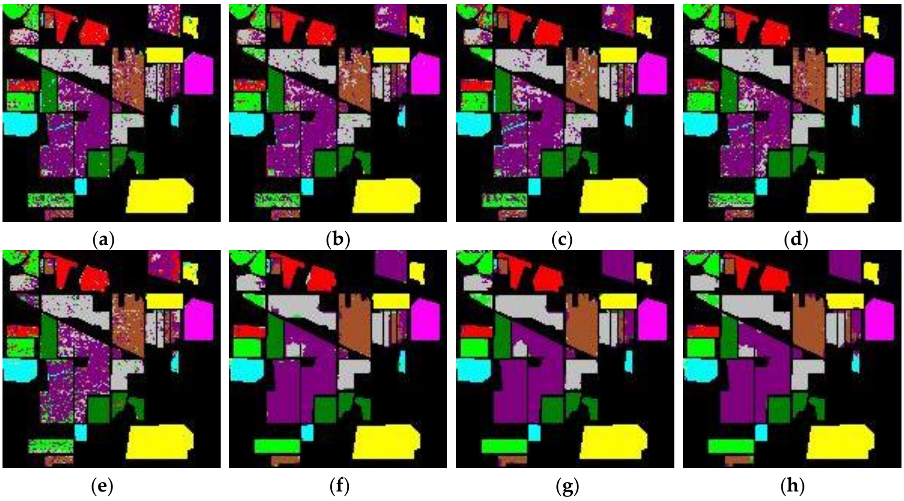


Figure 5.10: Classification maps for the Indian Pines dataset obtained by SVM classification after using (**a**) PCA, (**b**) PPCA, (**c**) FA, (**d**) LDA, (**e**) NWFE, (**f**) MFA, (**g**) DMFA, and (**h**) SMFA DR methods.

Table 5.6: The classification results (%) of different DR methods on the Indian Pines dataset, the best results are in bold typeface. The row of each class number (CN) is the mean accuracy ± standard deviation based on ten runs.

CN	PCA	PPCA	FA	LDA	NWFE	MFA	DMFA	SMFA
1	75.13 ± 2.39	81.22 ± 1.39	86.48 ± 1.93	89.31 ± 0.98	84.51 ± 2.25	93.00 ± 1.87	**97.49** ± 1.51	96.34 ± 1.32
2	83.33 ± 2.16	84.00 ± 4.21	89.30 ± 3.49	72.44 ± 3.97	82.82 ± 2.16	91.71 ± 1.05	**99.17** ± 0.89	95.61 ± 1.04
3	94.51 ± 1.62	94.77 ± 2.27	**97.48** ± 0.89	92.84 ± 2.62	91.50 ± 1.75	94.42 ± 1.08	97.35 ± 1.66	95.08 ± 1.51
4	96.09 ± 1.02	95.95 ± 0.99	97.65 ± 0.98	97.32 ± 1.62	98.51 ± 1.49	98.53 ± 0.40	98.39 ± 0.47	**98.81** ± 1.08
5	99.55 ± 0.14	99.77 ± 0.18	99.84 ± 0.16	99.55 ± 0.23	99.32 ± 0.16	99.83 ± 0.18	**99.86** ± 0.14	99.84 ± 0.18
6	75.03 ± 1.99	82.06 ± 2.27	82.50 ± 2.14	80.37 ± 2.89	82.43 ± 3.00	90.53 ± 0.68	95.48 ± 1.40	**97.36** ± 0.62
7	78.22 ± 1.85	77.62 ± 1.45	88.40 ± 1.70	82.85 ± 2.05	88.74 ± 2.11	95.56 ± 1.58	96.91 ± 1.22	**98.24** ± 1.42
8	83.54 ± 2.68	83.00 ± 2.34	89.21 ± 2.68	77.76 ± 2.74	79.57 ± 3.56	96.75 ± 1.34	98.74 ± 1.03	**99.28** ± 1.06
9	99.40 ± 0.32	99.83 ± 0.41	99.66 ± 0.26	99.57 ± 0.36	99.49 ± 0.55	99.83 ± 0.30	99.74 ± 0.54	**99.91** ± 0.46
AA	87.20 ± 0.35	88.69 ± 0.73	92.30 ± 0.67	88.00 ± 0.64	89.65 ± 0.88	95.59 ± 1.01	**98.14** ± 0.86	97.85 ± 0.51
OA	84.48 ± 0.35	85.98 ± 0.39	90.96 ± 0.60	87.20 ± 0.56	89.28 ± 0.82	95.59 ± 0.86	97.86 ± 0.72	**97.90** ± 0.60
KC	0.8173 ± 0.0040	0.8345 ± 0.0048	0.8940 ± 0.0072	0.8495 ± 0.0065	0.8738 ± 0.0096	0.9459 ± 0.0102	0.9749 ± 0.0085	**0.9753** ± 0.0070

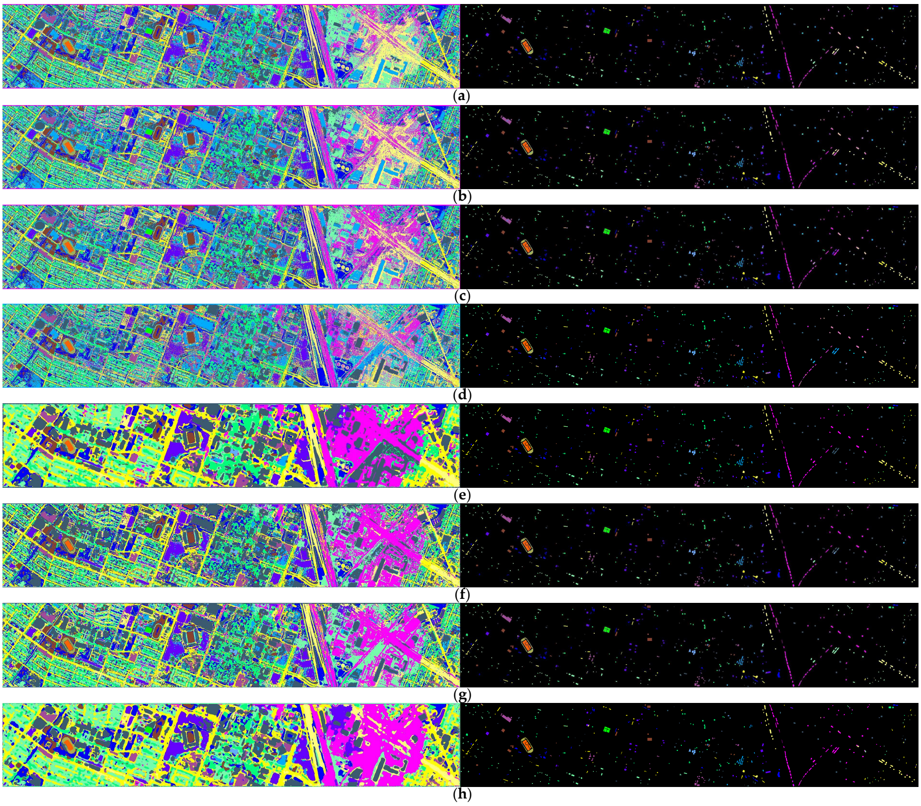


Figure 5.11: Classification maps for the Houston dataset obtained by SVM classification after using (**a**) PCA, (**b**) PPCA, (**c**) FA, (**d**) LDA, (**e**) NWFE, (**f**) MFA, (**g**) DMFA, and (**h**) SMFA DR methods.

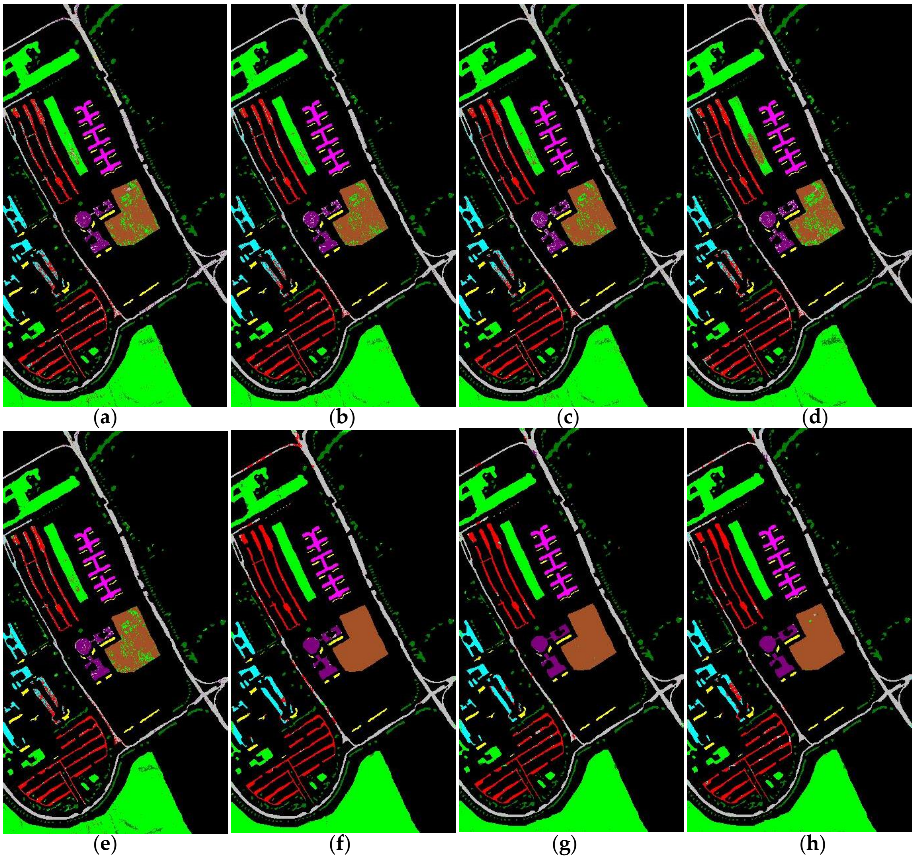


Figure 5.12: Classification maps for the University of Pavia dataset obtained by SVM classification after using (**a**) PCA, (**b**) PPCA, (**c**) FA, (**d**) LDA, (**e**) NWFE, (**f**) MFA, (**g**) DMFA, and (**h**) SMFA DR methods.

Table 5.7: The classification results (%) of different DR methods on the Houston dataset, the best results are in bold typeface. The row of each class number (CN) is the mean accuracy ± standard deviation based on ten runs.

CN	**PCA**	**PPCA**	**FA**	**LDA**	**NWFE**	**MFA**	**DMFA**	**SMFA**
1	98.05	96.20 ± 0.29	96.37	80.53	80.72 ± 0.45	98.98 ± 0.89	**99.31** ± 1.66	79.49 ± 1.37
2	96.01	**96.36** ± 0.13	95.97	80.92	82.99 ± 0.18	95.33 ± 1.63	95.24 ± 1.58	83.46 ± 1.57
3	**100**	99.97 ± 0.27	**100**	**100**	99.42 ± 0.58	99.76 ± 0.24	99.82 ± 0.19	99.48 ± 0.34
4	98.05	97.35 ± 1.36	97.72	92.90	89.49 ± 0.94	**99.89** ± 0.32	99.47 ± 1.78	89.30 ± 2.09
5	97.41	97.45 ± 0.16	97.11	96.97	98.86 ± 0.38	98.42 ± 0.82	98.11 ± 1.09	**99.96** ± 0.11
6	99.94	95.07 ± 0.46	95.30	93.71	95.11 ± 0.21	47.24 ± 0.80	**99.97** ± 0.27	92.31 ± 0.35
7	81.34	82.05 ± 0.31	87.91	84.80	79.76 ± 2.44	66.22 ± 1.38	**89.53** ± 3.02	82.37 ± 1.23
8	81.45	68.87 ± 0.17	84.18	71.13	72.46 ± 4.73	84.06 ± 1.14	81.82 ± 2.64	**90.03** ± 0.11
9	83.40	78.34 ± 0.60	78.08	72.52	75.54 ± 1.19	87.65 ± 1.04	**91.66** ± 0.66	83.95 ± 1.08
10	70.03	83.58 ± 0.48	65.32	73.17	81.27 ± 3.21	86.21 ± 0.24	66.67 ± 4.76	**97.97** ± 1.59
11	67.86	69.57 ± 1.86	57.04	85.39	**93.55** ± 2.10	68.21 ± 0.16	74.51 ± 0.59	88.99 ± 4.01
12	85.50	89.15 ± 5.12	82.68	82.52	85.11 ± 1.86	95.80 ± 2.28	94.07 ± 2.64	**96.25** ± 1.19
13	30.69	30.06 ± 1.33	43.85	70.53	70.88 ± 1.74	**92.52** ± 3.26	89.56 ± 2.31	65.26 ± 0.89
14	99.18	98.39 ± 0.24	89.05	99.19	99.60 ± 0.31	94.64 ± 0.83	97.24 ± 1.13	**99.96**± 0.22
15	**100**	99.41 ± 0.59	**100**	95.98	98.73 ± 0.50	99.54 ± 0.54	99.88 ± 0.14	91.97 ± 1.61
AA	85.93	85.50 ± 0.50	84.70	85.35	86.94 ± 0.47	87.68 ± 0.36	**91.81** ± 0.39	89.42 ± 0.69
OA	83.45	83.75 ± 0.59	82.05	83.59	85.35 ± 0.56	86.65 ± 0.42	88.72 ± 0.49	**89.40** ± 0.67
KC	0.8207	0.8240 ± 0.0065	0.8053	0.8223	0.8412 ± 0.0059	0.8552 ± 0.0046	0.8775 ± 0.0053	**0.8848** ± 0.0075

Table 5.8: The classification results (%) of different DR methods on the University of Pavia dataset, the best results are in bold typeface. The row of each class number (CN) is the mean accuracy ± standard deviation based on ten runs.

CN	**PCA**	**PPCA**	**FA**	**LDA**	**NWFE**	**MFA**	**DMFA**	**SMFA**
1	92.00 ± 1.01	93.16 ± 0.82	94.65 ± 0.58	93.20 ± 0.55	94.10 ± 0.49	98.07 ± 0.45	98.39 ± 0.66	**98.69** ± 0.27
2	95.71 ± 0.53	96.12 ± 0.25	96.43 ± 0.24	96.86 ± 0.41	97.52 ± 0.38	99.68 ± 0.10	99.02 ± 0.42	**99.79** ± 0.08
3	78.98 ± 2.06	82.23 ± 1.86	80.16 ± 1.72	76.13 ± 2.81	74.01 ± 4.00	90.15 ± 2.29	**98.39** ± 1.44	94.28 ± 1.93
4	94.24 ± 0.95	94.61 ± 0.82	93.86 ± 0.78	92.57 ± 0.90	89.63 ± 1.26	95.65 ± 0.36	**98.72** ± 0.95	96.27 ± 0.61
5	99.41 ± 2.27	98.99 ± 2.01	99.21 ± 1.23	99.17 ± 0.16	99.59 ± 0.13	99.89 ± 0.12	**99.93**± 0.35	99.81 ± 0.09
6	90.96 ± 2.01	91.80 ± 0.75	93.54 ± 1.09	81.11 ± 1.06	90.39 ± 1.72	98.63 ± 0.30	99.80 ± 0.88	**99.98** ± 0.30
7	85.40 ± 4.34	88.69 ± 1.98	90.08 ± 2.19	80.95 ± 1.35	83.12 ± 4.17	94.32 ± 0.76	99.75 ± 1.06	**99.92** ± 0.68
8	77.35 ± 2.14	83.85 ± 0.97	84.40 ± 1.61	86.33 ± 2.03	84.91 ± 2.30	94.63 ± 0.81	97.15 ± 1.12	**97.31** ± 1.54
9	91.55 ± 1.78	**99.89**± 0.58	99.31± 0.87	99.65 ± 0.18	99.88 ± 0.11	97.07 ± 0.93	99.64 ± 1.07	97.77 ± 0.58
AA	89.57 ± 0.53	92.27 ± 0.39	92.57 ± 0.48	89.55 ± 0.19	90.35 ± 0.66	96.47 ± 0.30	**98.98** ± 0.49	98.22 ± 0.25
OA	91.78 ± 0.29	93.32 ± 0.27	93.80 ± 0.29	91.85 ± 0.18	93.02 ± 0.32	97.90 ± 0.14	**98.87** ± 0.41	**98.87** ± 0.14
KC	0.8908 ± 0.0038	0.9112 ± 0.0036	0.9177 ± 0.0038	0.8314 ± 0.0023	0.9071 ± 0.0043	0.9722 ± 0.0019	**0.9850** ± 0.0055	**0.9850** ± 0.0019

The second experiment was performed on the Houston dataset (Appendix B.5). Figure 5.11 shows the classification maps obtained by different methods. From the figures, we can see that the proposed MFA, DMFA, and SMFA algorithms also outperform the other algorithms. Table 5.7 presents the quantitative classification results of the different DR methods. As shown in Table 5.7, the performance of DMFA and SMFA is better than MFA and much better than PCA, PPCA, FA, LDA, and NWFE. MFA, DMFA, and SMFA improve the OA by 4.60%, 6.67%, and 7.35% compared to other methods in the experiment, respectively. Table 5.7 also presents the STDs of classification results. It can be seen that FA and LDA present the most stable results. MFA, DMFA, and SMFA have slight fluctuation for each experiment and give relatively stable results.

The third and fourth experiments were performed on the University of Pavia (Appendix B.2) and Salinas (Appendix B.6) datasets. It should be noted that the NWFE

Table 5.9: The classification results (%) of different DR methods on the Salinas dataset, the best results are in bold typeface. The row of each class number (CN) is the mean accuracy ± standard deviation based on ten runs.

CN	PCA	PPCA	FA	LDA	MFA	DMFA	SMFA
1	99.91± 0.17	99.32± 0.02	99.29± 0.38	99.95 ± 0.05	**99.94**± 0.09	99.82 ± 0.08	99.89± 0.18
2	99.88 ± 0.23	99.97 ± 0.09	99.82 ± 0.14	99.79 ± 0.03	99.79 ± 0.12	**99.99**± 0.21	**99.99**± 0.12
3	98.88 ± 0.36	99.04 ± 0.29	99.21 ± 0.16	99.78 ± 0.12	99.83 ± 0.07	99.72 ± 0.21	**99.96**± 0.22
4	99.13 ± 0.48	**99.76** ± 0.42	**99.76** ± 0.30	99.12 ± 0.53	98.96 ± 0.50	97.93 ± 0.46	98.41 ± 0.18
5	99.05 ± 0.64	98.19 ± 0.56	99.13 ± 0.15	98.96 ± 0.32	99.50 ± 0.34	**99.92** ± 0.42	99.17 ± 0.20
6	99.94 ± 0.20	99.92± 0.09	99.03± 0.06	99.83 ± 0.08	99.23 ± 0.20	**99.98**± 0.14	99.97 ± 0.05
7	99.94 ± 0.19	99.89± 0.14	**99.95**± 0.20	99.94 ± 0.09	99.81 ± 0.10	99.91 ± 0.13	99.91 ± 0.12
8	88.88 ± 0.92	87.55 ± 1.01	84.53 ± 1.12	90.09 ± 1.29	97.73 ± 0.91	99.25 ± 0.04	**99.71** ± 0.60
9	99.44 ± 0.16	99.38 ± 0.14	99.34 ± 0.15	99.95 ± 0.20	99.91 ± 0.07	99.93 ± 0.29	**99.99**± 0.03
10	96.27 ± 0.53	98.19 ± 0.72	97.80 ± 0.82	98.61 ± 0.45	98.99 ± 0.44	99.49 ± 0.83	**99.76** ± 0.87
11	99.03 ± 0.80	99.04 ± 0.74	99.68 ± 0.47	98.86 ± 0.41	**99.97** ± 0.19	99.90 ± 1.08	**99.97**± 0.28
12	98.36 ± 0.14	98.86 ± 0.07	99.26 ± 0.15	**99.97**± 0.32	99.94 ± 0.49	99.71 ± 0.15	99.89 ± 0.13
13	99.52 ± 0.51	99.76 ± 0.25	99.76 ± 0.22	99.39 ± 0.29	99.43± 0.09	99.52 ± 0.60	**99.96**± 0.45
14	99.79 ± 1.02	98.43 ± 0.83	99.16 ± 0.58	96.78 ± 0.93	99.93 ± 1.23	99.27 ± 1.60	**99.99**± 0.15
15	84.51 ± 1.20	85.68 ± 1.54	83.23 ± 1.96	72.62 ± 2.44	97.38 ± 1.15	**98.03** ± 1.64	97.75 ± 2.26
16	99.38 ± 0.30	99.75 ± 0.74	99.58± 1.15	98.65 ± 0.44	99.76± 0.97	99.08 ± 0.63	**99.92**± 0.80
AA	97.62 ± 0.15	87.73 ± 0.12	97.54 ± 0.09	97.02 ± 0.12	99.49 ± 0.08	99.47 ± 0.26	**99.66** ± 0.14
OA	95.09 ± 0.17	95.08 ± 0.09	94.19 ± 0.12	93.91 ± 0.10	99.02 ± 0.20	99.40 ± 0.40	**99.53** ± 0.34
KC	0.9453 ± 0.0019	0.9452 ± 0.0010	0.9353 ± 0.0013	0.9321 ± 0.0011	0.9890 ± 0.0022	0.9933 ± 0.0045	**0.9948** ± 0.0038

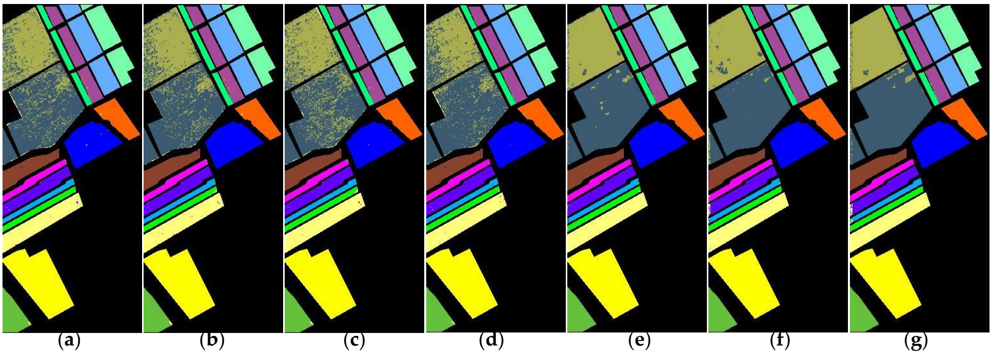


Figure 5.13: Classification maps for the Salinas dataset obtained by SVM classification after using (**a**) PCA, (**b**) PPCA, (**c**) FA, (**d**) LDA, (**e**) MFA, (**f**) DMFA, and (**g**) SMFA DR methods.

method does not work for the Salinas dataset. Therefore, in the experiments of the Salinas dataset, there are no experimental results for NWFE. Figures 5.12 and 5.13 show the classification maps obtained by different methods on the University of Pavia and Salinas datasets, respectively. From Figures 5.12 and 5.13, it can be seen that the classification maps obtained by MFA, DMFA, and SMFA are much better than PCA, PPCA, FA, LDA, and NWFE, respectively. Tables 5.8 and 5.9 present the quantitative classification results. As shown in Tables 5.8 and 5.9, the classification accuracies of the proposed MFA, DMFA, and SMFA methods are much better than PCA, PPCA, and FA methods. These results further demonstrate that, instead of using a single Gaussian distribution model, the performance of DR could be improved by considering the Gaussian mixture model. For the University of Pavia dataset, MFA, DMFA, and SMFA improve the OA by 6.05%, 7.02%, and 7.02% compared to other methods used in the experiment, respectively. For the Salinas dataset, MFA, DMFA, and SMFA improve the OA by 5.11%, 5.49%, and 5.62% compared to other methods used in the experiment, respectively. Moreover, in the experiments, DMFA and SMFA are clearly better than MFA, and DMFA and SMFA give similar and highest OA, AA, and KC and also achieve most of the top classification accuracy values for the individual classes. This also further indicates that MFA, DMFA, and SMFA could extract more effective information for classification from a complicated HSI. Tables 5.8 and 5.9 also give the STDs of classification results for different DR methods for the University of Pavia and Salinas datasets, respectively. It can be seen that all the methods give similar and relatively stable classification results. This also further demonstrates that MFA, DMFA, and SMFA are stable and effective DR methods.

5.2 Semi-Supervised MFA-Based Feature Extraction Methods

5.2.1 Introduction

MFA, DMFA, SMFA, and SDMFA are unsupervised feature extraction (FE) methods. They only use unlabeled or labeled samples to extract feature of HSI. In this chapter, we propose two semi-supervised FE methods, semi-supervised mixtures of factor analyzers (S^2MFA) and semi-supervised deep mixtures of factor analyzers (S^2DMFA), for HSI. For the S^2MFA method, first, the training set is created by randomly selecting some labeled samples and unlabeled samples of the HSI. Second, the training data is segmented using a Gaussian mixture model. Each segmented region is then fitted to a Gaussian distribution and simultaneously contains labeled and unlabeled samples. Third, a factor analyzer is used for each region to get a factor loading matrix. This matrix uses the labeled samples to improve the class discrimination and employs both labeled and unlabeled samples to preserve the local spatial features of the data. Also, the matrix transforms the original HSI to an optimal low-dimensional subspace to achieve DR. S^2DMFA is two-layer S^2MFA. This chapter extends our previous unsupervised MFA and unsupervised DMFA methods [51, 52] to semi-supervised MFA amd semi-supervised DMFA methods to effectively extract HSI features utilizing both labeled and unlabeled samples.

5.2.2 Proposed Semi-Supervised FE methods

S^{2}MFA

Let $\mathbf{x} \in \mathbb{R}^{D\times 1}$ denote a spectral vector that has a label y, $\mathbf{z} \in \mathbb{R}^{d\times 1}$ denotes a latent vector, $m \in \{1, ..., M\}$ denotes the component indicator variable of the M factor analyzers in S^{2}MFA. ω is a M-dimensional binary random vector, where $\omega_m = 1$ when the data is generated to the m-th Gaussian components and 0 otherwise. The standard multivariate normal prior on the latent factors is

$$p(\mathbf{z}|\omega_m) = p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}_d), \tag{5.15}$$

$$p(\omega_m) = \pi_m, \sum_{m=1}^{M} \pi_m = 1, \tag{5.16}$$

where $\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}_d)$ means that $\mathbf{z}$ is Gaussian vector with zero mean and $d \times d$ identity matrix $\mathbf{I}_d$ as the covariance matrix and π_m is the m-th mixing proportion.

The S^{2}MFA model is described by

$$p(\mathbf{x}, y|\mathbf{z}, \omega_m) = p(\mathbf{x}|\mathbf{z}, \omega_m)p(y|\mathbf{z}, \omega_m), \tag{5.17}$$

$$p(\mathbf{x}|\mathbf{z}, \omega_m) = \mathcal{N}(\mathbf{x}; \mathbf{W}_m\mathbf{z} + \boldsymbol{\mu}_m, \boldsymbol{\Psi}), \tag{5.18}$$

$$p(y|\mathbf{z}, \omega_m) = \mathcal{N}(y; \mathbf{w}_{ym}\mathbf{z} + \mu_{ym}, \sigma_y^2), \tag{5.19}$$

where the parameters of the m-th factor analyzer include a mixing proportion π_m, a $D \times d$ factor loading matrix $\mathbf{W}_m$, a $D \times 1$ mean $\boldsymbol{\mu}_m$, and a $D \times D$ diagonal matrix $\boldsymbol{\Psi}$ which represents the independent noise variances for each of the observed variables. σ_y^2 is the variance, $\mathbf{w}_{ym} \in \mathbb{R}^{1\times d}$ and $\mu_{ym} \in \mathbb{R}^{1\times 1}$ are defined similarly.

Labeled samples $(\mathbf{x}_i, y_i)$, $i = 1, ..., n$, and unlabeled samples $\mathbf{x}_i$, $i = n+1, ..., n+u$ are assumed and the parameters of the model are estimated using the expectation maximization algorithm [83].

The EM functional is given by

$$\begin{aligned} Q = \mathbb{E}_{\boldsymbol{\theta}^{\text{old}}}[\log \prod_{m=1}^{M} \{ \prod_{i=1}^{n+u} \{(2\pi)^{-\frac{D}{2}} \left|\boldsymbol{\Psi}\right|^{-\frac{1}{2}} \\ \times \exp[-\frac{1}{2}(\mathbf{x}_i - \widetilde{\mathbf{W}}_m\tilde{\mathbf{z}})^T\boldsymbol{\Psi}^{-1}(\mathbf{x}_i - \widetilde{\mathbf{W}}_m\tilde{\mathbf{z}})]\}^{\omega_m} \\ \times \prod_{i=1}^{n}\{(2\pi)^{-\frac{1}{2}}\sigma_y^{-1}\exp[-\frac{1}{2\sigma_y^2}(y_i - \tilde{\mathbf{w}}_{ym}\tilde{\mathbf{z}})^2]\}^{\omega_m}\}|\mathbf{x}, y], \end{aligned} \tag{5.20}$$

where $\mathbb{E}_{\boldsymbol{\theta}^{\text{old}}}[\bullet]$ denotes expectation with $\boldsymbol{\theta}^{\text{old}}$ fixed, $\tilde{\mathbf{z}} = [\mathbf{z}^T \ 1]^T$, $\widetilde{\mathbf{W}}_m = [\mathbf{W}_m \ \boldsymbol{\mu}_m]$, and $\tilde{\mathbf{w}}_{ym} = [\mathbf{w}_{ym} \ \mu_{ym}]$. Now let $h_{im} \triangleq \mathbb{E}_{\boldsymbol{\theta}^{\text{old}}}[\omega_m|\mathbf{x}_i, y_i]$, $\tilde{h}_{im} \triangleq \mathbb{E}_{\boldsymbol{\theta}^{\text{old}}}[\omega_m|\mathbf{x}_i]$, then

$$\begin{aligned}
Q = c - & \frac{n+u}{2}\log|\boldsymbol{\Psi}| + \sum_{i=1}^{n}\sum_{m=1}^{M}\{-\frac{1}{2}h_{im}\mathbf{x}_i^T\boldsymbol{\Psi}^{-1}\mathbf{x}_i \\
& + h_{im}\mathbf{x}_i^T\boldsymbol{\Psi}^{-1}\widetilde{\mathbf{W}}_m\mathbb{E}[\tilde{\mathbf{z}}|\mathbf{x}_i, y_i, \omega_m] \\
& - \frac{1}{2}h_{im}\text{tr}(\widetilde{\mathbf{W}}_m^T\boldsymbol{\Psi}^{-1}\widetilde{\mathbf{W}}_m\mathbb{E}[\tilde{\mathbf{z}}\tilde{\mathbf{z}}^T|\mathbf{x}_i, y_i, \omega_m])\} \\
& + \sum_{i=n+1}^{n+u}\sum_{m=1}^{M}\{-\frac{1}{2}\tilde{h}_{im}\mathbf{x}_i^T\boldsymbol{\Psi}^{-1}\mathbf{x}_i \\
& + \tilde{h}_{im}\mathbf{x}_i^T\boldsymbol{\Psi}^{-1}\widetilde{\mathbf{W}}_m\mathbb{E}[\tilde{\mathbf{z}}|\mathbf{x}_i, \omega_m] \\
& - \frac{1}{2}\tilde{h}_{im}\text{tr}(\widetilde{\mathbf{W}}_m^T\boldsymbol{\Psi}^{-1}\widetilde{\mathbf{W}}_m\mathbb{E}[\tilde{\mathbf{z}}\tilde{\mathbf{z}}^T|\mathbf{x}_i, \omega_m])\} \\
& - \frac{n}{2}\log\sigma_y^2 + \sum_{i=1}^{n}\sum_{m=1}^{M}\{-\frac{1}{2}h_{im}\sigma_y^{-2}y_i^2 \\
& + h_{im}\sigma_y^{-2}y_i\tilde{\mathbf{w}}_{ym}\mathbb{E}[\tilde{\mathbf{z}}|\mathbf{x}_i, y_i, \omega_m] \\
& - \frac{1}{2}h_{im}\sigma_y^{-2}\tilde{\mathbf{w}}_{ym}\mathbb{E}[\tilde{\mathbf{z}}\tilde{\mathbf{z}}^T|\mathbf{x}_i, y_i, \omega_m]\tilde{\mathbf{w}}_{ym}^T\},
\end{aligned} \tag{5.21}$$

where c is constant. The EM algorithm is based on iterating between the expectation step (E-step) and the maximization step (M-step) until convergence. The E-step and the M-step are as follows:

E-step: compute h_{im}, $\tilde{h}_{im}$, $\mathbb{E}[\mathbf{z}|\mathbf{x}_i, y_i, \omega_m]$, $\mathbb{E}[\mathbf{z}|\mathbf{x}_i, \omega_m]$, $\mathbb{E}[\mathbf{z}\mathbf{z}^T|\mathbf{x}_i, y_i, \omega_m]$, and $\mathbb{E}[\mathbf{z}\mathbf{z}^T|\mathbf{x}_i, \omega_m]$.

M-step: update $\boldsymbol{\theta}$, i.e., π_m, $\mathbf{W}_m$, $\mathbf{w}_{ym}$ $\boldsymbol{\mu}_m$, μ_{ym}, σ_y^2, and $\boldsymbol{\Psi}$ corresponding to the maximum of Q.

When $\forall\, i \in \{1, 2, ..., n\}$, $\mathbf{x}_i$ has a label y_i, the class discrimination can be improved as the factor loading matrix $\mathbf{W}_m$ is constrained by the label information in (5.18) and (5.19). The local spatial information can be preserved based on the labeled and unlabeled samples as the same classified pixels are segmented to the same segments using Gaussian mixture model. By combining the labeled with unlabeled samples together, S^{2}MFA achieves high class discrimination information and simultaneously preserves the sufficient local spatial information for HSI. The schematic of the S^{2}MFA is shown in Fig. 5.14 and Fig. 5.15.

S^{2}DMFA

For S^{2}DMFA, we also only consider two-layer S^{2}DMFA model. The algorithm of S^{2}DMFA is similar as S^{2}MFA. In this chapter, we omit the process of S^{2}DMFA.

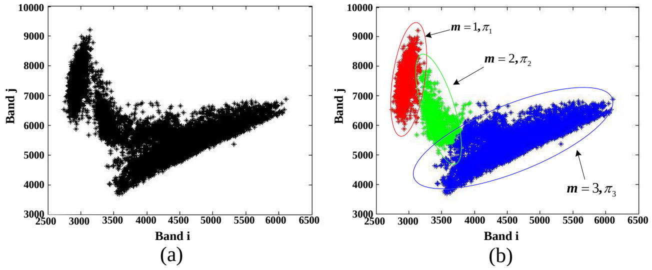


FIGURE 5.14: (a) The distribution of all the training samples of the Indian Pines dataset for bands i=21 and j=41 indicate non-Gaussian distribution. (b) Illustration of S^{2}MFA model with each ellipse representing a Gaussian component. S^{2}MFA has three components colored red ($m = 1$), green ($m = 2$) and blue ($m = 3$). Their mixing proportions are given by π_m.

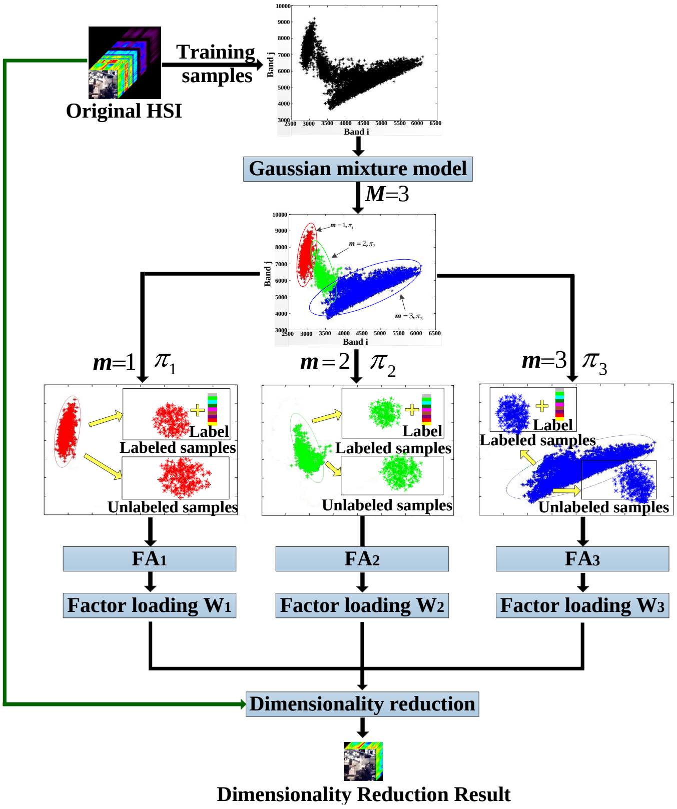


FIGURE 5.15: The schematic of the S^{2}MFA corresponding to Fig. 5.14(b).

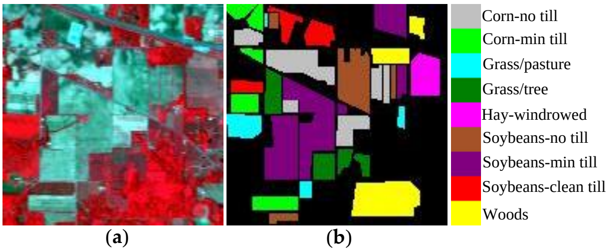


Figure 5.16: Indian Pines dataset. (a) Three-band false-color image. (b) Ground truth-map containing nine land-cover classes.

5.2.3 Experiments and Results

In this section, the performances of the S^{2}MFA method were validated using the classification of the support vector machine (SVM) classifier on the Indian Pines and Houston datasets.

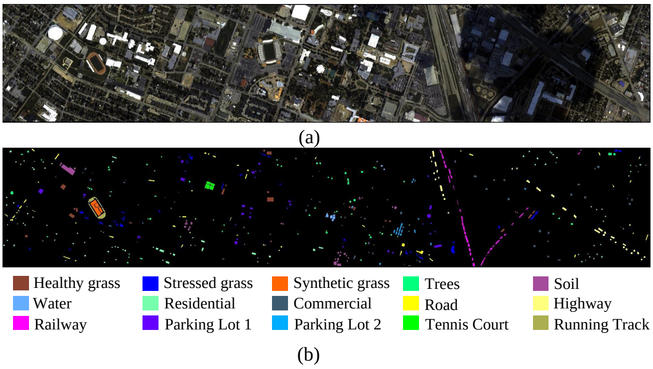
Table 5.10: Indian Pines dataset: Number of training and test samples.

NO.	Class name	Training samples	Test samples
1	Corn-no till	5	1429
2	Corn-min till	5	829
3	Grass/Pasture	5	492
4	Grass/Trees	5	742
5	Hay-windrowed	5	484
6	Soybean-min till	5	963
7	Soybean-min till	5	2463
8	Soybean-clean till	5	609
9	Woods	5	1289
	Total	45	9300

Experimental Datasets

The first dataset is the Indian Pines dataset (Appendix B.5). Fig. 5.16 shows the false-color composite of the Indian Pines image and the corresponding ground reference map, respectively.

The second experiment was performed on the Houston dataset (Appendix B.5). The HSI contains fifteen classes of interest. Fig. 5.17 shows the false-color composite of the Houston image and the corresponding ground reference map. The spatial positions and the number of training and test samples for each class of interest is fixed by the IEEE GRSS Data Fusion Contest in 2013. Table 5.11 provides information on the number of training and test samples for each class of interest. It is important to note that the standard sets of training and test samples were used for the dataset to make the results entirely comparable with most of the classification methods available in the literature.



(a)

(b)

Figure 5.17: Houston dataset. (a) Three-band false-color image. (b) Ground truth-map reference.

Table 5.11: Houston dataset: Number of training and test samples.

Class number	Class name	Training samples	Test samples
1	Healthy grass	198	1053
2	Stressed grass	190	1064
3	Synthetic grass	192	505
4	Trees	188	1056
5	Soil	186	1056
6	Water	182	143
7	Residential	196	1072
8	Commercial	191	1053
9	Road	193	1059
10	Highway	191	1036
11	Railway	181	1054
12	Parking Lot 1	192	1041
13	Parking Lot 2	184	285
14	Tennis Court	181	247
15	Running Track	187	473
	Total	2832	12197

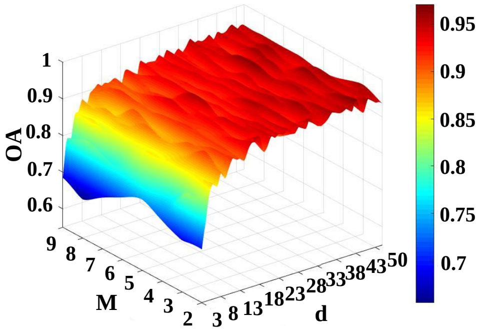


Figure 5.18: Overall accuracy (OA) versus the reduced dimensionality d and the number of mixture components M in the S^{2}MFA method with SVM classifier on the Indian Pines dataset (the number of training samples per class=50).

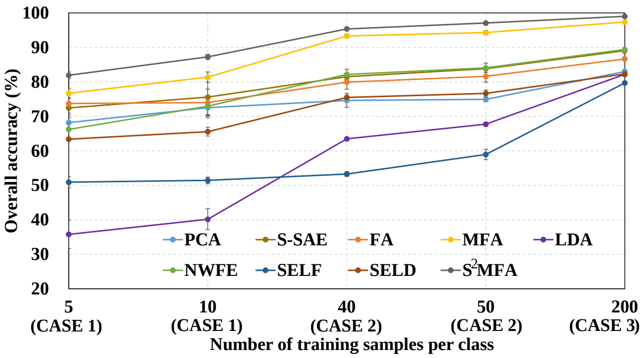


Figure 5.19: Classification results with different numbers of training samples for the Indian Pines dataset.

Experimental Setup: Indian Pines Dataset

The training set $\mathbf{X}$ consists of unlabeled samples subset $\mathbf{X}_{un}$ and labeled samples subset $\mathbf{X}_{lb}$, i.e., $\mathbf{X} = \mathbf{X}_{un} \cup \mathbf{X}_{lb}$, and $\mathbf{X}_{un} \cap \mathbf{X}_{lb} = \emptyset$. A number of unlabeled samples $u = 1500$ are randomly selected from the original image without labels to compose $\mathbf{X}_{un}$ [67]. Each class of the original image labeled training samples n_k are randomly selected to compose $\mathbf{X}_{lb}$. The impact of the training set size on the performance of S^{2}MFA is investigated in experiments using three different cases with different numbers of labeled training set $\mathbf{X}_{lb}$ [67]:

CASE 1, ill-posed condition: $n < D$ and $n_k < D$;

CASE 2, poor-posed condition: $n > D$ and $n_k < D$;

CASE 3, good-posed condition: $n > D$ and $n_k \geq D$.
Here n is the number of labeled training samples, D is the dimensionality of the input spectral bands of the original image. Five experiments were carried out on the Indian Pines dataset and separated to the three different cases, i.e., $n_k = 5$, $n_k = 10$, for CASE 1; $n_k = 40$, $n_k = 50$, for CASE 2; $n_k = 200$, for CASE 3. The training set for the semi-supervised methods (SELF, SELD, and S^{2}MFA) is $\mathbf{X}$ and for the supervised methods (LDA and NWFE) is $\mathbf{X}_{lb}$ and all the samples of the dataset is used for the unsupervised methods (PCA, S-SAE, FA, and MFA).

Experimental Setup: Houston Dataset

The experiment was performed on the Houston dataset. The training set for the semi-supervised methods (SELF, SELD, and S^{2}MFA) is $\mathbf{X}$. The training set $\mathbf{X}$ consists of unlabeled samples subset $\mathbf{X}_{un}$ and labeled samples subset $\mathbf{X}_{lb}$, i.e., $\mathbf{X} = \mathbf{X}_{un} \cup \mathbf{X}_{lb}$, and $\mathbf{X}_{un} \cap \mathbf{X}_{lb} = \emptyset$. A number of unlabeled samples $u = 1500$ are randomly selected from the original image without labels to compose $\mathbf{X}_{un}$. $\mathbf{X}_{lb}$ is the same as the training

Table 5.12: The classification results (%) of different DR methods on the Indian Pines dataset (the number of training samples per class is five), the best results are in bold typeface. The row of each class number (CN) is the mean accuracy ± standard deviation based on ten runs.

CN	PCA	S-SAE	FA	MFA	LDA	NWFE	SELF	SELD	S^2MFA
1	61.23 ± 24.88	**77.06** ± 1.80	59.55 ± 13.81	72.15 ± 9.91	33.66 ± 2.41	47.52 ± 10.89	17.57 ± 21.88	54.30 ± 12.81	56.54 ± 9.64
2	49.34 ± 12.06	66.74 ± 0.08	74.79 ± 32.92	66.83 ± 5.87	16.29 ± 7.92	53.80 ± 12.79	15.92 ± 10.11	39.69 ± 7.20	**77.20** ± 9.98
3	**95.33** ± 2.95	91.53 ± 2.28	85.16 ± 14.80	90.24 ± 8.28	48.37 ± 6.77	86.99 ± 11.86	64.84 ± 4.80	**95.33** ± 11.62	65.65 ± 12.42
4	91.78 ± 3.71	96.19 ± 1.36	96.09 ± 1.68	98.52 ± 0.92	24.53 ± 17.31	96.50 ± 7.94	61.59 ± 11.71	93.80 ± 7.27	**99.33** ± 2.32
5	99.24 ± 0.40	99.37 ± 0.14	99.12 ± 1.45	99.31 ± 5.03	51.65 ± 17.36	98.97 ± 2.61	86.36 ± 5.51	99.38 ± 0.24	**99.98** ± 0.21
6	68.22 ± 9.71	66.59 ± 4.95	58.98 ± 12.85	74.77 ± 12.23	45.07 ± 14.32	65.52 ± 12.11	57.01 ± 12.78	53.69 ± 5.09	**93.04** ± 10.03
7	50.18 ± 10.66	51.37 ± 1.11	54.04 ± 2.79	59.48 ± 6.89	28.62 ± 7.54	44.95 ± 3.99	57.65 ± 15.29	43.24 ± 8.34	**76.49** ± 3.96
8	65.19 ± 12.84	57.40 ± 4.06	**95.73** ± 9.72	69.79 ± 14.35	24.63 ± 6.53	75.37 ± 19.47	27.26 ± 11.99	53.20 ± 14.67	92.12 ± 12.56
9	87.98 ± 5.08	92.12 ± 1.64	91.70 ± 4.08	98.53 ± 13.82	58.42 ± 21.14	94.26 ± 1.51	79.44 ± 7.91	96.04 ± 4.39	**99.77** ± 2.31
AA	74.36 ± 1.98	77.60 ± 1.27	80.47 ± 3.99	81.28 ± 2.26	36.80 ± 4.45	73.76 ± 3.95	51.96 ± 2.07	69.85 ± 7.2	**84.45** ± 1.33
OA	68.19 ± 2.14	72.46 ± 1.02	73.72 ± 2.80	76.70 ± 1.43	35.78 ± 4.08	66.24 ± 2.41	50.92 ± 1.62	63.39 ± 12.81	**81.92** ± 1.01
KC	0.7436 ± 0.0247	0.6829 ± 0.0161	0.6982 ± 0.0330	0.7314 ± 0.0166	0.2618 ± 0.0425	0.6127 ± 0.0304	0.4269 ± 0.0181	0.5813 ± 0.1162	**0.7904** ± 0.0119

samples as shown in the Table 5.11. The training set for the supervised methods (LDA and NWFE) is $\mathbf{X}_{lb}$ and all the samples of the dataset is used for the unsupervised methods (PCA, S-SAE, FA, and MFA).

The proposed method is validated by the classification of the Indian Pines and Houston datasets using SVM classifier in the experiments. The SVM classifier is a supervised classification method, the LibSVM Toolbox for MATLAB version was applied with a radial basis function (RBF) kernel. The tuning parameter in SVM are obtained using five-fold cross-validation. The training of the classifier was performed using the labeled samples subset $\mathbf{X}_{lb}$. The classifier was tested by using the remaining labeled samples corresponding to the ground truth-map. The classification metrics used are overall accuracy (OA) (Appendix A.4), average accuracy (AA) (Appendix A.6), and Kappa coefficient (KC) (Appendix A.7), as well as standard deviation (STD). For comparison the following statistical and DR methods: PCA, S-SAE, FA, MFA, LDA, NWFE, SELF, SELD, and S^{2}MFA are used. Each experiment runs ten times, and the average and the standard deviation (STD) of these ten runs are reported.

Tuning Parameter Estimation and Assessment

For the S^{2}MFA algorithms, we need to estimate the number of mixture components M, and the dimensionality of latent factors d. In the experiments, $M \in \{2, 3, ..., M_c\}$, where M_c is the number of classes of experimental datasets, $d \in \{3, 4, ..., 50\}$, and five-fold cross-validation is used to obtain the optimal parameters M and d.

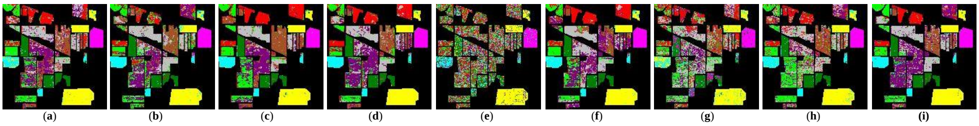


Figure 5.20: Classification maps for the Indian Pines dataset using five examples for each training class: (a) PCA, (b) S-SAE (c) FA, (d) MFA, (e) LDA, (f) NWFE, (g) SELF, (h) SELD, and (i) S^{2}MFA.

The effect of the M and d on the performance of S^{2}MFA is of interest. Since we are using classification accuracy, we investigated the effect of the M and d on OA. Fig.

5.18 shows the 3-dimensional surface of the OA of S^{2}MFA with respect to the M and d for the SVM classifier. It can be seen that the OA gradually increases in the beginning as d increases, and then keep stable with slight fluctuation as d increases. It can also be observed that the OA is insensitive to M.

Classification Results

For the Indian Pines dataset, five samples per class are randomly selected for training samples, and the rest is used for test samples. The amount of training and test samples per class are shown in Table 5.10. Fig. 5.20 shows the classification maps obtained by different methods. From the figures, we can see that S^{2}MFA gives good classification performance. Besides visual comparison, Table 5.12 presents the quantitative classification results for all the methods, and there it can be observed that S^{2}MFA achieves much higher classification accuracies than PCA, S-SAE, FA, MFA, LDA, and NWFE methods. This indicates that, compared to unsupervised and supervised FE methods, the semi-supervised S^{2}MFA method could extract more effective features for classification by simultaneously considering labeled and unlabeled samples. Moreover, S^{2}MFA is better than SELF and SELD and presents the highest OA, AA, and KC and achieves most of the top classification accuracy values for individual classes. S^{2}MFA improves the OA by 46.14% using SVM compared to other methods in the experiment. Table 5.12 also presents the STDs of the classification results for different DR methods. It can be seen that, S^{2}MFA gives relatively stable classification results compared to the other methods. Table 5.14 compares the CPU processing time (in seconds) used by different DR methods. All the methods are implemented using Matlab R2019a on a computer having Intel(R) Core(TM) i7-6700 processor (3.40GHz), 8.00 GB of memory and 64-bit Windows 10 Operating System. The running times for S^{2}MFA were 0.29, 0.43, and 0.76 s when the number of features are equal to 5, 20, and 50, respectively. It is worth noting that the running time for the semi-supervised methods (SELF, SELD, and S^{2}MFA) is affected considerably by the number of labeled and unlabeled training samples used and the supervised methods (LDA and NWFE) are affected considerably by the number of labeled training samples used and the unsupervised methods are affected by the size of the dataset.

The results of the impact of different training set sizes for the Indian Pines dataset are shown in Fig. 5.19. As can be seen from the figure that, when the number of labeled samples is very limited such as in CASE 1, the performances of all the methods are limited by the number of training samples. LDA performs much worse than the other methods because the number of extracted features in the LDA depends on the number of classes. When the number of labeled samples is increased such as in CASE 2 and 3, all the methods achieve better results. For all the three different cases, it can be seen that S^{2}MFA achieves much higher classification accuracies than the other methods. This further demonstrates that, by simultaneously considering the labeled and unlabeled samples, S^{2}MFA could preserve the local spatial information and improves the class discrimination of the data and thus could extract more effective information for classification from a complicated HSI.

Table 5.13: The classification results (%) of different DR methods on the Houston dataset, the best results are in bold typeface. The row of each class number (CN) is the mean accuracy ± standard deviation based on ten runs.

CN	PCA	S-SAE	FA	MFA	LDA	NWFE	SELF	SELD	S^2MFA
1	98.05	82.43	96.37	**98.98** ± 0.89	80.53	80.72 ± 0.45	82.43 ± 0.60	82.53 ± 0.03	82.43 ± 0.07
2	**96.01**	76.79	95.97	95.33 ± 1.63	80.92	82.99 ± 0.18	83.55 ± 0.13	83.18 ± 0.13	83.55 ± 0.53
3	**100**	92.87	**100**	99.76 ± 0.24	**100**	99.42 ± 0.58	**100**	**100**	**100**
4	98.05	89.49	97.72	**99.89** ± 0.32	92.90	89.49 ± 0.94	91.95 ± 0.40	90.44 ± 1.47	89.87 ± 0.94
5	97.41	**100**	97.11	98.42 ± 0.82	96.97	98.86 ± 0.38	98.67 ± 0.33	97.16 ± 0.27	**100**
6	**99.94**	95.80	95.3	47.24 ± 0.80	93.71	95.11 ± 0.21	95.11 ± 0.05	95.80 ± 2.47	94.41 ± 0.99
7	81.34	78.64	**87.91**	66.22 ± 1.38	84.80	79.76 ± 2.44	80.88 ± 1.12	74.81 ± 0.40	72.97 ± 0.46
8	81.45	**98.39**	84.18	84.06 ± 1.14	71.13	72.46 ± 4.73	54.70 ± 1.07	71.04 ± 0.34	82.06 ± 0.62
9	83.40	85.65	78.08	87.65 ± 1.04	72.52	75.54 ± 1.19	74.88 ± 2.27	63.08 ± 0.67	**89.97** ± 1.07
10	70.03	49.04	65.32	86.21 ± 0.24	73.17	81.27 ± 3.21	91.89 ± 2.39	57.92 ± 4.51	**95.95** ± 1.05
11	67.86	85.96	57.04	68.21 ± 0.16	85.39	93.55 ± 2.10	87.67 ± 1.74	82.83 ± 1.41	**94.51** ± 0.35
12	85.50	89.63	82.68	95.80 ± 2.28	82.52	85.11 ± 1.86	89.24 ± 0.07	66.19 ± 2.24	**96.64** ± 0.20
13	30.69	73.68	43.85	**92.52** ± 3.26	70.53	70.88 ± 1.74	75.79 ± 0.74	64.21 ± 0.25	77.90 ± 1.23
14	99.18	**100**	89.05	94.64 ± 0.83	99.19	99.60 ± 0.31	99.60 ± 0.04	98.79 ± 0.29	**100**
15	**100**	86.89	**100**	99.54 ± 0.54	95.98	98.73 ± 0.50	97.67 ± 0.45	97.89 ± 0.15	**100**
AA	85.93	85.68	84.70	87.68 ± 0.36	85.35	86.94 ± 0.47	86.94 ± 0.28	81.72 ± 0.08	**90.75** ± 0.42
OA	83.45	84.39	82.05	86.65 ± 0.42	83.59	85.35 ± 0.56	85.07 ± 0.38	79.09 ± 0.32	**89.78** ± 0.63
KC	0.8207	0.8304	0.8053	0.8552 ± 0.0046	0.8223	0.8412 ± 0.0059	0.8380 ± 0.0041	0.7730 ± 0.0035	**0.8891** ± 0.0069

Table 5.14: CPU processing times in seconds with different number of features (NF) for different DR methods applied on the Indian Pines dataset.

NF	PCA	S-SAE	FA	MFA	LDA	NWFE	SELF	SELD	S^2MFA
5	0.14	135.90	14.36	1.51	0.08	0.52	0.04	0.33	0.29
20	0.14	331.48	73.48	3.84	0.09	0.54	0.04	0.37	0.43
50	0.15	594.76	121.61	7.51	0.09	0.69	0.05	0.38	0.76

Table 5.15: CPU processing times in seconds with different number of features (NF) for different DR methods applied on the Houston dataset.

NF	PCA	S-SAE	FA	MFA	LDA	NWFE	SELF	SELD	S^2MFA
5	1.02	4362.81	18.19	50.64	0.78	4.81	118.29	47.56	2.86
20	2.86	7608.93	21.89	105.85	0.83	4.95	127.52	43.88	5.34
50	2.94	8520.14	28.36	320.58	0.88	5.29	124.69	45.60	20.14

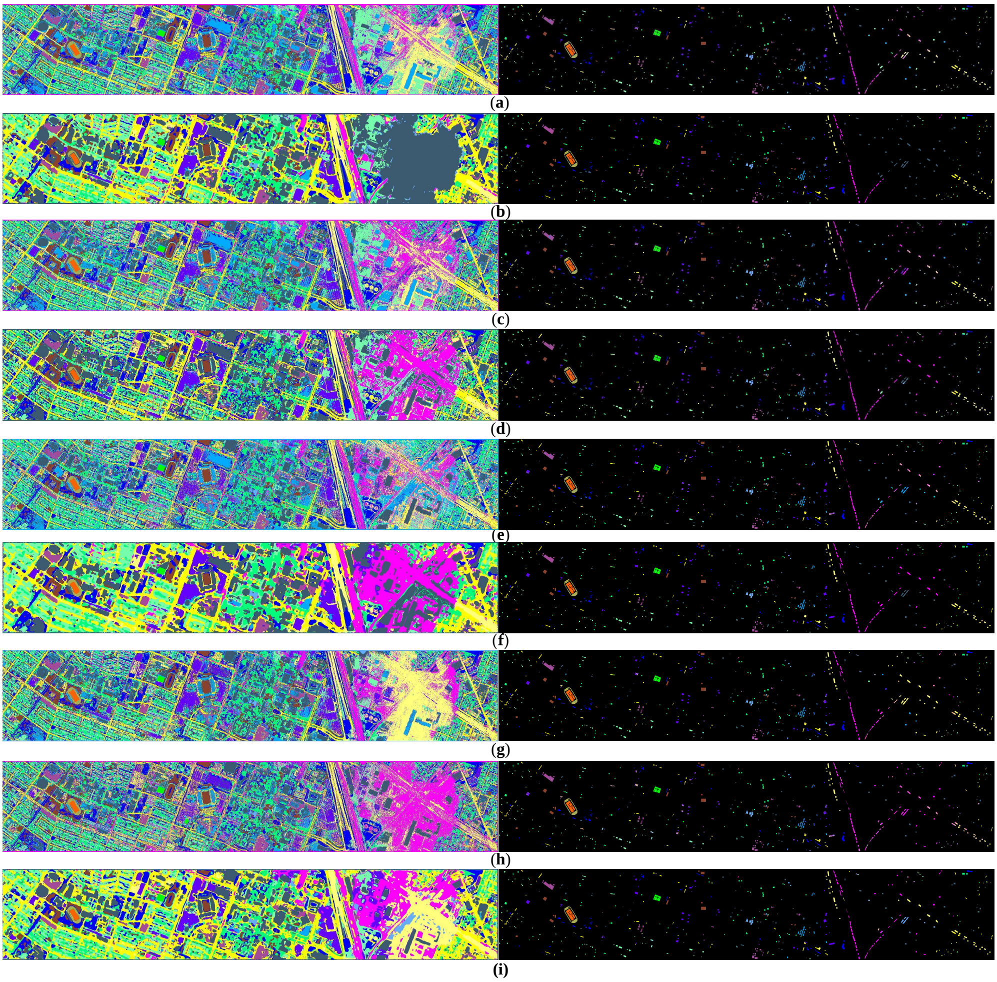


Figure 5.21: Classification maps for the Houston dataset obtained by SVM classification after using (a) PCA, (b) S-SAE, (c) FA, (d) MFA, (e) LDA, (f) NWFE, (g) SELF, (h) SELD, and (i) S^2MFA DR methods.

For the Houston dataset, Fig. 5.21 shows the classification maps obtained by different methods used in the chapter. From the figures, we can see that the proposed S^2MFA method outperforms all other methods. Table 5.13 presents the quantitative classification results of the different dimensionality reduction (DR) methods for Houston dataset. As shown in Table 5.13, the performance of S^2MFA is better than PCA, S-SAE, FA, MFA, NWFE, SELF, and SELD. S^2MFA improves the OA by 10.69% compared to other methods in the experiment. Table 5.13 also presents the standard deviations of the classification results. It can be seen that PCA, FA, and LDA present the most stable results. S^2MFA has slight fluctuation for each experiment and give relatively stable results. Table 5.15 compares the CPU processing time (in seconds) used by the different DR methods for the Houston dataset. It can be seen that the running times for S^2MFA was 2.86, 5.34, and 20.14 s when the number of features are equal to 5, 20, and 50, respectively.

5.3 Conclusions

In this chapter, six MFA low-rank based unsupervised, supervised, and semi-supervised FE methods were proposed for HSI. There are MFA, DMFA, SMFA, SDMFA, S^2MFA and S^2DMFA and were proposed for feature extraction of HSIs and were then used for classification of them.

MFA, DMFA, SMFA, SDMFA, S^2MFA and S^2DMFA are probabilistic DR methods, instead of assuming that a whole HSI obeys a Gaussian distribution, the methods use a Gaussian mixture model to extract more effective information for DR. The Gaussian mixture model is used for MFA to allow a low-dimensionality representation of the Gaussian. A two-layer MFA, DMFA, utilizes the samples from the posterior at the first layer to an MFA model at the second layer. MFA and DMFA are two unsupervised DR method. The methods are particularly suitable for DR of HSI with a non-normal distribution and unlabeled samples. SMFA and SDMFA are supervised DR methods and use labeled samples to extract features. SMFA and SDMFA can be effectively used to DR of HSI with a non-normal distribution and labeled samples.

S^2MFA and S^2DMFA are semi-supervised DR methods, which simultaneously consider labeled and unlabeled samples to extract features. S^2MFA uses a Gaussian mixture model to segment image to different parts, each part follows a Gaussian distribution and contains many labeled and unlabeled samples and uses a factor analyzer to get a factor loading matrix. This matrix uses labeled samples to improve the class discrimination and employs both labeled and unlabeled samples to preserve the local spatial features of the data and then is used for transforming the original HSI to an optimal low-dimensional subspace to achieve DR. S^2DMFA is a two-layer S^2MFA. S^2MFA and S^2DMFA are particularly suitable for DR of HSI with a complicated probability distribution and labeled and unlabeled samples.

Based on the six DR methods, we also proposed a framework for HSI classification, the overall accuracy of a classifier on validation samples is used to automatically determine the optimal number of features of DR for HSI classification. This framework can automatically extract the most effective feature for HSI classification. To validate the performance of DR, we conduct experiments in terms of SVM classification based on four real HSIs. The experimental results show that our proposed methods can give better results than statistical DR comparison methods.

Chapter 6

Conclusions

In this chapter, the conclusions and main contributions are reviewed. Future research topics are also discussed.

6.1 Main Contributions

In this thesis, two low-rank based and one sparse and low-rank based HSI denoising methods have been presented, a series of MFA low-rank based methods have been proposed for HSI denoising and feature extraction. The main contributions of this thesis are listed here.

6.1.1 Low-Rank Based Methods

Two low-rank based methods, non-local means low-rank approximation (NLMLRA) and wavelet-based block low-rank representations (WBBLRR), are presented. NLMLRA is an extension of the idea of [69], which focuses on grayscale images. The distinction of our proposed method is that we use full-band pixel-based patches to calculate the weighting function to take full advantage of the high spatial and spectral correlation and constructs a low-rank approximation function for denoising operator to improve the NLM denoised performance rather than only using spatial information alone as in [69]. To improve the practicability and reduce computational cost, Chebyshev polynomials are used in the NLMLRA. WBBLRR uses 3-D wavelet transformation to decompose HSI into different blocks, where each block utilizes a low-rank representations model to obtain the denoised block, and then uses inverse 3-D wavelet transformation for all the denoised blocks to obtain the denoised HSI.

NLMLRA and WBBLRR denoising methods were tested using simulated and real datasets. The simulated datasets are Washington DC Mall (WDCM) and University of Pavia (UP) datasets. The Gaussian noise is added to the simulated datasets. To evaluate the denoised results of the simulated datasets, the quantitative metrics are mean structural similarity (MSSIM) and peak signal-to-noise ratio (PSNR) are used. The proposed methods are shown to give better results than the comparison methods for HSI denoising.

6.1.2 Sparse and Low-Rank Based Method

A spectral-spatial transform-based sparse and low-rank representations (SSWSLRR) method using 3-D orthogonal transform (3-DOT) is proposed for HSI denoising. 3-D discrete orthogonal wavelet transform (3-D DWT) and 3-D discrete orthogonal cosine transform (3-D DCT) are two examples of 3-DOT appropriate for denoising since they concentrate the signal in few transform coefficients. SSSLRR-DWT uses 3-D DWT to be the 3-DOT to decompose noisy HSI to sparse wavelet coefficients. A weighted nuclear norm low-rank regularizer was proposed for the global image, which gives more flexibility than the nuclear norm. The weighted nuclear norm low-rank regularizer uses different weights to shrink the singular values of the wavelet coefficient matrix.

HSI has sparse and low-rank characteristics and can be preserved using the ℓ_1 penalty and the weighted nuclear norm low-rank penalty, respectively. Thus, a novel sparse and low-rank penalized model was proposed using both the ℓ_1 penalty and the weighted nuclear norm low-rank penalty. SSSLRR-DWT uses the sparse and low-rank penalized model can adaptively shrink wavelet coefficients and penalize the singular values of the wavelet coefficient matrix. Thus, SSSLRR-DWT uses the sparse and low-rank penalized model to remove noise effectively and recover the denoised image. The algorithm for SSSLRR-DWT was developed using ADMM. To analyze the effectiveness of different penalties for SSSLRR-DWT both in signal and wavelet domain, four different methods were analyzed. The analysis shows that SSSLRR-DWT uses both ℓ_1 penalty and weighted nuclear norm low-rank penalty in the wavelet domain yields better denoising results than the other methods. Moreover, a new SSSLRR-DCT denoising method was also proposed, which uses the 3-D discrete orthogonal cosine transform (3-D DCT) for SSSLRR. The analysis shows that both SSSLRR-DWT and SSSLRR-DCT have good denoising results.

In the experiments, two simulated and two real HSI datasets were used to demonstrate the effectiveness of the proposed method for HSI denoising. To evaluate the denoised results of the simulated datasets, the quantitative metrics are PSNR, MSSIM, the mean feature similarity (MFSIM), and the spectral angle mapper (SAM) are used. The experimental results show that the obtained PSNR, MSSIM, MFSIM, and SAM of the proposed SSSLRR method outperforms comparison methods. In the real datasets experiments, SSSLRR achieve satisfactory denoising results.

6.1.3 MFA Low-Rank Based Methods for HSI Denoising

Four MFA low-rank based HSI denoising methods are proposed. There are MFA, WMFA-2D, WMFA-3D, and LSSC-AMFA denoising mehtods. MFA utilizes Gaussian mixture model to segment the original HSI to different parts, where each part follows a Gaussian distribution and then utilizes a factor analyzer to get a low-rank factor loading matrix, and finally uses the inverse transformation of the low-rank factor loading matrix to get the hyperspectral data without noise. WMFA-2D is a MFA in the 2-D wavelet domain. In WMFA-2D, the whole HSI is firstly decomposed into sub-images (approximation coefficients (AC_1) sub-image and detail coefficients sub-images) using wavelet decomposition. Secondly, using all the sub-images to compose a new hyperspectral dataset with the same size as the original HSI. Thirdly, this new dataset is put in the MFA-2D algorithm to get a denoised dataset. Fourthly, the denoised dataset is split to different sub-images (new approximation coefficients (AC_2) sub-image and new detail coefficients sub-images) using the inverse way as the process of composition in the second step. Finally, using wavelet reconstruction transformation based on the approximation coefficients (AC_1) sub-image and new detail coefficients sub-images to reconstruct a new denoised HSI with the same size as the original HSI. WMFA-3D is a MFA in the 3-D wavelet domain and is similar as WMFA-2D. MFA, WMFA-2D, and WMFA-3D are used for removing Gaussian noise. HSIs are usually corrupted by different kinds of noise such as missing vertical lines (MVL), missing horizontal lines (MHL), missing pixels (MP), salt and pepper noise (SP), and Gaussian noise. The proposed LSSC-AMFA method, hierarchically, removes the mixed noises. Firstly, we develop a novel local spectral-spatial correlation (LSSC) method to remove the MVL and MHL noises. Then we propose an automatic mixtures of factor analyzers (AMFA) method to further remove the MP, SP, and Gaussian noises. The performance of the proposed methods are validated in experiments using both simulated and real HSI datasets. These experimental results show that the proposed methods outperform the state-of-the-art denoising methods in terms of PSNR for the simulated datasets. For the real dataset, experimental results show visual improvements.

6.1.4 MFA Low-Rank Based Methods for HSI Feature Extraction

Six MFA low-rank based unsupervised, supervised, and semi-supervised feature extraction (FE) methods are proposed. There are MFA, DMFA, SMFA, SDMFA, S^2MFA and S^2DMFA methods. MFA and DMFA are unsupervised FE methods. MFA is the same as it used for HSI denoising, the Gaussian mixture model is used for MFA to allow a low-dimensionality representation of the Gaussian. DMFA is a two-layer MFA and utlizes the sampples from the posterior at the first layer to an MFA in the second layer. The methods are particularly suitable for FE of HSI with a non-normal distribution and unlabeled samples. SMFA and SDMFA are supervised DR methods and use labeled samples to extract features. SMFA and SDMFA can be effectively used to FE of HSI with a non-normal distribution and labeled samples. S^2MFA and S^2DMFA are semi-supervised dimensionality reduction (DR) methods, which simultaneously consider labeled and unlabeled samples to extract features. S^2MFA uses a Gaussian mixture model to segment image to different parts, each part follows a Gaussian distribution and contains many labeled and unlabeled samples and uses a factor analyzer to get a factor loading matrix. This matrix uses labeled samples to improve the class discrimination and employs both labeled and unlabeled samples to preserve the local spatial features of the data and then is used for transforming the original HSI to an optimal low-dimensional subspace to achieve DR. S^2DMFA is a two-layer S^2MFA. S^2MFA and S^2DMFA are particularly suitable for DR of HSI with a complicated probability distribution and labeled and unlabeled samples. Based on the six dimensionality reduction methods, we also proposed a framework for HSI classification, the overall accuracy of a classifier on validation samples is used to automatically determine the optimal number of features of DR for HSI classification. This framework can automatically extract the most effective feature for HSI classification. To validate the performance of DR, we conduct experiments in terms of SVM classification based on real HSIs. The experimental results show that the proposed methods can give better results than comparison methods.

6.2 Further work

The work presented here can be extended or continued in a number of ways. Some possible future research topics are listed below.

- The proposed denoising methods in this thesis are used for HSI. These methods can be used for other different datasets, such as multispectral images, magnetic resonance images (MRI) and functional MRI.
- The proposed methods in this thesis can be used for other HSI applications, such as HSI classification, super-resolution, and unmixing.
- Tensor based low-rank model can be further developed for our SSWSLRR method.
- Combining CNN with statistic model for HSI denoising is our interest as a future research project.

Appendix A

Evaluation Metrics

Here are detailed evaluation metrics that are used to evaluate denoising and classification results.

A.1 Denoising Evaluation Metrics

To evaluate the denoised results of the simulated datasets, the quantitative metrics are the peak signal-to-noise ratio (PSNR), the mean structural similarity (MSSIM), the mean feature similarity (MFSIM), and the spectral angle mapper (SAM) are used.

A.1.1 PSNR

The definition of PSNR is

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\max^2(\mathbf{x})}{\frac{1}{P}\parallel \mathbf{x}-\widehat{\mathbf{x}}\parallel_2^2}\right), \tag{A.1}$$

where $\mathbf{x}$ is a vectorized the clean image of length P, $\widehat{\mathbf{x}}$ is the vectotized denoised image.

A.1.2 MSSIM and MFSIM

MSSIM is the bandwise mean of SSIM, the SSIM of ith band is defined as [133],

$$\mathrm{SSIM}_{i,j} = \frac{(2\mu_{x_{i,j}}\mu_{\widehat{x}_{i,j}}+c_1)(2\sigma_{x_{i,j}\widehat{x}_{i,j}}+c_2)}{(\mu_{x_{i,j}}^2+\mu_{\widehat{x}_{i,j}}^2+c_1)(\sigma_{x_{i,j}}^2+\sigma_{\widehat{x}_{i,j}}^2+c_2)}, \tag{A.2}$$

where $\mu_{x_{i,j}}$, $\mu_{\widehat{x}_{i,j}}$, $\sigma_{x_{i,j}}$, and $\sigma_{\widehat{x}_{i,j}}$ are mean and standard deviation for the reference and estimated images in a local 11×11 window centered at pixel i, respectively. $\sigma_{x_{i,j}\widehat{x}_{i,j}}$ is the cross-covariance between two images. $c_1=(K_1M)^2$ and $c_2=(K_2M)^2$, where $K_1=0.01$, $K_2=0.03$, and M is the dynamical range of the image that is set to 1 in this paper. PSNR and MSSIM assess the similarity of the target image and the reference image based on the mean square error and the structural consistency, respectively. MFSIM focuses on the perceptual consistency based on the phase congruency and gradient magnitude between the target image and the reference image [134]. The larger the PSNR, MSSIM, and FSIM are, the closer the target HSI is to the reference one.

A.1.3 SAM

SAM calculates the mean angle between spectrum vectors of the target image and the reference image across all spatial positions. The definition of SAM in degrees is

$$\mathrm{SAM} = \frac{1}{nB} \sum_{p=1}^{nB} \arccos \left(\frac{\mathbf{x}_{(p)}^{T} \widehat{\mathbf{x}}_{(p)}}{\|\mathbf{x}_{(p)}\| \|\widehat{\mathbf{x}}_{(p)}\|} \right) \frac{180}{\pi}, \tag{A.3}$$

where $\mathbf{x}_{(p)}$ and $\widehat{\mathbf{x}}_{(p)}$ are the reference image and estimated image at pixel p, respectively. Different from the former three metrics, the smaller the SAM is, the better does the target HSI estimate the reference one.

A.2 Classification Evaluation Metrics

To evaluate the classification results, the classification metrics used are overall accuracy (OA), class accuracy (CA), average accuracy (AA), and Kappa coefficient (KC).

A.2.1 OA

OA is the proportion of the correctly classified samples over all the test samples. We introduce confusion matrix ($\mathbf{C} = [c_{i,j}]$) to clearly explain OA. Table A.1 shows the confusion matrix for a 4-class classification problem. The OA is defined as

$$\mathrm{OA} = \frac{\sum_i c_{ii}}{\sum_{i,j} c_{i,j}}. \tag{A.4}$$

Table A.1: Confusion matrix for a 4-class classification problem

Predicted Class	Actual Class			
	Class 1	Class 2	Class 3	Class 4
Class 1	c_{11}	c_{12}	c_{13}	c_{14}
Class 2	c_{21}	c_{22}	c_{23}	c_{24}
Class 3	c_{31}	c_{32}	c_{33}	c_{34}
Class 4	c_{41}	c_{42}	c_{43}	c_{44}

A.2.2 CA

CA is the ratio of the correctly classified samples for each class and is defined as

$$\mathrm{CA}_j = \frac{c_{jj}}{\sum_i c_{i,j}}. \tag{A.5}$$

A.2.3 AA

AA is the average of all the CAs and is given as

$$\mathrm{AA} = \frac{1}{n} \sum_{i} \mathrm{CA}_{i}, \tag{A.6}$$

where n is the number of classes.

A.2.4 KC

KC is a classification statistic metric and is given by

$$\mathrm{KC} = \frac{p_0 - p_e}{1 - p_e}, \tag{A.7}$$

where $p_0 = \mathrm{OA}$, $p_e = \frac{\sum_i \|\mathbf{c}_{(i)}\|_1 \|\mathbf{c}_i\|_1}{\left(\sum_i c_{i,j}\right)^2}$.

Appendix B

Hyperspectral Data

This appendix describes the hyperspectral data sets used in this thesis.

B.1 Washington DC Mall

The Washington DC Mall dataset was collected by the Hyperspectral Digital Imagery Collection Experiment (HYDICE) sensor and has 1208 × 307 pixels with a spatial resolution of 2 m and 191 spectral bands. Figure B.1 shows the false-color composite of the Washington DC Mall image.

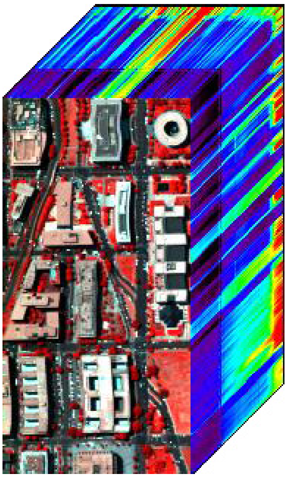


Figure B.1: Washington DC Mall dataset.

B.2 University of Pavia

The University of Pavia dataset was captured by the Reflective Optics System Imaging Spectrometer sensor over the city of Pavia, Italy. This image has 610×340 pixels with a spatial resolution of 1.3 m and 115 spectral bands coverage ranging from 0.43 μm to 0.86 μm. In the experiments, this data contains nine classes of interest and has 103 spectral bands after removing 12 noisy bands. Figure B.2 shows the false-color composite of the University of Pavia image and the corresponding ground reference map.

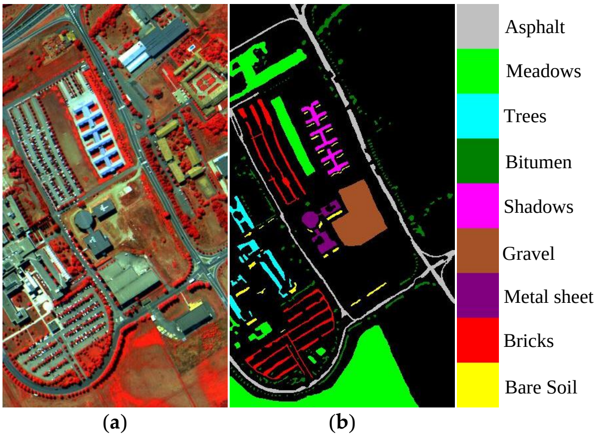


Figure B.2: University of Pavia dataset. (**a**) Three-band false-color image. (**b**) Ground truth-map reference.

B.3 Indian Pines

The Indian Pines dataset was collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor at Indian Pines. The image contains 145 × 145 pixels with a spatial resolution of 20 m and 220 spectral bands from 400 nm to 2500 nm. In the experiments, noisy bands and atmospheric vapor absorption bands are excluded leaving 200 spectral bands. Figure B.3 shows the false-color composite of the Indian Pines image and the corresponding ground reference map, respectively.

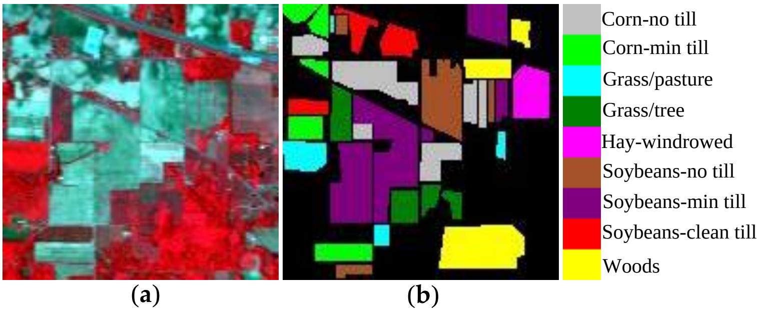


Figure B.3: Indian Pines dataset. (**a**) Three-band false-color image. (**b**) Ground truth-map containing nine land-cover classes.

B.4 Urban

The Urban is a 307 × 307 pixel HYDICE (Hyperspectral Digital Imagery Collection Experiment) hyperspectral image. The area is an urban landscape, showing trees, grass fields, houses, and roads. The dataset is composed of 210 spectral bands, has a spectral resolution of 10nm and covers the $400 - 2400$nm spectral range. Figure B.4 shows the false-color composite of the Urban image.

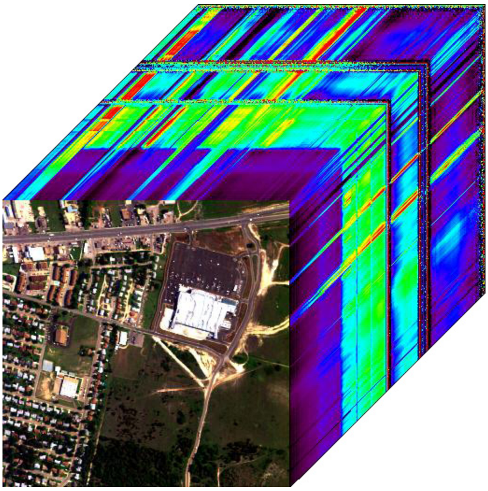


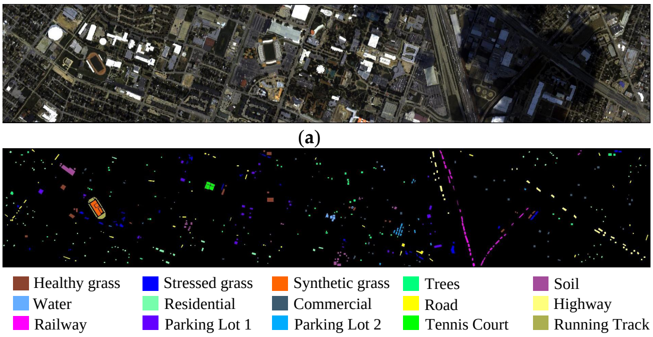
Figure B.4: Urban dataset.

B.5 Houston

The Houston dataset was provided by the IEEE Geoscience and Remote Sensing Society (GRSS) for the Data Fusion Contest in 2013. This image is of the University of Houston campus and the neighboring urban area. The dataset has 349 × 1905 pixels with a spatial resolution of 2.5 m and 114 spectral bands coverage ranging from 380 nm to 1050 nm. This HSI contains fifteen classes of interest. Figure B.5 shows the false-color composite of the Houston image and the corresponding ground reference map, respectively.

B.6 Salinas

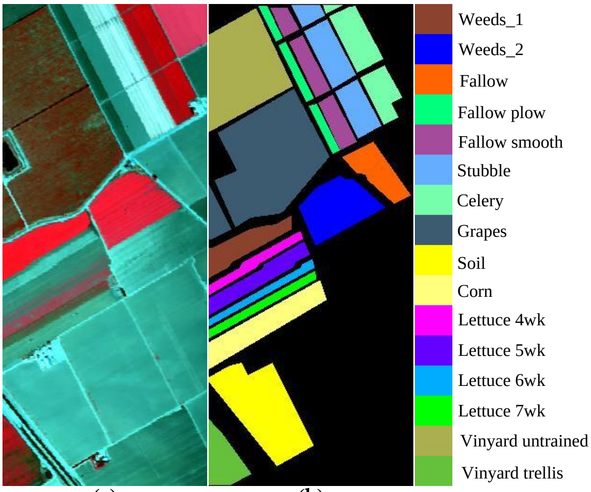
The Salinas dataset was acquired by the AVIRIS sensor over Salinas Valley, California. This image has 512 × 217 pixels with a spatial resolution of 3.7 m and 204 spectral bands after removing 20 water absorption bands. This image contains sixteen classes of interest. Figure B.6 shows the false-color composite of the Salinas image and the corresponding ground reference map.



(a)

(b)

FIGURE B.5: Houston dataset. (**a**) Three-band false-color image. (**b**) Ground truth-map reference.



(a) (b)

FIGURE B.6: Salinas dataset. (**a**) Three-band false-color image. (**b**) Ground truth-map reference.

Bibliography

[1] D. A. Landgrebe and E. Malaret, “Noise in remote-sensing systems: The effect on classification error,” *IEEE Trans. Image Process.*, , no. 2, pp. 294–300, 1986. [Cited on page 1]

[2] N. Acito, M. Diani, and G. Corsini, “Signal-dependent noise modeling and model parameter estimation in hyperspectral images,” *IEEE Trans. Image Process.*, vol. 49, no. 8, pp. 2957–2971, 2011. [Cited on page 1]

[3] Y. Qian and M. Ye, “Hyperspectral imagery restoration using nonlocal spectral-spatial structured sparse representation with noise estimation,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 6, no. 2, pp. 499–515, 2012. [Cited on page 1]

[4] Q. Yuan, Q. Zhang, J. Li, H. Shen, and L. Zhang, “Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1205–1218, 2018. [Cited on page 2]

[5] X. L. Zhao, W. H. Xu, T. X. Jiang, Y. Wang, and M. K. Ng, “Deep plug-and-play prior for low-rank tensor completion,” *Neurocomputing*, vol. 400, pp. 137–149, 2020. [Cited on page 2]

[6] W. He, N. Yokoya, L. Yuan, and Q. Zhao, “Remote sensing image reconstruction using tensor ring completion and total variation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8998–9009, 2019. [Cited on page 2]

[7] B. Zhao, J. R. Sveinsson, M. O. Ulfarsson, and J. Chanussot, “Hyperspectral images denoising based on mixtures of factor analyzers,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2020, pp. 1516–1519. [Cited on page 2]

[8] L. Zhuang and M. K. Ng, “Hyperspectral mixed noise removal by l_1-norm-based subspace representation,” *IEEE J. Sel.Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1143–1157, 2020. [Cited on page 2]

[9] B. Zhao, J. R. Sveinsson, M. O. Ulfarsson, and J. Chanussot, “Local spatial-spectral correlation based mixtures of factor analyzers for hyperspectral denoising,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2020, pp. 1488–1491. [Cited on page 2]

[10] M. Wang, Q. Wang, J. Chanussot, and D. Hong, “l_0-l_1 hybrid total variation regularization and its applications on hyperspectral image mixed noise removal and compressed sensing,” *IEEE Trans. Geosci. Remote Sens.*, 2021. [Cited on page 2]

[11] A. Maffei, J. M. Haut, M. E. Paoletti, J. Plaza, L. Bruzzone, and A. Plaza, “A single model cnn for hyperspectral image denoising,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2516–2529, 2020. [Cited on page 2]

[12] A. Buades, B. Coll, and J. M. Morel, “A non-local algorithm for image denoising,” in *Proc. of the IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2005, vol. 2, pp. 60–65. [Cited on page 2]

[13] D. L. Donoho and I. M. Johnstone, “Adapting to unknown smoothness via wavelet shrinkage,” *J. Am. Stat. Assoc.*, vol. 90, no. 432, pp. 1200–1224, 1995. [Cited on page 2]

[14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007. [Cited on pages 2 and 32]

[15] S. Gu, L. Zhang, W. Zuo, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” in *Proc. of the IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2862–2869. [Cited on pages 2 and 26]

[16] Y. Qian and M. Ye, “Hyperspectral imagery restoration using nonlocal spectral-spatial structured sparse representation with noise estimation,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 499–515, 2013. [Cited on page 2]

[17] L. Gao, B. Zhao, X. Jia, W. Liao, and B. Zhang, “Optimized kernel minimum noise fraction transformation for hyperspectral image classification,” *Remote Sens*, vol. 9, no. 6, pp. 548, 2017. [Cited on pages 2, 3, 76, and 84]

[18] Q. Yuan, L. Zhang, and H. Shen, “Hyperspectral image denoising employing a spectral–spatial adaptive total variation model,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3660–3677, 2012. [Cited on page 2]

[19] S. Chen, X. Hu, and S. Peng, “Hyperspectral imagery denoising using a spatial-spectral domain mixing prior,” *J. Comput. Sci. Technol.*, vol. 27, no. 4, pp. 851–861, 2012. [Cited on page 2]

[20] X. Huang, B. Du, D. Tao, and L. Zhang, “Spatial-spectral weighted nuclear norm minimization for hyperspectral image denoising,” *Neurocomputing*, vol. 399, pp. 271–284, 2020. [Cited on page 2]

[21] S. G. Chang, Bin Yu, and M. Vetterli, “Spatially adaptive wavelet thresholding with context modeling for image denoising,” *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1522–1531, 2000. [Cited on page 2]

[22] G. Chen and S. Qian, “Denoising of hyperspectral imagery using principal component analysis and wavelet shrinkage,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 973–980, 2010. [Cited on page 2]

[23] H. Othman and S. Qian, “Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 397–408, 2006. [Cited on page 2]

[24] B. Rasti, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, “Wavelet based hyperspectral image restoration using spatial and spectral penalties,” *Proc. SPIE*, vol. 8892, pp. 8892–1–8892–8, 2013. [Cited on page 3]

[25] B. Du, Z. Huang, N. Wang, Y. Zhang, and X. Jia, “Joint weighted nuclear norm and total variation regularization for hyperspectral image denoising,” *Int. J. Remote Sens.*, vol. 39, no. 2, pp. 334–355, 2018. [Cited on pages 3 and 23]

[26] Z. Huang, S. Li, L. Fang, H. Li, and J. A. Benediktsson, “Hyperspectral image denoising with group sparse and low-rank tensor decomposition,” *IEEE Access*, vol. 6, pp. 1380–1390, 2018. [Cited on pages 3 and 23]

[27] T. Xie, S. Li, and B. Sun, “Hyperspectral images denoising via nonconvex regularized low-rank and sparse matrix decomposition,” *IEEE Trans. Image Process.*, vol. 29, pp. 44–56, 2020. [Cited on pages 3 and 23]

[28] Y. Chang, L. Yan, H. Fang, and H. Liu, “Simultaneous destriping and denoising for remote sensing images with unidirectional total variation and sparse representation,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 6, pp. 1051–1055, 2014. [Cited on pages 3 and 23]

[29] W. He, Q. Yao, C. Li, N. Yokoya, Q. Zhao, H. Zhang, and L. Zhang, “Non-local meets global: An integrated paradigm for hyperspectral image restoration,” *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020. [Cited on pages 3 and 23]

[30] Q. Zhang, Q. Yuan, J. Li, X. Liu, H. Shen, and L. Zhang, “Hybrid noise removal in hyperspectral imagery with a spatial–spectral gradient network,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7317–7329, 2019. [Cited on pages 3 and 23]

[31] M. Wang, J. Yu, J. Xue, and W. Sun, “ Denoising of hyperspectral images using group low-rank representation,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 9, no. 9, pp. 4420–4427, 2016. [Cited on pages 3 and 23]

[32] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, “Hyperspectral image restoration using low-rank matrix recovery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4729–4743, 2013. [Cited on pages 3, 23, 53, and 61]

[33] W. He, H. Zhang, L. Zhang, and H. Shen, “Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation,” *IEEE J. Sel. Top. Appl. Earth. Obs. Remote Sens.*, vol. 8, no. 6, pp. 3050–3061, 2015. [Cited on pages 3, 23, 32, and 61]

[34] Y. Chen, Y. Guo, Y. Wang, D. Wang, C. Peng, and G. He, “Denoising of hyperspectral images using nonconvex low rank matrix approximation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5366–5380, 2017. [Cited on pages 3 and 23]

[35] T. Lu, S. Li, L. Fang, Y. Ma, and J. A. Benediktsson, "Spectral–spatial adaptive sparse representation for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 373–385, 2015. [Cited on pages 3 and 23]

[36] J. Li, Q. Yuan, H. Shen, and L. Zhang, "Noise removal from hyperspectral image with joint spectral–spatial distributed sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5425–5439, 2016. [Cited on pages 3 and 23]

[37] N. Liu, W. Li, R. Tao, and J. E. Fowler, "Wavelet-domain low-rank/group-sparse destriping for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10310–10321, 2019. [Cited on pages 3 and 23]

[38] L. Zhuang and J. M. Bioucas-Dias, "Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations," *IEEE J. Sel. Top. Appl. Earth. Obs. Remote Sens.*, vol. 11, no. 3, pp. 730–742, 2018. [Cited on pages 3, 23, and 32]

[39] B. Rasti, M. O. Ulfarsson, and P. Ghamisi, "Automatic hyperspectral image restoration using sparse and low-rank modeling," *IEEE Geosci. Remote. Sens. Lett.*, vol. 14, no. 12, pp. 2335–2339, 2017. [Cited on pages 3, 23, 32, and 61]

[40] Y. Chen, W. He, N. Yokoya, T. Huang, and X. Zhao, "Nonlocal tensor-ring decomposition for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1348–1362, 2019. [Cited on pages 3 and 23]

[41] J. Lin, T. Huang, X. Zhao, T. Jiang, and L. Zhuang, "A tensor subspace representation-based method for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, 2020. [Cited on pages 3 and 23]

[42] X. Gong, W. Chen, and J. Chen, "A low-rank tensor dictionary learning method for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 68, pp. 1168–1180, 2020. [Cited on pages 3 and 23]

[43] Y. Chen, W. He, N. Yokoya, and T. Z. Huang, "Hyperspectral image restoration using weighted group sparsity-regularized low-rank tensor decomposition," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3556–3570, 2019. [Cited on pages 3 and 23]

[44] J. Peng, Q. Xie, Q. Zhao, Y. Wang, L. Yee, and D. Meng, "Enhanced 3DTV regularization and its applications on HSI denoising and compressed sensing," *IEEE Trans. Image Process.*, vol. 29, pp. 7889–7903, 2020. [Cited on pages 3 and 23]

[45] C. Cao, J. Yu, C. Zhou, K. Hu, F. Xiao, and X. Gao, "Hyperspectral image denoising via subspace-based nonlocal low-rank and sparse factorization," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 12, no. 3, pp. 973–988, 2019. [Cited on pages 3 and 23]

[46] J. Al. Richards and J. A. Richards, *Remote sensing digital image analysis*, vol. 3, Springer, 1999. [Cited on pages 3 and 84]

[47] B. Rasti, P. Ghamisi, and M. O. Ulfarsson, "Hyperspectral feature extraction using sparse and smooth low-rank analysis," *Remote Sens.*, vol. 11, no. 2, pp. 121, 2019. [Cited on pages 3, 4, and 76]

[48] B. Zhao, L. Gao, W. Liao, and B. Zhang, "A new kernel method for hyperspectral image feature extraction," *Geo. Spat. Inf. Sci.*, vol. 20, no. 4, pp. 309–318, 2017. [Cited on pages 4 and 76]

[49] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. R. Stat. Soc.*, vol. 61, no. 3, pp. 611–622, 1999. [Cited on pages 4 and 76]

[50] D. N. Lawley, "A modified method of estimation in factor analysis and some large sample results," in *Uppsala Symp. Psychol Fac. Analys.* Taylor & Francis, 1953, vol. 17, pp. 35–42. [Cited on pages 4 and 76]

[51] B. Zhao, M. O. Ulfarsson, J. R. Sveinsson, and J. Chanussot, "Mixtures of factor analyzers and deep mixtures of factor analyzers dimensionality reduction algorithms for hyperspectral images classification," in *IEEE Trans. Geosci. Remote Sens. Symp.* IEEE, 2019, pp. 891–894. [Cited on pages 4 and 95]

[52] B. Zhao, M. O. Ulfarsson, J. R. Sveinsson, and J. Chanussot, "Unsupervised and supervised feature extraction methods for hyperspectral images based on mixtures of factor analyzers," *Remote Sens.*, vol. 12, no. 7, pp. 1179, 2020. [Cited on pages 4 and 95]

[53] H. Li, H. Li, and L. Zhang, "Quaternion-based multiscale analysis for feature extraction of hyperspectral images," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1418–1430, 2019. [Cited on page 4]

[54] B. Rasti, M. O. Ulfarsson, and J. R. Sveinsson, "Hyperspectral feature extraction using total variation component analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 6976–6985, 2016. [Cited on pages 4 and 76]

[55] J. Zabalza, J. Ren, J. Zheng, H. Zhao, C. Qing, Z. Yang, P. Du, and S. Marshall, "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging," *Neurocomputing*, vol. 185, pp. 1–10, 2016. [Cited on page 4]

[56] M. Li and B. Yuan, "2d-lda: A statistical linear discriminant analysis for image matrix," *Pattern Recognit. Lett.*, vol. 26, no. 5, pp. 527–532, 2005. [Cited on pages 4 and 76]

[57] B. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 5, pp. 1096–1105, 2004. [Cited on pages 4 and 76]

[58] Q. Du, "Modified fisher's linear discriminant analysis for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 503–507, 2007. [Cited on page 4]

[59] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, 2009. [Cited on page 4]

[60] B. Kuo, C. Li, and J. Yang, "Kernel nonparametric weighted feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1139–1155, 2009. [Cited on page 4]

[61] P. Zhang, H. He, and L. Gao, "A nonlinear and explicit framework of supervised manifold-feature extraction for hyperspectral image classification," *Neurocomputing*, vol. 337, pp. 315–324, 2019. [Cited on pages 4 and 76]

[62] S. A. Ahmadi, N. Mehrshad, and S. M. Razavi, "Supervised feature extraction method based on low-rank representation with preserving local pairwise constraints for hyperspectral images," *Signal Image and Video Process.*, vol. 13, no. 3, pp. 583–590, 2019. [Cited on pages 4 and 76]

[63] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4085–4098, 2010. [Cited on page 4]

[64] J. Xia, J. Chanussot, P. Du, and X. He, "(semi-) supervised probabilistic principal component analysis for hyperspectral remote sensing image classification," *IEEE J. Sel. Top. Appl. Earth. Obs. Remote Sens.*, vol. 7, no. 6, pp. 2224–2236, 2013. [Cited on page 4]

[65] H. Su, P. Du, and Q. Du, "Semi-supervised dimensionality reduction using orthogonal projection divergence-based clustering for hyperspectral imagery," *Opt. Eng.*, vol. 51, no. 11, pp. 111715, 2012. [Cited on page 4]

[66] M. Sugiyama, T. Idé, S. Nakajima, and J. Sese, "Semi-supervised local fisher discriminant analysis for dimensionality reduction," *Mach. Learn.*, vol. 78, no. 1, pp. 35–61, 2010. [Cited on page 4]

[67] W. Liao, A. Pizurica, P. Scheunders, W. Philips, and Y. Pi, "Semisupervised local discriminant analysis for feature extraction in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 184–198, 2012. [Cited on pages 4 and 101]

[68] G. Camps-Valls and D. Marsheva, T. V. B.and Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3044–3054, 2007. [Cited on page 4]

[69] V. May, Y. Keller, N. Sharon, and Y. Shkolnisky, "An algorithm for improving non-local means operators via low-rank approximation," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1340–1353, 2016. [Cited on pages 5, 6, 9, 10, and 109]

[70] S. K. Mohideen, S. A. Perumal, and M. M. Sathik, "Image de-noising using discrete wavelet transform," *Int. J. Netw. Secur.*, vol. 8, no. 1, pp. 213–216, 2008. [Cited on pages 6, 24, and 64]

[71] C. He, J. Xing, J. Li, Q. Yang, and R. Wang, "A new wavelet threshold determination method considering interscale correlation in signal denoising," *Math. Probl. Eng.*, vol. 2015, 2015. [Cited on pages 6 and 24]

[72] F. G. Meyer and X. Shen, "Perturbation of the eigenvectors of the graph laplacian: Application to image denoising," *Appl. Comput. Harmon. Anal.*, vol. 36, no. 2, pp. 326–334, 2014. [Cited on page 10]

[73] E. J. Candes, C. A. Sing-Long, and J. D. Trzasko, "Unbiased risk estimates for singular value thresholding and spectral estimators," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4643–4657, 2013. [Cited on page 15]

[74] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010. [Cited on page 24]

[75] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. on Inf. Theory*, vol. 41, no. 3, pp. 613–627, 1995. [Cited on page 25]

[76] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 119–133, 2012. [Cited on page 32]

[77] N. Renard, S. Bourennane, and J. Blanc-Talon, "Denoising and dimensionality reduction using multilinear tools for hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 138–142, 2008. [Cited on page 32]

[78] Y. Peng, D. Meng, Z. Xu, C. Gao, Y. Yang, and B. Zhang, "Decomposable nonlocal tensor dictionary learning for multispectral image denoising," in *Proce. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2949–2956. [Cited on page 32]

[79] Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery," *IEEE Trans. pattern anal. Mach. intell.*, vol. 40, no. 8, pp. 1888–1902, 2017. [Cited on page 32]

[80] J. M. Bioucas-Dias and J. Nascimento, "Hyperspectral subspace identification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, 2008. [Cited on pages 32, 65, 67, and 70]

[81] W. He, Q. Yao, C. Li, N. Yokoya, Q. Zhao, H. Zhang, and L. Zhang, "Non-local meets global: An integrated paradigm for hyperspectral denoising," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. [Cited on page 53]

[82] M. Wang, Q. Wang, and J. Chanussot, "Tensor Low-Rank Constraint and l_0 Total Variation for Hyperspectral Image Mixed Noise Removal," *IEEE J. Sel. Top. Signal Process.*, vol. 15, no. 3, pp. 718–733, 2021. [Cited on page 53]

[83] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers," Tech. Rep., Technical Report CRG-TR-96-1, University of Toronto, 1996. [Cited on pages 62 and 96]

[84] H. K. Aggarwal and A. Majumdar, “Hyperspectral image denoising using spatio-spectral total variation,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 442–446, 2016. [Cited on page 66]

[85] T. K. Moon, “The expectation-maximization algorithm,” *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, 1996. [Cited on pages 67, 77, and 82]

[86] J. C. Harsanyi and C. I. Chang, “Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach,” *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 779–785, 1994. [Cited on page 76]

[87] X. Jia and J. A. Richards, “Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 1, pp. 538–542, 1999. [Cited on page 76]

[88] G. Camps-Valls and L. Bruzzone, “Kernel-based methods for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, 2005. [Cited on page 76]

[89] Y. Chen, N. M. Nasrabadi, and T. D. Tran, “Hyperspectral image classification using dictionary-based sparse representation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, 2011. [Cited on page 76]

[90] C. Chang, Q. Du, T. Sun, and M. L. Althouse, “A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 6, pp. 2631–2641, 1999. [Cited on page 76]

[91] F. Feng, W. Li, Q. Du, and B. Zhang, “Dimensionality reduction of hyperspectral image with graph-based discriminant analysis considering spectral similarity,” *Remote Sens.*, vol. 9, no. 4, pp. 323, 2017. [Cited on page 76]

[92] X. Xu, J. Li, X. Huang, M. Dalla-Mura, and A. Plaza, “Multiple morphological component analysis based decomposition for remote sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 5, pp. 3083–3102, 2016. [Cited on page 76]

[93] L. Zhang, L. Zhang, D. Tao, and X. Huang, “On combining multiple features for hyperspectral remote sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 879–893, 2011. [Cited on page 76]

[94] A. E. Maxwell, T. A. Warner, and F. Fang, “Implementation of machine-learning classification in remote sensing: An applied review,” *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2784–2817, 2018. [Cited on page 76]

[95] M. Wieland and M. Pittore, “Performance evaluation of machine learning algorithms for urban pattern recognition from multi-spectral satellite images,” *Remote Sens.*, vol. 6, no. 4, pp. 2912–2939, 2014. [Cited on page 76]

[96] M. Pal and G. M. Foody, “Feature selection for classification of hyperspectral data by svm,” *IEEE Trans. . Remote Sens.*, vol. 48, no. 5, pp. 2297–2307, 2010. [Cited on page 76]

[97] G. Hughes, “On the mean accuracy of statistical pattern recognizers,” *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, 1968. [Cited on page 76]

[98] F. Melgani and L. Bruzzone, “Classification of hyperspectral remote sensing images with support vector machines,” *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, 2004. [Cited on page 76]

[99] L. Zhang, Y. Zhong, B. Huang, J. Gong, and P. Li, “Dimensionality reduction based on clonal selection for hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4172–4186, 2007. [Cited on page 76]

[100] F. Luo, H. Huang, Y. Duan, J. Liu, and Y. Liao, “Local geometric structure feature for dimensionality reduction of hyperspectral imagery,” *Remote Sens.*, vol. 9, no. 8, pp. 790, 2017. [Cited on page 76]

[101] H. Huang and M. Yang, “Dimensionality reduction of hyperspectral images with sparse discriminant embedding,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 9, pp. 5160–5169, 2015. [Cited on page 76]

[102] M. O. Ulfarsson, F. Palsson, J. Sigurdsson, and J. R. Sveinsson, “Classification of big data with application to imaging genetics,” *Proc. IEEE*, vol. 104, no. 11, pp. 2137–2154, 2016. [Cited on page 76]

[103] E. T. Gormus, N. Canagarajah, and A. Achim, “Dimensionality reduction of hyperspectral images using empirical mode decompositions and wavelets,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 6, pp. 1821–1830, 2012. [Cited on page 76]

[104] E. Esser, M. Moller, S. Osher, G. Sapiro, and J. Xin, “A convex model for non-negative matrix factorization and dimensionality reduction on physical space,” *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3239–3252, 2012. [Cited on page 76]

[105] P. Chen, L. Jiao, F. Liu, S. Gou, J. Zhao, and Z. Zhao, “Dimensionality reduction of hyperspectral imagery using sparse graph learning,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 3, pp. 1165–1181, 2016. [Cited on page 76]

[106] A. Plaza, P. Martinez, J. Plaza, and R. Perez, “Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 466–479, 2005. [Cited on page 76]

[107] Y. Dong, B. Du, L. Zhang, and L. Zhang, “Dimensionality reduction and classification of hyperspectral images using ensemble discriminative local metric learning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2509–2524, 2017. [Cited on page 76]

[108] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 844–856, 2012. [Cited on page 76]

[109] L. M. Bruce, C. H. Koger, and J. Li, "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 10, pp. 2331–2338, 2002. [Cited on page 76]

[110] B. Mojaradi, H. Abrishami-Moghaddam, M. J. V. Zoej, and R. P. Duin, "Dimensionality reduction of hyperspectral data via spectral feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 2091–2105, 2009. [Cited on page 76]

[111] Z. Wu, Y. Li, A. Plaza, J. Li, F. Xiao, and Z. Wei, "Parallel and distributed dimensionality reduction of hyperspectral data on cloud computing architectures," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 6, pp. 2270–2278, 2016. [Cited on page 76]

[112] G. Chen and S. Qian, "Denoising and dimensionality reduction of hyperspectral imagery using wavelet packets, neighbour shrinking and principal component analysis," *Int. J. Remote Sens.*, vol. 30, no. 18, pp. 4889–4895, 2009. [Cited on page 76]

[113] H. Du and H. Qi, "An fpga implementation of parallel ica for dimensionality reduction in hyperspectral images," in *IGARSS 2004. 2004 IEEE Int. Geosci. Remote Sens. Symp.* IEEE, 2004, vol. 5, pp. 3257–3260. [Cited on page 76]

[114] Z. Feng, S. Yang, S. Wang, and L. Jiao, "Discriminative spectral–spatial margin-based semisupervised dimensionality reduction of hyperspectral data," *IEEE Geosci.ence Remote Sens. Lett.*, vol. 12, no. 2, pp. 224–228, 2014. [Cited on page 76]

[115] S. Chen and D. Zhang, "Semisupervised dimensionality reduction with pairwise constraints for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 2, pp. 369–373, 2010. [Cited on page 76]

[116] X. Kang, S. Li, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, 2013. [Cited on page 76]

[117] H. Li, G. Xiao, T. Xia, Y. Y. Tang, and L. Li, "Hyperspectral image classification using functional data analysis," *IEEE Trans. Cy.*, vol. 44, no. 9, pp. 1544–1555, 2013. [Cited on page 76]

[118] P. Ghamisi, J. A. Benediktsson, and M. O. Ulfarsson, "Spectral–spatial classification of hyperspectral images based on hidden markov random fields," *IEEE Trans.Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2565–2574, 2013. [Cited on page 76]

[119] X. Liu and Y. Bo, "Object-based crop species classification based on the combination of airborne hyperspectral images and lidar data," *Remote Sens.*, vol. 7, no. 1, pp. 922–950, 2015. [Cited on page 76]

[120] T. Möckel, J. Dalmayne, H. Prentice, L. Eklundh, O. Purschke, S. Schmidtlein, and K. Hall, "Classification of grassland successional stages using airborne hyperspectral imagery," *Remote Sens.*, vol. 6, no. 8, pp. 7732–7761, 2014. [Cited on page 76]

[121] L. Pan, H. Li, Y. Deng, F. Zhang, X. Chen, and Q. Du, "Hyperspectral dimensionality reduction by tensor sparse and low-rank graph-based discriminant analysis," *Remote Sens.*, vol. 9, no. 5, pp. 452, 2017. [Cited on page 76]

[122] G. Licciardi and J. Chanussot, "Spectral transformation based on nonlinear principal component analysis for dimensionality reduction of hyperspectral images," *Eur. J. Remote Sens.*, vol. 51, no. 1, pp. 375–390, 2018. [Cited on page 76]

[123] R. E. Roger, "Principal components transform with simple, automatic noise adjustment," *Int. J. Remote Sens.*, vol. 17, no. 14, pp. 2719–2727, 1996. [Cited on page 76]

[124] C. M. Bishop, *Pattern Recognit. mach. learn.*, Springer, 2006. [Cited on page 76]

[125] Y. Zhang, X. Kang, S. Li, P. Duan, and J. A. Benediktsson, "Feature extraction from hyperspectral images using learned edge structures," *Remote sens. lett.*, vol. 10, no. 3, pp. 244–253, 2019. [Cited on page 76]

[126] B. Tu, N. Li, L. Fang, D. He, and P. Ghamisi, "Hyperspectral image classification with multi-scale feature extraction," *Remote Sens.*, vol. 11, no. 5, pp. 534, 2019. [Cited on page 76]

[127] X. Zhang, Y.i He, N. Zhou, and Y. Zheng, "Semisupervised dimensionality reduction of hyperspectral images via local scaling cut criterion," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 6, pp. 1547–1551, 2013. [Cited on page 76]

[128] S. Yang, P. Jin, B. Li, L. Yang, W. Xu, and L. Jiao, "Semisupervised dual-geometric subspace projection for dimensionality reduction of hyperspectral image data," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3587–3593, 2013. [Cited on page 76]

[129] H. Wu and S. Prasad, "Semi-supervised dimensionality reduction of hyperspectral imagery using pseudo-labels," *Pattern Recognit.*, vol. 74, pp. 212–224, 2018. [Cited on page 76]

[130] Y. Tang, R. Salakhutdinov, and G. Hinton, "Deep mixtures of factor analysers," *In Proceedings of the 29th Int. Conf. Mach. Learn.*, pp. 505–512, 2012. [Cited on page 83]

[131] J. A. Gualtieri and R. F. Cromp, "Support vector machines for hyperspectral remote sensing classification," in *27th AIPR Workshop: Adv. Comput. Vis. Pattern Recognit.* International Society for Optics and Photonics, 1999, vol. 3584, pp. 221–232. [Cited on page 85]

[132] C. Chang and C. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. [Cited on page 85]

[133] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004. [Cited on page 113]

[134] L. Zhang, L. Zhang, X. Mou, and D. Zhang, “Fsim: A feature similarity index for image quality assessment,” *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, 2011. [Cited on page 113]