Feature Extraction and Fusion for Classification of Remote Sensing Imagery

Kenmerkextractie en -fusie voor classificatie van remotesensingbeeldmateriaal

Renbo Luo

Promotors: Prof W. Philips, PhD, W. Liao, PhD, Prof Y. Pi, PhD
Doctoral thesis submitted in order to obtain the academic degrees of
Doctor of Computer Science Engineering (Ghent University) and
Doctor of Control Theory and Control Engineering (South China University of Technology)

GHENT
UNIVERSITY

Department of Telecommunications and Information Processing
Head of Department: Prof H. Bruneel, PhD
Faculty of Engineering and Architecture

Department of Automation and Network Engineering
Head of Department: Prof F. Luo, PhD
School of Automation Science and Engineering

Academic year 2016 - 2017

ISBN 978-94-6355-006-2
NUR 980
Wettelijk depot: D/2017/10.500/41

Members of the jury

Prof. Dr. Ir. Wilfried Philips (Ghent University, supervisor)
Dr. Ir. Wenzhi Liao (Ghent University, co-supervisor)
Prof. Dr. Ir. Youguo Pi (South China University of Technology, supervisor)
Prof. Dr. Ir. Stefaan Vandenberghe (Ghent University)
Prof. Dr. Ir. Lianfang Tian (South China University of Technology)
Prof. Dr. Ir. Weishi Zheng (Sun Yat-sen University)
Prof. Dr. Ir. Yuanqing Li (South China University of Technology, chairman)
Prof. Dr. Ir. Gert de Cooman (Ghent University, vice-chairman)
Prof. Dr. Ir. Qiliang Du (South China University of Technology, secretary)

Promoters:

Prof. Dr. Ir. Wilfried Philips
Dr. Ir. Wenzhi Liao
Prof. Dr. Ir. Youguo Pi

Research Group for Image Processing and Interpretation (IPI)
Department of Telecommunications and Information Processing (TELIN)
Faculty of Engineering and Architecture
Ghent University

Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium

Tel.: +32-9-264.34.12

Fax.: +32-9-264.42.95

Chairman: Prof. Dr. Ir. Yuanqing Li
Vice-Chairman: Prof. Dr. Ir. Gert de Cooman

This work came about in the context of a specialization fair of the CSC (China Scholarship Council) and Flemish Fund for Scientific Research (FWO-Flanders).

Acknowledgements

This work would never have been carried out without the guidance, help and support of supervisors, colleagues, families and friends. I would like to express my gratitude to the following people:

First and foremost, I would like to express my gratitude to my supervisors Youguo Pi, Wenzhi Liao and Wilfried Philips for the continuous support of my Ph.D study and related research. Thank you for your patience, motivation, immense knowledge, and useful relevant comments about my thesis. I would also like to thank Prof. Pi for the helpful guidance not only about my research work but also about my life in the last six years. My deepest gratitude goes to Wenzhi Liao. Thank you for your useful advices in dealing with numerous problems I encountered, even such as how to express sentences correctly in English, how to make a presentation, how to answer the comments of reviewers in more comfortable ways, and how to review a journal paper.

Specially, I am very thankful to Prof. Bart Goossens for translating the summary into Dutch. Special thanks to Prof. Paul Scheunders from the university of Antwerp, for your guidance and comments on our cooperated paper. Special thanks to Luk Brazle for helping me in English speaking and pronunciation.

My sincere gratitudes also go to all my colleagues and ex-colleagues at TELIN-IPI, Ghent University and South China University of Technology, for creating a nice atmosphere that makes work pleasant. I also wish to thank my Chinese friends: Xingzhe Xie, Shaoguang Huang, Hongyan Zhang, Jie Li, Junzhi Guan, Rui Wang, Pan Lian, Jiexiong Xie, Lingxiang Jiang, Chenglin Zuo, Qiang Chang, Zhixing Guo, Tao Liu, Chi Liu, thank you for having shared the joyful moments of spare time available to me in Ghent.

Further, I am thankful to the China Scholarship Council (CSC) and FWO for their funding in this research.

I am infinitely grateful to my families and relatives for their continuous love and support. I am grateful to my parents and my parents-in-law, for their trust, support and love. Last, but not least, my deepest gratitude and appreciation go to my wife, Tufen Hong, thank you for your selfless love, support and everlasting patience.

Renbo Luo
Ghent, February, 2017.

Samenvatting

Recente verbeteringen in remotesensingtechnologie maken het mogelijk om verschillende aspecten van objecten op aarde te meten, gaande van spectrale karakteristieken in multispectrale en hyperspectrale beelden, hoogteinformatie in Light Detection And Ranging (LiDAR) data, tot amplitude en fase in Synthetic Aperture Radar (SAR) systemen. Ondanks de rijkdom aan beschikbare informatie blijft de automatische interpretatie van remotesensingdata uitdagend.

Enorme hoeveelheden data, alsook de toenemende afmetingen bemoeilijken de mogelijkheid om de data te verwerken, wat problemen veroorzaakt in zowel de rekencomplexiteit als de opslagmiddelen. Bovendien veronderstellen classificatietechnieken in patroonherkenning vaak dat er genoeg trainingssamples beschikbaar zijn om nauwkeurige quantitatieve klassedescriptoren te leveren. Echter, in vele reële praktische toepassingen, is het verzamelen van grondwaarheidsdata vaak duur en tijdrovend. De beperkte trainingssamples kunnen dan Hughesverschijnselen veroorzaken, in het bijzonder voor de classificatie van remotesensingdata met een groot aantal dimensies. Ten slotte, verschillende gegevensbronnen hebben verscheidene voordelen en nadelen. Bijvoorbeeld, hyperspectrale beeldvorming biedt een overvloed aan waardevolle spectrale gegevens van verschillende objecten, maar kan geen onderscheid maken tussen verschillende objecten van hetzelfde materiaal. Daarnaast wordt hyperspectrale beeldvorming gemakkelijk beïnvloed door verschillende weersomstandigheden (variaties in helderheid). LiDAR-gegevens kunnen nuttige informatie bevatten over de grootte, structuur en hoogte van verschillende objecten, terwijl het moeilijk is om voorwerpen te discrimineren die qua hoogte gelijkaardig zijn, maar heel verschillend van aard zijn. Het onderzoek om aanvullende informatie uit meerdere databronnen te halen om de nauwkeurigheid erkenning van objecten te verbeteren is dan ook zeer uitdagend.

Om de bovenstaande uitdagende problemen aan te pakken, levert dit proefschrift een aantal bijdragen op vlak van kenmerkenextractie en datafusietechnieken om de classificatienauwkeurigheid van remote sensing beelden te verbeteren. In het algemeen kunnen onze voorgestelde methoden op een meer effectieve wijze kenmerken extraheren die leiden tot een hogere classificatienauwkeurigheid en een hogere efficiëntie in het verminderen van de rekencomplexiteit. Dit leidt tot mogelijke verbeteringen voor de verwerking van grote datasets. Een meer specifiek overzicht van onze bijdragen is als volgt:

- De eerste bijdrage van dit proefschrift bestaat uit een verkenning van gesuperviseerde kenmerkenextractiealgoritmen voor de classificatie van

hyperspectrale remotesensingbeelden door lokale geometrische structuren en labelinformatie te combineren. Meer specifiek stellen we de discriminerende gesuperviseerde buurbehoudende inbedding (DSNPE) en de gesuperviseerde plaatsbehoude projectie (PSLPP) voor. DSNPE incorporeert labelinformatie in een lineaire omgeving-behoudende extractiemethode, trekt naburige punten binnen dezelfde klasse dichter naar elkaar toe, terwijl naburige punten met verschillende labels verder van elkaar worden weggeduwd, tijdens de projectie van een hoogdimensionele kenmerkenruimte naar een lage kenmerkenruimte. PSLPP gebruikt eerst PCA om ruis en redundantie te verwijderen, en combineert dan vervolgens labelinformatie en de lokaliteit behoudende projectie om gelijkenissen tussen samples te construeren.

- Normaalgezien is het aantal gelabelde trainingssamples niet voldoende voor gesuperviseerde leertechnieken; als tweede bijdrage stellen we daarom nieuwe semi-gesuperviseerde kenmerkenextractiemethoden voor door beperkt gelabelde samples te combineren met een groot aantal ongelabelde samples. In de eerste instantie, verbeteren we de semi-gesuperviseerde lokale discriminantenanalysemethode (SELD) (die de gelabelde-gelabelde en de ongelabelde-ongelabelde relaties tussen samples modelleert) door toevoeging van de correlatie van gelabelde-ongelabelde samples, waarbij de connecties tussen een deel van de samples naar alle samples worden uitgebreid. Ten tweede, stellen we een semi-gesuperviseerde graph learning methode (SEGL) voor, die toelaat om een semi-gesuperviseerde graaf op te bouwen die de gelijkenissen tussen samples kan beschrijven. In onze semi-gesuperviseerde graaf, connecteren we gelabelde samples volgens hun labelinformatie en niet-gelabelde samples volgens hun dichtstebuurinformatie, en connecteren we de niet-gelabelde sample met gelabelde samples horende bij de dichtste buurtklasse. Bovendien, om beter de werkelijke verschillen en gelijkenissen tussen samples te modelleren, leggen we een gewogen grens vast tussen de geconnecteerde samples. Tot slot breiden we de semi-gesuperviseerde graafleermethode (SEGL) uit van het spectrale domain naar het spatiale domein, en bouwen we een semi-gesuperviseerde fusiegraaf door spectrale en spatiale informatie te combineren, met als doel om beter de correlaties tussen samples te modelleren eerder dan enkelvoudige informatie te gebruiken.

- Om complementaire informatie met multisensordata te combineren om zo de classificatieprestaties verder te verbeteren, stellen we ook een nieuw raamwerk voor om hyperspectrale en LiDAR beelden te fuseren, voor de classificatie van wolkoverdekte remotesensingscènes. In het voorgestelde raamwerk worden de wolkoverdekte en niet wolkoverdekte gebieden afzonderlijk behandeld. Eerst extraheren we een workschaduwmasker om de remotesensingscène te verdelen in twee gebieden (namelijk, wolkoverdekt en wolkvrij). Vervolgens classificeren we de niet-geschaduwde gebieden door verschillende kenmerken te integreren (bijvoorbeeld, spectraal uit rauwe

hyperspectrale data, spatiaal gegenereerd uit hyperspectrale beelden, en hoogtegegevens uit LiDAR data) met behulp van de beschikbare trainingssamples. Om wolkoverdekte gebieden te classificeren, genereren we nieuwe trainingsets van wolkoverdekte gebieden door de dichtste buren van de klasgemiddelden (verkregen van de LiDAR data) gebaseerd op zowel spectrale als spatiale kenmerken. De pixels van wolkoverdekte gebieden worden geclassificeerd met een gelijkaardige strategie als de schaduwvrije gebieden, terwijl de classifier getraind wordt op basis van nieuw gegenereerde trainingsamples. De uiteindelijke classificatiemap wordt verkregen door de classificatieresultaten van de schaduwvrije en wolkoverdekte gebieden samen te voegen. Het voorgestelde raamwerk maakt zo volledig gebruik van de voordelen van de verschillende gegevensbronnen.

- Onze laatste bijdrage bestaat uit het versnellen van niet-lineaire kenmerkenextractiemethodes door de voordelen van een grafische verwerkingseenheid (GPU) te benutten. Niet-lineaire kenmerkenextractiemethodes, zoals kernel principiële componentenanalyse (KPCA) zijn meer geschikt om niet-lineaire en hogere-orde distributies van de data te beschrijven, maar gaan gepaard met een relatief hogere rekencomplexiteit en een langere uitvoeringstijd. In deze dissertatie ontwikkelen we een efficiënte parallelle implementatie van het KPCA kenmerkenextractiealgoritme op GPU met behulp van de Jacket MATLAB Toolbox. Door de voorgestelde kenmerkenextractiemethodes in parallel toe te passen, verkrijgen we een significante versnelling (meer dan 100 keer) voor niet-lineaire kenmerkenextractiemethods (zoals KPCA), zonder in te boeten in classificatienauwkeurigheid.

Voor experimenten op echte datasets, vertonen de nieuwe technieken die ontwikkeld zijn in dit proefschrift een nauwkeurigheidsverbetering ten opzichte van een aantal state-of-the-art methoden. Bovendien hebben we aangetoond dat onze technieken efficiënt zijn.

Summary

Recent advances in remote sensing technology allow us to measure different aspects of objects on the Earth, from spectral characteristics in multispectral and hyperspectral images, to height information in the Light Detection And Ranging (LiDAR) data, to amplitude and phase in Synthetic Aperture Radar (SAR) systems. Despite the richness of information available, automatic interpretation of remote sensing data remains challenging.

Hugh amounts of data, as well as the increasing dimensions hamper the ability to process the big data, causing problems in both computational complexity and storage resources. What's more, classification techniques in pattern recognition typically assume that there are enough training samples available to obtain accurate class descriptions in quantitative form. However, in many real applications, collecting ground-truth is often expensive and time consuming in practical applications. The limited training samples may leads to the Hughes phenomenons when doing classification for high dimensionality of remote sensing data (e.g. hyperspectral imagery). Last but not least, different data sources have different advantages and shortages, such as hyperspectral imagery can provide plentiful and valuable spectral information of different objects of interest, but cannot distinguish different objects made of the same material, and is easily influenced by different weather conditions (cloudy); LiDAR data can provide useful information about the size, structure and elevation of different objects, while it is difficult to discriminate different objects which are similar in altitude but quite different in nature. How to extract complementary information from multi-source data to improve recognition accuracy of objects is still very difficult.

In order to address the challenging problems mentioned above, this dissertation focus on developing new feature extraction and data fusion techniques to improve the classification accuracy of remote sensing imagery. In general, our proposed methods can extract more effective features for higher classification accuracy and more efficiency in reducing the computational complexity, leading to potential improvements in processing of huge datasets. A more specific summary of our contributions can be highlighted in the following:

- The first contribution of this thesis is the exploration of supervised feature extraction algorithms for classification of hyperspectral remote sensing imagery by combining local geometrical structure and label information. In detail, discriminative supervised neighborhood preserving embedding (DSNPE) and principle component analysis (PCA)-based supervised locality preserving projection (PSLPP) are presented. DSNPE incorporates

the label information into a linear neighborhood preserving extraction method, pulls the neighboring points with the same class label closer, while simultaneously pushes the neighboring points with different labels far away from each other when projecting them from high dimensional feature space into low feature space. PSLPP first uses PCA to remove noisy and redundancy, and then combines label information and locality preserving projection to construct similarities between samples.

- Normally, the number of labelled training samples is not enough for supervised feature learning, our second contribution is the proposition of novel semi-supervised feature extraction methods by combining limited labeled samples and a large number of unlabelled samples. First, we improve the existing semi-supervised method by taking into account the correlations between labelled and unlabelled samples. Secondly, a semi-supervised graph learning (SEGL) method is proposed. The main contribution of SEGL is constructing a semi-supervised graph to model the similarities between samples, as labelled samples are connected according to their label information, unlabelled samples are connected by their nearest neighborhood information, and the connections between labelled and unlabelled samples are based on the distance between class center and unlabelled samples. All connected samples have been set a weighted edge to better model the actual differences and similarities between them. Lastly, we extend semi-supervised graph learning (SEGL) to both spectral and spatial domains, and build a semi-supervised fusion graph to simulate the correlations of samples.

- In order to combine the complementary information from multi-sensor data to improve classification performance, we propose a novel framework to fuse hyperspectral and LiDAR images for classification of the cloud-shadow mixed remote sensing scenes. In proposed framework, the cloud-shadow and non-shadow regions are processed separately. Firstly, we extract a cloud-shadow mask to divide the remote sensed scene into two parts (cloud-shadow and shadow-free). Then we classify shadow-free region by integrating multiple features (e.g. spectral from raw HS image, spatial generated from HS image, and elevation from LiDAR data), with available training samples. For classification of cloud-shadow region, we generate reliable training sample sets from cloud-shadow region by searching the nearest neighbors of each class center (obtained from LiDAR data) based on both spectral and spatial features. The pixels of cloud-shadow areas are classified with similar strategy to shadow-free region, while the classifier is trained by the new generated reliable training samples. The final classification map is produced by decision fusion of the classification results from both the shadow-free and cloud-shadow regions. The proposed framework makes full use of the advantages of different data sources.

- Our last contribution to speed up non-linear feature extraction meth-

ods by exploiting the advantages of GPU. Non-linear feature extraction methods, like kernel principle component analysis (KPCA) are more suitable to describe non-linear and higher-order distributions of the data, but with relatively large computational complexity and need long execution time. An efficient implementation of KPCA feature extraction algorithm on graphics processing unit (GPU) based on Jacket MATLAB Toolbox in parallel strategy is developed in this thesis. By using the proposed methods based on parallel strategy, we can speed up non-linear feature extraction methods (e.g. KPCA) significantly (more than 100 times), without losing classification accuracy.

Compared with some state of the art methods by the experiments on some real data sets, the novel techniques developed in this thesis demonstrates an improvement in terms of accuracies and have been proven to be efficient.

Contents

CONTENTS

List of Abbreviations

AA	Average Accuracy
AF	Attribute Filter
AP	Attribute Profile
APPR	Attribute Profile with Partial Reconstruction
CCIPCA	Candid Covariance-Free Incremental Principle Component Analysis
CPU	Central Processing Units
CUDA	Compute Unified Device Architecture
DSNPE	Discriminative Supervised Neighborhood Preserving Embedding
EAP	Extended Attribute Profile
EMAP	Extended Multi-Attribute Profile
EMP	Extended Morphological Profile
FE	Feature Extraction
GDA	Generalized Discriminant Analysis
GPKPCA	GPU-based Parallel Kernel Principle Component Analysis
GPU	Graphics Processing Units
HS	Hyperspectral
ICA	Independent Component Analysis
IC	Independent Component
ISEGL	Improved Semi-supervised Graph Learning
ISELD	Improved Semi-supervised Local Discriminant Analysis
KPCA	Kernel Principal Component Analysis
KPC	Kernel Principal Component

LDA	linear discriminant analysis
LiDAR	Light Detection and Ranging
LPP	Locality Preserving Projection
LapSVM	Laplacian Support Vector Machine
MAP	Multi-Attribute Profiles
MP	Morphological Profile
MRF	Markov Random Field
NDA	Nonparametric Discriminant Analysis
NN	Nearest Neighbor
NPE	Neighborhood Preserving Embedding
NWFE	Nonparametric Weighted Feature Extraction
OA	Overall Classification Accuracy
PCA	Principle Component Analysis
PC	Principal Component
PSLPP	PCA-based Supervised Locality Preserving Projection
RBF	Radial Basis Function
RF	Random Forest
SDA	Semi-supervised Discriminant Analysis
SEGL	Semi-supervised Graph Learning
SELF	Semi-supervised Local Fisher Discriminant Analysis
SELD	Semi-supervised Local Discriminant Analysis
SLPPCE	Semi-supervised Locality Preserving Projections with Compactness Enhancement
SSDR	Semi-Supervised Dimensionality Reduction
SSL	Semi-Supervised Learning
SSS	Small Sample Size
SVM	Support Vector Machine
TSVM	Transductive Support Vector Machine

1

Introduction

Remote sensing is crucial for various aspects of our life (from monitoring weather conditions, following possibly threatening storms or air pollution to safety and security). The general aim of this thesis is to develop new methods to extract discriminant efficient features from remote sensing data, and fuse the extracted features for land-cover/land-use classification. This Chapter presents a general framework of this thesis. First, the overview on the remote sensing field is introduced and the necessary background is reviewed. Then, the problems or challenges about remote sensing imagery processing are analysed. Last but not the least, the objectives and main contributions are summarized.

1.1 Remote Sensing

Remote sensing represents a set of techniques and algorithms, which are able to collect and interpret information regarding an object or phenomenon without making physical contact with the item under investigation [Campbell 02, Schott 07, Schowengerdt 07]. More particularly, remote sensing refers to the instrument-based technology and application for the detection, classification, and recognition of objects in the Earth [NAS , Richards 06], the Moon [Montopoli 07], the Mars [Roush 97] and other planets. Remote sensing techniques emerged with photography, and became popular with the invention of air-planes and then of satellites. From the 1950s, when the first artificial satellite was launched, remote sensing began to be used not only for military but also for civil operations. Advances in technology during recent decades have turned remote sensing into an essential technology. With research and development in the fields of electronics, informatics and signal processing, more and more remote sensing sensors/devices have been created and are able to acquire different types of information for a great number of applications. From a broader consideration, remote sensing devices include embedding structures for spacecraft, engineering life and atmospheric or geometric calibration, such as Radio Detection and Ranging (RADAR) [kre 15] sensors, Light Detection and Ranging (LiDAR) [LiD 13] sensors, x-ray units, Magnetic Resonance Imaging (MRI).

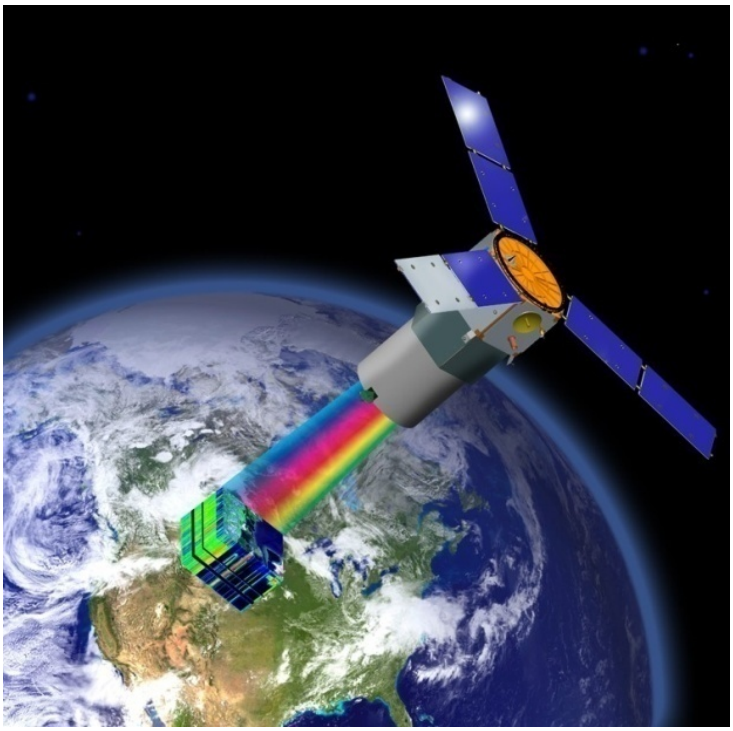


Figure 1.1: Remote sensing.

Remote sensing sensors often produce large amounts of data. From a physics point of view, these remote sensing data are electromagnetic radiation reflected from or emitted from the objects. Since most pixels (or pure objects) possess unique material characteristics, they can be distinguished by their different reflected radiation, allowing accurate classification or recognition. The details differ between sensors, such as multispectral/hyperspectral sensor [Dalponte 10]), LiDAR sensor and RADAR sensor. Most remote sensing sensors are carried by particular vehicles known as platforms, normally air-crafts or satellites in orbit for visual exploration (see Figure 1.1).

Thanks to the availability of a large number of sensors (with different functions and peculiarities), remote sensing data are used in many different applications, including urban distribution, agriculture monitoring, damage assessment, ice monitoring and forest inventories etc. Different applications use different sensors depending on what works best in the application. Hyperspectral and LiDAR data are two main remote sensing data sources we will exploit in the thesis. Nevertheless, the developed techniques in this thesis (e.g. method proposed in Chapter 4) could be also applied to other data sources or applications domains.

1.1.1 Hyperspectral Image

The idea of multispectral or hyperspectral imaging for remote sensing emerged at NASA's Jet Propulsion Laboratory in 1983, where the Airborne Visible InfraRed Imaging Spectrometer (AVIRIS) [Green 98] was developed for delivering high-dimensional data cubes with hundreds of contiguous spectral channels (bands) covering the wavelength region from 400 to 2500 nanometers. Multispectral remote sensors (or imaging spectroscopy) generate images of bands with relatively broad spectral widths, typically about 100 nanometers between 400 and 1100 nanometers (visible and near-infrared region), which limits their functionality for Earth observation purposes. While advanced hyperspectral sensor systems can acquire the detailed spectrum of reflected light throughout the visible, near-infrared, and mid-infrared portions of the electromagnetic spectrum, and produce huge data cube with large amount of spectral bands [Chang 03]. Figure 1.2 shows a hyperspectral image, in this hyperspectral image every pixel represents as a high-dimensional vector containing values corresponding to reflectance spectrum, so that the dimension of the vector (one pixel) is equal to the number of spectral bands.

From another perspective, the reflectance spectrum in one wavelength interval (spectral channel) can be considered as one gray scale image. The hyperspectral image can be seen as a stack of images (corresponding to different spectral channels) from the same area on the surface of the Earth. In other words, it forms a three dimensional data cube. Figure 1.3 shows an example of such a hyperspectral data cube. In the following, we assume a three dimensional hyperspectral data cube with $n_1 \times n_2$ pixels in the spatial domain, and d spectral bands. Such a cube can be treated in various ways:

1. Spectral perspective: In this case, a hyperspectral data cube includes large amount of pixels which represent specific regions of the Earth surface. Each pixel can be seen as a vector with multiple components, which correspond to the reflected radiation in specific spectral bands. This spectral information can be used to precisely distinguish different materials. The image in Figure 1.2 shows histograms of the values of specifics spectral components.

2. Spatial perspective: In this case, each spectral band is treated as a separate gray scale image. see Figure 1.3. In the spatial dimension, most neighboring pixels belong to the same object, particular for Very High Resolution (VHR) data.

Typically, hundreds or even thousands of spectral bands are available in a hyperspectral cube. This amount of spectral information available for each pixel of a scene increases the possibility of accurately distinguishing different physical materials. This is possible because different materials exhibit different spectral signatures. Figure 1.2 shows the spectral signatures of four different pixels from four materials, which show completely different behaviours in spectral domain. Unlike conventional Red, Green, and Blue (RGB) images,

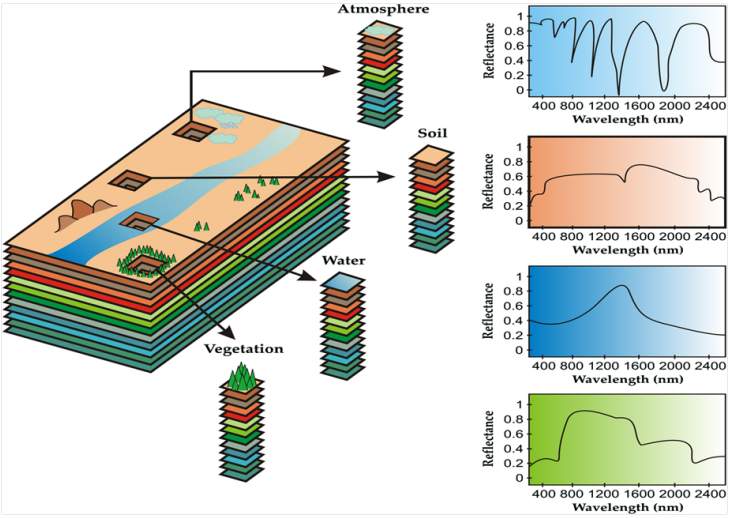


Figure 1.2: Spetral reflectance of different materials.

the rich spectral attributes of hyperspectral data allow practical applications such as: food quality inspection [Sun 10, Kelman 13, Qiao 15], medical sciences [Liu 07, Lu 14], mineralogy [Werff 06, Meer 12], military applications [Eismann 96, Heesung 05].

1.1.2 LiDAR Data

LiDAR data represent the distance between objects and the sensor, which is very different from hyperspectral data. LiDAR [LiD 13], which stands for Light Detection and Ranging, originated in the early 1960s, shortly after the invention of the laser. Combined with other data recorded by an airborne system, LiDAR can generate precise, three-dimensional information about the shape of the Earth and its surface characteristics, so it is a popular technology to make high-resolution maps.

The process of LiDAR can be simply summarized as: an airborne laser is pointed at a targeted area on the ground, then the beam of light (infrared, visible light, or ultraviolet light) is reflected by the surface it encounters, at this time, a LiDAR sensor (receiver detectors) and electronics will record this reflected light to measure a range (variable distances to Earth). When laser ranges are combined with scan angles, calibration data, position and orientation data generated from integrated GPS, a dense, detail-rich group of elevation points, called a "point cloud," will be produced. In the point cloud, each point has three-dimensional spatial coordinates (latitude, longitude, and height), and corresponds to a particular point on the Earth's surface where laser pulse was reflected from. As the elevation information collected by LiDAR sensors is

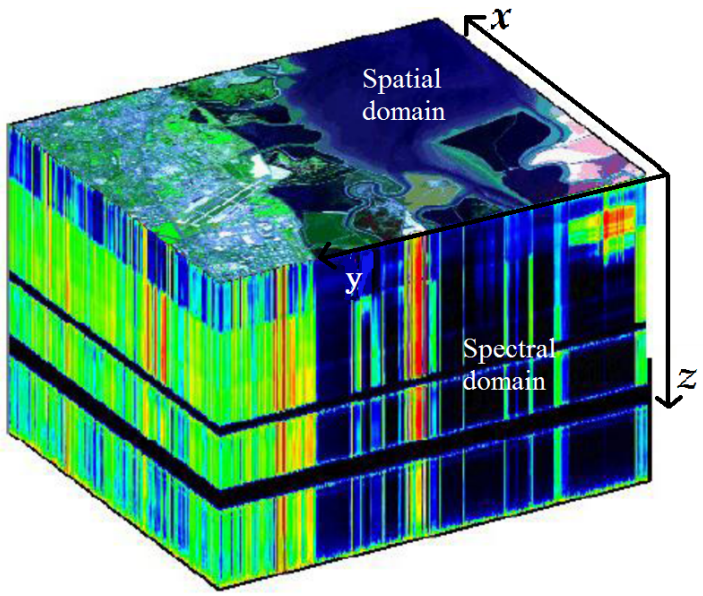


Figure 1.3: Hyperspectral cube (from NASA).

based on light pulses, it is not influenced by weather conditions, such as clouds and shadows, thus makes its potential combination with hyperspectral image on the applications of Earth observation.

1.2 Problem Statement

Detecting small targets can be very difficult, particularly in an unknown environment. In order to provide a better understanding of the processing (classification) of remote sensing data, we introduce a description of the full processing chain, as shown in Figure 1.5. This processing chain is widely adopted by most researchers and consists of three consecutive stages: data acquisition, feature extraction/fusion and classification. Data acquisition also involves preprocessing. As the original inputting data sets come from different sensors and have different nature, such as the high dimensionality of hyperspectral image and one dimensionality of LiDAR data, it is not good way to do classification directly. Therefore, feature extraction and fusion is a very necessary and important step. The details of feature extraction, fusion and classification of remote sensing imagery are introduced in following subsections.

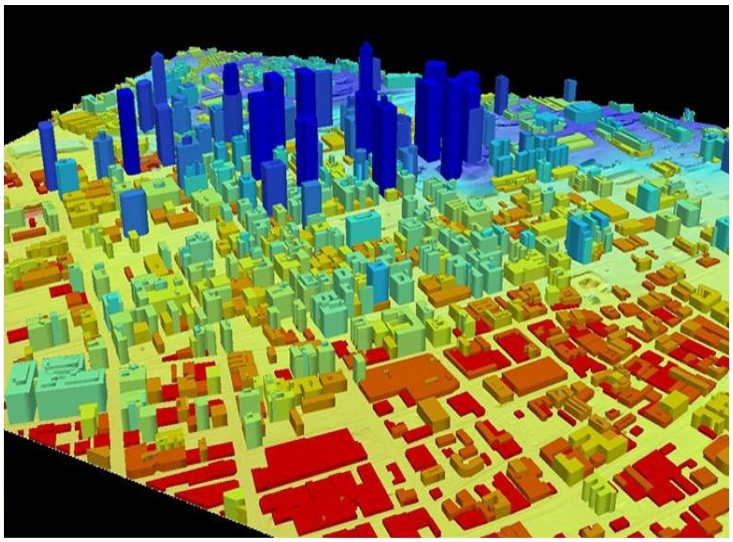


Figure 1.4: LiDAR image.

1.2.1 Feature Extraction and Fusion

Based on the idea of mining the useful information [Jia 13], feature extraction can be explained as finding a transformation to transform the high-dimensional data set to a low-dimensional feature space and reducing dimensionality. In processing of remote sensing imagery, feature extraction is the process of producing a small number of informative features by transforming the input data linearly or non-linearly to another new feature space [Fong 07, Zhou 15]. This is equivalent to (possibly non-linear) projection of the high-dimensional original data onto a low-dimensional subspace. The general concept of feature extraction will be employed in following.

Suppose $\mathbf{x}_i$ is original high-dimensional data point. Then the extracted features $\mathbf{z}_i$ in a low-dimensional projected subspace, are given by $\mathbf{z}_i = \mathbf{W}^T\mathbf{x}_i$, in the linear case. The main problem then is to find a suitable transform matrix $\mathbf{W}$ See Figure 1.6.

In fact, in order to reduce the dimensionality and get a few useful features, there is also another approach: selecting a suitable subset from the original features, which is called feature selection. The essential point relative to feature selection is to pursue an efficient search strategy to obtain an optimal subset to improve classification. While feature extraction does not use the original features directly, it transforms the original features to a new feature space and produces new features. The essential difference between the feature extraction and selection in visually can be seen in Figure 1.7. For example, in original data set, there are p measurements (features) in total. If l features are needed for classification, feature selection will select l features out of these p features,

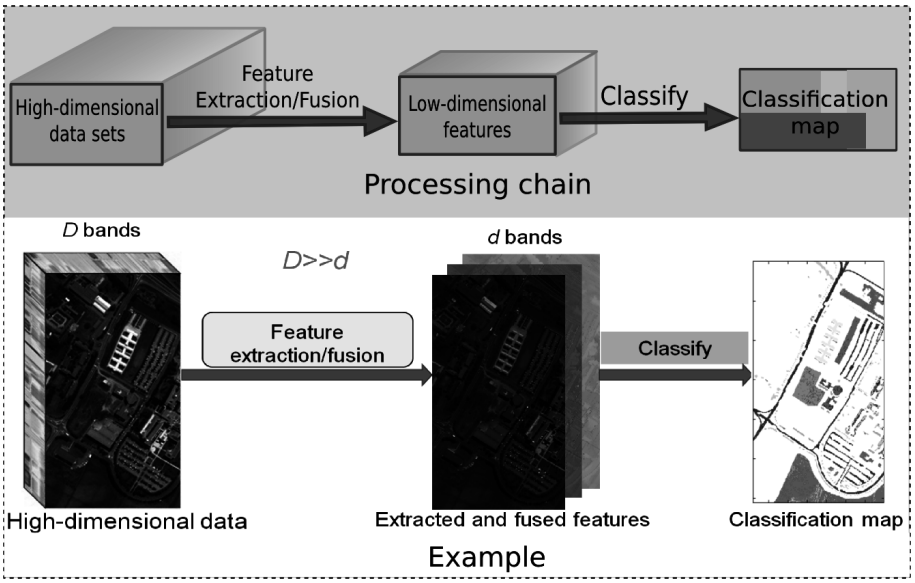


Figure 1.5: Remote sensing data processing chain and an example.

feature extraction will map these p features to l new features (in new feature space) which have correlations with all original p features. Compared with feature selection, the advantages of feature extraction are that no information from the original data is wasted.

The existing feature extraction methods can be grouped into unsupervised, supervised and semi-supervised cases. Unsupervised feature extraction methods do not require any prior knowledge on the training data [Serpico 07]. Supervised and semi-supervised methods require prior knowledge of the labelled class assigned to the different pixels. Supervised methods lead to better representation as they are computed taking into account the class information from pixels. By exploiting the useful label information of training samples, supervised methods can infer class separability. However, in many real-world applications, labelling large amounts of data may require considerable human resources and is time consuming. Therefore, the number of labelled data is usually very limited, while unlabelled data is available in large quantities at very low cost. As a result, semi-supervised methods, which aim at improved classification by utilizing both a large number of unlabelled and limited labelled samples in the training phase gained popularity in the machine learning community [Olivier 06, Zhang 07, Cai 07, Zhu 08, Liao 13].

Feature fusion in our thesis means integration of multiple features (fused in feature level) generated either from single data or multisensor data. The advantage of feature fusion is obvious. Different features extracted from single data or multi sensor data reflect different characteristic of patterns. By

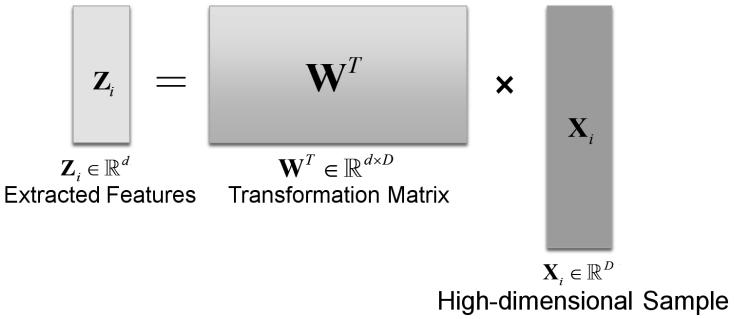


Figure 1.6: Feature extraction, $\mathbf{x}_i$ is original sample with D bands, $\mathbf{z}_i$ presents extracted features with d bands, $d << D$.

optimizing and combining these different features, feature fusion method preserves discriminant information from all features, while eliminates redundant information to certain degree. This is especially important in classification and recognition. In the feature fusion step, several feature vectors, such as spectral and spatial features from hyperspectral image, elevation features from LiDAR data, are stacked into one union-vector, and then a few useful features are extracted from the higher-dimension union-vector. Please see Figure 1.8 for visually and better understanding.

1.2.2 Classification

Given a set of observations (i.e., pixel vectors in a remote sensing image), the goal of classification is to assign a class label to every pixel in the image [Richards 06]. For visualization, the label represented by a color. Within this context, the input of a classification problem is remote sensing images and the objective is to create a thematic map. There are different categories of classification techniques depending on the availability of labelled training samples, including unsupervised methods, supervised methods, and semi-supervised methods. For the purpose of remote sensing imagery analysis, we focus on supervised classifiers: k-Nearest Neighbors (kNN) [Coomans 82], Support Vector Machine (SVM) [Chang 01] and Random Forests (RF) [Ho 95]. These types of methods are trained using a set of representative samples for each class, referred to as training samples. Before classification, a set of training samples for each class are used to partition the feature space into decision regions. To assess performance, the trained classifier is applied to a set of test pixels with known ground truth, the results in an estimate of the classification accuracy are used for performance assessment. As the experiments are carried out under the same conditions and classifiers, the difference in classification accuracy is attributed to the employed features, i.e., the efficiency of the extracted features from the corresponding approaches.

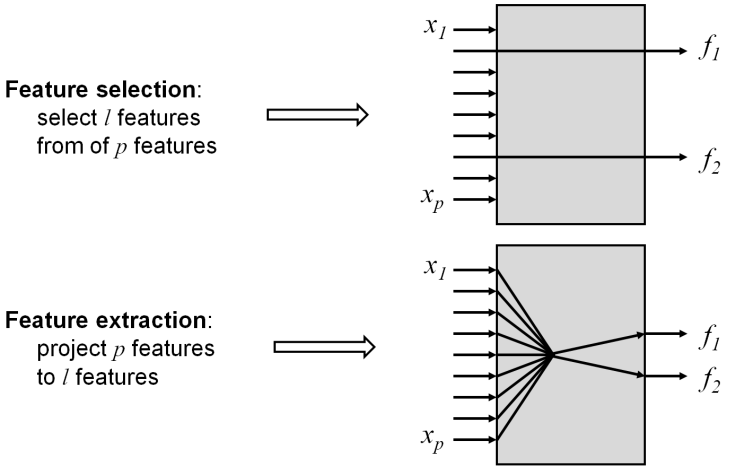


Figure 1.7: Different between feature extraction and feature selection, feature selection selects l features out of p features, while feature extraction maps p features to l new features (in new feature space) which have correlations with all original p features.

1.2.3 Challenges

In the last decade, many approaches for remote sensing image feature extraction and classification have been proposed and investigated. Although hyperspectral remote sensing data contain hundreds or even thousands of spectral bands and have many practical applications, the large number of spectral bands result in large data sets, and increase the processing complexity. Moreover, the primary challenge in remote sensing is how to extract and fuse the most appropriate features for specific applications. Several key issues are:

1. The large number of spectral bands leads to problems with storage resources and computational load. In order to process and analyze these huge data sets, super-computers and large data storage capacities are required. What's more, the original spectral bands in hyperspectral image contain high redundancy, especially for the adjacent bands, there are high correlations between them.

2. In practical applications, collecting ground-truth is often expensive and time consuming, as it requires a skilled expert agent to manually classify training examples. Thus, the *small sample size* (SSS) problem [Raudys 91], which states the number of available training samples is relatively much smaller than the dimensionality of the original data, is an important issue for high-dimensional hyperspectral image classification. This problem creates a challenge for conventional classification methods, especially for classifiers which are not robust to the Hughes phe-

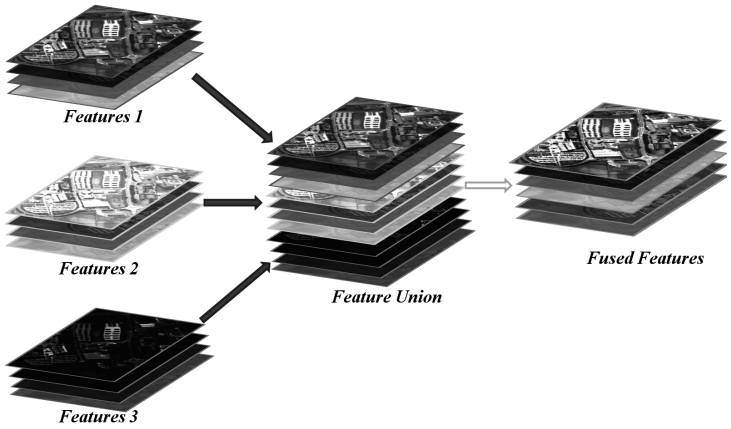


Figure 1.8: An example of feature fusion (fused in feature level). Suppose there are three types of features for a same region, feature fusion (in feature level) methods stack them together first, then map the high-dimensional stacking data set to a low-dimensional feature space and generate new features.

nomenon [Hughes 68]; these suffer from rapidly decreasing classification accuracy which increasing dimension or decreasing number of training samples, see Figure 1.9. In the real world, remote sensing hyperspectral images have more than hundred spectral bands, while the number of training samples is quite small. Therefore, addressing the SSS problem is essential for classification.

3. Most existing methods focus on performing the feature extraction in the spectral domain of the hyperpspectral data. Nevertheless, spatial domain processing is equally important and can be combined with spectral domain processing, to improve classification performance. However, some spectral-spatial feature extraction methods can not efficiently fuse spectral and spatial information [Khodadadzadeh 15, Zhou 15, Huang 13].

4. Different data sources have different advantages and disadvantages. For instance, hyperspectral images cannot distinguish different objects made of the same material, and are easily influenced by cloud and weather conditions. LiDAR data can provide useful information about the size, structure and elevation of different objects, but cannot discriminate well between objects with similar altitude but different materials. Optimal fusion of multi-source data for improving the accuracy of pattern recognition therefore is a big challenge.

In order to address the above challenging problems, the thesis focuses on feature extraction and fusion of remote sensing data, where the main objectives are to investigate and propose solutions to find efficient and discriminative

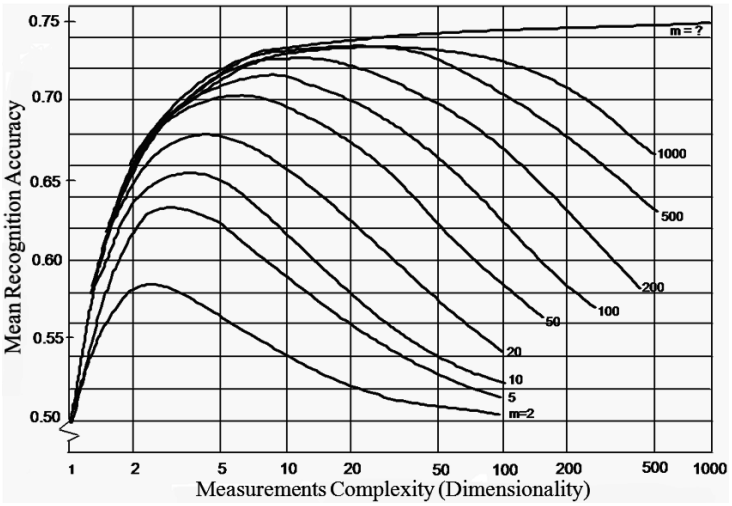


Figure 1.9: Hughes phenomenon, the recognition accuracy (Y-axis) will decrease from one point with increasing number of dimensionality (X-axis) or decreasing number of training samples (m).

features. The proposed methods are evaluated and compared to state-of-the-art methods from the literature, normally including the use of the original spectral features as an initial reference for bench marking. In order to evaluate the behaviour and performance of feature extraction methods on a consistent and comparable basis, different methods for feature extraction and data reduction are compared in the experiments under the same conditions, keeping the same data acquisition and classifiers, only changing different methods in the step of feature extraction/fusion.

1.3 Contributions and Publications

The work presented in this thesis aims at investigating and developing novel feature extraction and fusion techniques for classification of remote sensing imagery. In general, the developed techniques provide more effective features to enable an improved classification performances and more efficiency to reduce the computational complexity, leading to potential improvements in processing huge datasets. State-of-the-art techniques have already proven that the use of extracted features are effective for the classification of real data sets.

The main contributions of this thesis are:

- *Exploration of new supervised feature extraction algorithms for classification of hyperspectral remote sensing imagery.*

  Two novel supervised feature extraction algorithms, relying on labelled

samples to infer class separability. The first one is called discriminative supervised neighborhood preserving embedding (DSNPE), which incorporate the label information into a linear feature extraction named neighborhood preserving embedding (NPE). Similar to NPE, DSNPE preserves the local manifold and neighborhood structure, while projecting similar samples closer and dissimilar samples further apart on a lower dimensional feature space. Another proposed supervised feature extraction method is PCA-based supervised locality preserving projection (PSLPP), which combines principle component analysis (PCA), label information and locality preserving projections (LPP) together. In the proposed PSLPP method, principle component analysis is used to remove noise and redundancy, and label information and locality preserving projection are used to construct similarities between samples.

- *Proposition of new semi-supervised feature extraction methods, which combines a small number of labelled training samples with a large number of unlabelled training samples for feature extraction.*

  We propose a feature extraction method based on semi-supervised graph learning (SEGL), which aims to build a semi-supervised graph to describe the similarities between samples, especially the similarities between labelled and unlabelled ones. We also extend SEGL to both spectral and spatial domains and get better results. What's more, we improve semi-supervised local discriminant analysis (SELD) method for feature extraction of remote sensing scenes.

- *Definition of a novel framework to fuse hyperspectral and LiDAR images for classification of cloud-shadow mixed remote sensing scenes.*

  We propose a new framework to fuse cloud-shadowed hyperspectral and LiDAR data to increase classification performance, especially for cloud-shadow region. In our proposed methods, we process the cloud-shadow and shadow-free regions separately. Our main contribution is the development of a novel method to generate reliable training samples in the cloud-shadow regions. Classification is performed separately in the shadow-free (classifier is trained by the available training samples) and cloud-shadow regions (classifier is trained by our generated training samples) by integrating spectral (i.e. original HS image), spatial (morphological features computed on HS image) and elevation (morphological features computed on LiDAR) features. The final classification map is obtained by fusing the results of the shadow-free and cloud-shadow regions.

- *Exploiting the use of GPU to speed up non-linear feature extraction methods.*

  As a non-linear version of principle component analysis (PCA), kernel principle component analysis (KPCA) is more suitable to describe non-linear, higher-order and complex distributions. However, One disadvantage of KPCA is that its sequential implementations have long run time

due to their relatively large computational complexity. In order to speed up the computing process of KPCA, we implemented the KPCA algorithm to GPU to extract features from hyperspectral images, the experimental results reveal the GPU based parallel KPCA approach has the potential to improve computation speed.

In total, the research during this PhD resulted in several publications:

1. **Luo Renbo**, Liao Wenzhi, Zhang Hongyan, Zhang Liangpei, Pi Youguo, Scheunders Paul, Philips Wilfried, "Fusion of hyperspectral and LiDAR data for classification of cloud-shadow mixed remote sensing scene". IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. DOI:10.1109/JSTARS. 2017.2684085.(A1)

2. **Luo Renbo**, Liao Wenzhi, Huang Xin, Pi Youguo, Philips Wilfried, "Feature Extraction of Hyperspectral Images with Semi-Supervised Graph Learning". IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2016; 9(9): 4389-4399.(A1)

3. **Luo Renbo**, Liao Wenzhi, Zhang Hongyan, Pi Youguo, Philips Wilfried, "Spectral-Spatial Classification of Hyperspectral Images with Semi-Supervised Graph Learning". SPIE Remote Sensing. Sep.2016.(P1)

4. **Luo Renbo**, Liao Wenzhi, Zhang Hongyan, Pi Youguo, Philips Wilfried, "Classification of cloudy hyperspectral image and LIDAR data based on feature fusion and decision fusion". IEEE Geoscience and Remote Sensing International Symposium (IGARSS 2016). Jul. 2016. (P1)

5. **Luo Renbo**, Liao Wenzhi, Pi Youguo, Philips Wilfried, "An improved semi-supervised local discriminant analysis for feature extraction of hyperspectal image". Joint Urban Remote Sensing Event, Proceedings (JURSE 2015). Mar. 2015. p. 1-4. (P1)

6. **Luo Renbo**, Pi Youguo, "Supervised neighborhood preserving embedding feature extraction of hyperspectral imagery. Acta Geodaetica et Cartographica Sinica". 2014; 43(5): 508-513.(A2)

7. **Luo Renbo**, Pi Youguo, "GPU-based parallel kernel PCA feature extraction for hyperspectral images". International Conference on Remote Sensing and Wireless Communications (RSWC 2014). 2014. (P1)

8. **Luo Renbo**, Liao Wenzhi, Pi Youguo, "Discriminative supervised neighborhood preserving embedding feature extraction for hyperspectral-image classification". Telkomnika. 2012; 10(5): 1051-1056. (A2)

9. **Luo Renbo**, Liao Wenzhi, Pi Youguo, "Research on supervised LPP feature extraction for hyperspectral image". Remote Sensing Technology and Application. 2012; 27(6): 46-52. (A2)

10. Zhang Hongyan, He Wei, Liao Wenzhi, **Luo Renbo**, Zhang Liangpei, Pizurica Aleksandra, "Exploiting the low-rank property of hyperpsectral imagery: a technical overview". Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote sensing (WHISPERS 2016). Aug. 2016. (C1)

11. Liao Wenzhi, Pizurica Aleksandra, **Luo Renbo**, Philips Wilfried. "A comparison on multiple level features for fusion hyperspectral and LiDAR data". Joint Urban Remote Sensing Event, Proceedings (JURSE 2017), Mar. 2017. (C1)

12. Liao Wenzhi, Zhang Hongyan, Li Jie, Huang Shaoguang, Wang Rui, **Luo Renbo**, Pizurica Aleksandra, "Fusion of Spectral And Spatial Information for Land Cover Classification". IEICE Information and Communication Technology Forum (ICTF2016), 2016. (C1)

1.4 Structure of the Thesis

This rest of this dissertation is organized as follows.

Chapter 2 deals with supervised feature extraction algorithms. In particular, two proposed methods, discriminative supervised neighborhood preserving embedding (DSNPE) and PCA-based supervised locality preserving projection (PSLPP), will be presented.

Chapter 3 focuses on semi-supervised feature extraction methods. Three semi-supervised feature extraction methods for classification of hyperspectral remote sensing imagery are presented, including improved semi-supervised local discriminant analysis (ISELD), semi-supervised graph learning (SEGL) method and its implementation on fusing the spectral and spatial information.

In Chapter 4, we propose a new framework to fuse hyperspectral and LiDAR images for the classification of the cloud-shadow mixed remote sensing scenes. Experimental results on real remote sensing data are presented to demonstrate its efficiency.

Chapter 5 explains the GPU implementation of a non-linear feature extraction method. Experimental analysis and results demonstrate that the GPU based non-linear feature extraction method can speed up more than 100 times compared conventional CPU based methods.

Chapter 6 presents a general discussion of the work described in this thesis reviewing the main contributions of this research and presents perspectives on possible future developments of the work.

2

Supervised Feature Extraction of Remote Sensing Data

This chapter focus on developing supervised techniques to extract interesting and useful features for reliable classification.

In processing of remote sensing imagery, feature extraction is a necessary pre-processing step to produce a small number of informative features for classification. Two typical unsupervised feature extraction methods, neighborhood preserving embedding (NPE) and locality preserving projections (LPP), are wildly used in hyperspectral image processing recently. However, as they are essentially unsupervised, the label information have not been well used. In this Chapter, we will propose two new supervised feature extraction algorithms (integrating label information into unsupervised NPE and LPP) for classification of hyperspectral remote sensing imagery. By taking the label information into account, supervised feature extraction methods generally can generate better features for class discrimination than unsupervised methods.

The first proposed supervised feature extraction method is called discriminative supervised neighborhood preserving embedding (DSNPE), which incorporates the label information into a linear feature extraction approach named neighborhood preserving embedding (NPE). DSNPE aims at pulling the neighboring points with the same class label towards each other as near as possible, while simultaneously pushing the neighboring points with different labels away from each other as far as possible. What's more, similar to NPE, DSNPE can preserve the local manifold structure and the neighborhood structure.

The other proposed method is a PCA-based supervised locality preserving projection (PSLPP), which combines principle component analysis (PCA) and label information with LPP. In the proposed PSLPP method, two similarity matrices are first calculated, the similarity matrix represents the correlations between samples, the element corresponding to two samples in similarity matrix will be set to a weight between 0 and 1 based on their similarity in spectral

features or other information. The first similarity matrix is calculated based on the neighbors information in low-dimensional spectral space which is transformed by PCA to remove the noisy and redundancy. The second one is obtained based on training samples labels information. By exploiting above two similarity matrices, we can better quantify how likely it is that neighboring data points belong to the same or a different class, and find an optimal transformation matrix to project high-dimensional hyperspecral image to a lower dimensional subspace.

2.1 Introduction

To mitigate the small sample size (SSS) problem, which states the number of available training samples is relatively much smaller than the dimensionality of the sample space and leads Hughes phenomenon (for a limited number of training samples, the classification accuracy decreases as the dimension increases) [Hughes 68], feature extraction is usually an important preprocessing step for most hyperspectral image analysis. Feature extraction methods are developed to reduce the dimensionality of hyperspectral remote sensing image while keeping as much intrinsic information as possible: relatively few bands can represent most information of the hyperspectral data [Fong 07]. Even though the extracted features are then no longer directly related to physical material properties, they provide a compressed version of the original complete set of spectral bands, which still contains all required information to classify the data. Each band of the original hyperspectral data often contributes to the extracted low dimensional features. How much exactly is determined by the transformation matrix associated with a given feature extraction method.

Determination of how to transform original high-dimensional bands to a low-dimensional feature space (feature extraction) is the key issue in this Chapter. The techniques for feature extraction presented below can be categorized as unsupervised (global data oriented) and supervised (class data oriented) methods.

2.1.1 Unsupervised Feature Extraction Methods

Unsupervised feature extraction methods do not require any prior knowledge on ground truth (e.g., human annotations) for training data, whereas supervised feature extraction methods rely on labelled training data [Jia 13]. Two typical unsupervised feature extraction methods are principle component analysis (PCA) [Hotelling 33] and independent component analysis (ICA) [Hyvarinen 00]. As one of the best known unsupervised methods and widely used for hyperspectral images [Fong 07, Plaza 05], PCA tries to extract the features which are linear combinations of the input data. The number of extracted features (eigenvectors) and the coefficients (eigenvalues) in the linear combinations are computed based on analyzing the covariance matrix of the original training data [Jlliffe 86, Zubko 07]. The eigenvalues of the covariance matrix

are considered to be an indicator of the information content. Large values correspond to features with a large variance, which suggests a large information content; low values are considered to be mostly noise and therefore not very informative. Due to its low complexity and the absence of parameters, PCA has also been widely used other areas, as face recognition, data mining, and so on.

ICA is a statistical technique for separating independent signals from signal mixtures [Hyvarinen 00]. Compared with PCA, ICA is more powerful at finding the underlying factors or sources in cases where PCA fails. ICA defines a generative model for the observed multivariate data. In the model, the data variables are still assumed to be linear mixtures of some unknown latent variables, with the mixing system also being unknown. However, the latent variables are assumed to be non-Gaussian and mutually statistically independent [Hyvarinen 00]. Recently, Wang and Chang [Wang 06] proposed three ICA-based dimensionality reduction methods for hyperspectral data. Their experimental results have shown that their methods perform better than PCA, as there is no prioritization among components generated by the ICA due to the use of random initial projection vectors. Marchesi and Bruzzone applied ICA and kernel ICA for change detection in multitemporal remote sensing images [Marchesi 09]. In [Palmason 05], PCA/ICA and morphological transformations had been combined for the classification of hyperspectral images of urban areas.

2.1.2 Supervised Feature Extraction Methods

Supervised methods mainly rely on labelled samples to learn about class separability; using these labels, they can learn more discriminative features than unsupervised methods. Two widely used supervised feature extraction methods for hyperspectral images are Fisher's linear discriminant analysis (LDA) [Fukunaga 90] and nonparametric weighted feature extraction (NWFE) [Kuo 04]. LDA is a traditional parametric feature extraction technique that is based on the mean vector and covariance matrix of each (labelled) class. The ratio of within-class to between-class scatter matrices is used to formulate an effective criterion for class separability [Fukunaga 90]. The inherent limitations of LDA include its dependence on the distributions of classes being approximately Gaussian and its inability to handle cases where class data does not form a single cluster. When the distributions of classes are non normal like or multi-modal mixture distributions, the performance of LDA is not satisfactory. Furthermore, the maximum rank of the between-class scatter matrix is the number of classes (C) minus one, thus only a maximum of C-1 features can be extracted by LDA. Actually, using only C-1 features may not be sufficient for classification of hyperspectral data [Kuo 04], as the data distributions are often complicated (C-1 dimensional space is not enough to present) and not normal-like. Last but not least, LDA performs poorly when the within-class covariance is singular. This frequently occurs in high-dimensional but small sample size (SSS) classification problems [Raudys 91, Yang 10]. What's more,

when the number of training samples is small compared to the feature dimensionality, the estimates of second-order statistics may not be reliable at the class level, and thus the extracted features may not perform well [Kuo 04] for post applications.

In order to address the limitation of LDA (maximum C-1 features), Fukunaga *et al.* [Fukunaga 83] proposed nonparametric discriminant analysis(NDA), NDA defines a nonparametric between-class scatter matrix based on a critical finding that data points closer to the boundary between two classes in feature space are more important to learn proper classifiers than those far from the boundary. As a result, each sample should be given a distinct weight when extracting informative features. Nonparametric weighted feature extraction (NWFE) was proposed by Kuo [Kuo 04] in 2004. The main ideas of NWFE are putting different weights on every sample to compute the weighted center (feature vector) of each class first, and then compute the distance between samples and their weighted centers as a measure related to "closeness" to the boundary, see Figure 2.1. After that, the authors define nonparametric between-class and within-class scatter matrices, which put large weights on the samples close to the boundary and emphasize those samples far from the boundary, to obtain more than C-1 features. NWFE is developed in light of nonparametric discriminant analysis(NDA) [Fukunaga 83], introducing regularization techniques to achieve better performance for hyperspectral image classification than LDA and NDA [Kuo 07].

Some extensions to both LDA and NWFE have been proposed in recent years, such as modified Fisher's linear discriminant analysis [Kuo 07], regularized linear discriminant analysis [Bandos 09], modified nonparametric weight feature extraction using spatial and spectral information [Kuo 04]. With developments of kernel-based methods (which are based on mapping data from the original input feature space to a kernel feature space of higher dimensionality, and then solving a linear problem in that space [Camps-Valls 05]), some typical linear feature extraction methods are extended to nonlinear feature extraction. These kernel-based methods use a suitable kernel function to transform data, and increase the class separability in the kernel space (which is nonlinearly related to the input space). For instance, generalized discriminant analysis (GDA) [Baudat 00] and kernel local Fisher discriminant analysis (KLFDA) [Bandos 09] are nonlinear extensions of LDA. Kernel nonparametric weighted feature extraction (KNWFE) [Kuo 09] is the nonlinear version of NWFE. Similar as NWFE, cosine-based nonparametric feature extraction (CNFE), which employs cosine distance (which measures the cosine of the angle between two non zero vectors of an inner product space) to measure similarity instead of the Euclidean distance used by NDA and NWFE, was proposed by Yang *et al.* in 2010 [Yang 10]. Additionally, Huang and Kuo [Huang 10] proposed a double nearest proportion (DNP) feature extraction by constructing new scatter matrices based on a double nearest proportion structure. Double nearest proportion (DNP) method can better reduce the effect of overlap and emphasize the separability of class boundaries even when overlap occurs. De-

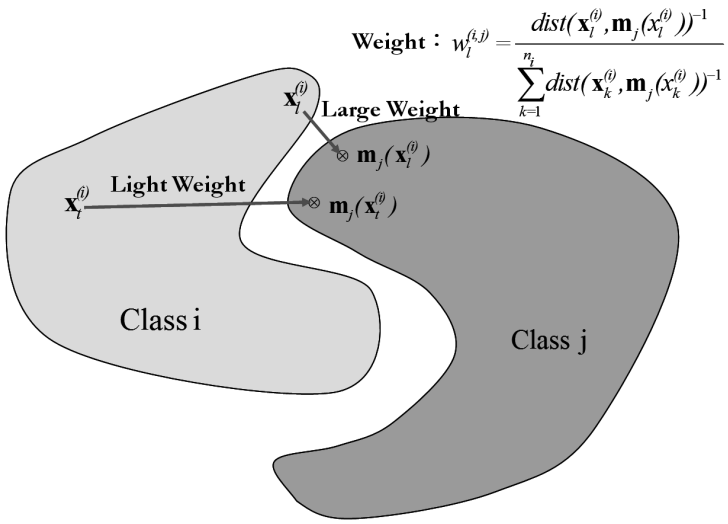


Figure 2.1: The idea of nonparametric weighted feature extraction (NWFE), $\mathbf{x}_l^{(i)}$ is lth sample from class i, $\mathbf{m}_j(\mathbf{x}_l^{(i)})$ demotes the mean of k nearest neighbors of $\mathbf{x}_l^{(i)}$ in class j, $dist(\cdot)$ is Euclidean distance. If $dist(\mathbf{x}_l^{(i)}, \mathbf{m}_j(\mathbf{x}_l^{(i)}))$ is small, then $\mathbf{x}_l^{(i)}$ is considered to be more close to the class boundary and gains large weight $w_l^{i,j}$ ($w_l^{i,j}$ is useful for calculating scatter matrix).

cision boundary feature extraction (DBFE) [Landgrebe 03], an early method proposed by Lee and Landgrebe specifically for feature extraction of hyperspectral images, seeks to find new features which are normal to class decision boundaries. It can extract both discriminate informative features and discriminate redundant features from the decision boundary. The approach uses the training samples directly to determine the location of the decision boundary and employs information about the decision hypersurfaces associated with a given classifier to define an intrinsic dimensionality for the classification problem. Then, the corresponding optimal linear mapping can be obtained. The goal of these supervised feature extraction methods (LDA, NWFE and their extensions GDA, DNP and DBFE) is to find a linear transformation that maximizes the between class scatter and minimizes the within class scatter [Jia 13]. The only difference between them is the different definition of within- and between-class scatter matrices, their general form can be represented as:

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmax}}\, tr(\mathbf{W}^T\mathbf{S}_w\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{S}_b\mathbf{W}) \tag{2.1}$$

$\mathbf{S}_w$ and $\mathbf{S}_b$ represent within- and between-class scatter matrices, the columns of $\mathbf{W}$ are the optimal features by optimizing above Fisher criterion. The projection $\mathbf{z}_i$ of an unknown sample $\mathbf{x}_i$ can then be evaluated by

$$\mathbf{z} = \mathbf{W}^T \mathbf{x}. \tag{2.2}$$

In order make the definition of these supervised methods be more clear, an example (Figure 2.2) has been drawn to show the different projections between PCA and these supervised methods (here take LDA as an example).

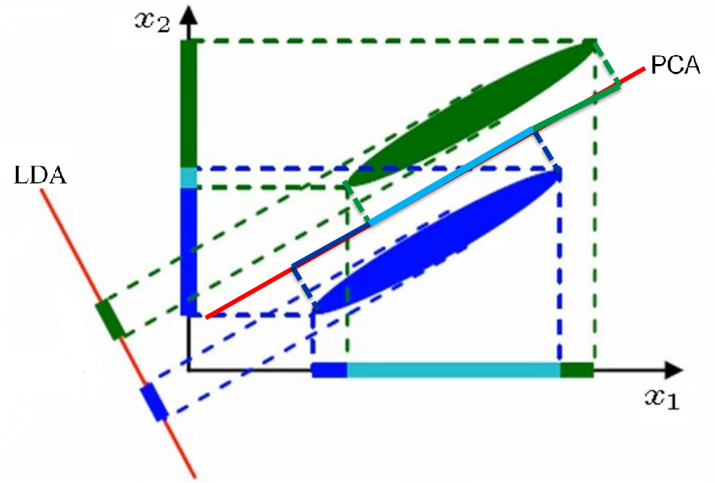


Figure 2.2: Different projections between PCA and LDA. In this example, PCA and LDA project all points with two dimensions into one dimension (line), PCA will find a project direction with largest scatter, while LDA will project samples with same labels close and samples with different labels faraway.

2.1.3 Proposed Supervised Local Feature Extraction Methods

Besides the supervised methods, several unsupervised local methods, which preserve the properties of local neighborhoods also have been proposed recently to reduce the dimensionality of hyperspectral images [Fong 07, Fang 14], such as Laplacian eigenmaps(LE) [Belkin 02] and locally linear embedding (LLE) [Roweis 00], their linearisation of locality preserving projections (LPP) [He 04] and neighborhood preserving embedding (NPE) [He 05]. By considering local geometrical structure information (as nearest neighbors), these local methods can preserve local neighborhood information and detect the manifold structure of data in the high-dimensional feature space, see Figure 2.3.

LPP aims to seek optimal projections where nearby points in the original high-dimensional space are likely to have similar projections in the low-dimensional feature space. Therefore, LPP preserves important local information of the original data in the low-dimensional representation. NPE focus on the preservation of the local manifold structure. Specifically, for each data

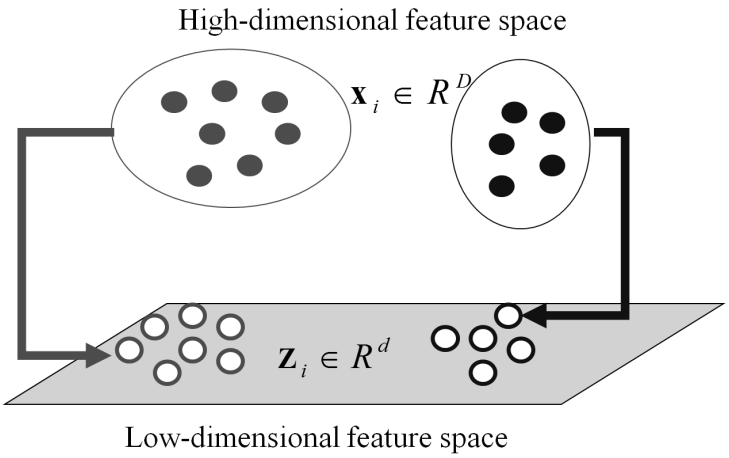


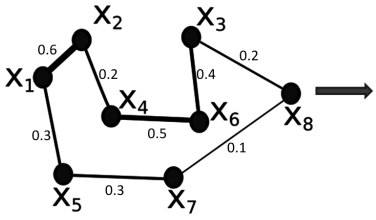
Figure 2.3: The main property of local methods, $\mathbf{x}_i$ is a point in high-dimensional feature space, $\mathbf{z}_i$ is the projection of $\mathbf{x}_i$ in low-dimensional feature space, local geometrical structure information will be kept after projection.

point, it is represented as a linear combination of the neighboring data points and the combination coefficients are specified in the weight matrix. We then find an optimal embedding such that the neighborhood structure can be preserved in the dimensionality reduced space.

LPP and NPE can be summarized as graph-based unsupervised feature extraction methods. Graph-based methods start by composing a graph where the nodes are points in high-dimensional space, and (weighted) edges reflect the similarity of nodes. The assumption is that nodes connected by a large-weight edge tend to belong to same class. The mathematical representation of graph is similarity matrix, thus can be seen from Figure 2.4. Both LPP and NPE try to build a graph to presents the corrections between samples, while with different criterion, more details will be discussed in following section.

However, as unsupervised feature extraction methods, both LPP and NPE do not make use of label information of samples. As a result they ignore the differences among neighbors which belong either same or different classes. To complement the shortages of LPP and NPE, we propose two supervised feature extraction methods by combining label information with them in this Chapter.

To enhance the classification performance of NPE, we propose a new algorithm termed discriminative supervised neighborhood preserving embedding (DSNPE) in section 2.2. NPE assumes that each data point could be represented as a linear combination of the neighboring data points, and the linear combination coefficients can be seen as the correlations between this data point with its nearest neighbors. When high-dimensional points are embedded into low-dimensional feature space, the correlations between points will be kept. Different from NPE, DSNPE first divide the nearest neighbors into two group:



(a) Graph

	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$	$\mathbf{X}_5$	$\mathbf{X}_6$	$\mathbf{X}_7$	$\mathbf{X}_8$
$\mathbf{X}_1$		0.6	0	0	0.3	0	0	0
$\mathbf{X}_2$	0.6		0	0.2	0	0	0	0
$\mathbf{X}_3$	0	0		0	0	0.4	0	0.2
$\mathbf{X}_4$	0	0.2	0		0	0.5	0	0
$\mathbf{X}_5$	0.3	0	0	0		0	0.3	0
$\mathbf{X}_6$	0	0	0.4	0.5	0		0	0
$\mathbf{X}_7$	0	0	0	0	0.3	0		0.1
$\mathbf{X}_8$	0	0	0.2	0	0	0	0.1	

(b) Similarity matrix

Figure 2.4: The graph and its mathematical representation similarity matrix, here we take eight points as an example, every point has two connected nearest neighbors, bolder line means more similar.

neighbors with same labels as center point and with different labels. Then each data point is represented as a linear combination of the neighboring data points from same class instead. At the same time, DSNPE attempts to separate neighbors from different classes as much as possible in the low-dimensional feature space. DSNPE offers three main benefits: 1) the algorithm takes into account both intraclass and interclass geometries so that it can achieve better performances in classification; 2) discriminability is effectively preserved in the algorithm because it takes into account label information of neighboring samples; and 3) in the subspace, our proposed method can project the neighboring samples from the same class nearer, while project neighboring samples with different labels further.

In the section 2.3, a PCA-based supervised locality preserving projection (PSLPP) feature extraction method will be proposed. In the proposed PSLPP method, PCA is first performed on the original high-dimensional data points to remove the noisy and redundancy. In order to better model the relationships of data points, we proposed a similarity matrix which contains the label and local neighborhood information. By exploiting the proposed similarity matrix, we get a transformation matrix to project high-dimensional HS image to a lower dimensional subspace.

2.2 Discriminative Supervised Neighborhood Preserving Embedding (DSNPE)

2.2.1 Neighborhood Preserving Embedding (NPE)

In this section, we will introduce a linear dimensionality reduction algorithm NPE. The generic problem of linear dimensionality reduction is the following: given a set of points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \in \mathbf{R}^D$, find a transformation matrix $\mathbf{W}$

that maps these n points to a set of points $\{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n\} \in \mathbf{R}^d$,$d << D$ by $\mathbf{z} = \mathbf{W}^T\mathbf{x}$.

NPE preserves local manifold structure when data points are projected from a high-dimensional feature space to a low-dimensional feature space. Provided there is sufficient data (such that the manifold is well-sampled), NPE expects each data point and its neighbors to lie on or close to a locally linear patch of the manifold. We characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. Reconstruction errors are measured by the cost function as follows:

$$\min \sum_{i}^{n} ||\mathbf{x}_i - \sum_{j}^{e} \mathbf{a}_{ij}\mathbf{x}_j||^2 \tag{2.3}$$

$$\sum_{j}^{e} \mathbf{a}_{ij} = 1.$$

which adds up the squared distances between all the data points and their reconstructions. The weights $\mathbf{a}_{ij}$ summarize the contribution of the jth data point to the ith reconstruction. Let $\mathbf{G}$ denote a graph with n nodes, the data point $\mathbf{x}_i$ corresponds to ith node. We add a directed edge from node $\mathbf{x}_i$ to $\mathbf{x}_j$ in the graph if $\mathbf{x}_j$ is one of the e nearest neighbors of $\mathbf{x}_i$, see Figure 2.4. Let $\mathbf{A}$ denote the weight matrix composed by $\mathbf{a}_{ij}$. To compute the weights $\mathbf{a}_{ij}$, we minimize the cost function subject to two constraints: first, that each data point $\mathbf{x}_i$ is reconstructed only from its neighbors, enforcing $\mathbf{a}_{ij} = 0$ if $\mathbf{x}_j$ does not belong to the set of neighbors of $\mathbf{x}_i$; second, that the rows of the weight matrix sum to one. Please see [Roweis 00] for the details about how to solve the above minimization problem.

If the original data points are mapped to a line, then, each data point on the line can be represented as a linear combination of its neighbors with the coefficients $\mathbf{a}_{ij} = 0$. Let $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n)^T$ be such a map, n denotes the total number of training samples. A reasonable criterion for choosing a map is to minimize the following cost function:

$$\min \sum_{i}^{n} ||\mathbf{z}_i - \sum_{j}^{e} \mathbf{a}_{ij}\mathbf{z}_j||^2 \tag{2.4}$$

This cost function, like the previous one, is based on locally linear reconstruction errors, but here we fix the weights $\mathbf{a}_{ij}$ [He 05] while optimizing the coordinates $\mathbf{z}_i$. In other words, the cost function seek an optimal transformation $\mathbf{z} = \mathbf{W}^T\mathbf{x}$ to minimize the combination error.

Following some algebraic formulations, the cost function can be expressed

as:

$$\begin{aligned}&\sum_{i}^{n}||\mathbf{z}_i-\sum_{j}^{e}\mathbf{a}_{ij}\mathbf{z}_j||^2\\=&\mathbf{z}^T(\mathbf{I}-\mathbf{A})^T(\mathbf{I}-\mathbf{A})\mathbf{z}\\=&\mathbf{w}^T\mathbf{X}(\mathbf{I}-\mathbf{A})^T(\mathbf{I}-\mathbf{A})\mathbf{X}^T\mathbf{w}\\=&\mathbf{w}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{w},\end{aligned}\tag{2.5}$$

where

$$\begin{aligned}\mathbf{X}&=\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n\}\\\mathbf{M}&=(\mathbf{I}-\mathbf{A})^T(\mathbf{I}-\mathbf{A})\\\mathbf{I}&=diag(1,1,\cdots,1)\end{aligned}$$

As $\mathbf{A}\neq\mathbf{I}$ (every point is linear combination of its k nearest neighbors, not include itself), it is easy to check that $\mathbf{M}$ is symmetric and semipositive definite.

In order to remove an arbitrary scaling factor in the projection, $\mathbf{z}_i$ is imposed an unit vector as:

$$\mathbf{z}^T\mathbf{z}=\mathbf{I}\rightarrow\mathbf{w}^T\mathbf{X}\mathbf{X}^T\mathbf{w}=\mathbf{I}\tag{2.6}$$

By combining equation (2.3) and (2.6), the transform matrix $\mathbf{W}$ can be obtained by solving the following cost function:

$$\mathbf{w}_{NPE}=\arg\min_{\mathbf{w}}\frac{\mathbf{w}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{w}}{\mathbf{w}^T\mathbf{X}\mathbf{X}^T\mathbf{w}}\tag{2.7}$$

In order to solve equation (2.7), it can be transferred as generalized eigenvector problem:

$$\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{w}=\lambda\mathbf{X}\mathbf{X}^T\mathbf{w}\tag{2.8}$$

The vectors $\mathbf{w}_1,\mathbf{w}_2,\cdots,\mathbf{w}_d$ which solve equation (2.8) are the columns of $\mathbf{W}_{NPE}$, as $\mathbf{W}_{NPE}=[\mathbf{w}_1,\mathbf{w}_2,\cdots,\mathbf{w}_d]$. The columns are ordered according to their eigenvalues $\lambda_1<\lambda_2<\cdots<\lambda_d$. Thus, the embedding is as follows:

$$\mathbf{x}_i\rightarrow\mathbf{z}_i=\mathbf{W}_{NPE}^T\mathbf{x}_i$$

2.2.2 Proposed DSNPE

As NPE is an unsupervised feature extraction methods, neglecting label information of training samples. For a data points $\mathbf{x}_i$, its nearest neighbors may contain points from other class, if we use those data points to linearly reconstruct $\mathbf{x}_i$, the reconstruction error will be very large, which will have a big influence for the cost function (as equation (2.3)) and its solution. Therefore, we proposed a supervised methods called DSNPE, which linearly reconstructs every data points with nearest neighbors from same class and pushes nearest neighbors from other classes away. By doing this, DSNPE can find a better transform matrix and extract more discriminative features than NPE. The idea of DSNPE is shown in Figure 2.5

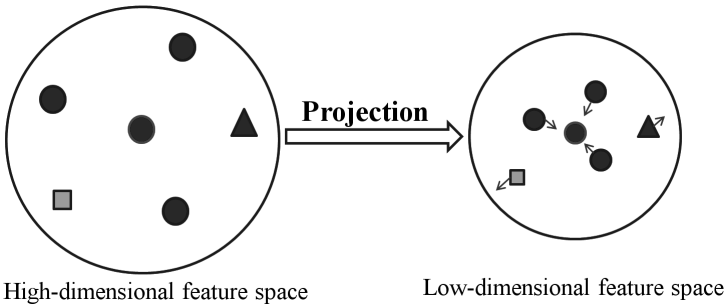


Figure 2.5: The idea of DSNPE: projecting the neighboring samples from the same class nearer, while projecting neighboring samples with different labels further.

Suppose $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{il_i}, \mathbf{x}_{i(l_i+1)}, \mathbf{x}_{i(l_i+2)}, \cdots, \mathbf{x}_{i(l_i+c_i)}\}$ are the e labelled nearest neighbors of $\mathbf{x}_i$. Some of these will have the same label as $\mathbf{x}_i$. We assume that the indexing is such that the neighbors with the same label are $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{il_i}\}$ whereas the neighbors from other class are $\{\mathbf{x}_{i(l_i+1)}, \mathbf{x}_{i(l_i+2)}, \cdots, \mathbf{x}_{i(l_i+c_i)}\}$, here $e = l_i + c_i$.

In contrast to NPE, DSNPE assumes that each sample can be reconstructed from samples having the same label only. That is, $\mathbf{x}_i$ can be linearly reconstructed from $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{il_i}\}$ as:

$$\mathbf{x}_i = a_{i1}\mathbf{x}_{i1} + a_{i2}\mathbf{x}_{i2} + \cdots + a_{il_i}\mathbf{x}_{il_i} + \varepsilon_i \tag{2.9}$$

where ε_i is the reconstruction error. Minimizing the error yields

$$\arg\min_{\mathbf{a}_i} \sum_i ||\varepsilon_i||^2 = \min \sum_i ||\mathbf{x}_i - \sum_{j=1}^{l_i} a_{ij}\mathbf{x}_{ij}||^2 \tag{2.10}$$

Same as the introduction of NPE, if the original data points are projected to a line so that each data point on the line can be represented as a linear combination of its neighbors with the coefficients a_{ij}. Let $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n\}$ be such a transformed points of $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, and assumes that $\mathbf{a}_i$ reconstructs both $\mathbf{x}_i$ from $(\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{il_i})$ in the high-dimensional space and $\mathbf{z}_i$ from $(\mathbf{z}_{i1}, \mathbf{z}_{i2}, \cdots, \mathbf{z}_{il_i})$ in the low-dimensional space (preserving local geometrical structure information). Then the equation (2.10) is transferred as:

$$\min \sum_i ||\mathbf{x}_i - \sum_j^{l_i} \mathbf{a}_{ij}\mathbf{x}_j||^2 \to \min \sum_i ||\mathbf{z}_i - \sum_j^{l_i} \mathbf{a}_{ij}\mathbf{z}_j||^2 \tag{2.11}$$

where $\mathbf{a}_i = \{\mathbf{a}_{i1}, \mathbf{a}_{i2}, \cdots, \mathbf{a}_{il_i}\}$, and $\sum_{j=1}^{l} a_{ij} = 1$. In order to optimize the coordinates $\mathbf{z}_i$, we fix the weights $\mathbf{a}_{ij}$ in a closed form as:

$$a_{ij} = \frac{d(\mathbf{x}_i, \mathbf{x}_{ij})^{-1}}{\sum_{t=1}^{l_i} d(\mathbf{x}_i, \mathbf{x}_{it})^{-1}}.$$

where $d(\mathbf{x}_i, \mathbf{x}_{it})$ denotes the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_{it}$. If the distance between $\mathbf{x}_i$ and $\mathbf{x}_{it}$ is small, this means $\mathbf{x}_{it}$ is very similar as $\mathbf{x}_i$, then its reconstruction coefficient will be close to 1; otherwise, it will be close to 0. For the other points, which do not belong to l_i nearest neighbors of $\mathbf{x}_i$, their weights are set to 0.

At the same time, we would like that neighboring samples with different labels $(\mathbf{x}_{i(l_i+1)}, \mathbf{x}_{i(l_i+2)}, \cdots, \mathbf{x}_{i(l_i+c_i)})$ are far from the given sample $\mathbf{x}_i$. To make this happen, we can maximize the sum of the distances between $\mathbf{z}_i$ and $\{\mathbf{z}_{i(l_i+1)}, \mathbf{z}_{i(l_i+2)}, \cdots, \mathbf{z}_{i(l_i+c_i)}\}$ in the low-dimensional feature space, as a result we have a maximum cost function as following:

$$\max \sum_i \sum_{t=1}^{c_i} ||\mathbf{z}_i - \mathbf{z}_{it}||^2 \tag{2.12}$$

Combining the embedding framework equation (2.11) and (2.12), the optimization problem can be resolved by converting them into the following ratio problem:

$$\min \frac{\sum_i ||\mathbf{z}_i - \sum_j^{l_i} \mathbf{a}_{ij}\mathbf{z}_j||^2}{\sum_i \sum_{t=1}^{c_i} ||\mathbf{z}_i - \mathbf{z}_{it}||^2} \tag{2.13}$$

We then look for a linear transform $\mathbf{z} = \mathbf{w}^T\mathbf{X}$, which optimizes this criterion, where the ith column vector of $\mathbf{X}$ is $\mathbf{x}_i$, and assumes that

$$\begin{aligned}
&2\frac{\sum_i ||\mathbf{z}_i - \sum_j^{l_i} \mathbf{a}_{ij}\mathbf{z}_j||^2}{\sum_i \sum_{t=1}^{c_i} ||\mathbf{z}_i - \mathbf{z}_{it}||^2} \\
=&\frac{\sum_i ||\mathbf{z}_i - \sum_j^{l_i} \mathbf{a}_{ij}\mathbf{z}_j||^2}{\frac{1}{2}\sum_i \sum_{t=1}^{c_i} ||\mathbf{z}_i - \mathbf{z}_{it}||^2} \\
=&\frac{\mathbf{z}^T(\mathbf{I}-\mathbf{A})^T(\mathbf{I}-\mathbf{A})\mathbf{z}}{\mathbf{z}^T(\mathbf{D}-\mathbf{B})\mathbf{z}} \\
=&\frac{\mathbf{w}^T\mathbf{X}(\mathbf{I}-\mathbf{A})^T(\mathbf{I}-\mathbf{A})\mathbf{X}^T\mathbf{w}}{\mathbf{w}^T\mathbf{X}(\mathbf{D}-\mathbf{B})\mathbf{X}^T\mathbf{w}} \\
=&\frac{\mathbf{w}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{w}}{\mathbf{w}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w}}
\end{aligned} \tag{2.14}$$

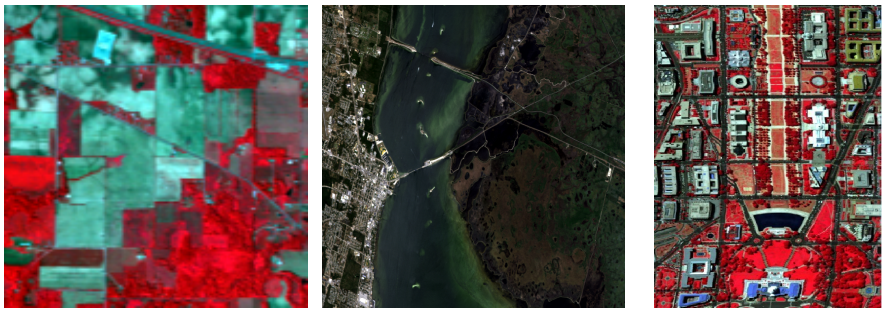
$$B_{ij} = \begin{cases} 1 & , \mathbf{x}_j \in knn(\mathbf{x}_i) \ \& \ y_i \neq y_j \\ 0 & , \text{otherwise} \end{cases}$$

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$$

$$\mathbf{M} = (\mathbf{I}-\mathbf{A})^T(\mathbf{I}-\mathbf{A})$$

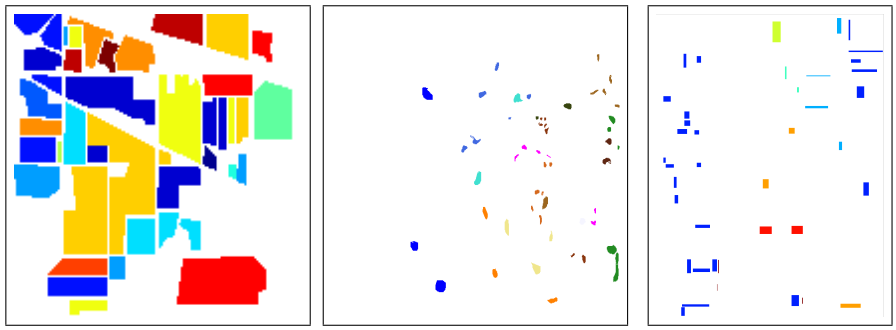
$$\mathbf{I} = diag(1, 1, \cdots, 1),$$

where $\mathbf{D}$ is a diagonal matrix; its entries are column(or row) sum of $\mathbf{B}$, $D_{ii} = \sum_j B_{ij}$, y_i denotes the label of $\mathbf{x}_i$, $knn(\mathbf{x}_i)$ is a set including e nearest neighbors



(a) Indian Pine **(b)** KSC **(c)** DC mall

Figure 2.6: False colour image (three bands were selected from original data sets and act as R,G,B) of experimental data sets.



(a) Indian Pine **(b)** KSC **(c)** DC mall

Figure 2.7: Groundtruth of experimental data sets.

of point $\mathbf{x}_i$. We can now reformulate the cost function 2.13 as follows:

$$\mathbf{w}_{DSNPE} = \arg\min_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{X}\mathbf{M}\mathbf{X}^T \mathbf{w}}{\mathbf{w}^T \mathbf{X}\mathbf{L}\mathbf{X}^T \mathbf{w}} \tag{2.15}$$

The transformation vector $\mathbf{w}$ that minimizes the objective function is obtained by solving a generalized eigenvalue problem:

$$\mathbf{X}\mathbf{M}\mathbf{X}^T \mathbf{w} = \lambda \mathbf{X}\mathbf{L}\mathbf{X}^T \mathbf{w} \tag{2.16}$$

Let the column vectors $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d$ be the solutions of equation (2.16) according to eigenvalues $\lambda_1 < \lambda_2 < \cdots < \lambda_d$. The optimal projection matrix $\mathbf{W}_{DSNPE}$ is given by $\mathbf{W}_{DSNPE} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d]$.

Table 2.1: Data Sets Used In The Experiments

No	India Pine		KSC		DC mall	
	Class name	Samples	Class name	Samples	Class name	Samples
1	Corn-no till	1434	Scrub	761	Roof	3834
2	Corn-min till	834	Willow swamp	243	Street	416
3	Corn	234	Cabbage palm hammock	256	Path	175
4	Soybeans-no till	968	Cabbage palm/oak hammock	252	Grass	1928
5	Soybeans-min till	2468	Slash Pine	161	Trees	405
6	Soybeans-clean till	614	Oak/broadleaf hammock	229	Water	1224
7	Alfalfa	54	Hardwood swamp	105	Shadow	97
8	Grass/pasture	497	Graminoid marse	431		
9	Grass/trees	747	Spartan marse	520		
10	Grass/moved	26	Cattail marse	404		
11	Hay-windrowed	489	Salt marse	419		
12	Oats	20	Mud flats	503		
13	Wheat	212	Water	927		
14	Woods	1294				
15	Bldg-Grass-Trees	380				
16	Stone-steel towers	95				
Total		10366		5211		8079

Table 2.2: Overall classification accuracy (OA%) and its corresponding number of extracted features (in brackets) of different feature extraction methods with different training sample size.

Data set	$\mathbf{n}_i$		PCA	NWFE	LPP	NPE	DSNPE
India Pine	20	OA	64.6(11)	**68.1(13)**	53.1(19)	63.1(17)	66.1(17)
		κ	0.598	**0.644**	0.481	0.584	0.615
	40	OA	69.1(13)	73.8(11)	64.1(14)	70.5(15)	**75.0(11)**
		κ	0.655	0.704	0.602	0.670	**0.718**
	100	OA	76.2(16)	82.9(11)	76.0(18)	77.8(17)	**84.3(11)**
		κ	0.734	0.810	0.731	0.751	**0.824**
KSC	20	OA	**69.6(17)**	67.7(9)	55.1(17)	54.8(15)	68.9(15)
		κ	**0.677**	0.658	0.515	0.527	0.673
	40	OA	80.5(17)	82.6(12)	68.0(16)	79.3(17)	**83.9(15)**
		κ	0.723	0.805	0.657	0.771	**0.819**
	100	OA	82.4(18)	87.4(12)	80.2(19)	84.3(16)	**89.4(15)**
		κ	0.805	0.860	0.785	0.831	**0.881**
DC mall	20	OA	77.5(9)	81.9(4)	64.4(11)	81.3(7)	**82.2(5)**
		κ	0.661	0.718	0.512	0.695	**0.720**
	40	OA	86.4(9)	87.5(6)	81.7(10)	87.2(6)	**88.9(5)**
		κ	0.792	0.799	0.698	0.798	**0.807**
	100	OA	89.8(10)	94.0(6)	86.5(11)	92.1(7)	**95.7(7)**
		κ	0.815	0.886	0.791	0.856	**0.904**

2.2.3 Experiments

2.2.3.1 Experimental Data sets and settings

Experimental Data sets: three hyperspectral images, namely Indian Pine, KSC and Washington DC Mall, are adopted in the experiments for evaluating the performance of the proposed method. Table 2.1 shows the number of labelled samples in each class for all the data sets. Note that the different colors in the cell denotes different classes in the classification maps.

Indian Pine: the Indian Pine hyperspectral data set was acquired by using the National Aeronautics and Space Administration's Airborne Visible Infrared Imaging Spectrometer (AVIRIS), mounted on an aircraft flown at 65000 ft altitude. It contains 145×145 pixels. Each pixel has 220 spectral bands, and the corresponding spatial resolution is approximately 20m. The false color image which composed by three bands composition of original hy-

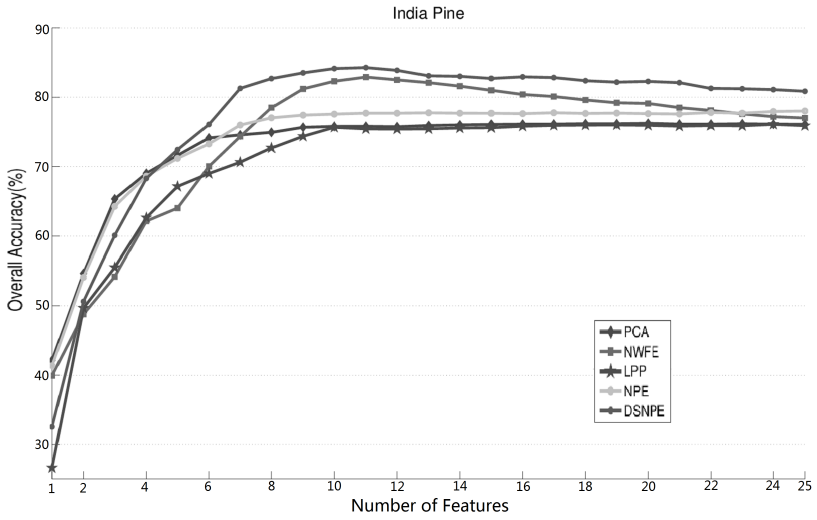


Figure 2.8: Average overall classification accuracy (OA%) with the number of extracted features increasing for different methods for Indian Pine. 100 labelled training samples are chosen randomly from each class.

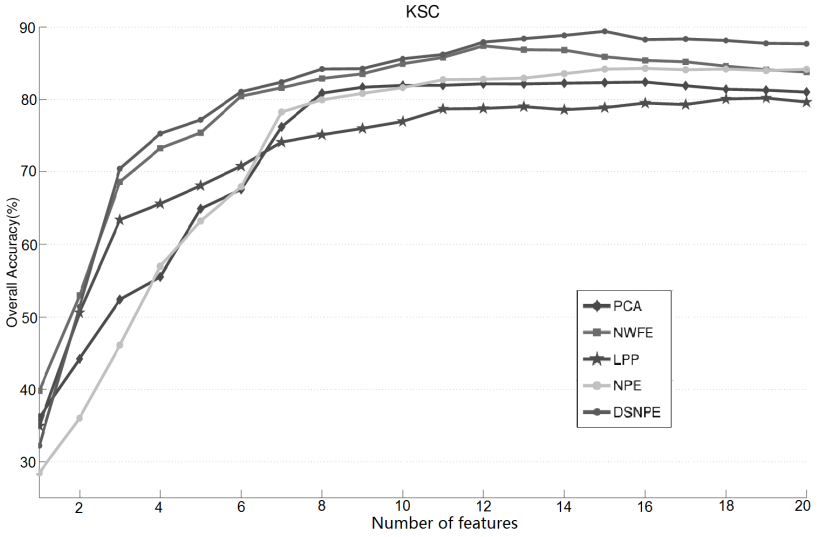


Figure 2.9: Average overall classification accuracy (OA%) with the number of extracted features increasing for different methods for KSC. 100 labelled training samples are chosen randomly from each class.

perspectral data is shown in Figure 2.6a. There are 16 different identified land-cover types in this region. The ground truth of the area associated to the employed 16 classes is shown in Figure 2.7a. 13 classes, which have

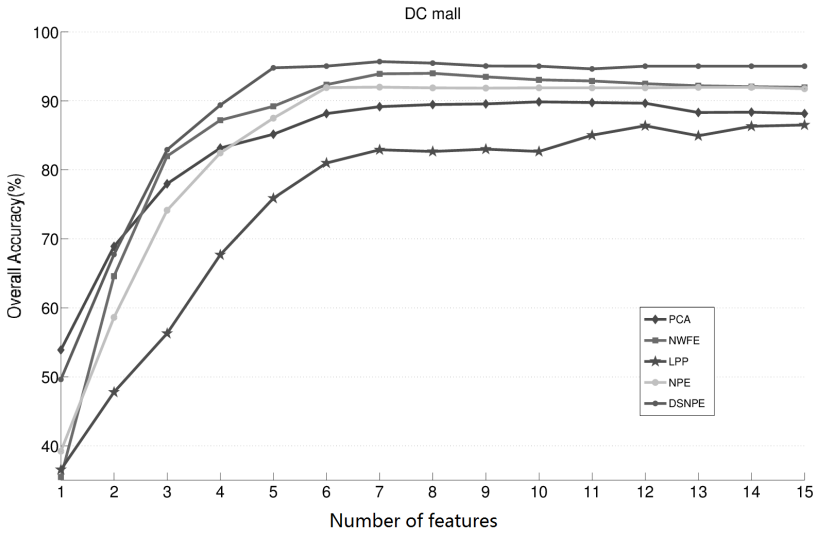


Figure 2.10: Average overall classification accuracy (OA%) with the number of extracted features increasing for different methods for DC mall. 100 labelled training samples are chosen randomly from each class.

more than 60 labelled samples, were selected for the experiments. The information about the land-cover types and corresponding available ground-truth pixels are listed in Table 2.1. For more details, the readers are referred to ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/.

KSC: the NASA AVIRIS instrument acquired data over the Kennedy Space Center (KSC), Florida, on March 23, 1996. AVIRIS acquires data in 224 bands of 10-nm width with center wavelengths from 400 to 2500 nm. The KSC data, acquired from an altitude of approximately 20 km, has a spatial resolution of 18 m. After removing water absorption and low-SNR bands, 176 bands were used for the analysis. Discrimination of land cover type is difficult due to the similarity of spectral signatures of some different vegetation types. For classification purposes, 13 classes representing the various land-cover types that occur in this environment were defined. The data set with 512×614 pixels is located on a different portion of the flight line and exhibits somewhat different characteristics. Figure 2.6b shows a false color image is composed of three bands of original hyperspectral data. Details of the thirteen land cover classes considered in the KSC area are shown in Table 2.1. For more information, visit http://www.csr.utexas.edu.

DC Mall: the Washington DC Mall data set is a Hyperspectral Digital Imagery Collection Experiment airborne hyperspectral image. This data set contains 1280 scan lines, and each line has 307 pixels. Every pixel contains 210 bands in the 0.4-2.4 μm region of the visible and IR spectrum. Since some water absorption channels (with less sprectral information but a lot of noise) are removed in the data-preprocessing procedure, only 191 channels are

preserved in this paper. A false color image composed of three bands of original hyperspectral data is shown in Figure 2.6c. The portion of this scene used in our experiments has dimensions of 550×307 pixels, as most testing samples are located in these region and less computing resources required. The name of the classes and the corresponding available samples (for training and testing) are listed in Table 2.1.

We compare our proposed DSNPE with the other three widely used unsupervised feature extraction methods namely PCA [Hotelling 33], LPP [He 04] and NPE [He 05], and supervised method NWFE [Kuo 04]. The classification accuracies are evaluated by two statistics, overall classification accuracy (OA) and Kappa coefficient(κ) (please see details in: *https* : *//en.wikipedia.org/wiki/Cohen's_kappa*). For the classifier, we used k-Nearest Neighbors (k-NN) classifier.

In order to explore the influences of the training sample size on the classification performance for each feature extraction method, the number of training sample for each class (n_i) is set to be 20, 40 and 100. If the number of known ground-truth pixels for one class is smaller than the number of training sample for each class (n_i), then half number of ground-truth pixels in that class would be selected as training samples, the remaining labelled samples act as testing samples.

In our experiments, we use overall classification accuracy (OA) to evaluate the feature extraction results. The results were averaged over 10 runs on different number of extracted features for each method.

2.2.3.2 Experimental Results

The experimental results are summarized in Figure 2.8 - Figure 2.10 and Table 2.2-Table2.5. Figure 2.8 - Figure 2.10 indicate averaged overall classification accuracy (OA) with increasing extracted number of features of different methods for India Pine, KSC, DC mall respectively. Table 2.3 - Table 2.5 show the OA of different methods for each class of the three data sets. The highest overall classification accuracy and the corresponding number of employed features (placed in parentheses), Kappa coefficient(κ) are listed in Table 2.2. The classification maps associated with DSNPE and other methods are shown in Figure 2.11 - Figure 2.13. From the experimental results mentioned above, we have the following findings:

1. The results confirm that supervised feature extraction methods, as NWFE and proposed DSNPE, can achieve better results in the classification of hyperspectral images, compared to the unsupervised feature extraction methods. The DSNPE-related classification results have the highest accuracies, because DSNPE takes into account both label information and local neighborhood information. This means that our proposed method DSNPE is able to extract more discriminative features than the both methods.

Table 2.3: OA% of different methods with 12 extracted features for each class of India Pine, with 100 labelled training samples from each class.

Class	PCA	NWFE	LPP	NPE	DSNPE
C1	73.3	**81.0**	71.1	62.2	69.0
C2	62.7	70.1	58.3	69.3	**73.3**
C3	54.2	70.8	56.7	57.5	**71.4**
C4	54.6	57.4	43.5	39.1	**63.1**
C5	88.5	91.6	85.1	85.8	**93.3**
C6	94.0	95.2	92.4	94.8	**97.0**
C7	80.0	89.5	85.0	80.0	**94.7**
C8	98.6	95.2	93.6	**99.3**	98.1
C9	92.9	92.3	78.6	**100.0**	92.3
C10	63.4	77.1	67.2	59.8	**78.3**
C11	70.8	78.8	76.7	71.6	**81.8**
C12	60.9	75.3	52.2	68.5	**77.6**
C13	99.5	99.4	98.4	**99.6**	98.3
C14	**95.3**	93.3	92.2	91.0	94.0
C15	56.7	56.3	34.8	**68.4**	47.4
C16	**91.4**	90.8	81.5	79.0	89.5
OA	75.8	82.7	75.6	77.4	**84.0**

2. From Table 2.2, it can be noted that the classification accuracies for all methods increases with the number of labelled training samples. If 100 labelled training samples are selected from each class, the OA of our proposed DSNPE can reach to 84.3%, 89.4% and 95.7% for India Pine, KSC, DC mall respectively. With higher classification accuracy, DSNPE has more potential in practical applications.

3. Figure 2.8 - Figure 2.10 shows that the OAs of NWFE and DSNPE increases first with the number of extracted features and then falls, whereas the proposed method DSNPE remains stable after that.

4. DSNPE is superior to NPE in all situations since the DSNPE considers not only the intraclass geometry but also the discriminative information derived from the interclass samples.

5. Table 2.3-2.5 show the OA of each class for three hyperspectral data sets. For most classes, DSNPE performs better than others, especially for class

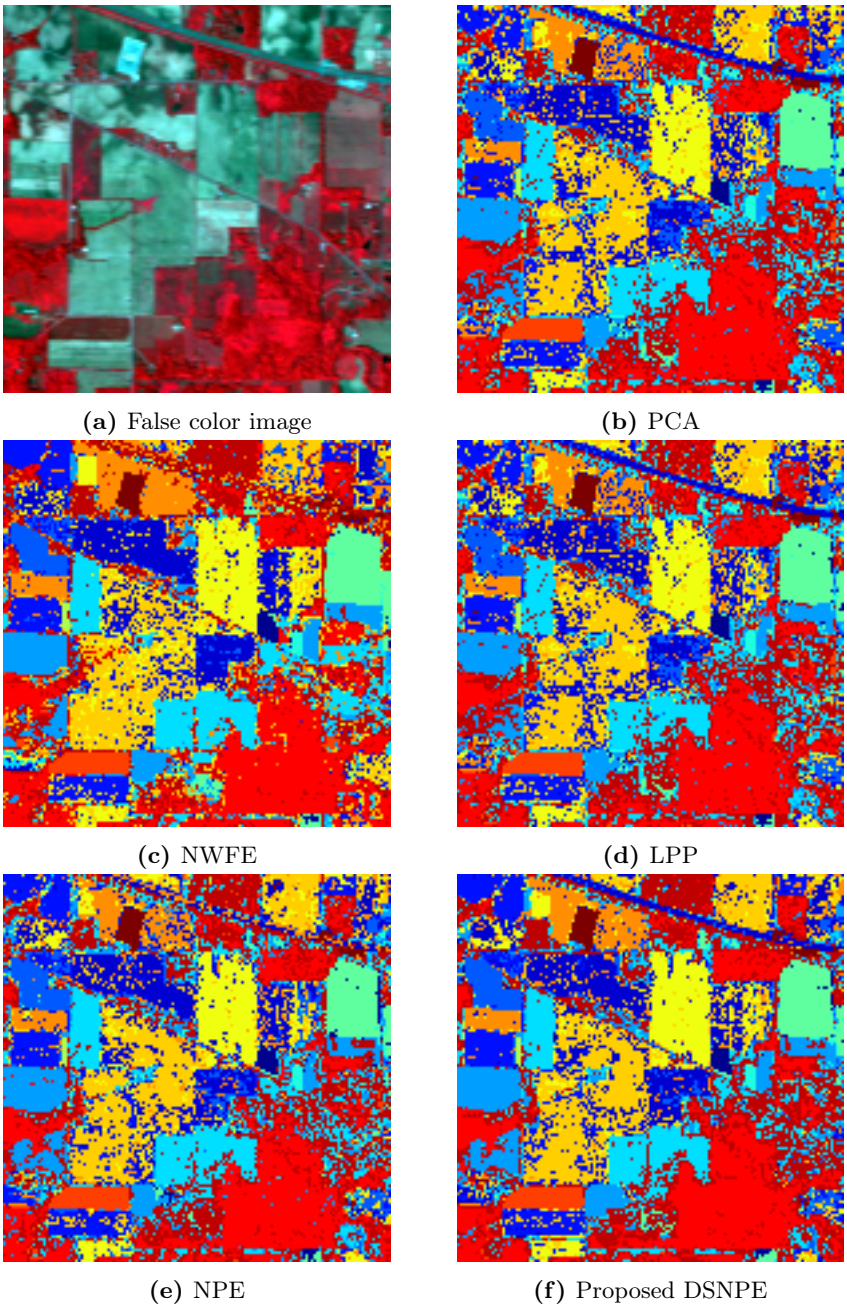
Table 2.4: Overall classification accuracy (OA%) of different methods with 12 extracted features for each class of KSC, with 100 labelled training samples from each class.

Class	PCA	NWFE	LPP	NPE	DSNPE
C1	75.7	85.3	73.1	84.3	**85.9**
C2	75.4	**89.6**	82.3	81.6	84.7
C3	60.6	69.4	**85.2**	67.0	69.9
C4	42.9	47.4	**50.5**	39.0	50.0
C5	52.1	59.4	48.8	65.4	**66.3**
C6	64.0	56.8	46.6	57.0	**69.8**
C7	72.3	80.0	**92.3**	92.0	88.9
C8	76.7	82.2	77.5	59.5	**87.9**
C9	88.5	88.3	84.0	80.0	**90.2**
C10	95.1	98.3	87.9	92.9	**98.5**
C11	93.9	**98.1**	89.2	94.1	97.2
C12	79.3	89.4	79.0	79.7	**90.5**
C13	**100.0**	99.0	97.7	99.3	**100.0**
OA	81.8	87.4	79.6	84.2	**89.1**

Table 2.5: Overall classification accuracy (OA%) of different methods with 6 extracted features for each class of DC Mall, with 100 labelled training samples from each class.

Class	PCA	NWFE	LPP	NPE	DSNPE
C1	84.6	92.4	81.4	89.5	**93.2**
C2	94.9	77.1	80.6	95.3	**99.3**
C3	98.2	97.4	96.7	**100.0**	**100.0**
C4	99.8	99.0	92.5	99.8	**100.0**
C5	**99.3**	99.0	89.4	97.0	98.9
C6	99.5	95.8	92.5	**99.8**	99.5
C7	99.0	**99.5**	77.8	93.4	94.1
OA	89.6	93.8	86.2	92.1	**95.7**

C2, C4, C7 in India Pine, class C6 and C8 in KSC, class C2 in DC mall.

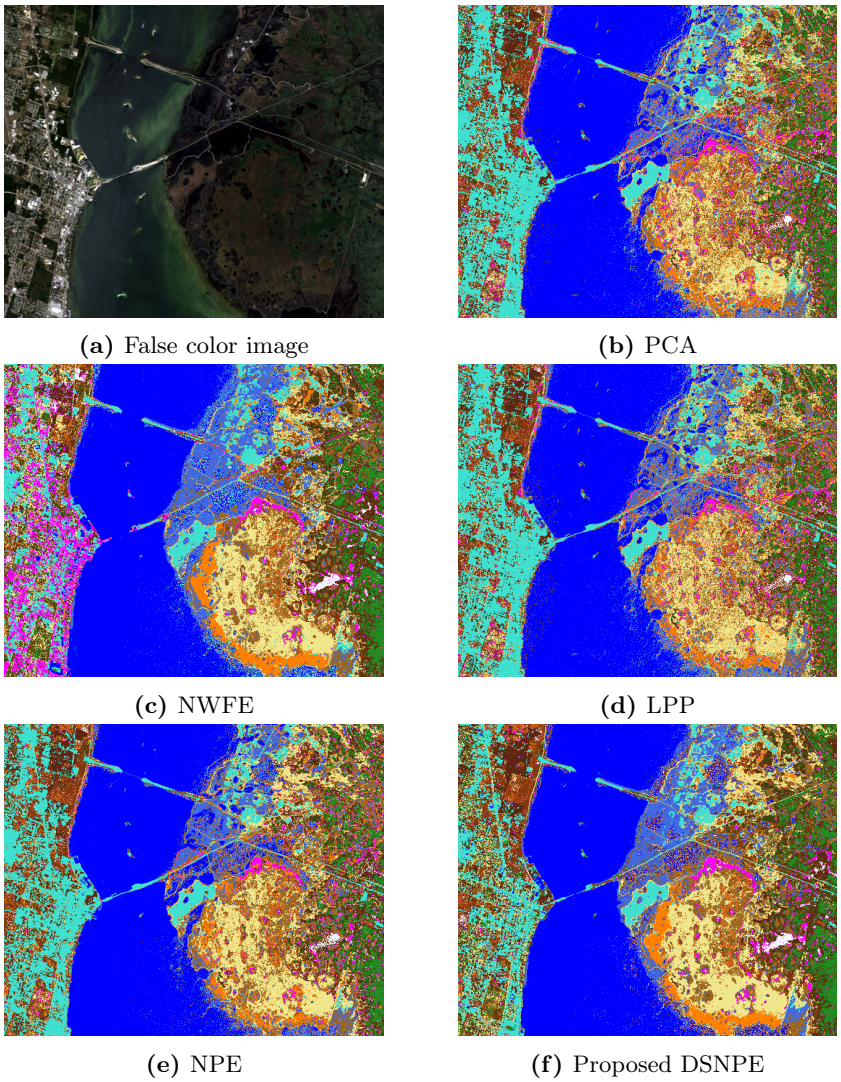


(a) False color image **(b)** PCA

(c) NWFE **(d)** LPP

(e) NPE **(f)** Proposed DSNPE

Figure 2.11: Classification maps of different feature extraction methods for Indian Pine with k-NN classifier ($n_i = 100$).



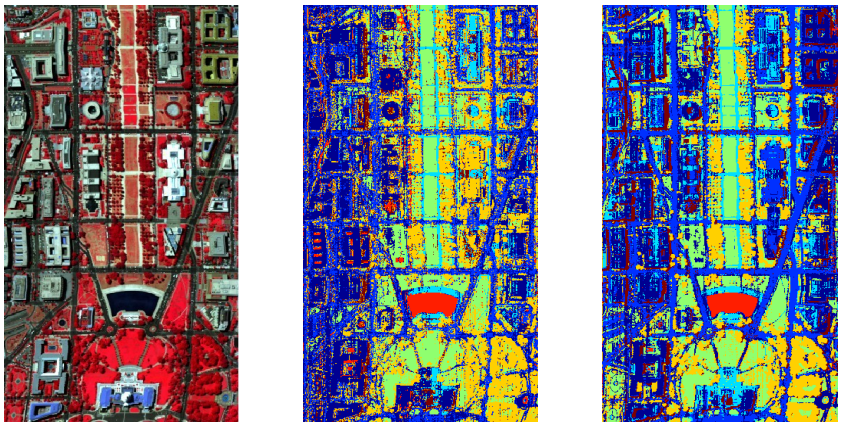
(a) False color image **(b)** PCA

(c) NWFE **(d)** LPP

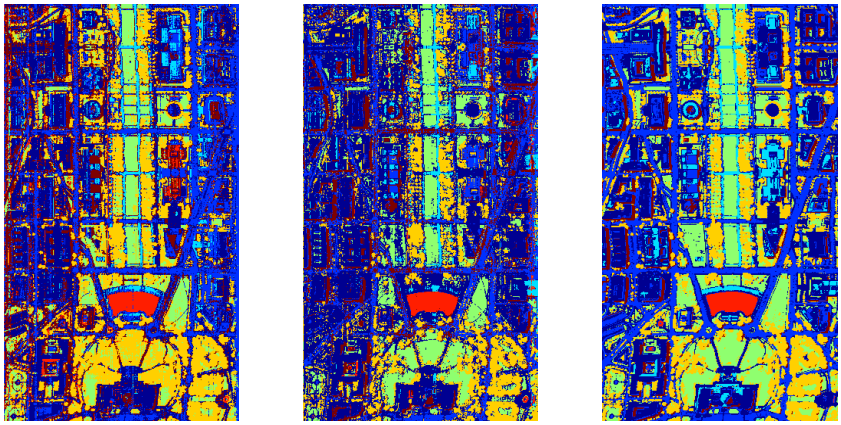
(e) NPE **(f)** Proposed DSNPE

Figure 2.12: Classification maps of different feature extraction methods for KSC with k-NN classifier ($n_i = 100$).

6. In order to compare the classification results visually, we randomly select 100 labelled training samples per class from each data set. The best classification maps of each methods are shown in Figure 2.11 and Figure 2.13 respectively. It can be seen that the classification maps of proposed DSNPE looks smooth and with less noisy, and this is specially clear for "Water" region near to the coastline in the classification maps of KSC.



(a) False color image **(b)** PCA **(c)** NWFE



(d) LPP **(e)** NPE **(f)** Proposed DSNPE

Figure 2.13: Classification maps of different feature extraction methods for DC Mall with k-NN classifier ($n_i = 100$).

2.3 PCA-based Supervised Locality Preserving Projection (PSLPP)

2.3.1 Locality Preserving Projection (LPP)

PCA aims to preserve the global geometrical structure (such as the data scatter in one direction) of feature space, and the LDA method aims to preserve the discriminating information in global (such as within- and between-class scatter of data). However, in many real world applications, both global and local geometrical/manifold structure information is important, but without using local manifold structures (similarity of data point with its nearest neighbors) may not be good. Locality preserving projection (LPP), which is a linear approxi-

mation of the LE [Belkin 02] method, can preserve the local manifold structure of samples. In LPP, the local manifold property is preserved based on the pairwise distances between nearby data points. As an unsupervised manifold dimension reduction algorithm, LPP seeks a low-dimensional representation of the data in which the distances between a data point and its e nearest neighbors are minimized.

Suppose $\mathbf{x}_i$ is a high-dimensional data point, and $\mathbf{z}_i$ is the low-dimensional representation of $\mathbf{x}_i$. Then the cost function of LPP is as follows:

$$\min \sum_{ij} ||\mathbf{z}_i - \mathbf{z}_j||^2 A_{ij} \tag{2.17}$$

where $\mathbf{A}$ is a data points spectral similarity matrix, see as Figure 2.4. This is done in a weighted manner (different pairwise distances corresponding different A_{ij}), as in the low-dimensional data representation, the distance between two nearest data points contributes more to the cost function than the distance between the second nearest neighbors. Two data points $\mathbf{x}_i$ and $\mathbf{x}_j$, if they are connected or within e nearest neighbors of each other, they have an associated weight A_{ij}. If neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart, the objective function with the big weight of A_{ij} will result in a heavy penalty. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close then $\mathbf{z}_i$ and $\mathbf{z}_j$ are close too.

Both LPP and NPE try to extract features by preserving local manifold structure information. However, NPE expects each projected data point can be represented as a linear combination of its neighbors with the coefficients, see cost function 2.3; LPP finds a transformation based on pairwise distances between data points and its nearest neighbors, see cost function in equation (2.3).

Normally similarity matrix A_{ij} is defined as follows:

$$A_{ij} = \begin{cases} \exp(\dfrac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{t}) & , \mathbf{x}_i \in knn(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in knn(\mathbf{x}_i) \\ 0 & , \mathbf{x}_i \notin knn(\mathbf{x}_j) \text{ and } \mathbf{x}_j \notin knn(\mathbf{x}_i) \end{cases} \tag{2.18}$$

or

$$A_{ij} = \begin{cases} \exp(\dfrac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{t}) & , ||\mathbf{x}_i - \mathbf{x}_j||^2 \leq \epsilon \\ 0 & , ||\mathbf{x}_i - \mathbf{x}_j||^2 > \epsilon \end{cases} \tag{2.19}$$

$knn(\mathbf{x}_i)$ denotes the nearest neighbors set of $\mathbf{x}_i$, ϵ is the radius to define the nearest neighbors of $\mathbf{x}_i$.

The cost function above can be reduced by the following algebra formula-

tion:

$$\begin{aligned}\frac{1}{2}\sum_{ij}(\mathbf{z}_i-\mathbf{z}_j)^2A_{ij} &= \frac{1}{2}\sum_{ij}(\mathbf{w}^T\mathbf{x}_i-\mathbf{w}^T\mathbf{x}_j)^2A_{ij}\\ &= \sum_i \mathbf{w}^T\mathbf{x}_iD_{ii}\mathbf{x}_i^T\mathbf{w}-\sum_{ij}\mathbf{w}^T\mathbf{x}_iA_{ij}\mathbf{x}_j^T\mathbf{w}\\ &= \mathbf{w}^T\mathbf{X}(\mathbf{D}-\mathbf{A})\mathbf{X}^T\mathbf{w}\\ &= \mathbf{w}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w}\end{aligned} \tag{2.20}$$

where $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n]$, $\mathbf{D}$ is a diagonal matrix with $D_{ii}=\sum_j A_{ij}$, $\mathbf{L}=\mathbf{D}-\mathbf{A}$ is the Laplacian matrix [Belkin 02].

The transformation vector $\mathbf{w}$ that minimizes the cost function is obtained by minimizing the generalized eigenvalue problem:

$$\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w}=\lambda\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{w} \tag{2.21}$$

Supposing that $\mathbf{w}_1,\mathbf{w}_2,\cdots,\mathbf{w}_d$ are the solutions of equation (2.21), sort them in ascending order according to their eigenvalues, namely $\lambda_1<\lambda_2<\cdots<\lambda_d$. Thus mapping the high-dimensional sample $\mathbf{x}_i\in\mathbb{R}^D$ to low-dimensional sample $\mathbf{z}_i\in\mathbb{R}^d$ as follows:

$$\mathbf{z}_i=\mathbf{W}^T\mathbf{x}_i \tag{2.22}$$

where $\mathbf{w}_i$ is a d-dimensional vector, and $\mathbf{W}=[\mathbf{w}_1,\mathbf{w}_2,\cdots,\mathbf{w}_d]$ is an $n\times d$ matrix.

2.3.2 Proposed PSLPP

In locality preserving projection (LPP), spectral similarity matrix $\mathbf{A}$ is only related to the neighborhood or the nearest neighbors, only nearest neighbors are connected and have non-zero elements A_{ij} in similarity matrix $\mathbf{A}$. However, there are two cases in reality: *case*1, some unlabelled points belong to same class but without connection as they are not within e nearest neighbors of each other; *case*2, some points are within e nearest neighbors of each other and connected but belong to different class. Due to the locality preserving property of LPP, the points corresponding to *case*1 would be faraway to each other in the reduced feature space, while the points corresponding to *case*2 would be faraway to each other. This will result in an unfavourable situation in pattern analysis especially in classification problem. With prior class label information, we propose a supervised approach PSLPP for hyperspectral image feature extraction. As the original high-dimensionality hyperspectral image is noisy and contains redundant information, the distances between samples are not reliable. As a result, it is better to remove noise and redundancy by PCA before the construction of similarity matrix. By omitting the smallest principal components (noisy), more than 99.9% information in the sense of reconstruction is kept. In our experiments, first 15 principal components (features) as extracted and used to construct similarity matrix.

Suppose that $\mathbf{x}_i^*$ is a representation of $\mathbf{x}_i$ transformed by PCA, y_i is the label of $\mathbf{x}_i$. Let $\mathbf{z}_i$ denote $\mathbf{x}_i$'s representation in the final low-dimensional space, with $\mathbf{z}_i = \mathbf{W}^T\mathbf{x}_i$, $\mathbf{W}$ is transformation matrix. Then the cost function of the proposed PSLPP is:

$$\min \sum_{ij} ||\mathbf{z}_i - \mathbf{z}_j||^2 (\frac{P_{ij} + S_{ij}}{2}) \tag{2.23}$$

where $\mathbf{P}$ is a similarity matrix based on nearest neighbor information, $\mathbf{S}$ is a similarity matrix based on label information. and they are defined as:

$$P_{ij} = \begin{cases} \exp(\frac{-\left\|\mathbf{x}_i^* - \mathbf{x}_j^*\right\|^2}{t}) & , \mathbf{x}_i^* \in knn(\mathbf{x}_j^*) \text{ or } \mathbf{x}_j^* \in knn(\mathbf{x}_i^*) \\ 0 & , \mathbf{x}_i^* \notin knn(\mathbf{x}_j^*) \text{ and } \mathbf{x}_j^* \notin knn(\mathbf{x}_i^*) \end{cases} \tag{2.24}$$

$knn(\mathbf{x}_i)$ means the nearest neighbors set of $\mathbf{x}_i$.

$$S_{ij} = \begin{cases} 1 & , y_i = y_j \\ 0 & , y_i \neq y_j \end{cases} \tag{2.25}$$

Similar to LPP, the cost function of proposed PSLPP can be reduced by the following algebra formulation:

$$\begin{aligned} \sum_{ij} (\mathbf{z}_i - \mathbf{z}_j)^2 (\frac{P_{ij} + S_{ij}}{2}) &= \frac{1}{2} \sum_{ij} (\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\mathbf{x}_j)^2 (P_{ij} + S_{ij}) \\ &= \mathbf{w}^T\mathbf{X}(\mathbf{D} - \mathbf{P} - \mathbf{S})\mathbf{X}^T\mathbf{w} \\ &= \mathbf{w}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w} \end{aligned} \tag{2.26}$$

$\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j (P_{ij} + S_{ij})$, $\mathbf{L} = \mathbf{D} - \mathbf{P} - \mathbf{S}$ is the Laplacian matrix. Besides, a constraint is imposed as follows:

$$\mathbf{z}^T D\mathbf{z} = 1 \rightarrow \mathbf{w}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w} = 1 \tag{2.27}$$

The transformation vector $\mathbf{w}$ that minimizes the object function is obtained by solving the minimizing optical problem:

$$\mathbf{w}_{PSLPP} = \arg\min_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{w}}{\mathbf{w}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w}} \tag{2.28}$$

the solutions of equation (2.28) $\mathbf{W}_{PSLPP} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d]$ is the optical transformation $n \times d$ matrix.

2.3.3 Experiments

2.3.3.1 Experimental Datasets and Settings

Experiments were run on two data sets, namely the Indian Pine and DC mall, which were reported in the previous Chapter.

Algorithm 1 Proposed PSLPP

1: Remove noisy and redundancy by PCA: projecting the original high-dimensional training samples $\mathbf{x}_i$ into lower dimensional space $\mathbf{x}_i^*$ (as 15 bands).
2: Construct the similarity matrix $\mathbf{P}$ based on LPP by using low-dimensional points $\mathbf{x}_i^*$, as equation (2.24).
3: Construct the similarity matrix $\mathbf{S}$ based on label information, as equation (2.25).
4: Calculate the transformation matrix $\mathbf{W}$ by equation (2.26) and (2.28).
5: Extract features through $\mathbf{z}_i = \mathbf{W}^T\mathbf{x}_i$.
6: Classify the testing samples with extracted features.

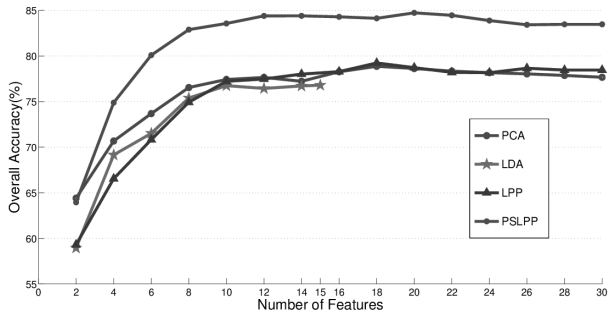
Experimental setup: in order to explore the influences of the training sample size on classification performance of different feature extraction methods, 5%, 10% and 15% (n_i) samples are randomly selected from each class as training sample set for Indian Pine data set, and 2%, 5% and 10% for DC mall data set. The testing data set is composed of the remaining samples with known ground truth pixels in the scene. In our experiments, we extract 1 to 30 features from Indian Pine data set, and 1 to 15 features from DC mall data set (except LDA, as it can extract maximum C-1 features, C is the number of class). Then, the testing accuracies for different values of the employed number of features are calculated. In this section, other three feature extraction methods, namely PCA, LDA and LPP, are included to compare with the performance of the proposed PSLPP. The overall classification accuracy (OA) is utilized to evaluate the classification performances. For the classifier, we choose two traditional classifiers: k-Nearest Neighbors (k-NN) and Support Vector Machine (SVM).

2.3.3.2 Experimental Results

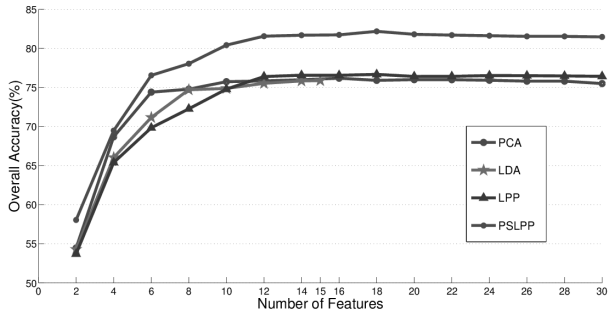
The experimental results are summarized in Figure 2.14 - Figure 2.16 and Table 2.6 - Table 2.8. Figure 2.14 indicates the performance of each method for India Pine and DC mall respectively, as the number of features increases when training sample is 10% from each class. Table 2.7 - Table 2.8 show the OA% of different methods for each class of the two data sets. The highest overall accuracies (OA) and the corresponding number of employed features (placed in parentheses) are listed in Table 2.6. In order to compare the classification results visually, the classification maps for these two data sets of each methods are shown in Figure 2.15 and Figure 2.16. From the experimental results mentioned above, we conclude the following:

1. Except for only one case ($N_i = 5\%$ in the Indian Pine data set), PCA has the highest accuracy, and the PSLPP-related classification results have the highest accuracies in all the considered situations.

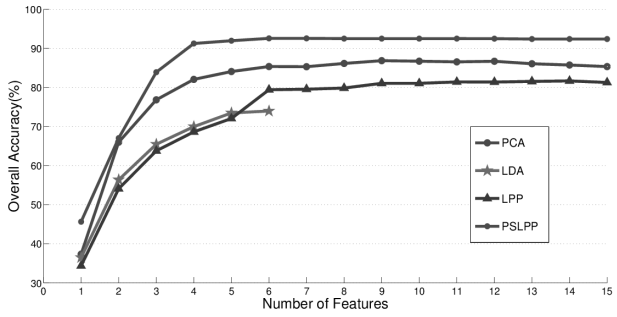
2. PSLPP performs better than LPP as the it takes into account not only the local manifold structure, but also label information, the label information



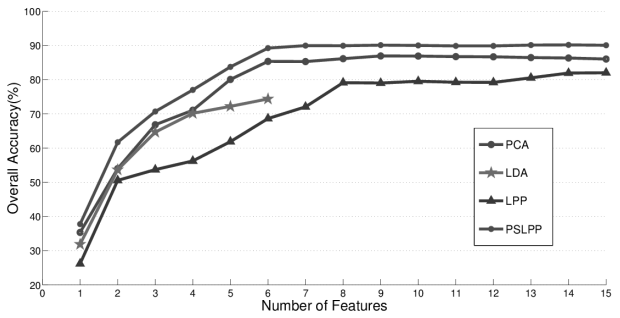
(a) Indian Pine with SVM classifier



(b) Indian Pine with k-NN classifier



(c) DC mall with SVM classifier



(d) DC mall with k-NN classifier

Figure 2.14: Averaged OA (%) with the number of extracted features increasing for different methods. 10% labelled training samples are chosen randomly from each class.

Table 2.6: OA% (optimal number of features) of different feature extraction methods with different training sample size.

Data set	Classifier	$\mathbf{n}_i$	PCA	LPP	LDA	Proposed PSLPP
India Pine	k-NN	5%	**64.5(18)**	61.4(16)	60.3(15)	64.3(14)
		10%	76.2(16)	76.7(18)	75.9(14)	**82.2(18)**
		15%	82.1(20)	79.8(20)	78.4(14)	**84.4(16)**
	SVM	5%	**67.8(16)**	63.5(14)	61.4(14)	65.6(16)
		10%	78.9(18)	79.2(18)	76.8(15)	**83.7(20)**
		15%	85.0(14)	84.4(18)	82.8(15)	**85.5(16)**
DC mall	k-NN	2%	82.2(10)	78.4(7)	72.9(6)	**85.1(9)**
		5%	87.0(9)	80.6(13)	75.2(6)	**90.2(13)**
		10%	88.8(10)	85.6(11)	80.7(5)	**92.4(12)**
	SVM	2%	84.1(10)	79.9(12)	73.3(5)	**87.8(9)**
		5%	86.9(9)	81.7(14)	74.0(6)	**92.6(11)**
		10%	89.5(8)	87.9(13)	81.1(6)	**93.9(13)**

can make the similar samples which belong to the same class more close to each other in the low-dimensional feature space.

3. As shown in Figure 2.14, PSLPP performs better than LDA (higher classification accuracy). Due to the restriction of the rank of the between-class scatter matrix, no more than $C-1$ (C is the number of the class) features can be extracted and used for LDA, while PSLPP does not have this limitation.

4. From all experimental results, SVM performs better than KNN, with at least 1% improvement in all the considered situations for Indian Pine and Dc Mall.

5. As can be seen from the classification accuracy curves, by increasing the number of features in the projection subspace (obtained by the proposed methods) the recognition accuracy of the proposed methods will not necessarily increase. Thus, in practice, the proposed methods will provide good performance without using a lot of features.

6. From Table 2.7-2.8, which show the OA of each class, we can see that PSLPP performs best than others, as class C5, C7, C12 in India Pine, class C1 and C3 in DC mall.

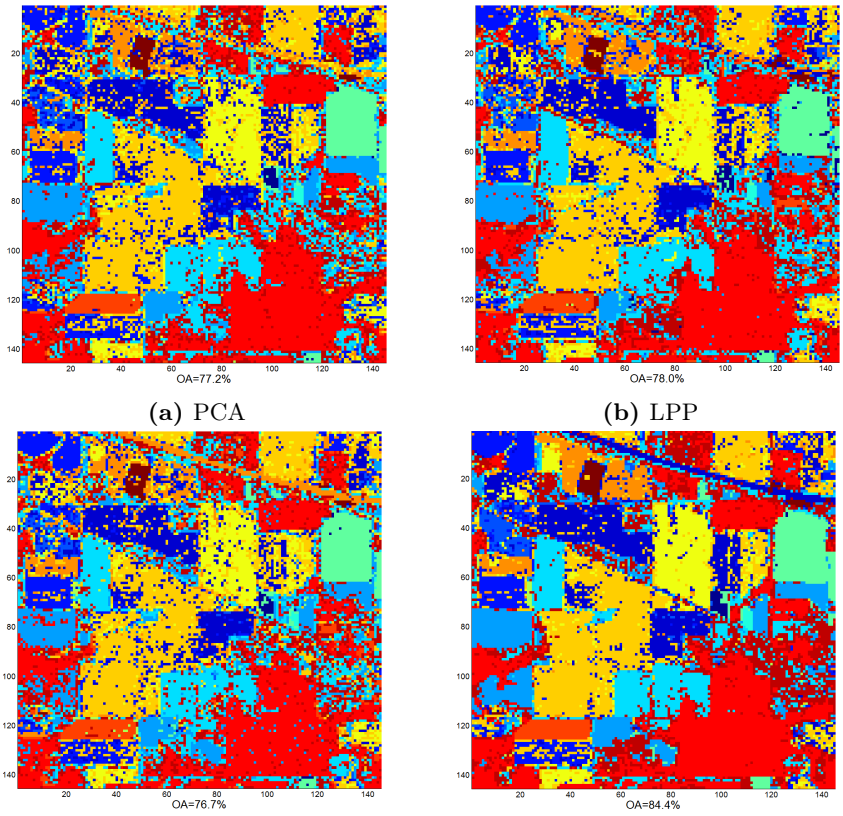
7. Based on the visual inspection of classification maps, generally, PSLPP has better performance than others.

Table 2.7: OA% of different methods with 12 extracted features for each class of India Pine, with 10% labelled training samples from each class and SVM classifier.

Class	PCA	LPP	LDA	Proposed PSLPP
C1	52.1	77.0	73.3	**77.5**
C2	58.3	63.7	62.7	**71.3**
C3	61.2	59.1	54.2	**59.0**
C4	49.6	53.7	54.6	**62.5**
C5	84.3	86.6	88.5	**92.1**
C6	**94.6**	90.9	94.3	94.1
C7	81.0	94.0	80.0	**100.0**
C8	**99.4**	89.7	98.6	98.2
C9	86.7	97.0	92.9	**100.0**
C10	**72.8**	71.6	63.4	65.8
C11	**77.4**	74.8	70.8	72.9
C12	49.0	65.7	60.9	**77.5**
C13	97.8	96.5	**99.5**	98.8
C14	**95.3**	89.9	93.3	93.4
C15	52.6	46.9	56.7	**69.1**
C16	86.2	86.9	**91.4**	85.9
OA	78.6	78.4	76.4	**83.5**

Table 2.8: OA% of different methods with 6 extracted features for each class of DC mall, with 10% labelled training samples from each class and SVM classifier.

Class	PCA	LPP	LDA	Proposed PSLPP
C1	83.6	76.9	79.6	**87.6**
C2	93.9	**98.5**	73.1	90.6
C3	97.2	93.1	96.6	**100.0**
C4	98.8	97.5	70.5	**99.8**
C5	98.3	98.3	70.0	**99.5**
C6	98.4	98.8	97.3	**100.0**
C7	**98.8**	94.2	62.3	**97.6**
OA	88.8	87.6	81.0	**93.4**



(a) PCA **(b)** LPP

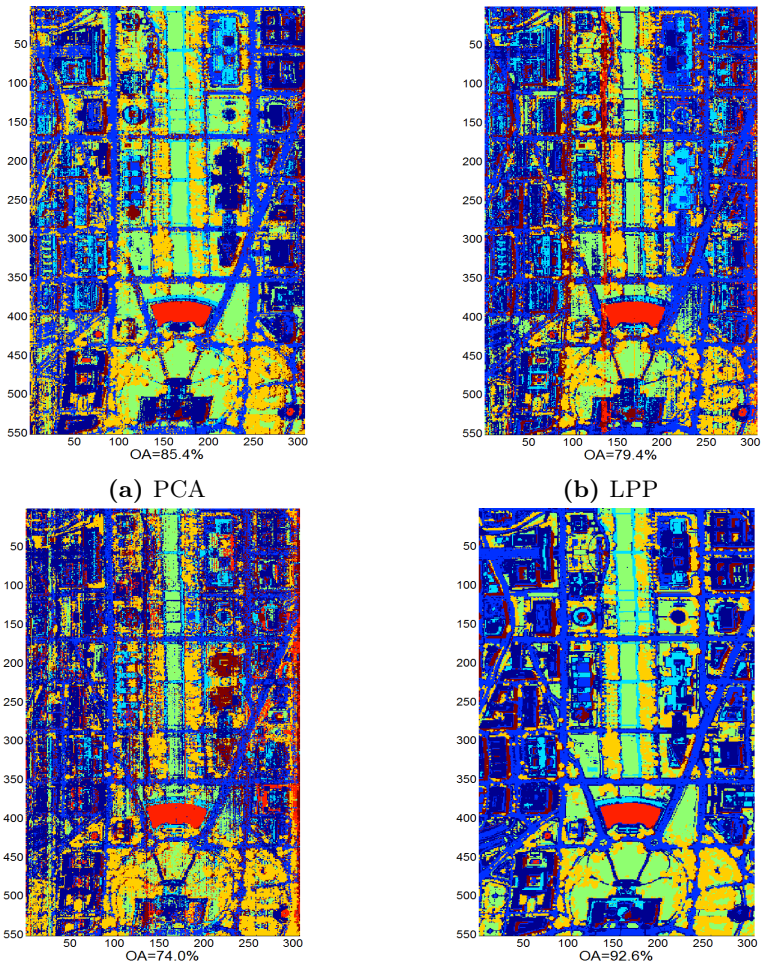
(c) LDA **(d)** Proposed PSLPP

Figure 2.15: Classification maps of different methods for Indian Ipine with SVM classifier

2.4 Conclusions

As unsupervised methods do not aim at extracting discriminant features, we presented two new supervised feature extraction approaches by introducing prior class label information during neighbourhood selection in the training stage in this Chapter. The first is a supervised version of neighborhood preserving embedding (NPE) for the classification of land-cover types in hyperspectral images. By using label information, our proposed supervised NPE performs better than NPE in classification. The results of experiments on the real images have demonstrated the effectiveness of the proposed algorithm. Compared with some representative dimensionality reduction algorithms, the proposed DSNPE has a very competitive performance with higher classification accuracy.

The second proposed method is named PCA-based supervised locality preserving projections (PSLPP), which is a supervised extension of locality pre-



(a) PCA **(b)** LPP

(c) LDA **(d)** Proposed PSLPP

Figure 2.16: Classification maps of different methods for DC mall with SVM classifier

serving projections (LPP). LPP seeks to preserve the local manifold structure (the similarity of point with its neighbors) which is usually more significant than the global manifold structure preserved by PCA, but does not take into account label information. PSLPP uses both local information and class information to model the similarity of the data and enhance the discriminant power of the data when mapping them into a low-dimensional space. Since both local manifold structure and representative label information are important for classification, PSLPP outperforms the traditional LPP, together with PCA, LDA, which tend to preserve the global manifold structure.

The research in this chapter lead to three journal publications as follows:

1. **Luo Renbo**, Pi Youguo, "Supervised neighborhood preserving embedding feature extraction of hyperspectral imagery". Acta Geodaetica et Cartographica Sinica. 2014; 43(5): 508-513.

2. **Luo Renbo**, Liao Wenzhi, Pi Youguo, "Discriminative supervised neighborhood preserving embedding feature extraction for hyperspectral-image classification". Telkomnika. 2012; 10(5): 1051-1056.

3. **Luo Renbo**, Liao Wenzhi, Pi Youguo, "Research on supervised LPP feature extraction for hyperspectral image". Remote Sensing Technology and Application. 2012; 27(6): 46-52.

3

Semi-supervised Feature Extraction of Remote Sensing Data

This chapter focus on developing semi-supervised techniques to extract interesting and useful features for reliable classification.

In many real applications, only few labelled samples are available for training, because manually labelling data is time consuming and fairly expensive. On the other hand, unlabelled samples are usually available in large quantities at very low cost. Semi-supervised feature extraction methods, which combine a limited set of labelled samples and general a larger set of unlabelled samples in training, can outperform both supervised and unsupervised methods in this case. For this reason, semi-supervised feature extraction methods have aroused a great deal of interest in the machine learning community, and have been successfully applied in hyperspectral (hyperspectral) image classification. This chapter explores semi-supervised feature extraction methods for hyperspectral image classification.

Firstly, we improve the previous method semi-supervised local discriminant analysis (SELD) [Liao 12] for feature extraction of hyperspectral images. The proposed improved semi-supervised local discriminant analysis (ISELD) method aims to find a projection which can preserve local neighborhood information and maximize the class discrimination of the data. Compared to the previous SELD, the proposed ISELD method better models the correlation between labelled and unlabelled samples. Experimental results on a real data demonstrate our approach outperforms others with increasing the training sample size changes.

We also propose a Semi-supervised Graph Learning (SEGL) method for feature extraction of hyperspectral remote sensing imagery. The proposed SEGL method aims to build a semi-supervised graph which can better model the spectral similarities of samples by weighting the edges, especially of labelled and unlabelled samples. In our semi-supervised graph, all training samples

are divided into two groups: labelled and unlabelled, then labelled samples are connected to each other by their labels (samples from same class will be connected), unlabelled samples are connected to each other by their nearest neighborhood information (the nearest neighbors will be connected). By sorting the mean distance between an unlabelled sample and the center of each class (get from labelled samples), we connect the unlabelled sample with all labelled samples belonging to its nearest neighborhood class. Moreover, the proposed SEGL better model the actual differences and similarities between samples, by assigning different weights to the different edges of connected samples. Experimental results on real hyperspectral images demonstrate advantages of our method compared to some related feature extraction methods.

The proposed SEGL model similarities of samples by only spectral information, without using their spatial location information (such as two samples may belong to same class if they close to each other in spatial location). Therefore, we also propose an improved version of SEGL by fusing spectral and spatial information for hyperspectral image classification. In this method, spectral, spatial and label information are taken into account to construct the semi-supervised fusion graph to better model the correlations between samples. The nodes of the fusion graph are connected according to not only their label information, but also their spectral-spatial nearest neighborhood information. As different feature sources have different statistical distributions, we project the high-dimensional spectral and spatial features into a much lower dimensional subspace separately and both based on the proposed semi-supervised graph. Thus, neighborhood information is preserved and discriminative features is enhanced. Experimental results on a real hyperspectral data demonstrate the efficiency of our proposed semi-supervised fusion method, with 2%-10% improvements compared with unsupervised and supervised feature extraction method.

This chapter is organized as follows. In Section 3.1, we introduce the related work. Section 3.2 details our proposed improved semi-supervised local discriminant analysis. We discuss the proposed feature extraction with semi-supervised graph learning (SEGL) in Section 3.3. Section 3.4 elaborates an improved version of semi-supervised graph learning (ISEGL) to fuse spectral and spatial information. Finally, Section 3.5 concludes this chapter.

3.1 Introduction

Techniques for machine learning can be categorized as supervised, unsupervised and semi-supervised learning methods according to how they combined labelled and unlabelled training data. Supervised learning methods rely on the existence of labelled samples to infer class separability, but they heavily depend on the quality of labelled training data sets; normally it is only useful to classify images with the same classes and taken under the same conditions as the labelled training data sets. Moreover, sample labelling is time consuming and with very high cost, resulting in low availability of labelled training samples. On the other hand, unsupervised methods deal with the cases where no labelled

samples are available. The exact relationship between clusters and classes is then unknown. Even the number of classes present in the data may then not be known.

3.1.1 Semi-supervised learning

Semi-supervised learning is a class of machine learning techniques that make use of both labelled and unlabelled data. In reality, although the acquisition of labelled data for a learning problem often requires a skilled human agent or a physical experiment, the labelling process is expensive, whereas the acquisition of unlabelled data is relatively much cheaper. Semi-supervised learning (SSL) [Olivier 06, Zhu 08, Bennet 99], which incorporates a small amount of labelled data with large number of unlabelled samples, can be of great practical value and gained popularity in the machine learning community, as SSL can produce significant improvement in classification accuracy [Camps-Valls 07, Chen 11, Liao 13]. A broader definition of semi-supervised learning includes regression and clustering as well, but we will not pursue that direction here.

In the case of two classes, semi-supervised learning assumes that each class has a Gaussian distribution (an easy example). The complete data set is assumed to be distributed according to a Gaussian mixture model. Given a large amount of unlabelled data, the mixture components can be identified with the expectation-maximization (EM) algorithm [Dempster 77]. Therefore, only a small number of labelled example is needed for each class, i.e., for each component in the mixture model. This model has been successfully applied to text categorization [Nigam 06]. A typical variant of semi-supervised learning is self-training: A classifier is first trained with labelled data. It is then used to classify the unlabelled data. The most confident unlabelled samples, together with their predicted labels, are added to the training set. The classifier is re-trained and the procedure is repeated. Note the classifier uses its own predictions to teach itself.

With the rising popularity of support vector machines (SVM), some semi-supervised variants of SVM, such as transductive SVM (TSVM) [Vapnik 98], emerged as an extension to standard SVM. TSVM exploits specific iterative algorithms which gradually search reliable separating hyperplanes in the kernel feature space (for a non-linear classification problem, instead of trying to fit a non-linear model, one can map the problem from the input space (non-linear) to a new (higher-dimensional) space (linear, called the **kernel feature space**) by doing a non-linear transformation using suitably chosen basis functions (as Radial basis function), and then use a linear model in the kernel feature space. The linear model in the kernel feature space corresponds to a non-linear model in the input space.). This is done within an active learning process that incorporates both labelled and unlabelled samples in the training phase [Bruzzone 06]. Intuitively, unlabelled data guides the decision boundary away from dense regions. However, most semi-supervised variants of SVM suffer from a high computational burden and consequently a limited number of labelled samples can be used for their training. This leads to a poor estimation of

the distribution of marginal data. Many heuristic approaches have been proposed to reduce the computational cost of TSVM. Bennet *et al.* proposed a mixed integer programming to find the labelling with the lowest objective function [Bennet 99]. However, the optimization is intractable for large data sets. A heuristic that iteratively solves a convex SVM objective function with alternate labelling of unlabelled samples was proposed in [Joachims 99]. However, this algorithm is only capable of dealing with a few thousand samples. What's more, the improved TSVM still has a cubic cost, and requires to store huge kernel matrices [Chapelle 05].

Recently, graph-based semi-supervised learning methods have attracted great attention. Graph-based methods start by composing a graph where the nodes are the labelled and unlabelled data points, and edges with different weights reflect the similarity of nodes. The assumption is that nodes connected by a large-weight edge tend to have the same label, and labels can propagation throughout the graph. Graph-based semi-supervised enjoy nice properties from spectral graph theory. They are provided with some available labelled information in addition to the unlabelled information, thus allowing to encode some knowledge about the geometry and the distribution of the data set.

In 2004, Zhou *et al.* proposed a semi-supervised learning method by designing a classifying function based on the assumption that nearby points are likely to have the same label and points on the same structure (typically referred to as a cluster or a manifold) are likely to have the same label. The approach in [Bandos 06] extended the semi-supervised graph-based method presented in [Zhou 04] to the classification of hyperspectral images. [Bandos 06] preserved the contextual information through the use of composite kernels, which have been recently revealed very useful to improve inductive support vector machines (SVMs) [Camps-Valls 06, Camps-Valls 07]. Semi-supervised kernel orthogonal subspace projection (KOSP) was proposed for target detection applications [Capobianco 09], which introduces an additional regularization term on the geometry of both labelled and unlabelled samples by using the graph Laplacian. The information from unlabelled samples was included in the standard KOSP by means of the graph Laplacian with a contextual unlabelled sample selection mechanism.

3.1.2 Semi-supervised Feature extraction

For the task of feature extraction, semi-supervised feature extraction methods try to find a projection by using very limited number of labelled samples and a large number of unlabelled samples [Bruzzone 09,Sugiyama 10,Zhu 08,Chen 11]. Some semi-supervised feature extraction methods make use of link information (must-link for same class samples and cannot-link for different class samples), such as Zhang *et al.* [Zhang 07] proposed a semi-supervised dimensionality reduction (SSDR) technique by utilizing the must-link and cannot-link constraints. Some semi-supervised feature extraction methods add a regularization term to preserve certain potential properties of the data, for example semi-supervised discriminant analysis (SDA) [Cai 07] added a regularizer into the

objective function of LDA; it makes use of a limited number of labelled samples to maximize class discrimination and employs both labelled and unlabelled samples to preserve the local properties of the data. Some semi-supervised feature extraction methods combine supervised methods with unsupervised ones using a trade-off parameter, such as semi-supervised local fisher discriminant analysis (SELF) [Sugiyama 10]. However, it may not be easy to specify the optimal parameter values in these semi-supervised methods, as mentioned in [Chen 11].

Recently Liao *et al.* [Liao 13] proposed semi-supervised local discriminant analysis (SELD) for feature extraction of hyperspectral image without parameters. Their method divided the data set into two sets: a labelled set and an unlabelled set. They employed the labelled samples in a supervised method (linear discriminant analysis, LDA) (connections were constructed between labelled samples) only to maximize the class discrimination. The unlabelled samples are used in unsupervised local linear feature extraction methods (connections were constructed between unlabelled samples), such as LPP, NPE, only to preserve the local neighborhood information. However, the connections between labelled and unlabelled samples are not well exploited in SELD for both class discrimination and local neighborhood information preservation.

3.1.3 Proposed Semi-supervised Feature Extraction

This chapter first proposed an improved semi-supervised local discriminant feature extraction (ISELD) [Luo 15] to reduce the dimensionality of the hyperspectral images. Compared with SELD, the proposed ISELD adds matrices to model the connection of labelled and unlabelled samples. We set an edge between labelled and unlabelled samples when an unlabelled sample is the closest to the cluster of labelled class. This way we model connections among all samples and preserve local neighborhood information when project high-dimensional data to a low-dimensional feature space by combining both labelled and unlabelled samples.

Our second proposed semi-supervised method aims to build a semi-supervised graph which can maximize the class discrimination and preserve the local neighborhood information by combining labelled and unlabelled samples (unsupervised graph means the graph is built by using unlabelled samples, semi-supervised graph means the graph is built by using both labelled and unlabelled samples). In our semi-supervised graph, we connect samples according to either label information (labelled samples) or their k-nearest neighbors (unlabelled samples). We connect an unlabelled sample with labelled samples in a class by minimizing the mean distance of the unlabelled samples to the selected labelled samples of each class. The proposed SEGL does feature extraction based on graph learning, this is different from SELD [Liao 13] and ISELD [Luo 15] which do feature extraction by maximizing between-class scatter and minimizing within-class scatter. Moreover, the proposed SEGL method does not set the same weights to the edges of the same class or samples within their k-nearest neighbors, as [Liao 13, Luo 15, Zhang 10] do, but employs weighted edges (with weights corresponding to the distance between samples).

This way we proposed a more general framework to build a semi-supervised graph, where the actual differences and similarities between samples are better modelled.

Actually, in high resolution hyperspectral images, not only detailed information on spectral reflectance characteristics of different materials can be used for classification, but also the spatial portrayal (e.g. structure of objects) with fine spatial resolution enables us to extract the spatial information (as pixels belong to the same object in spatial space), which increases the possibility of more precisely discriminating objects on the Earth's surface. Therefore, fusion of spectral and spatial features for land cover classification is an important research topic in hyperspectral remote sensing community.

the references [Zhou 15, Huang 13] explore the importance of joint spectral and spatial information for hyperspectral image analysis, and their results have demonstrated that combining spectral and spatial features can improve the accuracy of land cover classification. Zhou *et al.* [Zhou 15] integrated a spectral-domain regularized local preserving scatter matrix and a spatial-domain local pixel neighborhood preserving scatter matrix. Refs. [Huang 13], [Ghamisi 14b] and [Zhang 12] concatenate spectral and spatial features (morphological profiles) in a simple stacked architecture. Their methods treat spectral-spatial features equally and ignore complementary information provided by heterogeneous features, leading to even worse performances than using a single feature [Zhang 12, Huang 13].

In 2016, Liao *et al.* [Liao 16] proposed morphological attribute profiles with partial reconstruction (APPRs) which can separate the connected objects, these APPR can better model the spatial information of objects and are more robust on selecting values for different attributes. APPR also provides complementary information (in spatial) for spectral information for classification. However, in our proposed method SEGL, only spectral and label information are used. Spatial information (as in APPR) has not been considered. Therefore, based on the findings above, we improve the SEGL method by taking into account spectral, spatial (APPR) and label information. In this improved SEGL (ISEGL), labelled samples are connected still according to either label information, while unlabelled samples are connected according to their k-nearest neighbors not only in spectral feature space but also in spatial feature space. Furthermore, we link an unlabelled sample with all labelled samples in the class which is closest to this unlabelled sample in both spectral and spatial feature space. Thus, our proposed method better models the correlations between samples and preserves local geometrical structure information in both spectral and spatial feature spaces. What's more, as spectral features in raw hyperspectral image and APPR generated from hyperspectral image have different meanings and properties, and the information contained in them is not equally represented, if they are stacked first and then transformed to a low-dimensional space together, some important features would be lost or mixed. Therefore we extract the low dimensional spectral features from hyperspectral image and spatial features from APPR separately based on the proposed semi-supervised

graph, and finally fused the extracted spectral and spatial features for classification.

Table 3.1: Some notations used in this Chapter.

Notations	Description
hyperspectral image	raw data cube: $M \times N \times D$, $M \times N$ is the size of image, D is number of bands
n	number of labeled training samples
m	number of labeled training samples
C	number of classes
n_c	number of training samples in class c
$\mathbf{x}_i^L$	ith labelled training sample in hyperspectral image, $\mathbf{x}_i^L \in \mathbb{R}^D$
$\mathbf{x}_j^U$	jth unlabelled training sample in hyperspectral image, $\mathbf{x}_i^U \in \mathbb{R}^D$
y_i	label of sample $\mathbf{x}_i^L$
c_j	the class closest to unlabelled sample $\mathbf{x}_j^U$
$\mathbf{X}^{(c_j)}$	the labelled training samples set in class c_j
$m_c(\mathbf{x}_j^U)$	mean distance between $\mathbf{x}_j^U$ and class c
$\mathbf{A}$	$(n+m) \times (n+m)$ adjacency matrix between samples
$A_{i,j}^{LU}$	the edge between $\mathbf{x}_i^L$ and $\mathbf{x}_j^U$

3.2 Improved Semi-supervised Local Discriminant Analysis (ISELD)

Semi-supervised local discriminant analysis (SELD) [Liao 13] combines unsupervised linear local feature extraction and supervised linear discriminant analysis (LDA) methods in a way that adapts automatically to the fraction of the samples without any parameters. This approach employs the labelled samples through only the supervised linear discriminant analysis (LDA) and the unlabelled ones through only unsupervised method, then combined them both in a non-linear way, which makes full use of the advantages of both approaches. However, SELD has not exploited the relationships between labelled and unlabelled samples. Therefore, we proposed an improved SELD by adding the relationships between labelled and unlabelled samples, and better model

the similarities between samples. Before detailing the proposed method, we will first introduce SELD briefly.

3.2.1 Semi-supervised Local Discriminant Analysis (SELD)

Suppose a training data set $\mathbf{X} = \{\mathbf{X}_{labelled}, \mathbf{X}_{unlabelled}\} = \{\mathbf{x}_1^L, \mathbf{x}_2^L, \cdots, \mathbf{x}_n^L, \mathbf{x}_{n+1}^U, \mathbf{x}_{n+2}^U, \cdots, \mathbf{x}_{n+m}^U\}$, with labelled set $\mathbf{X}_{labelled} = \{\mathbf{x}_i^L\}_{i=1}^n$, y_i is the label of $\mathbf{x}_i^L$ and $y_i \in \{1, 2, \cdots, C\}$, C is the number of classes, and unlabelled set $\mathbf{X}_{unlabelled} = \{\mathbf{x}_{n+i}^U\}_{i=1}^m$, n is the number of labelled samples, m is the number of unlabelled samples, the class c has n_c samples with $\sum_{k=1}^C n_c = n$. One sample $\mathbf{x}_i^L$ or $\mathbf{x}_i^U$ corresponding one pixel in hyperspectral image which with D bands, then it can be seen as a vector with D elements, that is $\mathbf{x}_i^L \in \mathbb{R}^D, \mathbf{x}_i^U \in \mathbb{R}^D$. Assume that the labelled samples in $\mathbf{X}_{labelled} = \{\mathbf{x}_1^L, \mathbf{x}_2^L, \cdots, \mathbf{x}_n^L\}$ are ordered according to their labels, with the data matrix of the cth class $\mathbf{X}^{(c)} = \{\mathbf{x}_1^{(c)}, \mathbf{x}_2^{(c)}, \cdots, \mathbf{x}_{n_c}^{(c)}\}$, where $\mathbf{x}_i^{(c)}$ is ith sample in cth class. Then the labelled set can be expressed as $\mathbf{X}_{labelled} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(C)}\}$.

In order to find a projection which can preserve local neighborhood information and maximize the class discrimination of the data. Liao *et al.* [Liao 13] proposed SELD by combining an unsupervised method (from the class of local linear feature extraction methods, such as neighborhood preserving embedding (NPE)) and a supervised method LDA without any tuning parameters. The main idea of this approach is first to divide the samples into two sets: labelled and unlabelled. Then, the labelled samples are used to discover the global class discriminant of the data by LDA, while the unlabelled samples are used to preserve the local neighborhood spatial structure by NPE.

In the supervised part, the objective function of LDA can be written as:

$$\mathbf{w}_{LDA} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{X}_{labelled} \overline{\mathbf{C}^{LL}} (\mathbf{X}_{labelled})^T \mathbf{w}}{\mathbf{w}^T \mathbf{X}_{labelled} \underline{\mathbf{C}^{LL}} (\mathbf{X}_{labelled})^T \mathbf{w}} \tag{3.1}$$

where $\overline{\mathbf{C}^{LL}} = \mathbf{P}_{n\times n}$, $\underline{\mathbf{C}^{LL}} = \mathbf{I}_{n\times n} - \mathbf{P}_{n\times n}$, and matrix $\mathbf{P}_{n\times n}$ is defined as:

$$\mathbf{P}_{n\times n} = \begin{bmatrix} \mathbf{P}^{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^{(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P}^{(C)} \end{bmatrix}$$

$\mathbf{P}^{(c)}$ is a $n_c \times n_c$ matrix with all the elements equal to $\frac{1}{n_c}$.

Equivalently, in the unsupervised part, the NPE component which only uses the unlabelled samples is formulated:

$$\mathbf{w}_{NPE} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{X}_{unlabelled} \overline{\mathbf{C}^{UU}} (\mathbf{X}_{unlabelled})^T \mathbf{w}}{\mathbf{w}^T \mathbf{X}_{unlabelled} \underline{\mathbf{C}^{UU}} (\mathbf{X}_{unlabelled})^T \mathbf{w}} \tag{3.2}$$

where $\overline{\mathbf{C}^{UU}} = \mathbf{I}_{m\times m}$, $\underline{\mathbf{C}^{UU}} = (\mathbf{I}_{m\times m} - \mathbf{M}_{m\times m})^T(\mathbf{I}_{m\times m} - \mathbf{M}_{m\times m})$, and $\mathbf{M}$ denotes the weight matrix with M_{ij} being the edge from unlabelled sample $\mathbf{x}_i$ to unlabelled sample $\mathbf{x}_j$.

In order to make full use of the strengths of both two methods without parameter optimization, SELD used a natural way to combine them. First by fixing the matrix $\overline{\mathbf{S}}_{SELD}$ and $\underline{\mathbf{S}}_{SELD}$ as follows:

$$\begin{aligned}\overline{\mathbf{S}}_{SELD} &= \mathbf{X}_{labelled}\overline{\mathbf{C}^{LL}}\mathbf{X}_{labelled}^T + \mathbf{X}_{unlabelled}\overline{\mathbf{C}^{UU}}\mathbf{X}_{unlabelled}^T \\ &= [\mathbf{X}_{labelled}\ \mathbf{X}_{unlabelled}]\begin{bmatrix}\overline{\mathbf{C}^{LL}} & \mathbf{0} \\ \mathbf{0} & \overline{\mathbf{C}^{UU}}\end{bmatrix}[\mathbf{X}_{labelled}\ \mathbf{X}_{unlabelled}]^T\end{aligned}$$

$$\begin{aligned}\underline{\mathbf{S}}_{SELD} &= \mathbf{X}_{labelled}\underline{\mathbf{C}^{LL}}\mathbf{X}_{labelled}^T + \mathbf{X}_{unlabelled}\underline{\mathbf{C}^{UU}}\mathbf{X}_{unlabelled}^T \\ &= [\mathbf{X}_{labelled}\ \mathbf{X}_{unlabelled}]\begin{bmatrix}\underline{\mathbf{C}^{LL}} & \mathbf{0} \\ \mathbf{0} & \underline{\mathbf{C}^{UU}}\end{bmatrix}[\mathbf{X}_{labelled}\ \mathbf{X}_{unlabelled}]^T\end{aligned}$$

and then

$$\mathbf{w}_{SELD} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{X}\overline{\mathbf{C}}_{SELD}\mathbf{X}^T\mathbf{w}}{\mathbf{w}^T\mathbf{X}\underline{\mathbf{C}}_{SELD}\mathbf{X}^T\mathbf{w}} \tag{3.3}$$

where, $\mathbf{X} = \{\mathbf{X}_{labelled}, \mathbf{X}_{unlabelled}\}$, and the relationship matrix $\overline{\mathbf{C}}_{SELD}$ and $\underline{\mathbf{C}}_{SELD}$ are given as:

$$\overline{\mathbf{C}}_{SELD} = \begin{bmatrix}\overline{\mathbf{C}^{LL}} & \mathbf{0} \\ \mathbf{0} & \overline{\mathbf{C}^{UU}}\end{bmatrix}, \underline{\mathbf{C}}_{SELD} = \begin{bmatrix}\underline{\mathbf{C}^{LL}} & \mathbf{0} \\ \mathbf{0} & \underline{\mathbf{C}^{UU}}\end{bmatrix} \tag{3.4}$$

3.2.2 Proposed ISELD

As we can see from equation (3.3), SELD [Liao 13] infers class by only using labelled samples with matrices $\overline{\mathbf{C}^{LL}}$ and $\underline{\mathbf{C}^{LL}}$. To preserve local neighborhood information preservation, SELD utilizes only unlabelled samples through the matrices $\overline{\mathbf{C}^{UU}}$ and $\underline{\mathbf{C}^{UU}}$. The correlation matrices of labelled and unlabelled samples are set to 0 in SELD. This means the relationship between labelled and unlabelled samples is not well modelled by SELD. When very limited labelled samples are available (cannot effectively express class discrimination) or a small number of unlabelled samples (cannot effectively express local geometrical structure) are selected, neither class discrimination nor local neighborhood information can be well exploited. Therefore, we propose an Improved SELD (ISELD) method, in which the correlation matrices $\overline{\mathbf{C}}_{ISELD}$ (for between-class scatter matrix) and $\underline{\mathbf{C}}_{ISELD}$ (for within-class scatter matrix) are defined as:

$$\overline{\mathbf{C}}_{ISELD} = \begin{bmatrix}\overline{\mathbf{C}^{LL}} & \overline{\mathbf{C}^{LU}} \\ (\overline{\mathbf{C}^{LU}})^T & \overline{\mathbf{C}^{UU}}\end{bmatrix}, \underline{\mathbf{C}}_{ISELD} = \begin{bmatrix}\underline{\mathbf{C}^{LL}} & \underline{\mathbf{C}^{LU}} \\ (\underline{\mathbf{C}^{LU}})^T & \underline{\mathbf{C}^{UU}}\end{bmatrix}$$

here $\overline{\mathbf{C}^{LL}}$ and $\overline{\mathbf{C}^{LL}}$ are defined as the same to SELD, $\overline{\mathbf{C}^{LU}}$ and $\underline{\mathbf{C}^{LU}}$ are $n \times m$ matrices to model the correlation of labelled and unlabelled samples for between- and within-class scatter matrix, respectively.

In order to well define $\overline{\mathbf{C}^{LU}}$ and $\underline{\mathbf{C}^{LU}}$, we first try to find the nearest class of each unlabelled sample $\mathbf{x}_j^U$. Suppose class c_j represents the class that $\mathbf{x}_j^U$ is closest to, and $\mathbf{X}^{(c_j)}$ is a set including all labelled samples in class c_j, $c_j \in [1, C]$. $\mathbf{X}^{(c_j)}$ is obtained as follows:

$$c_j = \arg\min_c m_c(\mathbf{x}_j^U), c = 1, 2, \cdots, C \tag{3.5}$$

where $m_c(\mathbf{x}_j^U)$ denotes mean distance between unlabelled sample $\mathbf{x}_j^U$ and all labelled samples in class c.

$$m_c(\mathbf{x}_j^U) = \frac{1}{n_c} \sum_{t=1}^{n_c} d(\mathbf{x}_j^U, \mathbf{x}_t^{(c)}) \tag{3.6}$$

$d(\mathbf{x}_i, \mathbf{x}_j)$ is Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, if $m_k(\mathbf{x}_j^U)$ is smaller, it means $\mathbf{x}_j^U$ is closer to class c, and more similar to the samples in class c.

In SELD [Liao 13], the similarity of labelled samples $\mathbf{x}_j^L$ and $\mathbf{x}_j^L$ are modelled by $\overline{C}_{L_i L_j}$ (for between-class scatter) and $\underline{C}_{L_i L_j}$ (for within-class scatter), and they are set to $(1/n_c)$ and $(-1/n_c)$ respectively if $\mathbf{x}_i^L$ and $\mathbf{x}_j^L$ belong to the same class, otherwise they will be set to 0. If the unlabelled sample $\mathbf{x}_j^U$ is closest to class c_j, it could be semi-defined that $\mathbf{x}_j^U$ belong to class c_j, then $\overline{C_{i,j}^{LU}}$ and $\underline{C_{i,j}^{LU}}$ could be set to $(1/n_c)$ and $(-1/n_c)$ if $\mathbf{x}_i^L$ belongs to class c_j, same as SELD do. As a result, $\overline{C_{i,j}^{LU}}$ and $\underline{C_{i,j}^{LU}}$ can be written in mathematics as:

$$\overline{C_{i,j}^{LU}} = \begin{cases} 1/n_c & , \mathbf{x}_i^L \in \mathbf{X}^{(c_j)} \\ 0 & , \mathbf{x}_i^L \notin \mathbf{X}^{(c_j)} \end{cases} \tag{3.7}$$

$$\underline{C_{i,j}^{LU}} = \begin{cases} -1/n_c & , \mathbf{x}_i^L \in \mathbf{X}^{(c_j)} \\ 0 & , \mathbf{x}_i^L \notin \mathbf{X}^{(c_j)} \end{cases} \tag{3.8}$$

$\overline{C_{i,j}^{LU}}$ and $\underline{C_{i,j}^{LU}}$ denote the similarity of labelled sample $\mathbf{x}_i^L$ and unlabelled sample $\mathbf{x}_j^U$ for between- and within-class scatter matrix respectively, $\mathbf{X}^{(c_j)}$ is the group of class c_j which is closest to $\mathbf{x}_j^U$.

Given all of these notations, the ISELD transformation matrix $\mathbf{W}_{ISELD}$ can be obtained by solving the following cost function:

$$\mathbf{w}_{ISELD} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^T \overline{\mathbf{S}}_{ISELD} \mathbf{w}}{\mathbf{w}^T \underline{\mathbf{S}}_{ISELD} \mathbf{w}} \tag{3.9}$$

where

$$\overline{\mathbf{S}}_{ISELD} = [\mathbf{X}_{labelled}, \mathbf{X}_{unlabelled}] \begin{bmatrix} \overline{\mathbf{C}^{LL}} & \overline{\mathbf{C}^{LU}} \\ (\overline{\mathbf{C}^{LU}})^T & \overline{\mathbf{C}^{UU}} \end{bmatrix} [\mathbf{X}_{labelled}, \mathbf{X}_{unlabelled}]^T = \mathbf{X}\overline{\mathbf{C}}_{ISELD}\mathbf{X}^T$$

$$\underline{\mathbf{S}}_{ISELD} = [\mathbf{X}_{labelled}, \mathbf{X}_{unlabelled}] \begin{bmatrix} \underline{\mathbf{C}^{LL}} & \underline{\mathbf{C}^{LU}} \\ (\underline{\mathbf{C}^{LU}})^T & \underline{\mathbf{C}^{UU}} \end{bmatrix} [\mathbf{X}_{labelled}, \mathbf{X}_{unlabelled}]^T = \mathbf{X}\underline{\mathbf{C}}_{ISELD}\mathbf{X}^T$$

To obtain the projection matrix, we solve the generalized eigenvalue problem of the proposed ISELD method as:

$$\overline{\mathbf{S}}_{ISELD}\mathbf{w} = \lambda\underline{\mathbf{S}}_{ISELD}\mathbf{w} \tag{3.10}$$

The vectors $\{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d\}$ which solve equation (3.10) are the columns of $\mathbf{W}_{ISELD}$. The columns are ordered according to their eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_d$. Thus, the embedding is as follows:

$$\mathbf{x}_i \rightarrow \mathbf{z}_i = \mathbf{W}_{ISELD}^T\mathbf{x}_i$$

Through this projection, the local neighborhood information of the data can be best preserved, while simultaneously the class discrimination is maximal.

3.2.3 Experiments and Results

The hyperspectral image dataset we used in the experiments is the University of Pavia. It is an urban image that was captured by the ROSIS optical sensor around the Engineering School at the University of Pavia. This hyperspectral image contains 103 bands, 9 classes, and 610×340 pixels with a spatial resolution of 1.3 meter. For the classifier, we adapt the k-Nearest Neighbor (here k equals to 1) as classifier. In order to investigate the impact of the labelled samples on the classification accuracy, we randomly select the labelled samples from the training set with the sample size corresponding to different cases: 10, 20, 40, 80 per class. 1500 samples are randomly selected from the original hyperspectral image, similarly as [Liao 13]. Each experiment was repeated 5 times. In our experiments, we change the number of extracted features from 2 to 20, and record the best result of each method.

The experimental results for different methods are summarized in Figure 3.1-Figure 3.2 and Table 3.2-Table 3.3. Table 3.2 shows that ISELD performs best when classifying the class 'trees', "metal sheets", "soil" and "bitumen". It also can be seen from Table 3.3 that the classification accuracies of PCA and NPE remains stable as the number of labelled samples increases, this is due to their unsupervised nature. The supervised method NWFE outperforms the unsupervised methods, when the number of labelled samples is small, the highest overall classification accuracy (OA) is 71.8%, which is much higher than

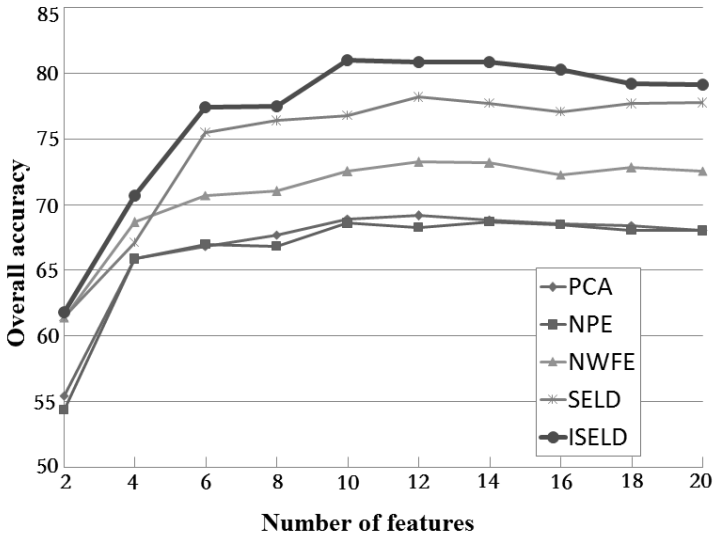


Figure 3.1: Comparison of overall classification accuracy OA, as the number of the extracted features increases using 40 labelled samples. Each experiment was repeated 5 times, the average is acquired.

Table 3.2: Overall classification accuracy (OA%) for each class by using different feature extraction methods with 40 labelled training samples per class.

Class	Train/Test	PCA	NPE	NWFE	SELD	ISELD
Asphalt	40/6631	67.8	68.1	**70.4**	68.2	69.2
Meadows	40/18649	62.6	62.5	65.9	**80.7**	79.7
Gravel	40/2099	62.7	63.0	74.3	67.9	**77.5**
Trees	40/3064	94.2	94.2	94.3	94.3	**95.7**
Metal sheets	40/1345	99.7	99.7	99.9	**100.0**	**100.0**
Soil	40/5029	67.2	67.5	72.1	85.6	**86.6**
Bitumen	40/1330	86.6	87.3	88.7	85.2	**88.9**
Bricks	40/3682	76.7	76.6	**78.5**	74.4	76.6
Shadow	40/947	98.9	97.9	98.8	97.8	**99.3**
OA(%)	-	69.6	68.7	73.8	78.9	**81.4**
AA(%)	-	79.6	79.7	82.5	84.5	**85.4**

Table 3.3: Overall classification accuracy (OA%) and optimal number of features (in bracket) for different feature extraction methods with different training sample size.

Methods	$n_k = 10$	$n_k = 20$	$n_k = 40$	$n_k = 80$
PCA	68.4(8)	69.2(10)	69.6(12)	69.9(12)
NPE	66.7(14)	68.0(12)	68.7(16)	69.8(10)
NWFE	71.8(10)	72.5(14)	73.8(12)	74.9(12)
SELD	74.6(8)	77.8(8)	78.9(12)	81.2(16)
ISELD	**77.6(10)**	**80.2(8)**	**81.4(10)**	**83.1(10)**

PCA and NPE. Even when the number of extracted features is as low as 2, the results look encouraging (above 60%). However, NWFE performs worse than semi-supervised methods. The semi-supervised method SELD performs well and its classification results keep stable when the number of features reaches 6. When the number of labelled training samples per class gets to 80, its highest OA could be over 81%.

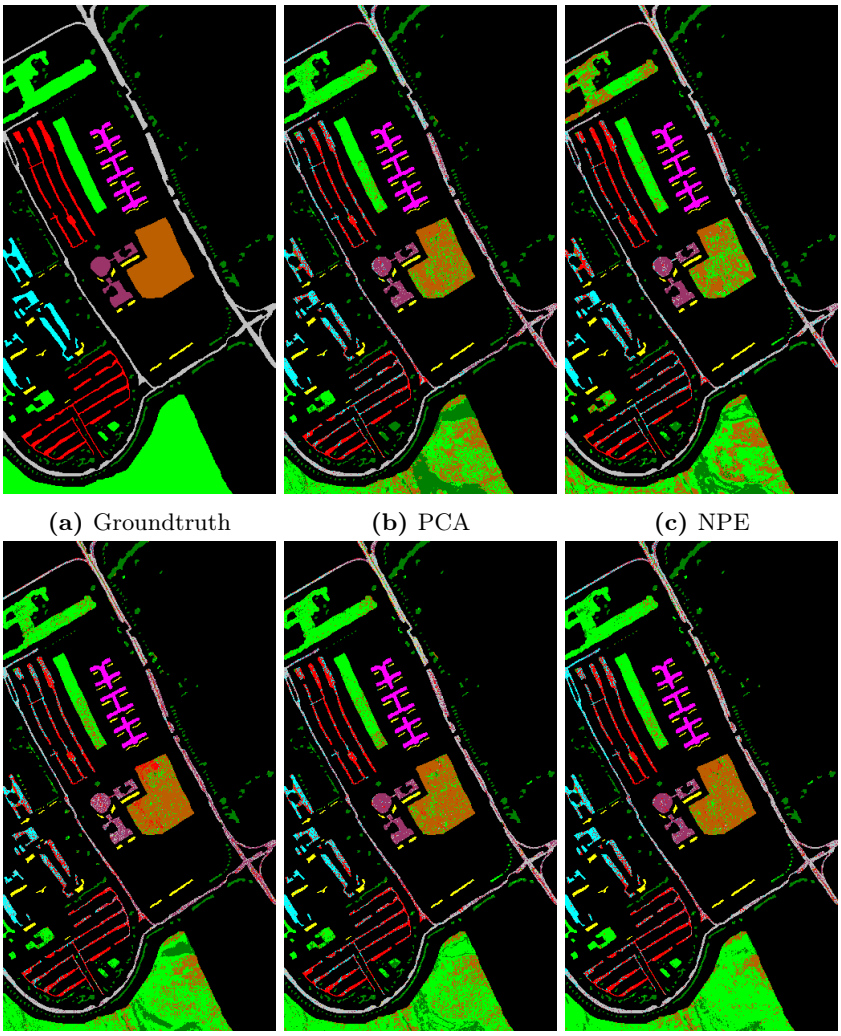
By modelling the correlation of labelled and unlabelled samples, the proposed method ISELD performs much better than SELD even when very limited labelled samples. We can see from Table 3.3 that the highest OA for our method are above 80% even with 20 labelled samples per class, which is much better than the other methods.

3.3 Semi-supervised Graph Learning (SEGL)

3.3.1 Proposed SEGL

In many applications, labelled samples are typically used to enhance class discrimination, but are always very limited. Unlabelled samples, on the other hand, are much easier accessible. The idea behind semi-supervised feature extraction methods [Liao 13] and [Luo 15] is to infer class discrimination from labelled samples, as well as the local neighborhood information from unlabelled samples. This section details our proposed semi-supervised graph learning method (SEGL) for feature extraction in hyperspectral images.

We exploit the label information and local neighborhood information through our proposed semi-supervised graph, and our proposed semi-supervised graph is defined as $\mathbf{G} = (\mathbf{X}, \mathbf{A})$, $\mathbf{X} = \{\mathbf{X}_{labelled}, \mathbf{X}_{unlabelled}\} = \{\mathbf{x}_1^L, \mathbf{x}_2^L, \cdots, \mathbf{x}_n^L, \mathbf{x}_{n+1}^U, \mathbf{x}_{n+2}^U, \cdots, \mathbf{x}_{n+m}^U\}$ is a set of nodes which connected by a set of edges $A_{i,j}$, $A_{i,j}$ is the edge (with a weight between 0 and 1) between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. The basic goal of our proposed method is to find a transformation matrix $\mathbf{W} \in \mathbb{R}^{D \times d}$, which can transform the data $\mathbf{x}_i$ from a high-dimensional feature space into a low-dimensional data $\mathbf{z}_i = \mathbf{W}^T \mathbf{x}_i$. The transformation matrix $\mathbf{W}$ can be calculated from cost function as following:



(a) Groundtruth **(b)** PCA **(c)** NPE

(d) NWFE **(e)** SELD **(f)** ISELD

Figure 3.2: Classification maps of the different methods. 40 labelled samples per class were randomly selected from the training set.

$$\arg\min_{\mathbf{w}}\left(\sum_{i,j=1}^{n+m}||\mathbf{W}^T\mathbf{x}_i-\mathbf{W}^T\mathbf{x}_j||^2A_{ij}\right) \tag{3.11}$$

Motivated by [Liao 13] and [Luo 15], which divide training samples into two groups, and then define between- and within-class scatter matrices by combing information from these two groups, we proposed a new semi-supervised feature

extraction method in the view of graph learning. In our proposed method, a semi-supervised graph is built to model different correlations between samples as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{LL} & \mathbf{A}^{LU} \\ \mathbf{A}^{UL} & \mathbf{A}^{UU} \end{bmatrix} \tag{3.12}$$

$\mathbf{A}$ is an adjacency matrix, the top left part $\mathbf{A}^{LL}$ is an $n \times n$ matrix that models the correlations between labelled samples, the bottom right part $\mathbf{A}^{UU}$ is an $m \times m$ matrix that models the correlations between unlabelled samples. Two labelled samples are connected if they belong to same class, two unlabelled samples are connected if they are within the k-nearest neighbors of others. Therefore, $\mathbf{A}^{LL}$ and $\mathbf{A}^{UU}$ can be defined as:

$$A_{i,j}^{LL} = \begin{cases} 1 & , y_i = y_j \\ 0 & , y_i \neq y_j \end{cases} \tag{3.13}$$

$$A_{i,j}^{UU} = \begin{cases} 1 & , \mathbf{x}_j^U \in knn(\mathbf{x}_i^U) \text{ or } \mathbf{x}_i^U \in knn(\mathbf{x}_j^U) \\ 0 & , \mathbf{x}_j^U \notin knn(\mathbf{x}_i^U) \text{ and } \mathbf{x}_i^U \in knn(\mathbf{x}_j^U) \end{cases} \tag{3.14}$$

where $knn(\mathbf{x}_i^U)$ denotes a set of unlabelled samples that are within the k nearest neighbors of $\mathbf{x}_i^U$.

The adjacency matrices $\mathbf{A}^{LU}$ and $\mathbf{A}^{UL}$ contain the connection between labelled and unlabelled samples, $\mathbf{A}^{UL} = (\mathbf{A}^{LU})^T$, as $\mathbf{A}$ is a symmetric matrix. Suppose the labelled sample $\mathbf{x}_i^L$ belongs to class c_j, the $n \times m$ adjacency matrix $\mathbf{A}^{LU}$ is defined as:

$$A_{i,j}^{LU} = \begin{cases} 1 & , \mathbf{x}_i^L \in \mathbf{X}^{(c_j)} \\ 0 & , \mathbf{x}_i^L \notin \mathbf{X}^{(c_j)} \end{cases} \tag{3.15}$$

$$c_j = \arg\min_c m_c(\mathbf{x}_j^U), c = 1, 2, \cdots, C \tag{3.16}$$

$$m_c(\mathbf{x}_j^U) = \frac{1}{n_c} \sum_{t=1}^{n_c} d, (\mathbf{x}_j^U, \mathbf{x}_t^{(c)}) \tag{3.17}$$

where c_j represents the class that $\mathbf{x}_j^U$ is closest to, and $\mathbf{X}^{(c_j)}$ is a set including all labelled samples in class c_j, $c_j \in \{1, \ldots, C\}$. $m_c(\mathbf{x}_j^U)$ denotes the mean distance of an unlabelled sample $\mathbf{x}_j^U$ to all labelled samples in class c. $d(\mathbf{x}_i, \mathbf{x}_j)$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. A smaller $m_c(\mathbf{x}_j^U)$ means that $\mathbf{x}_j^U$ is more similar to labelled samples from class c and more likely to belong to class c, we set edges between an unlabelled sample $\mathbf{x}_j^U$ and all labelled samples $\mathbf{x}_j^L$ in class c. Here we assume that the distribution of each class is unimodal [Sugiyama 07].

In the above definition of the adjacency matrix $\mathbf{A}$, if two nodes are connected, their edges $A_{i,j}$ are set to 1. If we set the same edges for pairwise samples, the connected pairwise samples would make same contribution the definition of the cost function (equation (3.11)). The actual differences and similarities between samples are not well modelled. Therefore we set a weighted edge between connected nodes by combing the affinity matrix $\mathbf{F}$ with adjacency matrix $\mathbf{A}$, to make the most similar connected samples be the closest, and less similar connected samples be more separated when projecting them onto the low-dimensional feature space. The final similarity matrix $\mathbf{A}$ in our proposed semi-supervised graph $\mathbf{G} = (\mathbf{X}, \mathbf{A})$ is defined as follows:

$$\mathbf{A} = \mathbf{F} \odot \mathbf{A} \tag{3.18}$$

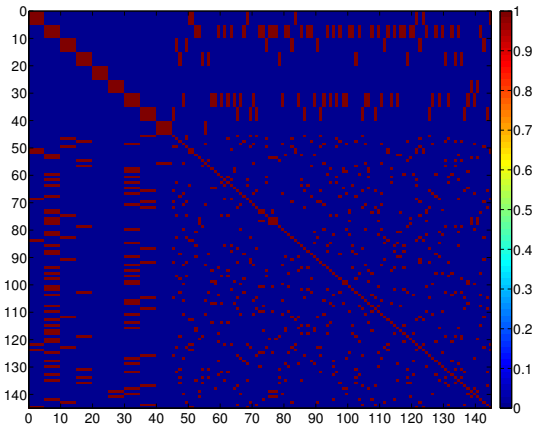
where "$\odot$" denoting element-wise multiplication, $F_{i,j}$ is the affinity between $\mathbf{x}_i$ and $\mathbf{x}_j$. In this paper we use the local scaling heuristic [Zelnik 05] as the definition of affinity matrix $\mathbf{F}$, i.e.,

$$F_{i,j} = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\delta_i \delta_j}\right). \tag{3.19}$$

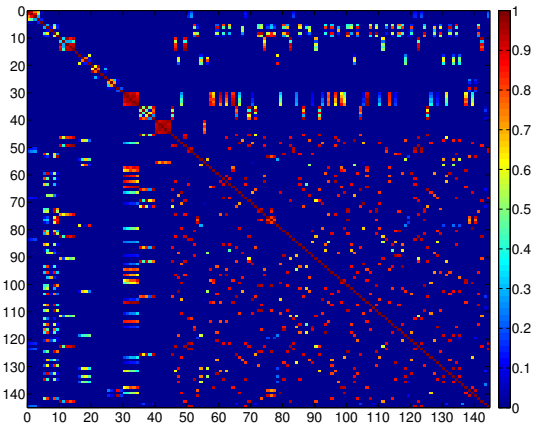
$\|\mathbf{x}_i - \mathbf{x}_j\|$ means Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, the parameter δ_i controls the local "scaling" around $\mathbf{x}_i$ defined by $\delta_i = \left\|\mathbf{x}_i - \mathbf{x}_i^k\right\|$, and $\delta_j = \left\|\mathbf{x}_j - \mathbf{x}_j^k\right\|$, $\mathbf{x}_i^k$ is the k-th nearest neighbour of $\mathbf{x}_i$. A heuristic choice of $k = 7$ was shown to be useful through experiments [Zelnik 05]. $F_{i,j}$ is large if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close", and $F_{i,j}$ is small if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "far apart".

Figure 3.3 shows the graph constructed by our proposed semi-supervised method. The graph was constructed by selecting 5 labelled samples per class and 100 unlabelled samples randomly from the University of Pavia data set. Without F, the edges of all connected samples are equally set to 1, as shown Figure 3.3(a). However, in real applications, samples even from the same class, have spectral differences, as shown Figure 3.3(b). If we set all the weights of the samples from the same labelled class to the same value as SELD [Liao 13] does, the differences and similarities cannot be well modelled. With our proposed semi-supervised graph, the differences and similarities are much better modelled. This means two labelled samples $\mathbf{x}_i$ and $\mathbf{x}_j$, that are closer to each other have a larger connection weight A_{ij}. On the contrary, if they are far away from each other or mislinked, their connection weight A_{ij} would be smaller, which reduces the negative influence of mislinking, as Figure 3.3 (b). Therefore, adding weights for connected pairwise samples as equation (3.18) can better model correlations of samples.

After the adjacency matrix $\mathbf{A}$ is conformed, by simple algebra formulation,



(a) Graph with same weights



(b) Proposed graph with different weights

Figure 3.3: Semi-Supervised graph.

the cost function (equation (3.11)) can be reduced to:

$$\begin{aligned}\frac{1}{2}\sum_{ij}(\mathbf{z}_i-\mathbf{z}_j)^2A_{ij} &= \frac{1}{2}\sum_{ij}(\mathbf{w}^T\mathbf{x}_i-\mathbf{w}^T\mathbf{x}_j)^2A \\ &= \sum_i \mathbf{w}^T\mathbf{x}_i D_{ii}\mathbf{x}_i^T\mathbf{w} - \sum_{ij}\mathbf{w}^T\mathbf{x}_i A_{ij}\mathbf{x}_j^T\mathbf{w} \\ &= \mathbf{w}^T\mathbf{X}(\mathbf{D}-\mathbf{A})\mathbf{X}^T\mathbf{w} \\ &= \mathbf{w}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w}\end{aligned} \tag{3.20}$$

where $\mathbf{D}$ is a diagonal matrix with $D_{i,i} = \sum_{j=1}^{m+n} A_{i,j}$, $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the Laplacian matrix [Belkin 02]. Matrix $\mathbf{D}$ provides a natural measure on the data points, the bigger the value $D_{i,i}$ (corresponding to $\mathbf{z}_i$) is, the more "important" is $\mathbf{z}_i$. Therefore, in order to avoid degeneracy, we impose a constraint as follows:

$$\mathbf{W}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{W} = \mathbf{I}, \tag{3.21}$$

$\mathbf{I}$ is the identity matrix.

The transformation matrix a that minimizes the cost function is given by the minimum eigenvalue solution to the generalized eigenvalue problem:

$$\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{w} = \lambda\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{w} \tag{3.22}$$

Supposing that $\mathbf{w}_1, \mathbf{w}_2, \cdots$ are the solutions of above equation, the optimal transformation matrix $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d\}$ is made up by d eigenvectors associated with the least d eigenvalues $\lambda_1 < \lambda_2 < \cdots < \lambda_d$ of above generalized eigenvalue problem. Thus, the high-dimensional sample $\mathbf{x}_i \in \mathbb{R}^D$ can be mapped to low-dimensional sample $\mathbf{z}_i \in \mathbb{R}^d$ as $\mathbf{z}_i = \mathbf{W}^T\mathbf{x}_i$.

3.3.2 Experiments and Results

3.3.2.1 Hyperspectral Image Data Sets

Four real hyperspectral data sets are used in our experiments: 'University of Pavia', 'Pavia Center', 'Botswana' and 'Kennedy Space Center'. Table 3.4 shows the number of labelled samples in each class for all the data sets. Note that the color in the cell denotes different classes in the classification maps (Figure 3.5 and Figure 3.6). The data sets University of Pavia and Pavia Center, are from urban areas in the city of Pavia, Italy. The data were collected by the ROSIS (Reflective Optics System Imaging Spectrometer) sensor, with 115 spectral bands in the wavelength range from 0.43 to 0.86 μm, and very fine spatial resolution of 1.3 meters by pixel. For more details of University of Pavia, please see section 3.2.3 and Table 3.4.

Pavia Center (*PCenter*): The data with 1096×715 pixels was collected over Pavia city center, Italy. It contains 102 spectral channels after removal of 13 noisy bands. Nine groundtruth classes were considered in experiments, see Table 3.4.

Kennedy Space Center (*KSC*): the data set was acquired by NASA AVIRIS instrument over the KSC, Florida in 1996 and consists of 224 bands of 10-nm width with center wavelengths from 0.4-2.5 μm. The data, acquired from an altitude of approximately 20 km, have a spatial resolution of 18 m/pixel. Several spectral bands were removed from the data due to noise and water absorption phenomena, leaving a total of 176 bands to be used for the analysis. For classification purposes, 13 classes representing the various land cover types that occur in this environment were defined for the site, see Table 3.4. For more information, see website http://www.csr. utexas.edu/hyperspectral/.

Table 3.4: Data Sets Used In The Experiments

No	University of Pavia		KSC		Botswana		Pavia Centre	
	Class name	Sample	Class name	Sample	Class name	Sample	Class name	Sample
1	Asphalt	6631	Scrub	761	Water	270	Water	65971
2	Meadows	18649	Willow swamp	243	Hippo grass	101	Trees	7598
3	Gravel	2099	Cabbage palm hammock	256	Floodplain Grass1	251	Meadows	3090
4	Trees	3064	Cabbage palm/oak hammock	252	Floodplain Grass2	215	Bricks	2685
5	Metal sheets	1345	Slash Pine	161	Reeds	269	Soil	6584
6	Soil	5029	Oak/broadleaf hammock	229	Riparian	269	Asphalt	9248
7	Bitumen	1330	Hardwood swamp	105	firescar2	259	Bitumen	7287
8	Bricks	3682	Graminoid marse	431	Island interior	203	Tiles	42826
9	Shadow	947	Spartan marse	520	Acacia woodlands	341	Shadow	2863
10			Cattail marse	404	Acacia shrublands	248		
11			Salt marse	419	Acacia grasslands	305		
12			Mud flats	503	Short mopane	181		
13			Water	927	Mixed mopane	268		
14					Exposed soils	95		
Total		42686		5211		3248		148155

Botswana (*Botswana*): The data set was acquired over the Okavango Delta, Botswana in May 31, 2001 by the NASA EO-1 satellite, with 30 m/pixel resolution over a 7.7-km strip in 242 bands covering the 0.4-2.5 μm portion of the spectrum in 10-nm windows. Uncalibrated and noisy bands that cover water absorption features were removed, leaving a total of 145 radiance channels to be used in the experiments. The data consist of observations from 14 identified classes intended to reflect the impact of flooding on vegetation, see Table 3.4. For more information, see http://www.csr.utexas.edu/hyperspectral/.

3.3.2.2 Experimental Setup

The training set $\mathbf{X}$ is composed of labelled subset $\mathbf{X}_{labelled}$ and unlabelled subset $\mathbf{X}_{unlabelled}$ (such that $\mathbf{X} = \mathbf{X}_{labelled} \cup \mathbf{X}_{unlabelled}$, and $\mathbf{X}_{labelled} \cup \mathbf{X}_{unlabelled} = \phi$). In order to analyze the influence of the size of labelled samples on classification accuracy, a number of unlabelled samples $m = 2000$ was randomly selected from the image parts with no labels to compose $\mathrm{X}_{unlabelled}$, and labelled subset $\mathrm{X}_{labelled}$ was made of labelled training samples which was randomly selected from labelled data with the samples size corresponding different case: 20 , 40 , 80 samples per class, respectively. The training of the classifiers was carried out using the labelled subset $\mathrm{X}_{labelled}$. The remaining labelled samples were used as the test set. We compare the classification accuracies using the proposed SEGL method with results from the following methods: PCA [Schott 07]; LPP [He 04]; NWFE [Kuo 04]; SDA [Cai 07], where the parameter α is optimized with fivefold cross-validation within the given set $\{0.1, 0.5, 2.5, 12.5, 62.5\}$; SELF [Sugiyama 10], of which the parameter β is chosen from $\{0, 0.1, 0.2, ..., 0.9, 1\}$ by fivefold cross-validation; SLPPCE [Zhang 10]; SELD [Liao 13] and ISELD [Luo 15].

We used three common classifiers: 1-Nearest Neighbor (1NN), Support Vector Machines (SVM) and Random Forest (RF). The SVM classifier with RBF kernels has two parameters: the penalty factor C and the RBF kernel widths γ, we optimized C within the given set $\{10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$ and γ within the given set $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$ by five-fold cross validation, the RF classifier with 200 trees. All classifiers were evaluated against the test set. Meanwhile, we use overall classification accuracy (OA) to evaluate the feature extraction results. The results were averaged over ten runs on different number of extracted features from 1 to 20, and the average OA was recorded for each method. The number of nearest neighbors was set to 8.

3.3.2.3 Results on Different Number of Labelled Training Samples

Table 3.5, Table 3.6 and Table 3.7 display the classification accuracies of testing data with different distinct labelled samples size: 20, 40 and 80 per class, respectively. The best average accuracy of each data set (in column) is highlighted in bold. From these tables, we conclude:

1. The results confirm that most semi-supervised feature extraction methods achieve better results in the classification of hyperspectral images,

Table 3.5: Overall classification accuracy (OA%) and optimal number of features (in bracket) by using different feature extraction approaches with labelled training sample size 20 per class.

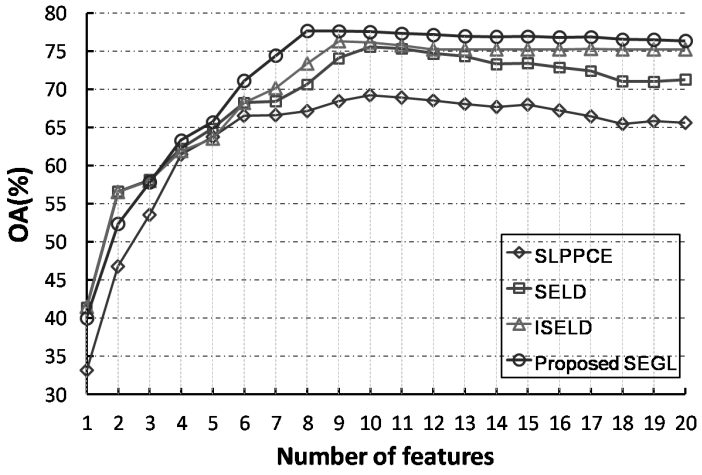
Feature Extraction	Classifier	Data Set			
		UPavia	*KSC*	*Botswana*	*PCentre*
PCA	1NN	67.38(11)	77.90(14)	86.18(14)	94.77(11)
	SVM	70.21(8)	85.66(18)	92.72(8)	96.08(12)
	RF	73.46(11)	86.40(12)	92.16(7)	95.63(11)
LPP	1NN	66.97(18)	77.93(19)	85.82(13)	94.78(13)
	SVM	71.80(9)	86.12(17)	92.15(12)	96.14(11)
	RF	72.46(18)	85.46(19)	92.13(9)	95.97(11)
NWFE	1NN	71.34(9)	85.63(17)	90.07(13)	95.92(11)
	SVM	72.29(8)	89.64(13)	92.15(10)	96.47(14)
	RF	75.84(9)	89.20(15)	92.41(12)	96.09(7)
SDA	1NN	52.67(7)	73.16(8)	72.89(11)	79.86(14)
	SVM	51.62(8)	73.87(10)	72.98(11)	79.55(12)
	RF	55.72(9)	70.56(8)	71.97(12)	79.33(8)
SELF	1NN	61.42(18)	78.79(18)	83.84(18)	93.64(13)
	SVM	63.93(19)	85.98(17)	79.20(14)	93.38(10)
	RF	67.98(18)	85.33(15)	82.64(12)	94.34(6)
SLPPCE	1NN	68.96(8)	87.18(13)	89.26(16)	93.81(8)
	SVM	67.40(8)	86.14(19)	84.04(12)	92.43(10)
	RF	66.83(9)	83.63(12)	86.33(11)	92.28(6)
SELD	1NN	77.27(17)	88.82(19)	93.91(16)	95.44(12)
	SVM	75.71(11)	89.44(16)	91.20(10)	95.42(12)
	RF	75.53(8)	88.78(13)	91.85(6)	95.59(8)
ISELD	1NN	78.66(11)	90.74(18)	93.94(18)	96.60(9)
	SVM	76.20(9)	91.09(15)	93.71(12)	96.52(7)
	RF	75.66(8)	89.54(15)	93.41(12)	96.04(7)
SEGL	1NN	**79.40(9)**	90.81(15)	**94.58(12)**	96.54(7)
	SVM	77.57(8)	**92.25(12)**	94.14(13)	**96.70(8)**
	RF	76.67(8)	89.52(13)	93.74(13)	96.29(6)

Table 3.6: Overall classification accuracy (OA%) and optimal number of features (in bracket) by using different feature extraction approaches with labelled training sample size 40 per class.

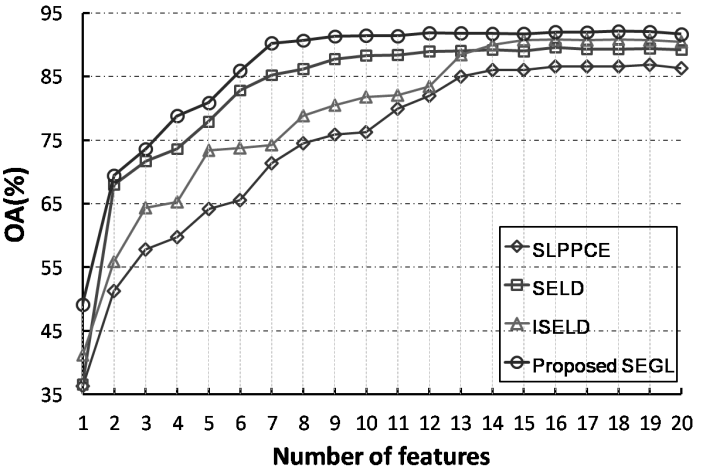
Feature Extraction	Classifier	Data Set			
		UPavia	*KSC*	*Botswana*	*PCentre*
PCA	1NN	69.46(12)	81.39(19)	87.98(9)	95.33(12)
	SVM	74.77(8)	88.53(9)	93.30(8)	96.78(11)
	RF	73.88(11)	88.87(18)	93.79(11)	96.13(11)
LPP	1NN	69.43(18)	81.20(18)	87.59(9)	95.33(13)
	SVM	77.14(12)	89.13(14)	92.42(11)	96.92(10)
	RF	74.20(19)	88.91(9)	93.26(8)	96.23(12)
NWFE	1NN	73.34(9)	87.76(14)	90.44(8)	96.32(14)
	SVM	76.62(8)	91.83(12)	93.72(8)	96.87(9)
	RF	77.03(11)	91.15(13)	93.98(12)	96.64(8)
SDA	1NN	62.64(8)	84.98(9)	85.12(12)	91.43(13)
	SVM	63.64(9)	86.75(10)	85.45(13)	89.99(14)
	RF	66.97(9)	84.98(8)	84.87(12)	90.68(12)
SELF	1NN	68.75(18)	81.86(17)	88.58(9)	95.54(11)
	SVM	75.85(19)	89.76(15)	90.54(12)	96.75(6)
	RF	75.97(12)	89.87(11)	92.79(10)	96.86(9)
SLPPCE	1NN	73.34(8)	89.77(13)	91.93(13)	95.31(9)
	SVM	68.16(10)	89.55(18)	88.70(10)	94.73(7)
	RF	67.56(11)	89.98(10)	91.34(12)	94.73(8)
SELD	1NN	79.03(10)	91.12(18)	94.69(12)	96.10(9)
	SVM	76.20(10)	92.03(15)	93.26(11)	96.68(8)
	RF	77.50(9)	91.94(12)	93.69(12)	96.45(7)
ISELD	1NN	79.79(10)	92.65(17)	94.52(18)	97.07(9)
	SVM	76.26(9)	92.78(15)	94.31(12)	96.90(6)
	RF	76.79(8)	91.91(9)	94.89(12)	96.49(7)
SEGL	1NN	**81.22(9)**	93.02(12)	**95.47(13)**	97.19(7)
	SVM	78.16(8)	**93.19(13)**	95.39(12)	**97.29(7)**
	RF	79.14(8)	92.35(12)	95.33(12)	96.59(6)

Table 3.7: Overall classification accuracy (OA%) and optimal number of features (in bracket) by using different feature extraction approaches with labelled training sample size 80 per class.

Feature Extraction	Classifier	Data Set			
		UPavia	*KSC*	*Botswana*	*PCentre*
PCA	1NN	70.23(11)	83.31(19)	89.71(9)	95.97(12)
	SVM	76.81(9)	90.50(18)	94.45(17)	96.61(11)
	RF	76.13(11)	91.13(14)	95.32(13)	96.60(8)
LPP	1NN	70.17(12)	84.77(13)	89.4(12)	95.94(13)
	SVM	77.95(18)	89.82(8)	94.02(15)	95.52(9)
	RF	75.82(19)	90.90(12)	95.33(11)	96.56(8)
NWFE	1NN	74.42(12)	90.03(17)	90.91(12)	96.37(7)
	SVM	77.18(9)	93.42(9)	94.90(11)	96.39(8)
	RF	78.74(8)	92.57(19)	95.53(11)	96.00(6)
SDA	1NN	70.76(8)	91.74(12)	91.98(10)	95.45(13)
	SVM	70.96(10)	91.57(10)	91.87(19)	96.89(9)
	RF	71.34(9)	91.38(8)	92.57(13)	95.43(12)
SELF	1NN	69.56(16)	90.50(12)	90.13(11)	95.89(9)
	SVM	82.74(18)	92.88(11)	94.12(14)	96.45(12)
	RF	80.56(12)	92.89(13)	95.86(12)	96.59(10)
SLPPCE	1NN	74.36(10)	92.87(18)	94.41(9)	96.40(9)
	SVM	72.91(10)	92.58(14)	93.22(13)	96.23(9)
	RF	78.53(11)	92.26(15)	95.60(13)	96.22(11)
SELD	1NN	81.95(11)	93.43(18)	95.14(12)	97.25(8)
	SVM	82.03(11)	93.71(13)	94.97(15)	97.20(8)
	RF	82.26(9)	93.80(12)	95.60(11)	96.92(9)
ISELD	1NN	82.47(9)	94.05(12)	94.99(14)	97.16(11)
	SVM	79.40(10)	94.22(15)	95.75(12)	97.20(8)
	RF	79.75(9)	94.09(10)	95.66(12)	96.86(7)
SEGL	1NN	**83.57(10)**	94.69(13)	96.44(15)	97.18(8)
	SVM	82.26(8)	**95.37(12)**	96.02(12)	**97.64(7)**
	RF	83.51(9)	93.65(12)	**97.13(13)**	96.95(7)



(a) University of Pavia



(b) KSC

Figure 3.4: Averaged OA (%) with the number of extracted features increasing for different semi-supervised feature extraction method with SVM classifier. 40 labelled training samples are chosen randomly from each class.

comparing the unsupervised or supervised feature extraction methods. The classification accuracy is higher when the number of labelled training samples increases. Especially for the proposed SELG method, on the University of Pavia data set, when the labelled training size is small (20 labelled training samples per class), the best average OA is 79.40%, and if we choose 80 labelled training samples from each class, the best average OA reaches to 83.57%, which has more than 4% improvements.

2. The semi-supervised feature extraction methods SELD [Liao 13], ISELD [Luo 15] and the proposed method SEGL, which divide the samples into two sets (labelled and unlabelled) first, infer class discrimination from labelled samples and keep local neighborhood information from unlabelled samples, perform better than other semi-supervised methods (SELF [Sugiyama 10], SLPPCE [Zhang 10]). This suggests that dividing the samples into two group (labelled group and unlabelled group) first, and then achieving different goals on different group (labelled samples used for inferring class discriminant and unlabeled samples used for preserving local manifold structure) is an effective way in semi-supervised learning.

3. By connecting unlabelled and labelled samples in the semi-supervised graph and employing weighted edges between samples, SEGL outperforms other semi-supervised methods. Compared with semi-supervised graph learning method SLPPCE, our proposed SEGL method is 10% better on the University of Pavia data set, and more than 3% on the KSC data set.

4. The SVM classifier is more efficient (with higher classification results) on the data sets KSC and Pavia Centre, however 1NN classifier obtains the highest average accuracy in most cases for the other two data sets. When the features were extracted by the unsupervised methods PCA and LPP, or by the supervised methods NWFE, the RF and SVM classifiers perform much better than the 1NN classifier. On the other hand, the SVM classifier needs more time for classification than RF and 1NN classifier from the experiments.

5. The proposed SEGL method contains the best average OA on the four data sets among all results of all experiments. In Table 3.5, the best average OA results in the University of Pavia and KSC data sets are 79.40% (SEGL with 1NN classifier) and 92.25% (SEGL with SVM classifier), respectively. This is at least 1% better than others. In Table 3.6, only the results of proposed methods SEGL with 1NN classifier exceed 81% in the University of Pavia. For the other three data sets, SEGL still produces the best results: 93.19%, 95.47% and 97.29% respectively. In Table 3.7, SEGL with SVM classifier has best results in KSC and Pavia Center data sets, 95.37% and 97.64% respectively. SEGL with RF classifier gets the

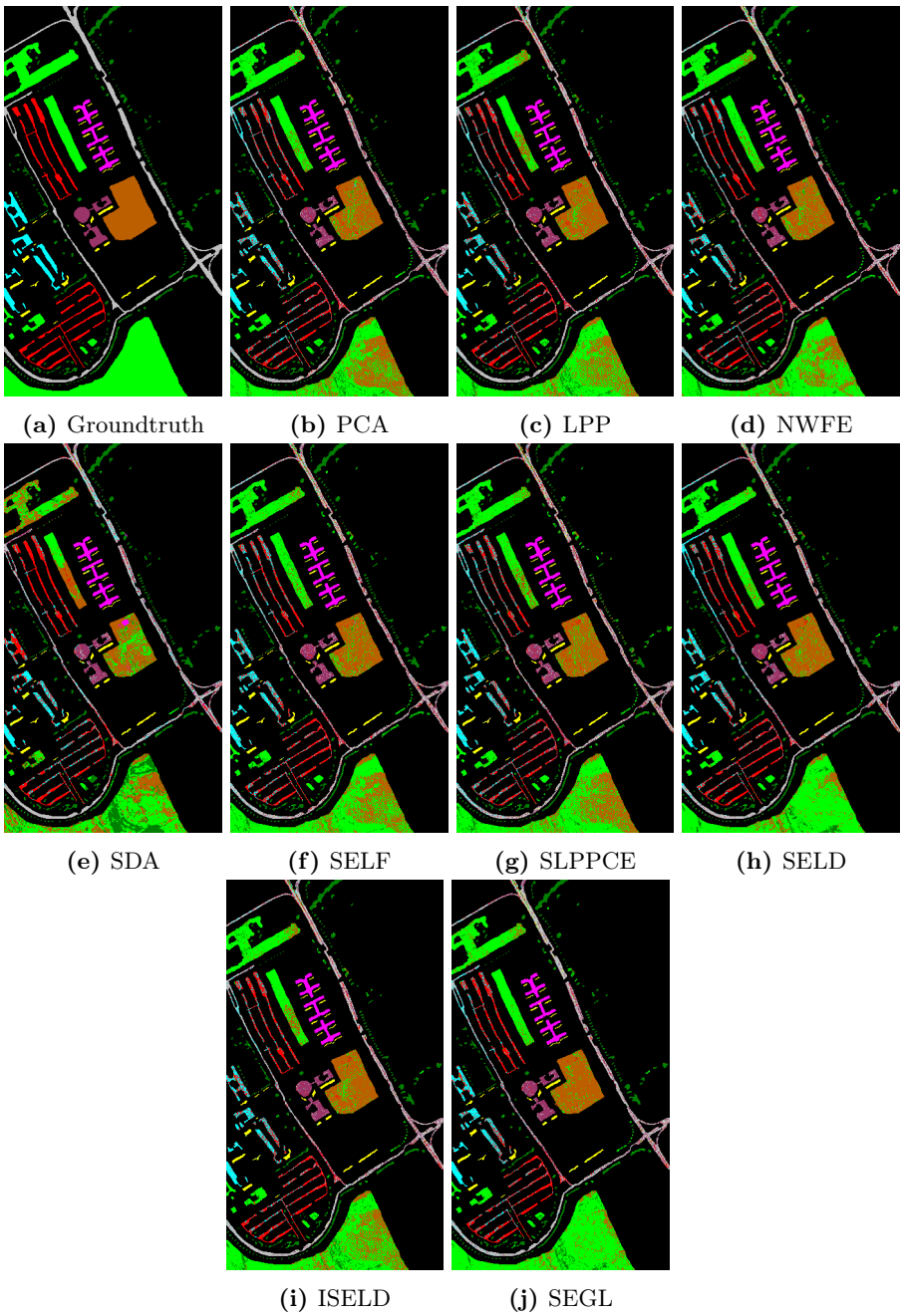
highest OA 97.13% in Botswana data set, and SEGL with 1NN classifier gets the highest OA 83.57% in University of Pavia data set.

Figure 3.4 shows the average OA of several semi-supervised learning methods in function of the number of extracted features. It can be seen that the proposed SEGL with SVM classifier has better performance than other methods on the University of Pavia and KSC data sets. With increasing number of extracted features, the OA of SEGL first improves an then remains constant or slightly decreases. The optimal number of features extracted by our proposed method is 8 for University of Pavia and 12 for KSC with SVM classifier for University of Pavia and KSC, respectively. However, automatic selection of the optimal number of features is still very challenging for most methods [Fauvel 08]. The optimal value depends on the distribution of the data sets, the training samples and the classifiers, see Table 3.5, 3.6, 3.7. Many approaches select the optimal number of features according to the cumulative variance [Fauvel 08]. However, these approaches do not always work well, as discussed in [Fauvel 08] [Liao 16].

In order to compare the classification results visually, we randomly select 40 labelled training samples per class from University of Pavia and KSC data sets. For the SVM classifier, the best classification maps of each methods are shown in Figure 3.5 and Figure 3.6 respectively. It can be seen that the classification maps of proposed SEGL looks smooth on the University of Pavia data set, and this is specially clear for the class "*Meadows*" and "*Soil*". In the classification maps of KSC, the proposed method SEGL also yields good classification result, and outperforms other feature extraction methods in the "*Water*" region near to the coastline, also in the "*Salt marsh*" parts located in the center of "*Water*" region.

3.3.2.4 Results on Different Number of Unlabelled Training Samples

This experiment investigates the influence of the unlabelled sample size on the classification performances. The choice of the number of unlabelled samples is also a very important step in the semi-supervised methods. A large number of unlabelled training samples increases computational complexity, while a small number of unlabelled samples is not sufficient to present the local geometrical structure or distributions of data sets. We choose 20 labelled training samples from each class to compose the labelled subset $X_{labelled}$, the number of unlabelled subset $X_{unlabelled}$ was evaluated from 500 to 5000 with a step 500. Figure 3.7 shows the average classification accuracies (OA%) tends with SVM classifier when increasing the number of unlabelled training samples for various feature extraction methods. As can be seen, the classification accuracy first improves with the number of training samples and then remains constant as more and more unlabelled samples are used. The average OA of the proposed SEGL method improves about 2% when the number of unlabelled training samples is increased from 500 to 5000 on these two data sets. As on the KSC data set, the

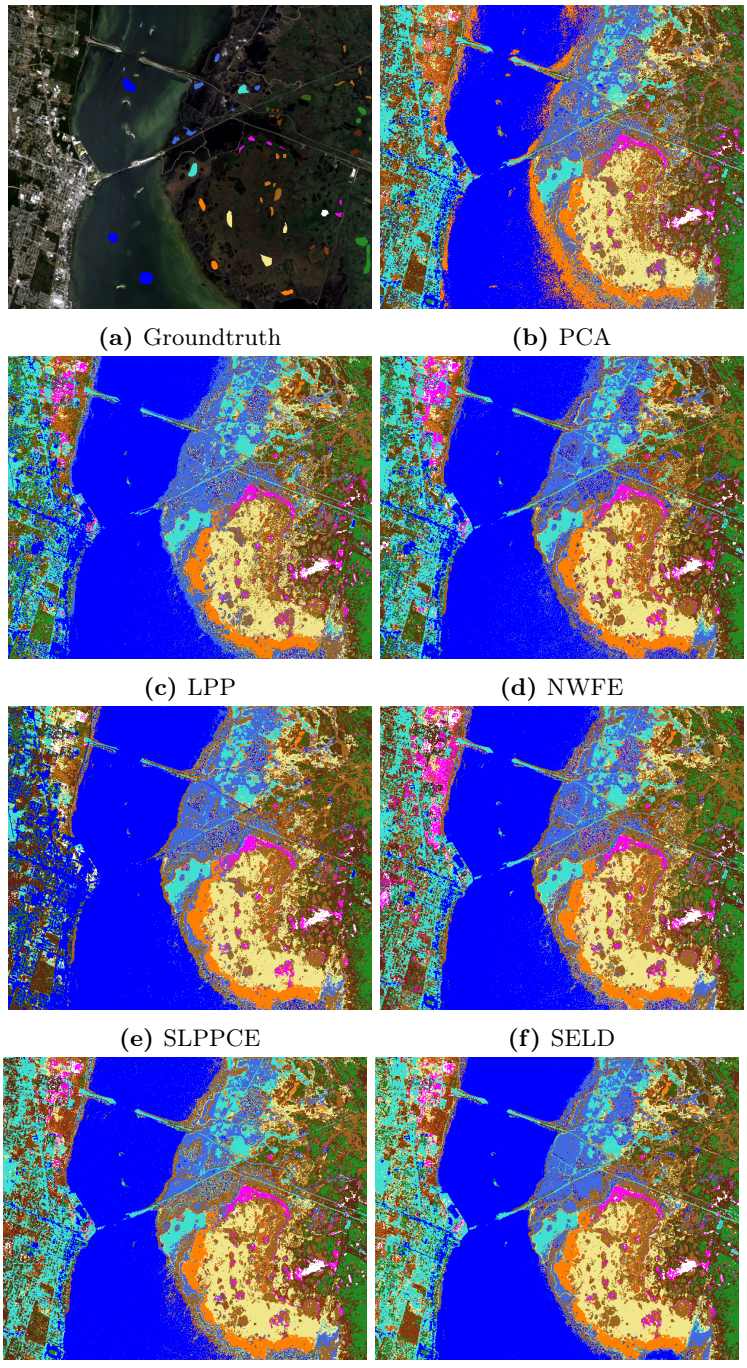


(a) Groundtruth **(b)** PCA **(c)** LPP **(d)** NWFE

(e) SDA **(f)** SELF **(g)** SLPPCE **(h)** SELD

(i) ISELD **(j)** SEGL

Figure 3.5: Classification maps of the different methods with SVM classifier for University of Pavia. 40 labelled samples per class were randomly selected from the training set.



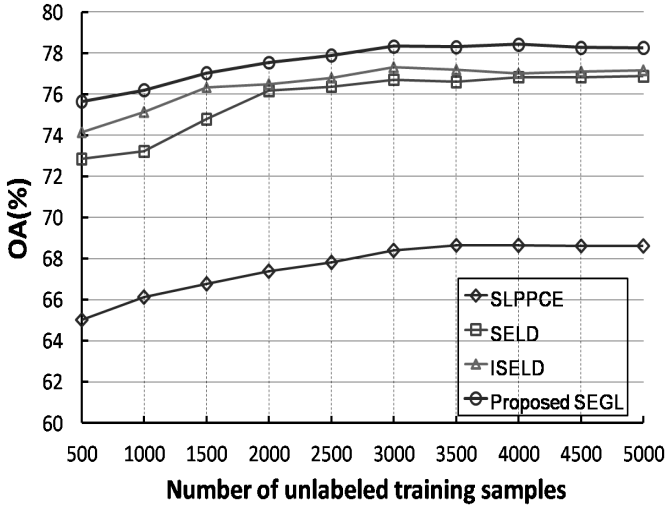
(a) Groundtruth **(b)** PCA

(c) LPP **(d)** NWFE

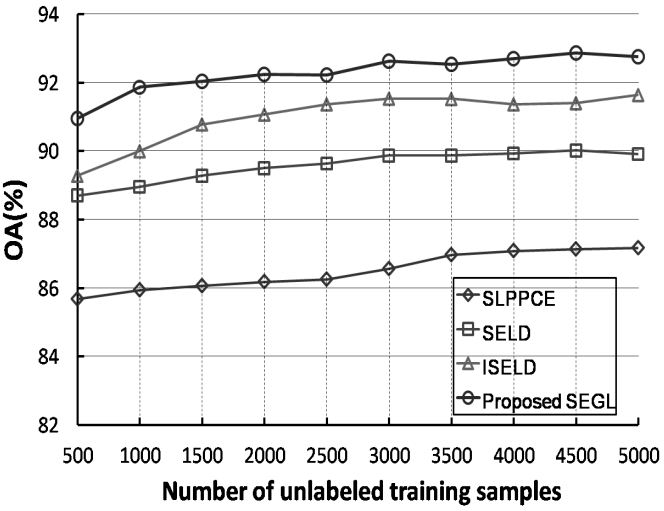
(e) SLPPCE **(f)** SELD

(g) ISELD **(h)** Proposed SEGL

Figure 3.6: Classification maps of the different methods with SVM classifier for KSC. 40 labelled samples per class were randomly selected from the training set.



(a) University of Pavia



(b) KSC

Figure 3.7: Average classification accuracies (OA%) with an SVM classifier for various semi-supervised feature extraction techniques, with various number of unlabelled training samples.

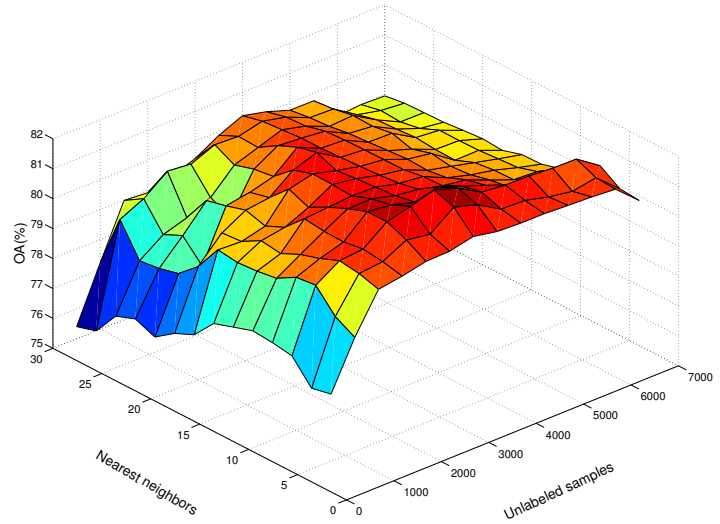
average OA of ISELD method reaches t 89% when only 500 unlabelled training samples were chosen, while the OA reaches 92.86% when using 4500 unlabelled samples.

3.3.2.5 Results on Different Number of Nearest Neighbors

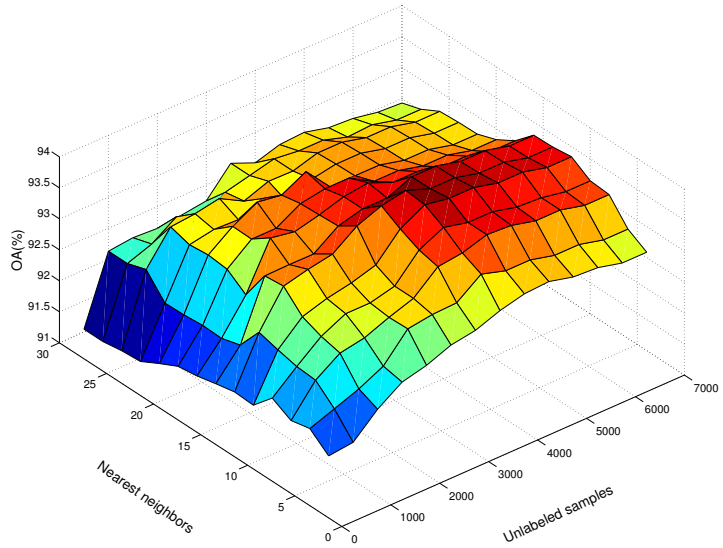
The number of nearest neighbors (k) is an important parameter in our proposed semi-supervised graph. On the one hand, when k is too small, the local manifold structure information may not be properly modelled. On the other hand, a too large k (keep unlabelled samples constant) leads to misclassification. To investigate the effect of the number of nearest neighbors and unlabelled samples on the classification accuracy, we take University of Pavia and KSC data sets as examples in our experiments. 40 labelled training samples were selected from each class with five-fold cross validation to compose the labelled subset $\mathbf{X}_{labelled}$, the number of unlabelled subset $\mathbf{X}_{unlabelled}$ was evaluated from 500 to 7000 with a step 500, the number of nearest neighbors was changed from 4 to 30 with a step 2.

Figure 3.8 shows the correlations between classification results and two parameters: number of nearest neighbors (k) and number of unlabelled samples (m), with SVM classifier. As can be seen, when m is set to 500, the average OA increases at first and then decreases when the k is changed from 4 to 30. This indicates that the increase of k, with fixed number of m, will misclassify many unlabelled samples, leading to poor classification performance. When we keep k constant, the OA will first increase then fall down as the number of unlabelled samples increases. This means if k (or m) is set to a larger value, the possibilities of wrong linked would be increased, i.e. some unlabelled samples which belong to different classes in reality would be linked, as a result the performances of the proposed method would be degraded. We can also see that when the number of unlabelled samples is less than 2000, the classification results changes a lot with different number of nearest neighbors (k). This is because the distribution of nearest neighborhood unlabelled samples is sparse (less density), the change of k has a big effect on the average distance between a sample and its kth nearest neighbors. Consequently, if k is fixed, the effect of k on classification decreases as the number of unlabelled data increases. Therefore in our proposed method, the number of nearest neighbors (k) should be changed in accordance with the number of unlabelled samples (m). Furthermore, the results show that selecting training samples with cross validation can improve the classification performances of our proposed method.

Figure 3.9 shows the correlations between classification results and two parameters: number of nearest neighbors (k) and number of features, with SVM classifier. The number of nearest neighbors was changed from 4 to 30 with a step 2, the number of extracted feature was evaluated from 2 to 20 with a step 1. It can be seen that a larger k (with fixed number of the extracted features) increases the possibility of misclassification, leading to poor classification performances. What's more, the optimal numbers of features will increase as the raise of nearest neighbors.

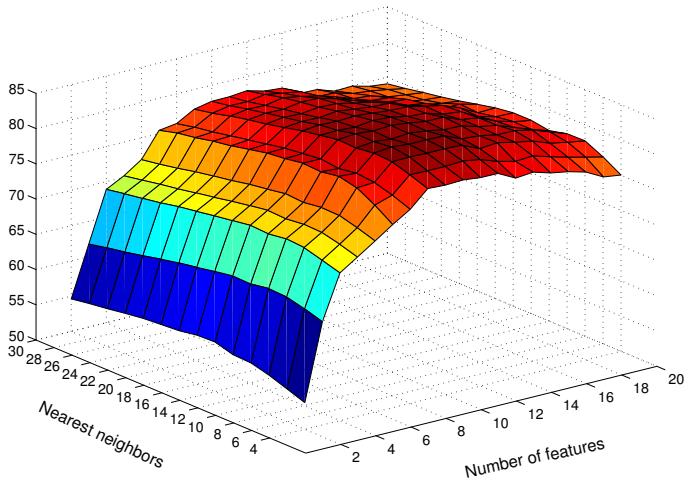


(a) University of Pavia

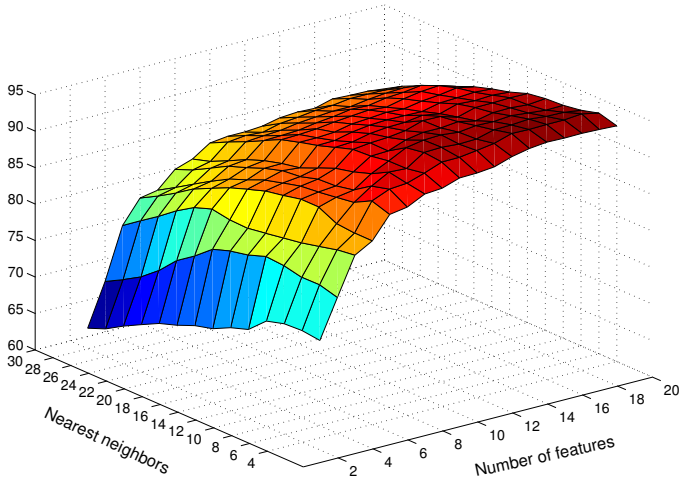


(b) KSC

Figure 3.8: The OA (%) as a function of the number of nearest neighbors and unlabelled samples, with 40 labelled samples per class and 8 extracted features.



(a) University of Pavia



(b) KSC

Figure 3.9: The OA (%) as a function of the number of nearest neighbors and number of features, with 40 labelled samples per class and 2000 unlabelled samples.

3.4 Improved Semi-supervised Graph Learning (ISEGL)

In our previous work [Luo 16c], we proposed a semi-supervised graph learning method for feature extraction, but it only uses spectral and label information,

without considering spatial or neighbor information. In this section, we present a novel semi-supervised graph leaning method which takes into account spectral, spatial and label information for classification of hyperspectral imagery. In our semi-supervised fusion graph, samples are connected according to either label information (labelled samples) or their k-nearest neighbors in both spectral and spatial nearest neighbors (unlabelled samples). Furthermore, we link a unlabelled sample with all labelled samples in a class which is closest to this unlabelled sample in both spectral and spatial feature space. Thus, our proposed method better models the similarities between samples and preserves local manifold structure both in spectral and spatial feature space. What's more, for one sample, if its spectral features (the spectrum of hyperspectral image) and spatial features (morphological attribute profiles with partial reconstruction (APPR)) are stacked into one vector and transformed to a low-dimensional feature space, some important features or information would be lost or mixed, as different types of features have different distributions (feature spaces) and meanings, the joint feature space is highly non-linear, and cannot be modelled well as a single linear subspace. Therefore we extract the low dimensional spectral features from hyperspectral images and spatial features from morphological attribute profiles with partial reconstruction (APPR) separately based on the proposed semi-supervised fusion graph, and fuse the extracted spectral and spatial features for classification only afterwards. In the following section, we will briefly introduce the morphological attribute profiles with partial reconstruction (APPR) first, and then detail our proposed method.

3.4.1 Morphological Attribute Profiles With Partial Reconstruction

The attribute profiles (APs) are obtained by applying a sequence of attribute filters (AFs) to a gray-level image [Mura 10a]. AFs are operators defined in the mathematical morphology framework which operate by merging connected components at different levels in the image according to some measure computed on the components (i.e., attributes). However, being connected filters, AFs [Ouzounis 07, Salembier 09] together with operators based on geodesic reconstruction [Soille 03], suffer from the problem that regions of different objects (e.g., buildings and roads and roads and parking lots are connected) are sometimes connected by spurious links and will then be considered a single object. This is called "over-reconstruction" in [Bellens 08]. This phenomenon might lead to some unexpected results for remote sensed images. To overcome the limitation of over-reconstruction in geodesic reconstruction [Soille 03], the approach in [Bellens 08] proposed a partial reconstruction for morphological opening and closing and better modeled the shape and size of objects in an image.

In 2016, Liao *el at.* [Liao 16] proposed a novel framework for morphological APs with partial reconstruction (APPR) and extended it to the classification of high-resolution hyperspectral images. The approach first applies morphologi-

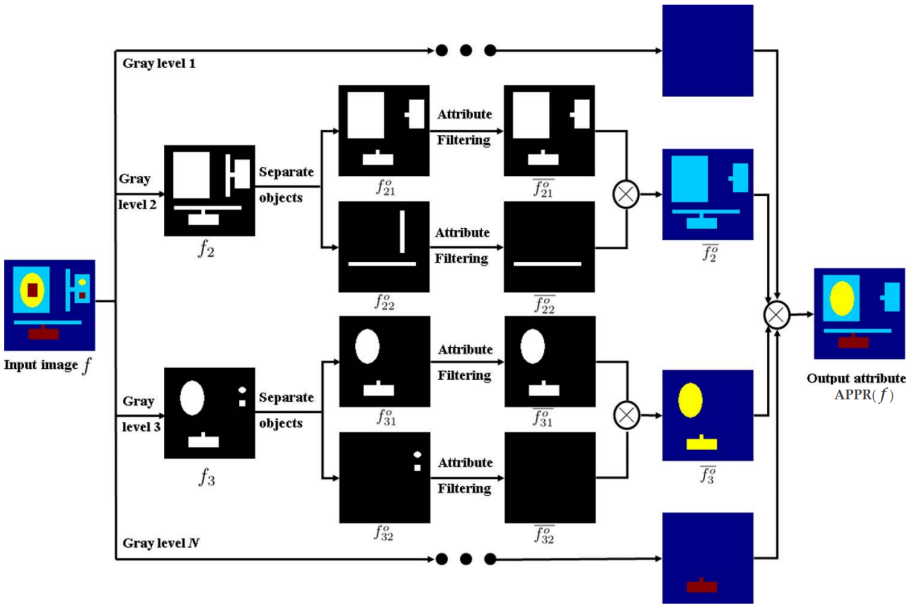


Figure 3.10: Framework for morphological APPR

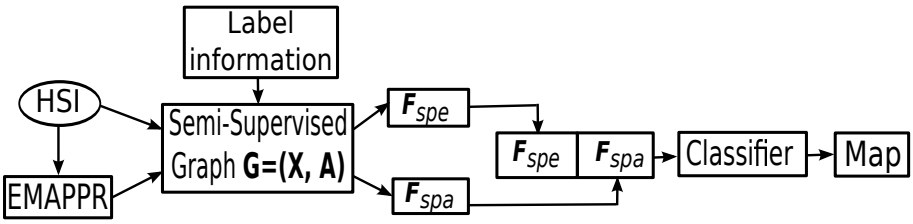


Figure 3.11: Proposed framework

cal filters with partial reconstruction [Bellens 08] to separate connected objects (e.g., roads and parking lots) of a binary image (i.e., at one gray level) into two disjoint parts, and each part of the separated objects are included in the two resulting binary images. Then, they apply AFs to these two binary images. Finally, they integrate all the residuals of the filtered images and get the final output image by repeating this for all gray levels. Figure 3.10 shows their proposed framework for morphological APs with partial reconstruction (APPR).

Table 3.8: Additional notations used in this section.

Notations	Description
APPR	attribute profiles (B profiles) generated from hyperspectral image: $M \times N \times B$
$\mathbf{x}_i^L = \{\mathbf{x}_i^{L(spe)}; \mathbf{x}_i^{L(spa)}\}$	ith labelled training sample, $\mathbf{x}_i^{L(spe)} \in \mathbb{R}^D$ is from hyperspectral image, $\mathbf{x}_i^{L(spa)} \in \mathbb{R}^B$ is from APPR, $\mathbf{x}_i^L \in \mathbb{R}^{D+B}$
$\mathbf{x}_j^U = \{\mathbf{x}_j^{U(spe)}; \mathbf{x}_j^{U(spa)}\}$	jth unlabelled training sample, $\mathbf{x}_j^{U(spe)} \in \mathbb{R}^{(D)}$ is from hyperspectral image, $\mathbf{x}_j^{U(spa)} \in \mathbb{R}^B$ is from APPR, $\mathbf{x}_i^U \in \mathbb{R}^{D+B}$
c_j^{spe}	the class nearest to sample $\mathbf{x}_j^U$ in spectral feature space
c_j^{spa}	the class nearest to sample $\mathbf{x}_j^U$ in spatial feature space
d	the number of extracted features
$Fspe$	spectral features extracted from hyperspectral image: $M \times N \times d$
$Fspa$	spatial features extracted from APPR: $M \times N \times d$

3.4.2 Proposed ISEGL

In this section, we detail our proposed method, Figure 3.11 shows the proposed framework, the hyperspectral images are transformed by principal component analysis (PCA), and the first few important principal components (PCs) are used as base images to calculated the morphological attribute profiles with partial reconstruction (APPR) [Liao 16]. Then semi-supervised fusion graph is obtained based on spectral (as spectrum of hyperspectral image), spatial (as APPR) and label information. After that, we extract low dimensional spectral features ($Fspe$) from the hyperspectral image with the proposed semi-supervised graph, and extract low dimensional spatial features ($Fspa$) from APPR based on semi-supervised fusion graph. Finally the efficient $Fspe$ and $Fspa$ are fused for classification. In the following, the semi-supervised construction of the proposed spectral-spatial graph will be discussed in details.

Let us define $\mathbf{x}_i = \{\mathbf{x}_i^{spe}; \mathbf{x}_i^{spa}\}$, $\mathbf{x}_i^{spe}$ and $\mathbf{x}_i^{spa}$ denote the spectral (spectrum of hyperspectral image) and spatial (APPR) information of ith samples, Furthermore, let $\mathbf{X}_{labelled} = \{(\mathbf{x}_i^L, y_i)\}_{i=1}^n$, $\mathbf{y} = \{y_i\}_{i=1}^n$, $y_i \in \{1, 2, \cdots, C\}$, $\mathbf{x}_i^L$ means the ith sample in the labelled samples set, y_i is the label of $\mathbf{x}_i^L$, C is the number of classes. Finally let $\mathbf{X}_{unlabelled} = \{\mathbf{x}_{n+1}^U, \mathbf{x}_{n+2}^U, \cdots, \mathbf{x}_{n+m}^U\}$, $\mathbf{x}_j^U$ denotes jth sample in the unlabelled samples set, n and m denote the number

of labelled and unlabelled training samples.

We exploit the label information and spectral-spatial local neighborhood information through our proposed semi-supervised graph, which is defined as $\mathbf{G} = (\mathbf{X}, \mathbf{A})$, $\mathbf{X} = \{\mathbf{X}_{labelled}, \mathbf{X}_{unlabelled}\} = \{\mathbf{x}_1^L, \mathbf{x}_2^L, \cdots, \mathbf{x}_n^L, \mathbf{x}_{n+1}^U, \mathbf{x}_{n+2}^U, \cdots, \mathbf{x}_{n+m}^U\}$ is a set of nodes which connected by a set of edges $A_{i,j}$, $A_{i,j}$ is the edge (with a weight between 0 and 1) between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. The basic goal of our proposed method is to find two transformation matrices $\mathbf{W}_{spe}$ and $\mathbf{W}_{spa}$, which can transform the data $\mathbf{x}_i = \{\mathbf{x}_i^{spe}; \mathbf{x}_i^{spa}\}$ in high-dimensional feature space into low-dimensional sample $\mathbf{z}_i = \{\mathbf{z}_i^{spe}; \mathbf{z}_i^{spa}\}$, $\mathbf{z}_i^{spe}$ and $\mathbf{z}_i^{spa}$ are obtained by $\mathbf{z}_i^{spe} = \mathbf{W}_{spe}^T \mathbf{x}_i^{spe}$ and $\mathbf{z}_i^{spa} = \mathbf{W}_{spa}^T \mathbf{x}_i^{spa}$. The transformation matrices $\mathbf{W}_{spe}$ and $\mathbf{W}_{spa}$ can be optimized as follows:

$$\mathbf{W}_{spe} = \arg\min_{\mathbf{W}} \left(\sum_{i,j=1}^{n+m} ||\mathbf{W}^T \mathbf{x}_i^{spe} - \mathbf{W}^T \mathbf{x}_j^{spe}||^2 A_{ij} \right) \tag{3.23}$$

$$\mathbf{W}_{spa} = \arg\min_{\mathbf{W}} \left(\sum_{i,j=1}^{n+m} ||\mathbf{W}^T \mathbf{x}_i^{spa} - \mathbf{W}^T \mathbf{x}_j^{spa}||^2 A_{ij} \right). \tag{3.24}$$

In many applications, labelled samples are used to enhance class discrimination, but the number of them is always very limited (labelling samples is time consuming and expensive). Unlabelled samples, on the other hand, are much easier accessible. The idea behind semi-supervised feature extraction methods [Liao 13] is to infer class discrimination from labelled samples (labels samples help to provide semantic meaning to clusters and the discriminant of different classes), and to preserve local manifold structure from unlabelled samples. As in [Luo 15], we define our proposed semi-supervised graph to model different correlations between samples as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{LL} & \mathbf{A}^{LU} \\ \mathbf{A}^{UL} & \mathbf{A}^{UU} \end{bmatrix}, \tag{3.25}$$

where $\mathbf{A}$ is the adjacency matrix for all training samples; two nodes are adjacent if they are connected, i.e. if $A_{ij} \neq 0$. The adjacency matrix $\mathbf{A}^{LL}$ is an $n \times n$ matrix that models the correlations between labelled samples. If two samples belong to the same class, we create an edge between them. The adjacency matrix $\mathbf{A}^{UU}$, which is a $m \times m$ matrix, models the correlations between unlabelled samples. We connect two unlabelled samples if they are within the k-nearest neighbors of each other both in spectral and spatial feature space. $\mathbf{A}^{LL}$ and $\mathbf{A}^{UU}$ are defined respectively as:

$$A_{i,j}^{LL} = \begin{cases} 1 & , y_i = y_j \\ 0 & , y_i \neq y_j \end{cases} \tag{3.26}$$

$$A_{i,j}^{UU} = \begin{cases} 1 & , \mathbf{x}_j^{U(spe)} \in knn(\mathbf{x}_i^{U(spe)}) \text{ and } \mathbf{x}_j^{U(spa)} \in knn(\mathbf{x}_i^{U(spa)}) \\ 0 & , \mathbf{x}_j^{U(spe)} \notin knn(\mathbf{x}_i^{U(spe)}) \text{ or } \mathbf{x}_j^{U(spa)} \notin knn(\mathbf{x}_i^{U(spa)}) \end{cases} \tag{3.27}$$

where $knn(\mathbf{x}_i^{U(spe)})$ denotes the set of samples that are within the k nearest neighbors of $\mathbf{x}_i^{U(spe)}$. The adjacency matrices $\mathbf{A}^{LU}$ and $\mathbf{A}^{UL}$ contain the connection between labelled and unlabelled samples, $\mathbf{A}^{UL} = (\mathbf{A}^{LU})^T$, as $\mathbf{A}$ is a symmetric matrix. Suppose the labelled sample $\mathbf{x}_i^L$ belong to class c_j, the $n \times m$ adjacency matrix $\mathbf{A}^{LU}$ is defined as:

$$A_{i,j}^{LU} = \begin{cases} 1 & , \mathbf{x}_i^L \in \mathbf{X}^{(c_j)}, c_j = c_j^{spe} \text{ and } \mathrm{c_j} = \mathrm{c_j^{spa}} \\ 0 & , otherwise \end{cases} \tag{3.28}$$

where $\mathbf{X}^{(c_j)}$ is a set including all labelled samples in class c_j, c_j represents the class closest to $\mathbf{x}_j^U$ in both spectral and spatial feature space, c_j^{spe} and c_j^{spa} denote the class closest to $\mathbf{x}_j^U$ in spectral and spatial space, respectively. The details to find the closest class are explained in [Luo 15]. With this approach, the connected labelled and unlabelled samples have similar spectral and spatial features, and belong to the same class with high probability.

Table 3.9: Overall classification accuracies (OA)(%) of different schemes with different training size n_c/per class.

Methods	$n_k = 10$	$n_k = 20$	$n_k = 40$	$n_k = 80$
Raw	69.6	73.1	76.3	79.6
APPR	81.5	84.9	88.9	91.0
PCA	75.0	85.6	91.9	93.1
NWFE	87.4	91.2	93.7	95.8
Proposed	**88.1**	**93.8**	**96.6**	**97.6**

3.4.3 Experiments and Results

The hyperspectral image dataset we used in the experiments was the University of Pavia; see section 3.2.3 and Table 3.4. The SVM classifier with radial basis function (RBF) kernels is utilized in our experiments. The parameters of SVM classifier are set the same as in our previous work [Luo 15]. In order to investigate the impact of the labelled samples on the classification accuracy, we randomly select the labelled samples from the training set with the sample size corresponding to different cases: 10, 20, 40, 80 per class. 1500 samples were randomly selected from the original hyperspectral image, similarly as [Liao 13].

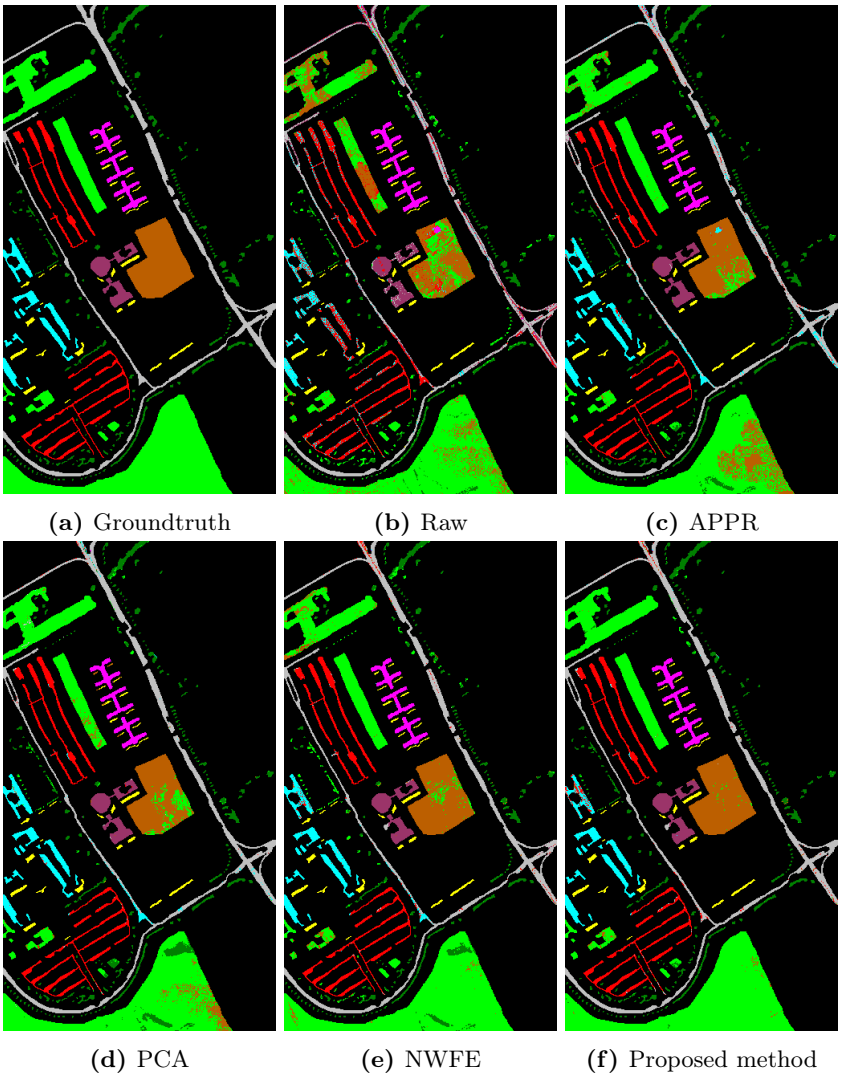
Table 3.10: Overall Classification accuracies(OA)(%) for different classes with 40 labelled training samples per class.

	Train/Test	Raw	APPR	PCA	NWFE	Proposed
No. of features	-	103	180	15+15	15+15	15+15
Asphalt	40/6631	69.6	88.3	**94.9**	94.4	94.7
Meadows	40/18649	78.5	83.8	89.1	94.2	**99.3**
Gravel	40/2099	60.5	96.2	**98.0**	95.4	95.5
Trees	40/3064	75.6	96.4	93.6	79.6	**97.3**
Metal sheets	40/1345	98.4	99.6	**99.7**	98.8	99.4
Soil	40/5029	65.0	86.8	85.8	**93.0**	85.8
Bitumen	40/1330	85.5	98.8	98.2	96.2	**99.0**
Bricks	40/3682	84.5	98.7	98.4	98.1	**99.2**
Shadow	40/947	**100**	**100**	94.4	**100**	**100**
OA(%)	-	76.3	88.9	91.9	93.7	**96.6**
AA(%)	-	79.7	94.3	94.7	94.4	**96.7**

Each experiment was repeated 5 times. In our experiments, the proposed method has been compared with PCA and NWFE [Kuo 04], and 15 spectral features are extracted from hyperspectral image and 15 spatial features are extracted from APPR by each method. **Raw** means the original hyperspectral image, **APPR** is all morphological attribute profiles, **PCA** and **NWFE** use both 15 spectral features and 15 spatial features.

The experimental results for different methods are summarized in Table 3.9-Table 3.10 and Figure 3.12. In the table, the optimal results are highlighted in bold. The results show that using only spectral or only spatial features is not sufficient for a reliable classification: fusion improves the classification performance, with the highest OA 97.6% for the proposed method. By comparing the results reported in Table 3.9, it is easy to infer that the proposed semi-supervised method outperforms unsupervised PCA and supervised NWFE, as it combines the advantages of labelled and unlabelled samples. When the number of labelled training samples is only 20 per class, the classification accuracy still reaches to 93.8%. Table 3.10 shows that the proposed method performs well, especially on classifying Meadows, bitumen and bricks, the classification accuracies are all above 99%. In particular, the improvements of proposed method in OA are 2.9%-20.3% compared to the schemes of others.

In order to compare the classification results visually, we randomly select 40 labelled training samples per class from University of Pavia data set, and classification maps of each schemes are shown in Figure 3.12. It can be seen that



(a) Groundtruth **(b)** Raw **(c)** APPR

(d) PCA **(e)** NWFE **(f)** Proposed method

Figure 3.12: Classification maps of the different schemes for KSC, with 40 labelled training samples per class.

the classification maps of proposed method has higher quality looks smooth, and this is specially clear for class "*Meadows*" and "*Soil*".

3.5 Conclusions

In this chapter, we discussed semi-supervised approaches for feature extraction of hyperspectral remote sensing imagery. First, we proposed an improved lo-

cal discriminant semi-supervised feature extraction (ISELD) for hyperspectral image. The proposed ISELD builds correlation matrices of labelled and unlabelled samples, and offers better class discrimination and better preservation of local neighborhood information. When a small number of labelled samples is available, the performance of our approach is outstanding compared to the other methods.

Secondly, we presented a new feature extraction method with semi-supervised graph learning, and applied it to classification of hyperspectral images. The proposed method connects labelled samples according to their label information, connects unlabelled samples by their k-nearest neighbors information. For connections of labelled and unlabelled samples, we find the nearest class for each unlabelled sample first, then connect the unlabelled sample with labelled samples belonging to its nearest class. Last but not least, the proposed SEGL method set weighted edges to connected samples by utilizing distance information between samples. This way our proposed SEGL method can better models the connections between samples through a general semi-supervised graph than state-of-the-arts. Compared to some related feature extraction methods on four hyperspectral data sets, our proposed SEGL has better performance (with higher classification accuracies).

The last contribution of this chapter is to propose an improved version of SEGL to fuse spectral and spatial information in a semi-supervised way for feature extraction of hyperspectral image. The spatial information (carried in morphological features with partial reconstruction), spectral and label information are first combined to build an optimal semi-supervised graph. By exploiting the fused semi-supervised graph, we then get transformation matrices to project high-dimensional hyperspectral image and morphological features to their lower dimensional subspaces separately, the final classification map is obtained by concentrating the lower-dimensional spatial and spectral features together as an input of SVM classifier. Experimental results on the classification of the real hyperspectral data show the efficiency of the proposed method.

The research in this chapter lead to one journal publication and three proceedings as follows:

1. **Luo Renbo**, Liao Wenzhi, Huang Xin, Pi Youguo, Philips Wilfried, "Feature Extraction of Hyperspectral Images with Semi-Supervised Graph Learning". IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2016; 9(9): 4389-4399.

2. **Luo Renbo**, Liao Wenzhi, Zhang Hongyan, Pi Youguo, Philips Wilfried, "Spectral-Spatial Classification of Hyperspectral Images with Semi-Supervised Graph Learning". SPIE Remote Sensing. Sep.2016.

3. **Luo Renbo**, Liao Wenzhi, Pi Youguo, Philips Wilfried, "An improved semi-supervised local discriminant analysis for feature extraction of hyperspectal image". Joint Urban Remote Sensing Event, Proceedings (JURSE 2015). Mar. 2015. p. 1-4.

4

Classification based on Joint Cloud-shadow HS and LiDAR Data

In the previous Chapters, we proposed supervised and semi-supervised feature extraction methods for classification, but all these methods only use hyperspectral data. In most piratical applications, hyperspectral data are not enough for pattern recognition and classification. For instance, hyperspectral data cannot distinguish different objects made from similar materials. If hyperspectral data are shadowed by clouds, the objects in shadow region will be difficult to recognize (shadow region becomes dark as its low radiance). Recent advances in Light Detection And Ranging (LiDAR) sensors allow gathering more useful and complementary information for Earth observation. Specifically, LiDAR can provide elevation information, and is not influenced by clouds and shadows (as LiDAR detect objects on Earth surface based on pulsed laser). For an increased classification performance, fusion of hyperspectral and LiDAR data recently attracted interest but the topic is quite challenging. Most existing classification methods which fuse hyperspectral and LiDAR data suffer from a poor performance in cloud-shadow regions because of lack of sufficient training data and inadequate combination of the advantages of different source data.

In this Chapter, we propose a new framework to fuse hyperspectral and LiDAR data for classification of remote sensing scenes mixed with shadows due to partial cloud cover. We process the cloud-shadow and shadow-free regions separately. Our main contribution is the development of a novel method to generate reliable training samples in the cloud-shadow regions. Classification is performed separately in the shadow-free (classifier is trained with the available training samples) and cloud-shadow regions (classifier is trained by our generated training samples). Our method integrates spectral (i.e. original hyperspectral image), spatial (morphological features computed on hyperspectral image) and elevation (morphological features computed on LiDAR) features. The final classification map is obtained by fusing the results of the shadow-free

and cloud-shadow regions. Experimental results on a real hyperspectral and LiDAR dataset demonstrate the effectiveness of the proposed method, as the proposed framework improves the overall classification accuracy with 4% for whole scene and 10% for shadow-free regions over the other methods.

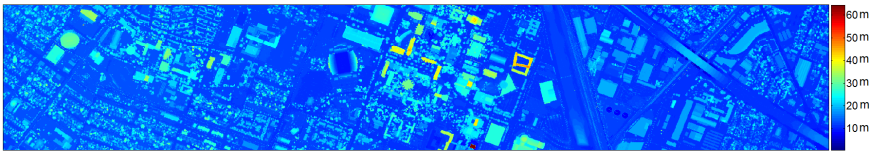
4.1 Introduction

Recent advances in sensor technology allow us to measure different aspects of the objects on the Earth's surface, e.g. the spectral reflectance using hyperspectral images, and height information using Light Detection And Ranging (LiDAR) data [Bruce 13]. Nowadays, hyperspectral images of both high spatial and spectral resolutions are available and can provide valuable spectral information for land use/cover applications [Bioucas-Dias 13]. However, their use is still limited in very complex scenes in which many objects are made of similar materials (e.g. roofs, parking lots and roads). Moreover, optical in nature, hyperspectral sensors suffer from cloudy weather conditions. On the other hand, LiDAR data provides complementary information related to the size, structure and elevation of different objects [Jung 14], but fails to discriminate between different objects that are similar in altitude while quite different in nature (e.g. grass field and swimming pool). Therefore, using a single data source (either hyperspectral or LiDAR data) alone might not be sufficient to obtain reliable classification results.

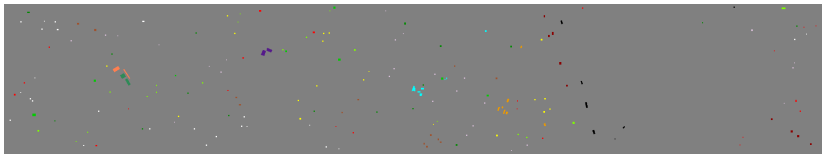
Due to an increased availability of hyperspectral and LiDAR data from overlapping areas, the fusion of hyperspectral and LiDAR data has recently been explored intensively. In [Gu 15], Gu *et al.* proposed a multiple-kernel learning (MKL) model to integrate heterogeneous features from hyperspectral images and LiDAR data for urban area classification. Elakshe *et al.* [Elakshe 08] explored the fusion of hyperspectral and LiDAR data for coastal mapping by using hyperspectral imagery to discriminate between road and water pixels, and LiDAR data to detect and create a vector layer of building polygons. Dalponte *et al.* [Dalponte 08] investigated the joint use of hyperspectral and LiDAR data for the classification of complex forest areas. Yokoya *et al.* [Yokoya 14] fused hyperspectral images and LiDAR data for landscape visual quality assessment and enabled the prediction of landscape quality from any viewpoint using large-scale remote sensing observations. In [Naidooa 12], classification of eight common savanna tree species was performed by fusing hyperspectral and LiDAR data with an automated Random Forest modelling approach. Shimoni *et al.* proposed a score-level fusion approach to detect stationary vehicles under shadows in [Shimoni 11], where detection scores from both hyperspectral and LiDAR data are derived separately and combined with a simple sum rule. From these literatures [Gu 15]- [Liao 14], it can be concluded that the combination of hyperspectral and LiDAR data can contribute to a more comprehensive interpretation of ground objects, as hyperspectral and LiDAR data contain different and complementary information for same objects.

As the footprint of one object often contains more than one pixel and thus

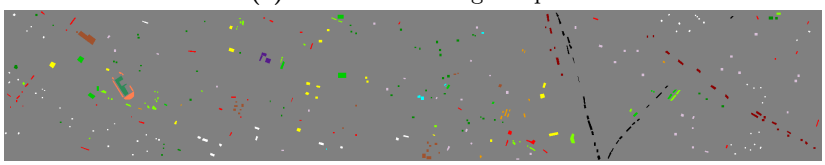
(a) False RGB image of hyperspectral data



(b) LiDAR image



(c) Available training samples



(d) Test samples

Figure 4.1: Experimental data set

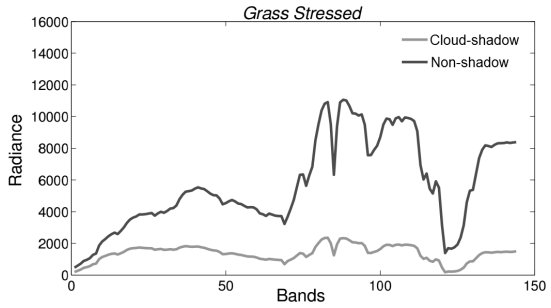
a high spatial correlation is expected between neighboring pixels, and many approaches [Chen 14]- [Liao 12] exploit this correlation to improve the fusion of hyperspectral and LiDAR data and to address the "salt-and-pepper" noise in classification. In [Pedergnana 12], Pedergnana *et al.* applied morphological attribute profiles (MAPs) [Mura 10b] to model hyperspectral and LiDAR data, and fused multiple feature sources in a stacked architecture. Recently, Khodadadzadeh *et al.* [Khodadadzadeh 15] developed a new strategy to fuse hyperspectral and LiDAR data by stacking multiple types of features (spatial and spectral features from hyperspectral, elevation features from LiDAR). The above mentioned methods have demonstrated that combining spectral, spatial and elevation features further boosts the accuracy of land cover classification maps. However, stacking the high dimensional spectral and morphological features directly (storing data in a vector) may lead to the curse of dimensionality problem [Hughes 68] and excessive computation time. What's more, the joint higher dimensional feature space is highly non-linear, the stacked vector may not be easy to process as a single linear subspace.

In 2013, the Data Fusion Technical Committee of the IEEE Geoscience and

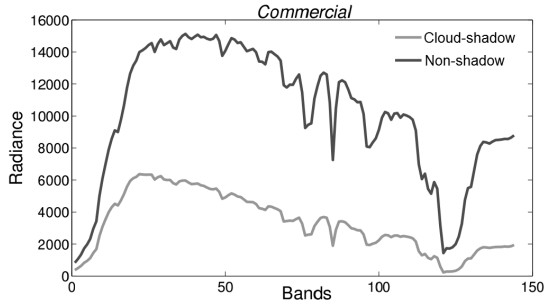
Remote Sensing Society (GRSS) organized a contest involving two types of data sources: a cloud-shadow hyperspectral image and a LiDAR derived digital surface model (DSM) [Hyp 13], see Fig 4.1. The competition was established to stimulate the development of advanced methods to fuse hyperspectral and LiDAR data for classification [Debes 14]. More than 900 researchers from universities, national labs, space agencies and corporations across the globe registered to the contest. The contest data sets contain many regions with shadows due to cloud cover (as hyperspectral image). As cloud shadows weaken most of the spectral reflectance (shadowed region has much lower radiance compared with shadow-free region), most objects in cloud-shadow regions become dark, thus prevent accurate land cover mapping [Zhu 14]. Moreover, as the clouds emerge and move irregularly and unpredictably, it is very difficult to label training samples and acquire remote sensing images at the same time, and to prepare two distinct sets of training samples for shadow-free and cloud-shadow regions in a remote sensing scene. Typically, most of the pixels in the cloud-shadow regions will be misclassified when only using training samples selected from shadow-free regions, as the spectral reflectance information of samples located within and out of the cloud-shadow are totally different, see Figure 4.2.

In order to improve classification performance by fusing hyperspectral image and LiDAR data, a graph-based fusion method [Liao 15] was proposed, in this method, the problem of multi-sensor data fusion is solved by projecting all features (spectral, spatial, and elevation) into a low-dimensional subspace, on which neighborhood relationships among data points (i.e., with similar spectral, spatial, and elevation characteristics) in the original space are maintained. Debes *et al.* [Debes 14] proposed a two-stream classification framework which combined the hyperspectral and LiDAR data by a parallel process that involves both unsupervised and supervised classification. Ghamisi *et al.* [Ghamisi 16] fused hyperspectral and LiDAR data by using extinction profiles and deep convolutional neural networks and achieved improved classification results. However, even though all the available training samples located in shadow-free regions were used to train the classifiers, the classification performances of the cloud-shadow regions were not satisfactory [Khodadadzadeh 15]- [Zhong 16].

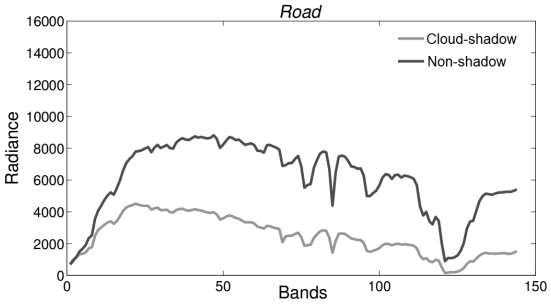
In this Chapter, we propose a novel framework to fuse hyperspectral and LiDAR data for classification of remote sensing scenes mixed with cloud-shadow. The proposed method performs classification separately on the cloud-shadow and shadow-free regions. We solve the problem of missing training samples in the cloud-shadow regions by developing a method to generate reliable training samples. This method is based on the truth that in the cloud-shadow regions different feature sources can be seen as features from different aspects for same objects, and that elevation features from LiDAR data are more reliable than spectral features from hyperspectral image as LiDAR sensor is not influenced by shadows. We then classify the shadow-free (using the available training samples) and cloud-shadow regions (using the generated training samples) separately by integrating spectral, spatial and elevation features, obtained by exploiting attribute profiles [Mura 10a]. The final classification map is produced



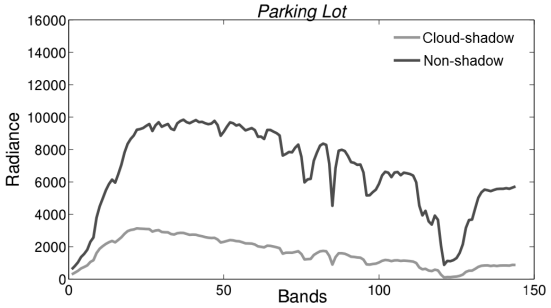
(a)



(b)



(c)



(d)

Figure 4.2: Radiance of different materials.

by decision fusion of the obtained cloud-shadow and shadow-free maps.

We can also interpret our proposed framework from the viewpoint of domain adaptation [Tuia 16]. The shadow-free region can be seen as the source domain, whereas the cloud-shadow region can be seen as a target domain. The labeled training set is only available for the source domain. According to the condition that the source and target domains share the same set of classes and elevation features (LiDAR), we make use of the information from the source domain to generate training samples for the classification of the target domain.

The remainder of this Chapter is organized as follows: section 4.2 describes the proposed framework, with a detailed description of every part of the proposed method. The experimental results on real urban cloud-shadow hyperspectral images and LiDAR data are presented and discussed in Section 4.3. Finally, the conclusions of the Chapter are drawn in Section 4.4.

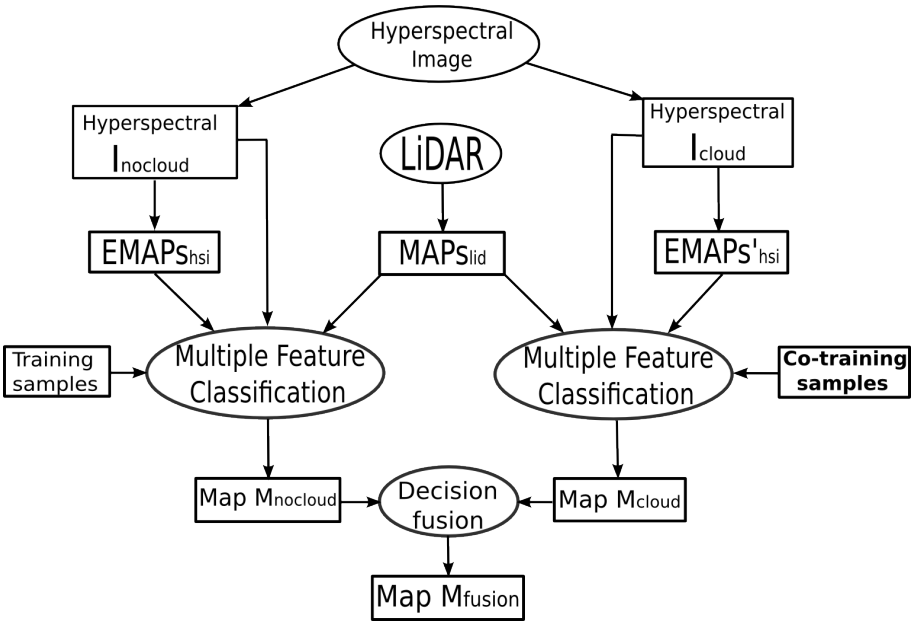


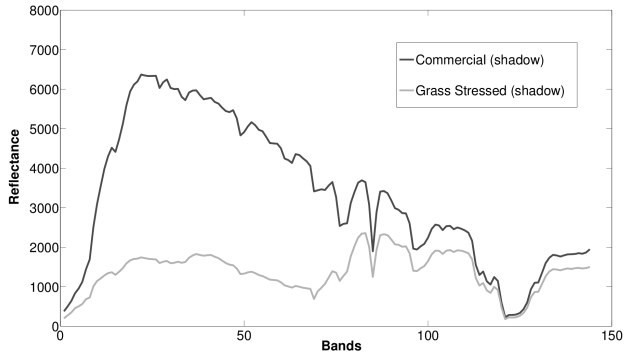
Figure 4.3: Flowchart of the proposed framework, here hyperspectral $I_{nocloud}$ and hyperspectral I_{cloud} denote the shadow-free and cloud-shadow regions of hyperspectral image, $EMAPs_{hsi}$ and $MAPs_{lid}$ mean morphological profiles extracted from hyperspecral and LiDAR data. 'Co-training samples' are generated training samples by our methods for cloud-shadow regions.

4.2 Proposed Framework

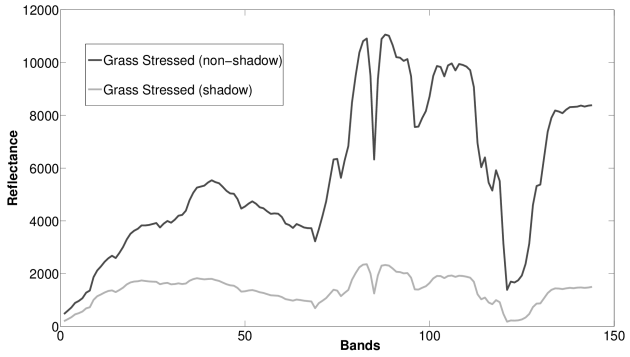
Clouds heavily distort the sun's reflectance, information analysis based on such distorted optical images is not always reliable. Meanwhile, collection of train-

ing data is preferably done on the ground, and since the clouds emerge and move irregularly and unpredictably, it is very difficult to obtain remote sensing images and label training samples from both cloud-shadow and shadow-free regions at the same time. If remote sensing images collection and samples labelling happen not at the same time, we can't sure the labelled samples locate in cloud-shadow or shadow-free region as clouds move unpredictably. Moreover, users prefer to select training samples from shadow-free regions for better visualization and interpretation. For example in Figure 4.1c, all training samples were collected from shadow-free regions. When classifying a remote sensing scene with a classifier, trained on a shadow-free training set only, the results on the cloud-shadow regions will be very poor, the main reason being that objects made of the same material have different spectral reflectance in cloud-shadow and shadow-free regions (Figure 4.4). However, it is important to notice that within the cloud-shadow regions, objects made of different materials have different spectral signatures (Figure 4.4), indicating that these regions still contain sufficient distinctive information. Some notations used throughout this Chapter are summarized in Table 4.1.

Therefore, we propose a novel framework for the classification of remote sensing scenes containing cloud shadows. In the proposed framework, as shown in Figure 4.3, we first divide the hyperspectral image into two different parts: cloud-shadow (hyperspectral I_{cloud}) and shadow-free (hyperspectral $\mathrm{I}_{nocloud}$) regions. $\mathrm{EMAPs}_{\mathrm{hsi}}$ and $\mathrm{MAPs}_{\mathrm{lid}}$ denote the additional spatial and elevation information extracted from hyperspectral and LiDAR data by attribute profiles [Mura 10a], respectively. For the classification of shadow-free regions hyperspectral $\mathrm{I}_{nocloud}$, we fuse multiple features using a similar framework as in [Fauvel 08]. To reduce the redundancy of both the original spectral data and the spatial information (e.g. two continuous bands in hyperspectral image have high correlation and contain redundancy, two morphological profiles with high correlation in $\mathrm{EMAPs}_{\mathrm{hsi}}$ and $\mathrm{MAPs}_{\mathrm{lid}}$ also contain redundancy), we first use feature extraction (FE) techniques to extract relevant information from each single feature source. Then we concatenate all extracted features together, and use these as input for a classifier to obtain the classification map of shadow-free region. In order to generate a classification map of cloud-shadow regions map M_{cloud}, we propose a novel method to generate some samples from cloud-shadow regions as training samples for classification of this region, which will be detailed in the following subsection. Using these generated training samples, the pixels in the cloud-shadow region can be classified by integrating spectral and morphological features computed on the hyperspectral image and elevation or morphological features computed on the LiDAR data, just as in the case of shadow free regions. Last but not least, the final classification map of a cloud mixed remote sensing scene is obtained by fusing map $\mathrm{M}_{nocloud}$ and map M_{cloud}.



(a)



(b)

Figure 4.4: (a) Reflectance of different materials in cloud-shadow region; (b) Reflectance of similar materials in cloud-shadow and shadow-free region.

4.2.1 Morphological Attribute Profiles

For the classification of very high-resolution remote sensing data, spatial information (e.g. the size and shape of objects) has been widely exploited [Pesaresi 01]- [Benediktsson 05]. To model the spatial information from hyperspectral images and LiDAR data, Pesaresi *et al.* [Pesaresi 01] build so-called morphological profiles (MP). As an extension of the concept of MP, attribute profiles (APs) [Mura 10a] provide a multilevel characterization of an image by the sequential application of morphological attribute filters, which model different specifications of the structural information contained in the scene, such as length, area and shape of objects. In [Mura 11], [Ghamisi 14b], extended multi-attribute profiles (EMAPs) were developed to extract abundant spatial information in hyperspectral images. All above literatures prove that attribute profiles (APs) and extended multi-attribute profiles (EMAPs) can model different attributes (as size, shape and standard deviation) of objects flexibility, and these profiles make an obvious contribution to the classification.

Table 4.1: Some notations used in this Chapter.

Notations	Description
HS image	raw data cube: $M \times N \times D$, D is number of bands
$\text{EMAPs}_{\text{hsi}}$	attribute profiles (134 profiles) from HS image: $M \times N \times 134$
MAPs_{lid}	attribute profiles (67 profiles) from LiDAR image: $M \times N \times 67$
n	number of labeled training samples
C	number of classes
y_i	label of ith training sample
$y_i^{'}$	label of ith sample in map M_{lid}
$\mathbf{x}_i^{spe}$	ith sample (column vector) in HS image, $\mathbf{x}_i^{spe} \in \mathbb{R}^D$
$\mathbf{x}_i^{spa}$	ith sample (column vector) in $\text{EMAPs}_{\text{hsi}}$, $\mathbf{x}_i^{spa} \in \mathbb{R}^{134}$
$\mathbf{x}_i^{lid}$	ith sample (column vector) in MAPs_{lid}, $\mathbf{x}_i^{lid} \in \mathbb{R}^{67}$
$\mathbf{x}_i^{Sta}$	$\mathbf{x}_i^{Sta} = \{\mathbf{x}_i^{spe}; \mathbf{x}_i^{spa}\} \in \mathbb{R}^{D+134}$
d	the number of extracted features from each data source
$Fspe$	spectral features extracted from HS image: $M \times N \times d$
$Fspa$	spatial features extracted from $\text{EMAPs}_{\text{hsi}}$: $M \times N \times d$
$Flid$	elevation features extracted from MAPs_{lid}: $M \times N \times d$
$\mathbf{G} = \{g_{ij}\}$	cloud-shadow mask
$\mathbf{X}_{c(k)}^{'}$	co-training samples for class c in kth iteration
$n_{c(k)}$	number of samples in $\mathbf{X}_{c(k)}^{'}$
$\mathbf{m}_{c(k)}^{spe}$	center of $\mathbf{X}_{c(k-1)}^{'}$ in spectral feature space, $\mathbf{m}_{c(k)}^{spe} \in \mathbb{R}^D$
$\mathbf{m}_{c(k)}^{spa}$	center of $\mathbf{X}_{c(k-1)}^{'}$ in spatial feature space, $\mathbf{m}_{c(k)}^{spa} \in \mathbb{R}^{134\times 1}$

Attribute profiles (AP) generate a multi-level decomposition of the input image based on morphological attribute filters, which can properly extract and model the spatial information of the adjacent pixels with progressively higher threshold values. Suppose $\lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ ($\lambda_i < \lambda_j$ with $i < j$) is a sequence of predefined criteria for morphological attribute filters (an attribute profile is an image filtered according to λ_i) used in a gray scale image g, then an AP of g can be defined as:

$$\mathbf{AP}(g) = \{\phi_n(g), \cdots, \phi_1(g), g, \varphi_1(g), \cdots, \varphi_n(g)\}, \tag{4.1}$$

where ϕ_i and φ_i denote the attribute thinning and thickening operations with reference values λ_i.

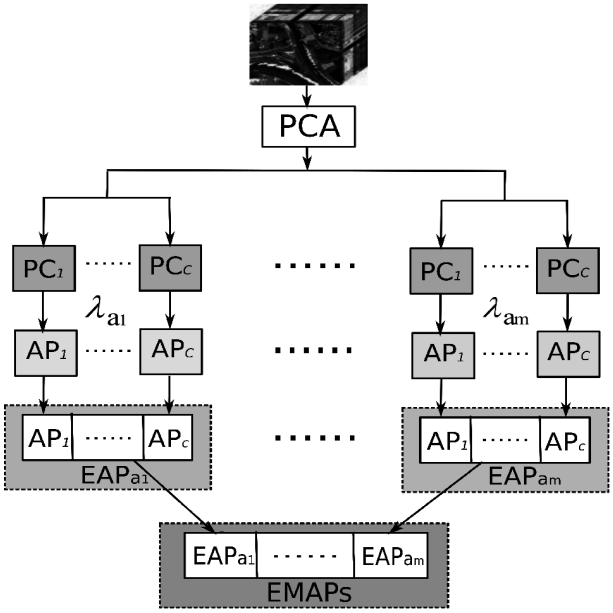


Figure 4.5: General architecture of EMAPs, λ_{a_i} denotes the predefined conditions of attribute a_i.

The above AP only works on a gray scale image. In order to extend the concept of the AP to hyperspectral images, one possible way is to perform a feature reduction approach (such as PCA) on the input data and then apply APs to the first principle components [Ghamisi 14a]. Let PC_i $i = \{1, ..., c\}$ denotes the first principle components of hyperspectral image, then the extended-AP (EAP) is given by:

$$\mathbf{EAP} = \{\mathbf{AP}(PC_1), \mathbf{AP}(PC_2), \cdots, \mathbf{AP}(PC_c)\} \tag{4.2}$$

The presented EAPs model the size and structure of different objects based on one attribute. If more attributes (e.g. area, diagonal of bounding box, length and standard deviation) are considered, extended multi-attribute profiles (EMAPs) can be denoted as:

$$\mathbf{EMAPs}_{hsi} = \{\mathbf{EAP}_{a_1}, \overline{\mathbf{EAP}}_{a_2}, \cdots, \overline{\mathbf{EAP}}_{a_m}\} \tag{4.3}$$

where a_i is a generic attribute (e.g. length, area and shape) and $\overline{\mathbf{EAP}} = \mathbf{EAP} \setminus (PC_1, PC_2, \cdots, PC_c)$. Removing PCs from $\mathrm{EAP}_{a_i}, i > 1$ is necessary for avoiding redundancy since the original components PC_i are present in each EAP. Figure 4.5 shows the general architecture of $\mathrm{EMAPs}_{\mathrm{hsi}}$. $\mathrm{EMAPs}_{\mathrm{hsi}}$ can be seen as a data cube by stacking attribute profiles (grey images) obtained above, and every pixel in $\mathrm{EMAPs}_{\mathrm{hsi}}$ is a vector.

Attribute filters can also be applied in LiDAR image to produce attribute profiles and model elevation features. An attribute thinning acts on bright

objects (for LiDAR image, the bright regions are actually areas with high elevation, such as the top of a roof), while thickening acts on dark (low height) objects. For example, an attribute thinning deletes bright objects that are smaller than the threshold λ_i. By computing a series of attributes, a complete attribute profile is built, carrying information about the elevation information of objects in the image. Let L denotes the LiDAR image, which is treated as a gray scale image where the value of a pixel denotes the altitude at that point. Then the AP of L can be defined as:

$$\mathbf{AP}(L) = \{\phi_n(L), \cdots, \phi_1(L), L, \varphi_1(L), \cdots, \varphi_n(L)\} \tag{4.4}$$

As the LiDAR image has only one single band (elevation of objects), we use the term multi-APs (MAPs) instead of extended multi-attribute profiles (EMAPs) to model the spatial information in LiDAR image, with exploiting different attribute filters. Then the MAPs generated from LiDAR image ($\mathbf{MAPs}_{lid}$) can be expressed as:

$$\mathbf{MAPs}_{lid} = \{\mathbf{AP}_{a_1}(L), \mathbf{AP}_{a_2}(L), \cdots, \mathbf{AP}_{a_m}(L)\}, \tag{4.5}$$

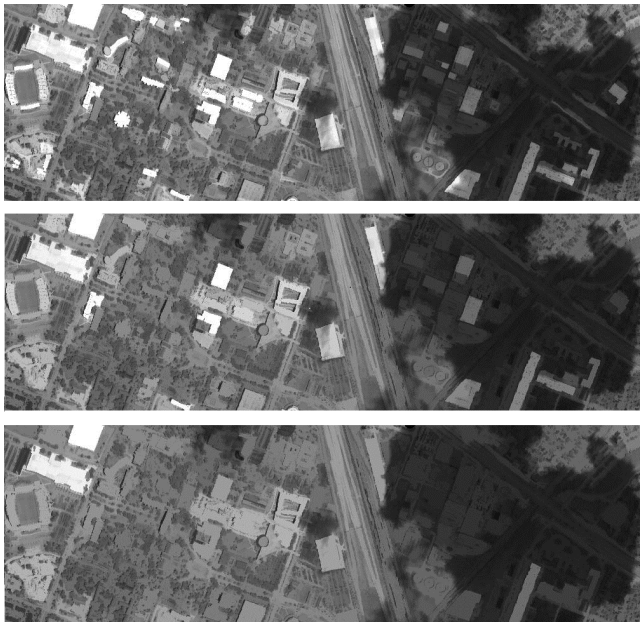
where $a_i, i = \{1, \cdots, m\}$ are the attributes including the area, diagonal of bounding box, length and standard deviation, etc.

Figure 4.6 shows some of the obtained APs on the hyperspectral and LiDAR images. The objects of the hyperspectral image under cloud shadow appear to be darker. Moreover, many objects (even in different categories) exhibit similar intensities (Figure 4.6a). APs of LiDAR data are clearly less influenced by the cloud (Figure 4.6b), small objects disappear as the scale increases. With EMAP, additional spatial and elevation can be extracted.

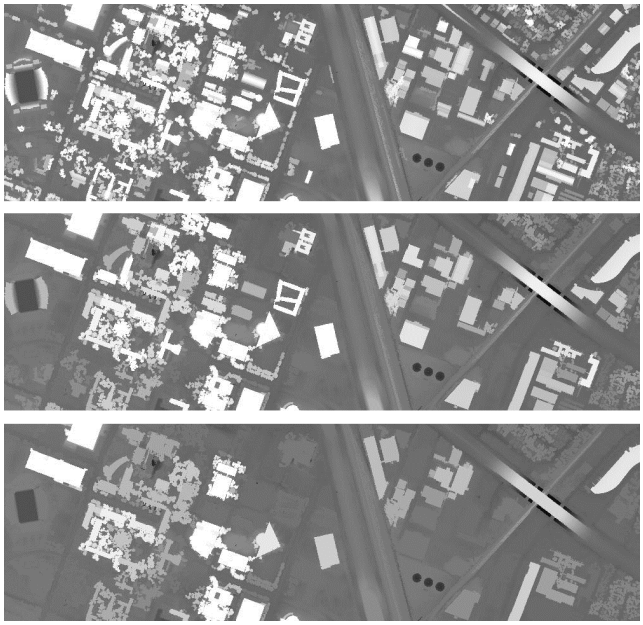
4.2.2 Multiple Feature Classification

From hyperspectral and LiDAR data, three types of feature sources can be obtained: spectral values from the original spectrum of the hyperspectral image, $\text{EMAPs}_{\text{hsi}}$ from the hyperspectral image and MAPs_{lid} from the LiDAR data, all of them having high dimensionality. If these features are stacked together directly, the dimensionality of this stacked vector will be very large, thus leading to the problem of the curse of dimensionality. Moreover, the stacked vector will contain redundant information and noise. Therefore, we propose to use feature extraction (FE) methods before stacking, to reduce the dimensionality of the spectral features. Concretely, we compute the low dimensional features $\text{EMAPs}_{\text{hsi}}$ and MAPs_{lid} before fusing the extracted low-dimensional features together for classification. Here the non-parametric weighted feature extraction (NWFE) methid [Kuo 04] is chosen, as it is proven to be efficient to extract discriminative features for classification of hyperspectral image [Ghamisi 14a].

Figure 4.7a shows the proposed multiple feature classification strategy. First, the original hyperspectral data is transformed by NWFE to obtain a reduced set of effective spectral features ($Fspe$) that contains the spectral information of the hyperspectral data. In parallel, the hyperspectral image is

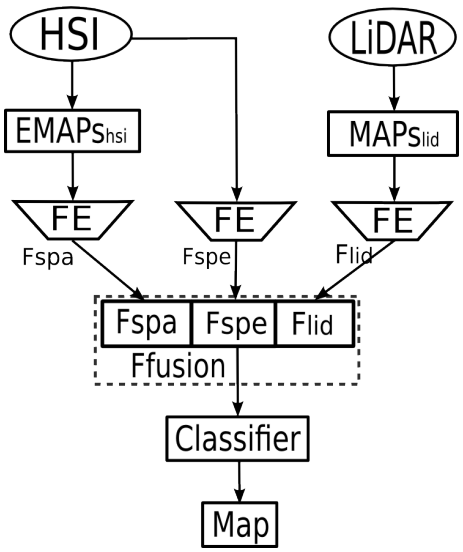


(a) APs of hyperspectral image with first PC

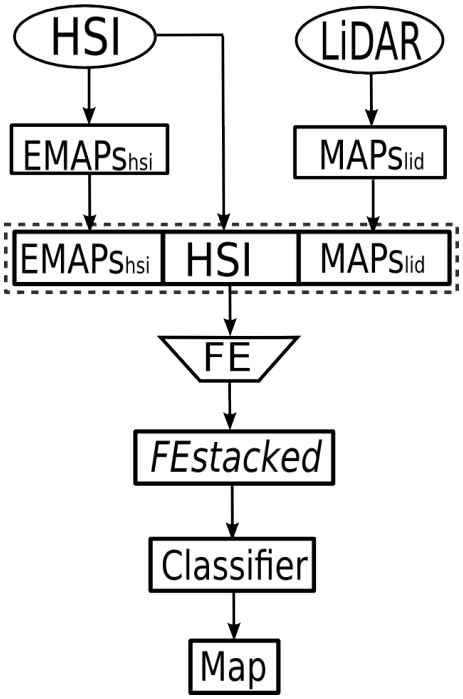


(b) APs of LiDAR data

Figure 4.6: Attribute thinning with "area" attribute. From up to down, the area size was set to 200, 500, and 1000 respectively. (a) APs of hyperspectral image with first PC; (b) APs of LiDAR data.



(a) Proposed multiple feature classification



(b) Multiple feature classification proposed in [Khodadadzadeh 15]

Figure 4.7: Multiple feature classification

transformed by PCA, and the first few important PCs that correspond to 99% of the cumulative variance are used to construct the $\text{EMAPs}_{\text{hsi}}$. If there are c PCs, each AP is composed of n thickening and n thinning transformations of the corresponding PC for each attribute and the number of attributes is m, then there are in total $c \times (m \times (2n) + 1)$ features in the EMAPs. In order to reduce redundancy and noise, avoid the curse of dimensionality and save processing time, NWFE is applied to extract an effective feature set ($Fspa$) from $\text{EMAPs}_{\text{hsi}}$ before classification. On the LiDAR data, exactly the same is done to extract an effective elevation feature set ($Flid$) from the MAPs_{lid} . Finally, the obtained $Fspe$, $Fspa$ and $Flid$ are concatenated into one stacked vector $Ffusion$.

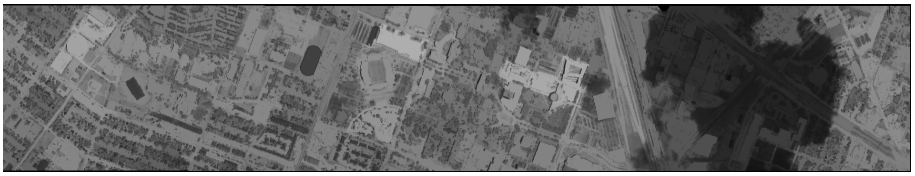
Another strategy would be to make up a large stacked vector from the spectral features, the spatial profiles of $\text{EMAPs}_{\text{hsi}}$ and elevation profiles of MAPs_{lid} [Khodadadzadeh 15], and then extract effective features from this large stacked vector (Figure 4.7b). In that case however, since the different feature sources have different distributions, the information will be not equally represented in the stacked vector, and some important features may get lost or mixed if we project stacked features from different sources into a low-dimensional feature space together. We verified experimentally that the first strategy is the more effective one.

4.2.3 Cloud-shadow Detection

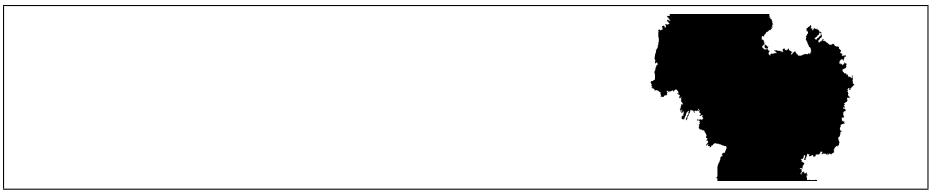
Cloud shadow cannot be always avoided during the acquisition of optical remote sensing data. The presence of cloud and shadow complicates the analysis of remote sensing data, leading e.g. to false detection of land cover change [Zhu 14], biased estimation of Normalized Difference Vegetation Index (NDVI) values, and mistakes in classification tasks. Therefore, the detection of cloud shadow is an initial and important step [Arvidson 01]. Actually, many approaches have been developed to detect cloud shadow, such as geometry-based methods [Luo 08] and the Fmask algorithm [Zhu 12, Frantz 15].

Since cloud-shadow detection is not the primary goal of this work, we will apply a simple method based on area attribute filters [Mura 11] to detect big cloud-shadow regions, because in our specific case study area, the area of the cloud-shadow is much larger and darker than other ground objects. By increasing the thresholds of the area attribute, more and more bright objects are filtered out, leaving finally the largest dark cloud shadow region (Figure 4.8b). The cloud-shadow mask is then obtained by binarizing the result (Figure 4.8c). In fact, there is a very small cloud-shadow region present at the top center-right of the image. We just choose the large main cloud-shadow as an example here, as all shadowed testing samples are located in this large cloud-shadow region. Denote $\mathbf{G} = \{g_{ij}\}$ as the cloud-shadow mask, with pixel values $g_{ij} = 0$ in the cloud-shadow region and $g_{ij} = 1$ in the shadow-free region.

(a) False RGB image of hyperspectral data



(b) Area attribute thinning with area size 3000 of hyperspectral data



(c) Extracted cloud-shadow map

Figure 4.8: Cloud map detection.

4.2.4 Co-training Samples Generation

In this section, we describe a new method to generate and label a separate training set (called co-training samples) for the cloud-shadow regions. Since LiDAR data are not influenced by clouds (based on laser pulse), our proposed method uses single elevation information (i.e., *Flid*, see section 4.2.2) to obtain an initial classification map (M_{lid}). However, single elevation information from LiDAR data is not sufficient for a reliable classification, as many objects from different class in urban areas are of similar height. Therefore, we combine the spectral and spatial information from the hyperspectral image with M_{lid} to generate new co-training samples from cloud-shadow regions in the following way (Figure 4.9a):

Suppose $\mathbf{X}^{Sta} = \{\mathbf{x}_i^{Sta}\}_{i=1}^{n}$ denote the set of samples, $\mathbf{x}_i^{Sta} = \{\mathbf{x}_i^{spe}; \mathbf{x}_i^{spa}\}$, $\mathbf{y}^{'} = \{y_i^{'}\}_{i=1}^{n}$, $\mathbf{x}_i^{spe}$ and $\mathbf{x}_i^{spa}$ denote the spectral information in the hyperspectral image and spatial information in $\text{EMAPs}_{\text{hsi}}$ of the ith pixel respectively, $y_i^{'} \in \{1, \cdots, C\}$ denotes the label of pixel i in the classification map (M_{lid}, see Figure 4.9a) obtained by the LiDAR feature source.

In fact, multiple feature sources (i.e., the original hyperspectral image, $\text{EMAPs}_{\text{hsi}}$ from hyperspectral image and MAPs_{lid} from LiDAR image) can be seen as information from different aspects for pixels. For two samples, if their information are similar from all aspects, we assume they belong to same

class and share same labels. Let $\mathbf{X}^{'}_{c(1)}$ be a set of initial co-training samples (selected based on Map M_{lid} and information from hyperspectral data) which belong to class c. $\mathbf{X}^{'}_{c(1)}$ can be obtained as follows:

$$\mathbf{X}^{'}_{c(1)} = \{\mathbf{x}_i^{Sta} : \mathbf{x}_i^{spe} \in knn(\mathbf{m}^{spe}_{c(1)})\, and\, \mathbf{x}_i^{spa} \in knn(\mathbf{m}^{spa}_{c(1)})\}, \tag{4.6}$$

where

$$\mathbf{m}^{spe}_{c(1)} = \frac{1}{n_{c(0)}} \sum_{i=1}^{n_{c(0)}} \mathbf{x}_i^{spe}, \text{with } y_{\mathrm{i}}^{'} = \mathrm{c}, \tag{4.7}$$

$$\mathbf{m}^{spa}_{c(1)} = \frac{1}{n_{c(0)}} \sum_{i=1}^{n_{c(0)}} \mathbf{x}_i^{spa}, \text{with } y_{\mathrm{i}}^{'} = \mathrm{c}, \tag{4.8}$$

$\mathbf{m}^{spe}_{c(1)}$ and $\mathbf{m}^{spa}_{c(1)}$ can be seen as the initial center of class c in spectral feature space and spatial feature space respectively, $n_{c(0)}$ is the number of initial generated co-training samples in class c, $knn(\mathbf{m}_{c(1)})$ denotes the set of k-nearest neighbors of $\mathbf{m}_{c(1)}$. Here k-nearest neighbors are selected based on Euclidean distance, as Euclidean distance is simple and widely used in k-nearest neighbors searching. In this way, the generated co-training samples for each class have similar spectral, spatial and elevation information. For details see Figure 4.9.

However, as map M_{lid} is obtained only based on a single elevation feature source, the classification accuracies for some classes are relatively low, leading to less accurate class centers. As a result, the co-training samples generated based on equation (4.6) are not reliable. In order to solve this problem, we iteratively update the class centers, similar as in a mean shift algorithm. Suppose $\mathbf{X}^{'}_{c(k)}$ is a set of training samples belonging to class c, generated in the kth iteration. This set can be obtained from the training set at the $(k-1)$th iteration through the following criterion:

$$\mathbf{X}^{'}_{c(k)} = \{\mathbf{x}_i^{Sta} : \mathbf{x}_i^{spe} \in knn(\mathbf{m}^{spe}_{c(k)})\, and\, \mathbf{x}_i^{spa} \in knn(\mathbf{m}^{spa}_{c(k)})\}, \tag{4.9}$$

where

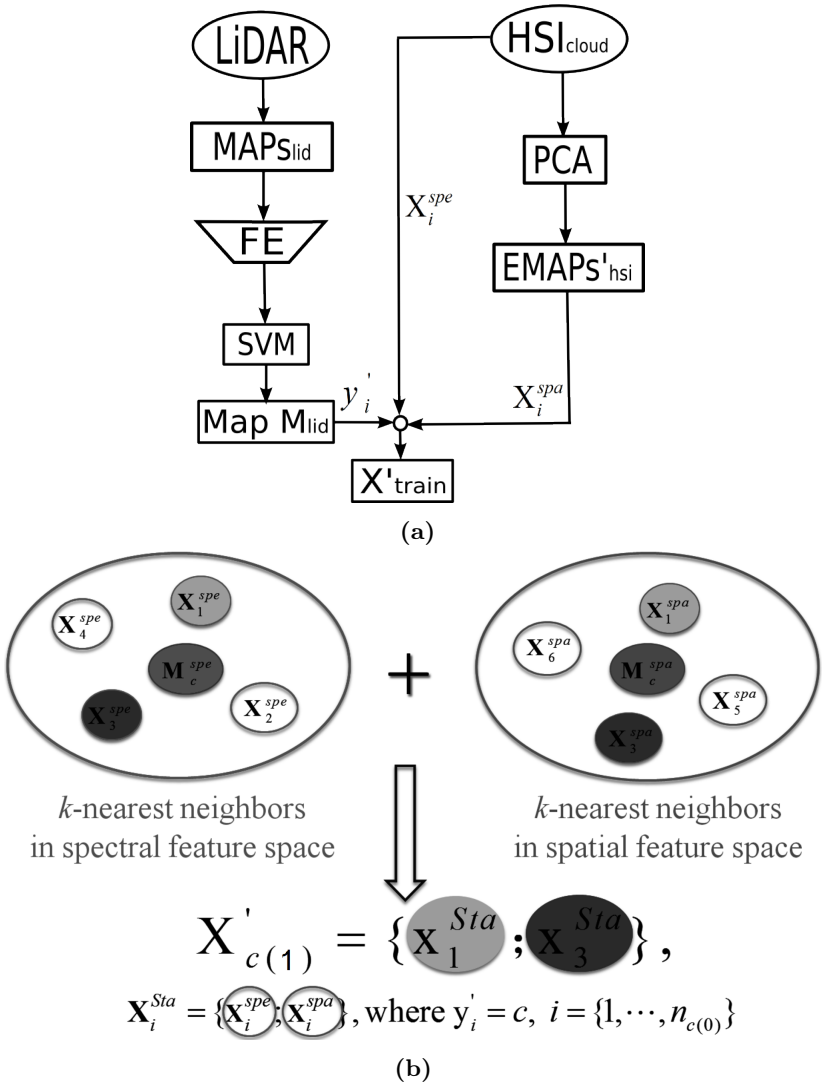
$$\mathbf{m}^{spe}_{c(k)} = \frac{1}{n_{c(k-1)}} \sum_{i=1}^{n_{c(k-1)}} \mathbf{x}_i^{spe}, \text{with } \mathbf{x}_{\mathrm{i}}^{\mathrm{Sta}} \in \mathbf{X}^{'}_{\mathrm{c(k-1)}}, \tag{4.10}$$

$$\mathbf{m}^{spa}_{c(k)} = \frac{1}{n_{c(k-1)}} \sum_{i=1}^{n_{c(k-1)}} \mathbf{x}_i^{spa}, \text{with } \mathbf{x}_{\mathrm{i}}^{\mathrm{Sta}} \in \mathbf{X}^{'}_{\mathrm{c(k-1)}}, \tag{4.11}$$

$\mathbf{m}^{spe}_{c(k)}$ and $\mathbf{m}^{spa}_{c(k)}$ denote the centers of $\mathbf{X}^{'}_{c(k-1)}$ in spectral feature space and spatial feature space, respectively, $n_{c(k-1)}$ is the number of samples in $\mathbf{X}^{'}_{c(k-1)}$. The iteration procedure can be stopped by introducing two thresholds ε^{spe} and ε^{spa}. When

$$\mid \mathbf{m}^{spe}_{c(k)} - \mathbf{m}^{spe}_{c(k-1)} \mid < \varepsilon^{spe} \;\&\; \mid \mathbf{m}^{spa}_{c(k)} - \mathbf{m}^{spa}_{c(k-1)} \mid < \varepsilon^{spa}, \tag{4.12}$$

the center of co-training samples in each class are stable. We define the final co-training samples set as: $\mathbf{X}^{'}_{train} = \{\mathbf{X}^{'}_{1(k)}, \mathbf{X}^{'}_{2(k)}, \cdots, \mathbf{X}^{'}_{c(k)}\}$. The algorithmic procedure of the proposed co-training samples selection method is formally stated in Algorithm 2.



(a)

$$\mathbf{X}'_{c(1)} = \{\mathbf{x}_1^{Sta}; \mathbf{x}_3^{Sta}\},$$

$$\mathbf{X}_i^{Sta} = \{\mathbf{x}_i^{spe}; \mathbf{x}_i^{spa}\}, \text{where } \mathrm{y}'_i = c,\ i = \{1, \cdots, n_{c(0)}\}$$

(b)

Figure 4.9: (a) Co-training samples generation; $\mathbf{x}_i^{spe}$ and $\mathbf{x}_i^{spa}$ represent spectral reflectance and $\mathrm{EMAPs}_{\mathrm{hsi}}$ of the ith pixel; (b) $\mathbf{x}_i^{Sta} = \{\mathbf{x}_i^{spe}; \mathbf{x}_i^{spa}\}$; the pixel is labeled c in map M_{lid} ($y_i^{'} = c$); $\mathbf{m}_c^{spe}$ and $\mathbf{m}_c^{spa}$ are the centers of class c in spectral and spatial feature space, here $\mathbf{x}_1^{Sta}$ and $\mathbf{x}_3^{Sta}$ are nearest neighbors of the center of class c, both in spectral and spatial feature space, and selected as candidate co-training samples.

4.2.5 Classification Map Fusion

After obtaining the new co-training samples under cloud-shadow regions, multiple features classification is applied in the same way as for cloud-free regions,

Algorithm 2 Co-training samples generation algorithm

Input: Samples under cloud shadow $\mathbf{X}^{Sta} = \{\mathbf{x}_i^{Sta}\}_{i=1}^{n}$, $\mathbf{x}_i^{Sta} = \{\mathbf{x}_i^{spe}; \mathbf{x}_i^{spa}\}$, and their labels $\mathbf{y}^{'} = \{y_i^{'}\}_{i=1}^{n}$ in Map M_{lid}.
Calculate the initial spectral center of every class $\mathbf{m}_{c(1)}^{spe}(c \in \{1, \cdots, C\})$ via equation (4.7).
Calculate the initial spatial center of every class $\mathbf{m}_{c(1)}^{spa}(c \in \{1, \cdots, C\})$ via equation (4.8).
Find the common nearest neighbors $\mathbf{X}_{c(1)}^{'}$ via equation (4.6).
$k = 1$
Loop
    Update $\mathbf{m}_{c(k)}^{spe}$ via equation (4.10).
    Update $\mathbf{m}_{c(k)}^{spa}$ via equation (4.11).
    if $(\mid \mathbf{m}_{c(k)}^{spe} - \mathbf{m}_{c(k-1)}^{spe} \mid < \varepsilon^{spe}$ *and* $\mid \mathbf{m}_{c(k)}^{spa} - \mathbf{m}_{c(k-1)}^{spa} \mid < \varepsilon^{spa})$ **then**
        break Loop
    end if
    Update $\mathbf{X}_{c(k)}^{'}$ via equation (4.9).
    $k \leftarrow k + 1$
End Loop
Output: Final generated co-training samples $\mathbf{X}_{train}^{'} = \{\mathbf{X}_{1(k)}^{'}, \mathbf{X}_{2(k)}^{'}, \cdots, \mathbf{X}_{C(k)}^{'}\}$.

the only difference being the way the training samples were obtained, where in the cloud-free regions, we use available training samples outside of the cloud mask, while in the cloud-shadow regions, we apply our proposed co-training samples generation procedure. The final classification map is obtained by fusion of the two maps: M_{cloud} and $\mathrm{M}_{nocloud}$.

$$M_{fusion} = g_{i,j} M_{cloud} + \overline{g}_{i,j} M_{nocloud} \tag{4.13}$$

where $\overline{g}_{i,j}$ is the logical inverse of $g_{i,j}$.

4.3 Experiments

4.3.1 Data Description

In 2013, the Data Fusion Technical Committee of the IEEE Geoscience and Remote Sensing Society (GRSS) organized a contest involving two types data sources: a cloud-shadow hyperspectral image and a LiDAR derived digital surface model (DSM), both at the same spatial resolution (2.5m) [Hyp 13]. The competition was established to devise advanced methods for fusion and classification of hyperspectral and LiDAR data [Debes 14]. This data set was captured by the NSF-funded Center for Airborne Laser Mapping (NCALM) using the compact airborne spectrographic imager (CASI-1500) on June 2012 over the

University of Houston campus and its neighboring urban area. The hyperspectral image has 144 spectral bands with a wavelength range from 380 to 1050 nm. The whole scene of the data contains 349×1905 pixels. The ground truth provided for this data set contains 15 classes, summed up in Table 4.2, also mentioning between brackets the available numbers of training/test samples. The false color image and LiDAR image are shown in Figure 4.1a and Figure 4.1b, the distribution of training and test samples are shown in Figure 4.1c and Figure 4.1d. The given scene contains a large cloud-shadow region (see Figure 4.1a), which distorts the spectral reflectance of objects in the hyperspectral image (darkening effect). More information can be found in [Hyp 13].

4.3.2 Experimental Setup

The input hyperspectral image is transformed by principal component analysis (PCA), and the first two principal components are kept since they contain almost all of the variance in the hyperspectral image (cumulative variation of more than 99%). For the feature extraction (FE), we use the non-parametric weighted feature extraction (NWFE) method [Kuo 04], as it has been shown to be efficient in many applications [Ghamisi 14a]. To generate the EMAPs, four attributes are considered: 1) (a) area λ_a (related the size of the objects); 2) (s) standard deviation λ_s (as a measure of homogeneity of the objects); 3) (d) diagonal of the box bounding the objects λ_d; 4) (i) moment of inertia λ_i (as a measure of the elongation of the objects).

For the purpose of generating enough attribute profiles from hyperpspectral image and LiDAR data [Ghamisi 14b], we select large amount (≥ 10) threshold values for each attribute filters as follows:

λ_a=[50 100 200 300 500 700 1000 1500 2000 2500 3000 4000];
λ_s=[5 10 15 20 25 30 35 40 50 60];
λ_d=[5 10 25 50 75 100 150 200 300 400 500];
λ_i=[0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1].

The SVM classifier [Chang] with radial basis function (RBF) kernels is applied, containing two parameters: the penalty factor C and the RBF kernel widths γ. C is optimized within the given set $\{10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$ and γ is optimized within the given set $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$ by five-fold cross validation.

In order to validate the efficiency of our proposed framework, we compare the classification results by using following features as an input for SVM classifier:

1. Original hyperspectral image (Raw_{hsi});
2. spectral features **Fspe** extracted from the hyperspectral image by NWFE;
3. Spatial features **Fspa** extracted from $\text{EMAPs}_{\text{hsi}}$ by NWFE;
4. Elevation features **Flid** extracted from MAPs_{lid} by NWFE;

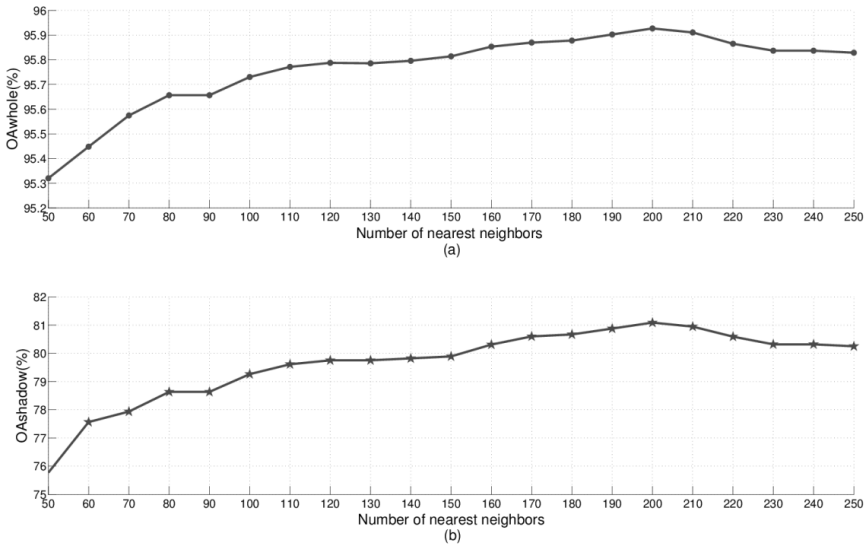


Figure 4.10: (a) OA for the whole scene with increasing number of nearest neighbors; (b) OA for shadow area with increasing number of nearest neighbors.

5. Stacked features **FEstacked**, stacking all spectral features, $\text{EMAPs}_{\text{hsi}}$ and MAPs_{lid} first, similar as the approach of [Khodadadzadeh 15], and then extracting low-dimensional features; by NWFE, as shown in Figure 4.7(b);

6. Fusion of stacked features **Fspe**, **Fspa** and **Flid** but only using the original training samples **Ffusion**; this is the proposed approach without the co-training samples generation procedure;

7. Features from the generalized graph-based fusion method **Fggf**, the same as in the approach of [Liao 15].

The classification results are quantitatively evaluated by measuring the overall classification accuracy (OA), the average accuracy (AA), Kappa coefficient (κ) on the test samples, shown in Figure 4.1d.

4.3.3 Effect of Number of Nearest Neighbors for Co-training Generation

The number of nearest neighbors of each class center (e) is an important parameter in the co-training samples generation procedure. On the one hand, when e is too small, the number of co-training samples will be insufficient. On the other hand, a large e will lead to mislabeling of co-training samples, as some samples with different labels will be included in the nearest neighbors if e is too large. To investigate the effect of the number of nearest neighbors on the

classification accuracy, we performed classification experiments with different numbers of e. The number of nearest neighbors was increased from 50 to 250 with a step size of 10. Figure 4.10 shows the OA for the whole scene and for the shadow area in function of an increasing number of nearest neighbors. As can be seen, the average OA increases as the number of nearest neighbors grows from 50 to 200, and then decreases with more nearest neighbors. This indicates that if e is set to a small value, the generated number of co-training samples will be too small to train the classifier well; if e is set to a large value, the possibility of mislabeling co-training samples increases, leading to poor classification performances. For this data set, we have set the number of nearest neighbors to 200 in our experiments.

4.3.4 Classification Results on the data set

This section mainly explores the performance (classification accuracy) of the proposed method, compared to the other methods. The resulting accuracies are reported in Tables 4.2, 4.3 and 4.4, and the classification maps are shown in Figure 4.11 for visual comparison. From the tables and figures, we conclude the following:

1. The proposed framework improves all results in terms of the overall accuracy (OA), the average accuracy (AA), the Kappa coefficient (κ) and the quality of the classification map. On the shadow-free region, it outperforms the state of the art at least 2% in the overall classification accuracy, in the cloud-shadow region, the improvements are dramatic. On the whole scene, the proposed framework improves the OA with 3.87%-20.10% over the other schemes.

2. In general, it can be observed that fusion of multiple features (spectral, spatial and elevation features) leads to better classification performances in comparison with using one single type of features. This shows that the chosen sets of features are efficient and fusing them exploits the information contained in both data sources.

3. When investigating the classification accuracies for each class separately in Table 4.4, it can be clearly noticed that, when single features are used, the Raw_{hsi} approach produces better results on class 'Tree', whereas the $Fspa$ scheme performs better on classes 'water', 'Residential' and 'Road'. However spectral or spatial features from the hyperspectral image perform poor on classes 'Commercial' and 'Railway'. On the contrary, $Flid$, extracted from the LiDAR data performs much better on these two classes. Classification accuracies for most classes improves by fusing those three features, especially for classes 'Residential', 'Road' and 'Parking Lot 2'. The generalized graph-based fusion method ($Fggf$) [Liao 15] improves the classification accuracy on classes 'Grass Stressed', 'Tree' and 'Highway'. The proposed framework obtains the best classification accuracies on 9 of the 15 classes.

Table 4.2: Classification accuracies for the shadow-free regions obtained by the different methods.

Features	Raw$_{hsi}$	$Fspe$	$Fspa$	$Flid$	$FEstacked$	$Ffusion$	$Fggf$	Proposed
Number of Features	144	15	15	15	45	45	22	45
Grass Healthy (198/875)	98.86	98.40	99.54	69.71	99.66	**100.00**	99.77	**100.00**
Grass Stressed (190/906)	96.36	96.58	89.51	71.52	97.90	98.34	**99.34**	98.34
Grass Synthetis (192/505)	99.80	**100**	**100**	94.85	**100**	**100**	**100**	**100**
Tree (188/986)	98.49	95.54	89.55	74.54	98.99	97.26	**99.19**	97.26
Soil (186/1056)	97.92	98.58	98.30	83.14	99.43	**99.83**	99.81	**99.83**
Water (182/143)	95.10	95.10	**99.30**	84.62	95.80	95.80	95.80	95.80
Residential (196/992)	82.36	79.44	83.17	84.58	92.14	**98.99**	94.46	**98.99**
Commercial (191/622)	72.67	82.80	63.50	92.44	93.73	**93.89**	91.64	**93.89**
Road (193/1040)	79.35	76.66	80.50	66.28	86.74	**94.43**	89.34	**94.43**
Highway (191/710)	86.62	93.66	96.06	93.52	96.34	**99.72**	91.13	**99.72**
Railway (181/902)	91.57	90.47	87.58	96.78	96.78	**99.22**	95.01	**99.22**
Parking Lot 1 (192/1041)	84.34	78.19	88.18	66.38	89.15	**97.98**	81.08	**97.98**
Parking Lot 2 (184/265)	77.36	76.23	78.11	68.68	86.04	**87.92**	80.00	**87.92**
Tennis Court (181/247)	99.60	99.60	**100**	99.60	**100**	**100**	**100**	**100**
Running Track (187/473)	97.25	97.04	**100**	58.35	98.52	98.73	98.73	98.73
OA (%)	90.28	89.77	89.42	79.03	95.24	**97.91**	94.37	**97.91**
AA (%)	90.58	90.55	90.22	80.33	95.37	**97.47**	94.35	**97.47**
κ	0.894	0.889	0.885	0.773	0.948	**0.977**	0.939	**0.977**

Table 4.3: Classification accuracies for the cloud-shadow regions obtained by the different methods.

Features	Raw_{hsi}	$Fspe$	$Fspa$	$Flid$	$FEstacked$	$Ffusion$	$Fggf$	Proposed
Number of Features	144	15	15	15	45	45	22	45
Grass Healthy (85/178)	0.00	0.00	0.00	55.06	0.00	0.00	0.00	**84.26**
Grass Stressed (102/158)	0.00	0.00	0.00	51.89	0.00	0.00	**91.78**	89.24
Tree (89/70)	0.00	0.00	0.00	**100.00**	0.00	0.00	**100.00**	**100.00**
Residential (75/80)	1.25	71.25	**90.00**	58.75	0.00	66.25	63.74	77.50
Commercial (121/431)	29.23	0.00	0.00	82.60	69.37	49.19	84.74	**88.17**
Road (71/19)	0.00	0.00	0.00	**63.16**	0.00	0.00	52.63	57.89
Highway (92/326)	0.00	51.23	0.00	46.63	15.34	23.93	**76.07**	69.49
Railway (111/152)	4.61	71.05	15.79	92.11	91.45	92.11	**93.42**	92.76
Parking Lot 2 (27/20)	5.00	**10.00**	0.00	5.00	0.00	0.00	0.00	**10.00**
OA (%)	9.42	22.31	6.70	66.71	34.96	34.54	71.97	**81.15**
AA (%)	4.45	22.61	11.75	61.56	20.52	36.60	62.49	**74.37**
κ	0.160	0.134	0.111	0.679	0.206	0.230	0.717	**0.796**

Table 4.4: Classification accuracies for the whole image obtained by the different methods.

Features	Raw$_{hsi}$	$Fspe$	$Fspa$	$Flid$	$FEstacked$	$Ffusion$	$Fggf$	Proposed
Number of Features	144	15	15	15	45	45	22	45
Grass Healthy (1053)	82.15	81.77	82.72	57.93	82.81	83.00	82.91	**97.34**
Grass Stressed (1064)	82.05	82.24	76.22	60.90	83.36	83.74	**99.44**	96.99
Grass Synthetis (505)	99.80	**99.80**	**100**	94.85	**100**	**100**	**100**	**100**
Tree (1056)	92.90	89.20	83.62	76.23	92.42	91.48	**99.24**	97.44
Soil (1056)	97.92	98.58	98.30	83.14	99.43	99.81	99.81	**99.83**
Water (143)	95.10	95.10	**99.30**	84.62	95.80	95.80	95.80	95.80
Residential (1072)	76.31	78.82	83.68	82.65	85.26	97.11	91.42	**97.38**
Commercial (1053)	54.89	48.91	37.51	**92.97**	83.76	75.59	92.50	91.55
Road (1059)	78.00	75.35	79.13	66.48	85.27	92.82	88.76	**93.86**
Highway (1036)	59.36	80.31	65.83	70.17	70.85	75.87	84.85	**90.15**
Railway (1054)	79.03	87.67	77.23	96.39	97.25	**99.15**	95.73	98.58
Parking Lot 1 (1041)	84.34	78.19	88.18	66.38	89.15	97.69	81.08	**97.69**
Parking Lot 2 (285)	72.28	71.58	72.63	63.86	80.00	77.89	74.39	**82.46**
Tennis Court (247)	99.60	99.60	**100**	99.60	**100**	**100**	**100**	**100**
Running Track (473)	97.25	97.04	**100**	58.35	98.52	98.73	98.73	98.73
OA (%)	80.78	81.96	79.70	75.82	88.16	90.44	92.05	**95.92**
AA (%)	83.40	84.29	82.96	76.97	89.55	91.25	92.31	**95.65**
κ	0.793	0.805	0.779	0.738	0.871	0.896	0.914	**0.958**

4. From the results reported on the shadow-free region (Table 4.2) and the whole image (Table 4.4), one can infer that fusing the features extracted from each source (hyperspectral image, $\text{EMAPs}_{\text{hsi}}$ and MAPs_{lid}) works better than using the features extracted from the stacked vector of the original hyperspectral image, $\text{EMAPs}_{\text{hsi}}$ and MAPs_{lid}, with an improvement of almost 3%. The main reason for this is that, because of their different nature, when fusing features from different sources and then projecting them on a lower dimensional space, information gets mixed up and lost.

5. By comparing the classification maps in Figure 4.11 and classification accuracies on the cloud-shadow region (Table 4.3), we can see that most of the objects under the cloud-shadow region are not well classified when only using the training samples located in the shadow-free region. Some objects in the cloud-shadow region are classified better by using features extracted from LiDAR data, because the elevation information contained in the morphological features of LiDAR data is not influenced by the cloud. For many other objects, the results are not good as the elevation information is not sufficiently discriminative. Taking all feature sources into consideration does not much improve the classification accuracy for most of the classes. The proposed framework leads to an improved classification of most classes, due to the selection and use of specific training samples in the cloud-shadow region.

As the described data set [Hyp 13] is very popular and open access, it has been used in many recentl state of the art comparisons, such as in [Khodadadzadeh 15], [Liao 15], [Bao 16] and [Zhong 16]. Compared with the experimental results from these references, the proposed scheme performs better on either cloud-shadow or shadow-free regions, with overall classification accuracy 97.91% and 81.15% respectively. This proves the proposed fused features are effective and distinguishable, and generating new training samples from the cloud-shadow regions is an efficient solution.

4.4 Conclusion

In this Chapter, we described a new method for classification of cloud mixed remote sensing scenes by fusion of hyperspectral and LiDAR data. The proposed method generates new samples as co-training samples in the cloud-shadow regions and classifies shadow-free and cloud-shadow regions separately using their own sets of training samples. In order to better combine hyperspectral and LiDAR data, additional spatial (EMAP_{hsi}) and elevation (MAP_{lid}) features are extracted from hyperspectral and LiDAR data, and effectively integrated without any regularization or weight parameters. Experimental results on the classification of the real cloud-shadow hyperspectral and LiDAR data show the efficiency of the proposed framework. In addition, the proposed approach can be thought of as a general framework, in which the feature extraction step can

be replaced by any other technique (kernel PCA, supervised or semi-supervised feature extraction, ...), possibly to improve classification accuracies. Moreover, the proposed framework is completely open and flexible in its capacity to integrate additional types of (e.g. infrared) features.

Recent Earth observation missions (Landsat series from NASA, Sentinel series from ESA) boost the use of the multi-sensor remote sensing imagery. However, cloud/shadow effects cannot be avoided in the optical sensors. Other sensors (e.g., synthetic aperture radar, thermal infrared, LiDAR, etc.) can provide complementary information for these cloud/shadow regions. The proposed framework is applicable for fusion of optical hyperspectral images and other images (e.g., SAR, thermal infrared), where multi-sensor images are available.

The research in this chapter lead to one journal publication and one proceeding as follows:

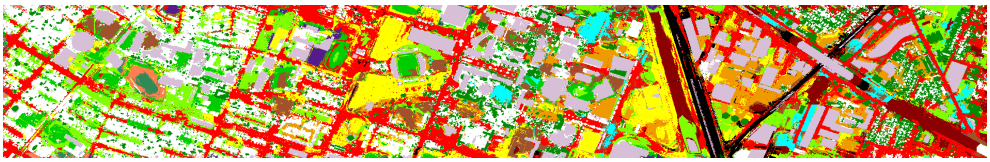
1. **Luo Renbo**, Liao Wenzhi, Zhang Hongyan, Zhang Liangpei, Pi Youguo, Scheunders Paul, Philips Wilfried, "Fusion of hyperspectral and LiDAR data for classification of cloud-shadow mixed remote sensing scene". IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. (under reviewing)

2. **Luo Renbo**, Liao Wenzhi, Zhang Hongyan, Pi Youguo, Philips Wilfried, "Classification of cloudy hyperspectral image and LIDAR data based on feature fusion and decision fusion". IEEE Geoscience and Remote Sensing International Symposium (IGARSS 2016). Jul. 2016.

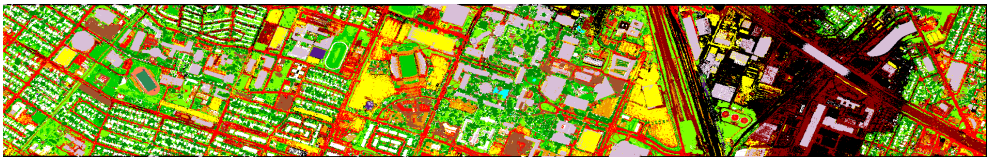
(a) RGB

(b) $Fspe$

(c) $Fspa$

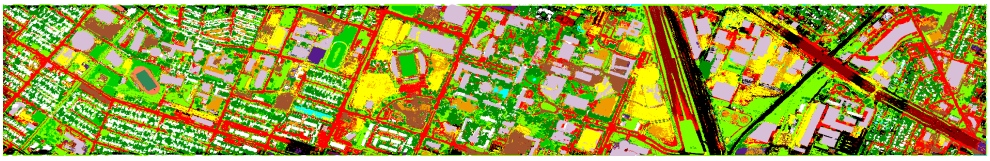


(d) $Flid$

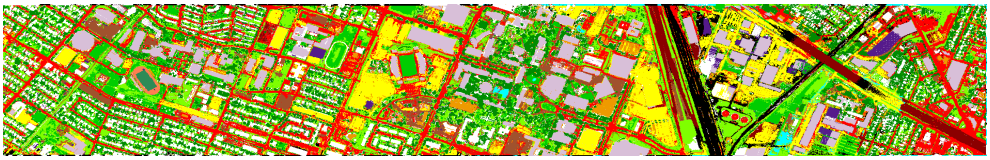


(e) $FEstacked$

(f) $Ffusion$



(g) $Fggf$



(h) Proposed

Figure 4.11: Classification maps produced by the described schemes.

5

GPU-Acceleration for Non-linear Feature Extraction

In chapters 2 and 3, we discuss supervised and semi-supervised feature extraction methods, all of them belong to linear methods. In this chapter, we will focus on non-linear methods, and provide solutions to accelerate the non-linear methods because of its computational complexities. One of traditional and widely used non-linear feature extraction methods is kernel principle component analysis (KPCA). However, the sequential implementations of KPCA (in central processing units (CPU)) require long processing time due to its relatively large computational complexity, such as the calculation and Eigen-decomposition of Gram matrix. In this chapter, a parallel version of KPCA based on graphics processing units (GPU) is presented and used for features extraction of hyperspectural images. Experiments are conducted using a hyperspectral data set, the results reveal that GPU-based parallel KPCA (GPKPCA) approach has great potential to improve computation speed without losing classification accuracy. The acceleration effect will be much more obvious, with bigger data.

5.1 Introduction

As explained in chapter 1, it is possible nowadays to collect hyperspectral images with hundreds of bands [Heesung 05,Lu 14]. Hyperspectral images contain much more information than regular RGB images, but most of their information content can still be summarized into a small number of the well chosen features [Qiao 15]. However, this initial processing and analysis of hyperspectral images is computationally intensive [Plaza 11, Christophe 11]. Thus, the development of computationally efficient techniques to extract the useful information quickly and accurately from large hyperspectral image data sest is becoming more and more important in Earth observation [Plaza 11].

Principal component analysis (PCA) is one of most traditional and popular feature extraction methods in the applications of hyperspectral images. It extracts features (the first several component principles) defined by analysing the covariance matrix of the original data. However, as a linear feature extraction method, PCA uses only second-order statistics, which limits its performance in many cases. Instead of computing the corresponding covariance matrix, the Candid covariance-free incremental principal component analysis (CCIPCA) [Weng 03] computes the principal components of a sequence of samples incrementally without estimating the covariance matrix (so covariance-free). To do this, CCIPCA keeps the scale of observations and computes the mean of observations incrementally, which is an efficient estimate for some well known distributions (e.g., Gaussian). CCIPCA is an iterative method and performs better than PCA from the experiments in [Weng 03] in most cases.

However, PCA and CCIPCA depend on linear projection, for classification on non-linearly separable data sets, linear feature extraction methods will perform poorly, as it is more difficult to separate non-linear data sets when projecting them into lower-dimensional feature space. In the last decade, a large number of non-linear techniques for dimensionality reduction have been proposed to address this problem. For an overview, see, e.g., [Arunasakthi 14, Jia 13]. In particular for real world data, the non-linear dimensionality reduction techniques may offer an advantage, because real world data is likely to form a highly non-linear manifold in the hypersectral data space. As a non-linear version of PCA, Kernel Principle Component Analysis (KPCA) [Scholkopf 98] is more suitable to describe non-linear, higher-order and complex distributions. In [Fauvel 09], KPCA performs better than PCA in terms of accuracy when extracting features from hyperspectral images. However, in order to capture non-linear kernel principal components, a large number of training samples are required, particularly for high dimensional data, this leads to serious computational load problems [Michiel 01].

To improve the compute efficiency, several parallel computing technologies, such as supercomputers, clusters, distributed computing, multicore central processing units (CPU), field-programmable gate arrays (FPGAs) and graphics processing units (GPU), have been used in hyperspectral data processing algorithms [Christophe 11, Lee 11]. Among these acceleration schemes, GPU-processing is quickly evolving as a standardized solution in hyperspectral processing due to its low cost and weight, portability and excellent computing performance for on-board processing [Plaza 11]. GPU has been wildly applied to huge remote sensing data analysis, including target detection [Bernabe 13], feature extraction [Qu 13, Du 13] and unmixing [Agathos 14, Chouzenoux 14].

However, to harness the compute power of GPUs, researchers need to express their algorithms in terms of texture operations, which is not easy. The Compute Unified Device Architecture (CUDA) was introduced in 2007 and OpenCL in 2009, to provide simpler GPU programming models. CUDA is a program language proposed by Nvidia on its G80 GPU series. Because of the benefits from the availability of numerous libraries, CUDA is becoming more

and more popular. This language is specific to one vendor and its hardware but still commonly used. OpenCL supports more hardware and provides a standard for general purpose parallel programming across GPU. This makes it easier for software developers to access powerful heterogeneous processing platforms portably and efficiently [Wu 15]. Actually, both CUDA and OpenCL exploit the concept of kernel. A kernel is a series of operations that will typically be applied to one pixel. Each kernel will be handled by one of the numerous GPU processors. Due to the C-like programming of CUDA and OpenCL, the learning curve to benefit from the GPU is significantly flattened [Kirk 10]. Many research papers demonstrate excellent implementations for a wild range of problems [Harish 07, Satish 09, Che 08].

In order to make the use of GPU more simple and practical, AccelerEyes LLC developed Jacket Matlab toolbox (www.accelereyes.com) which with Matlab language (with *name.m* files). The Jacket Matlab toolbox takes care of packaging Matlab data into Jacket's GPU data structure, and transforms Matlab code into GPU functions. The interpretative nature of the Matlab language is maintained by providing real-time, transparent access to the GPU compiler. To make it easier and convenience for developers to use, the functions and operations of GPU implemented within Jacket Matlab toolbox are transparent, and their calls are almost similar to the CPU Matlab implementation. Thus, Jacket is allowing a high level language as Matlab to utilize GPU without writing C or C++ code. Benefiting from the development of GPU and its operation language, GPU has been applied more and more on hyperspectral images analysis, such as classification [Christophe 11], band selection [Yang 11], endmember spectral unmixing [Chouzenoux 14].

In this Chapter, we explore GPU-accellerated non-linear feature extraction from high-dimensional hyperspectral images. Specifically, we developed an efficient GPU implementation of the KPCA feature extraction algorithm based on Jacket's Matlab Toolbox and compare a CPU with a GPU implementation.

The remainder of this Chapter is organized as follows. Section 5.2 briefly introduces the related background as GPU strategies and KPCA. Section 5.3 describes a new and fully optimized GPU-based KPCA implementation. Section 5.4 evaluates the proposed GPU-based parallel KPCA implementation in terms of computational performance by experiments. Section 5.5 concludes this Chapter with some remarks.

5.2 Related Background

5.2.1 GPU Architecture

As shown in Figure 5.1, GPU and CPU have different architectures. The architecture of CPU is single instruction single data stream (SISD) or multiple instruction multiple data stream (MIMD) for dual or quad-cores. For instance, the typical MIMD system includes an internet network, multiple processors and multiple memory blocks. During the data processing, each machine processor

Figure 5.1: CPU and GPU strategies

executes its own instruction pipeline. The strategy of CPU is to make the workload (one computed thread) run as fast as possible. Moreover, the CPU's efficiency depends on instruction/data pre-fetching, caching and speculative execution.

Different from CPU, the typical GPU architecture is organized into an array of highly threaded streaming multiprocessors (SMs), where each multiprocessor is characterized by a single instruction multiple data architecture (SIMD), i.e., in each clock cycle, each processor executes the same instruction while operating on multiple data streams. Each SM has a number of streaming processors that share a control logic and instruction cache and have access to a local shared memory and to local cache memories in the multiprocessor, while the multiprocessors have access to the global GPU (device) memory. GPUs can be abstracted in terms of a stream model, under which all data sets are represented as streams.

There is a hierarchy of parallelism on GPU. Parallelism comes in two flavours: outer, asynchronous parallelism between thread groups or thread blocks; and inner, synchronous parallelism within a thread block. Certain number of thread composition thread pieces, and a certain number of threads block is composed of one-dimensional or two-dimensional thread block grid. The same block thread can cooperate with each other, through shared memory to share data, and its implementation to coordinate access to memory. A block of all threads must be located in the same processor core; the number of threads per block is limited by memory resources in processor core. A kernel function may be executed by the thread blocks which have the same size, the number of threads that execute the kernel function should be equal to each block's thread number plus the number of blocks, and we call these for thread block grid. If thread blocks needed to execute independently, it must be able to perform in any order whether parallel or sequential execution. The block number of thread in a grid is usually limited by the processing of the data size, rather than by the hardware processor number, the former may far exceed than latter in quantity. Figure 5.2 shows the thread block grid in GPU.

In GPU, threads executing within a multiprocessor can share and communicate using the local memory, while threads executing on different multipro-

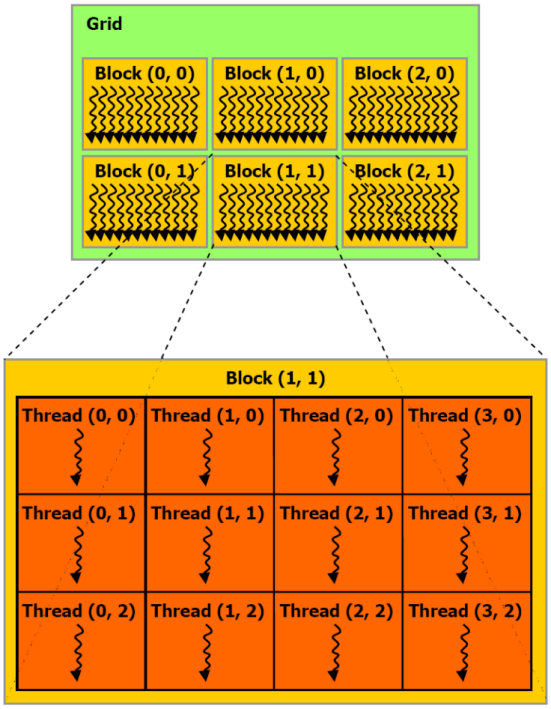


Figure 5.2: Thread block grid in GPU

cessors cannot communicate or synchronize. All the threads of a thread block will always be assigned as a group to a single multiprocessor, while different thread blocks can be assigned to different multiprocessors, the main differences between GPU and CPU can be seen in Table 5.

Table 5.1: Comparison between GPU and CPU.

GPU	**CPU**
composed of hundreds of cores that can handle thousands of threads simultaneously	composed of few cores with lots of cache memory that can handle a few software threads at a time
single instruction multiple data (SIMD)	single instruction single data stream(SISD)
high latency tolerance	low latency tolerance
most die surface for integer and fp units	few die surface for integer and fp units
optimized for computational and memory-intensive problems	optimized for caching or controlling flow operations

When porting one algorithm from CPU to GPU, the major challenge is how to take advantage of the parallel architecture. In order to increase the efficiency of calculation, CPU usually combines several parallelization techniques which still work sequentially, such as out-of-order execution, branch prediction and super-scalar. However, these techniques increase the computing complexity of the CPU. Moreover, the number of CPUs embedded on a single chip is limited. In contrast, GPUs can simplify each processing unit and pack thousands processing units on the chip, so it fits the data parallelism very well. For algorithms with high inherent parallelism and when the latency of each thread can be masked, the GPU will performs well. Specifically this is true when applying the same operation in parallel to all pixels of the image.

5.2.2 KPCA

Principal Component Analysis (PCA) [Hotelling 33] performs feature extraction through analyzing the covariance matrix of the original data. The eigenvalues of the covariance matrix are considered to be an indicator of the information content. Large values suggest more information content and low values indicate the presence of mostly noise. Suppose $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$ denotes the matrix of original data, where $\mathbf{x}_i \in \mathbb{R}^D$ (column vector), $i = 1, 2, \cdots, n$, n represents the total number of samples, and all data are centered as $\sum_{i=1}^{n} \mathbf{x}_i = 0$. In mathematical terms, PCA is a basis transformation to diagonalize an estimate of the covariance matrix of the centered data $\mathbf{X}$, defined as:

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \tag{5.1}$$

$\mathbf{C}$ is the covariance matrix the centered data $\mathbf{X}$ with $D \times D$. By solving the Eigen-decomposition of $\mathbf{C}$ as:

$$\mathbf{C}\mathbf{w} = \lambda \mathbf{w} \tag{5.2}$$

new coordinates in the Eigenvector basis $\{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d\}$ are d Eigenvectors of equation (5.2)), i.e. the orthogonal projections onto the Eigenvectors, are obtained and called principal components. For a testing sample $\mathbf{x}_i$, its principal components are $\mathbf{z}_i = \mathbf{W}^T \mathbf{x}_i$, $\mathbf{z}_i = \{z_{i1}, z_{i2}, \cdots, z_{id}\} \in \mathbb{R}^d$, z_{i1} is new coordinate in the Eigenvector basis $\mathbf{w}_1$.

For non-linear data sets, it is difficult to separate them with a linear hyperplane (one less dimension than the dimension of original feature space), but they may be separable if they are transformed into a higher or infinite dimensional ($\mathcal{F}$) Hilbert space. As shown in Figure 5.3, samples (belong two classes) with two-dimension distributes as two circles, it is very difficult to separate them with a line. However, it is very easy to separate these two class with a plane in three-dimensional feature space. Assume for the moment that there exists a function which can transform the original data into a higher or infinite dimensional ($\mathcal{F}$) Hilbert space $\mathbb{H}$ [Scholkopf 05]:

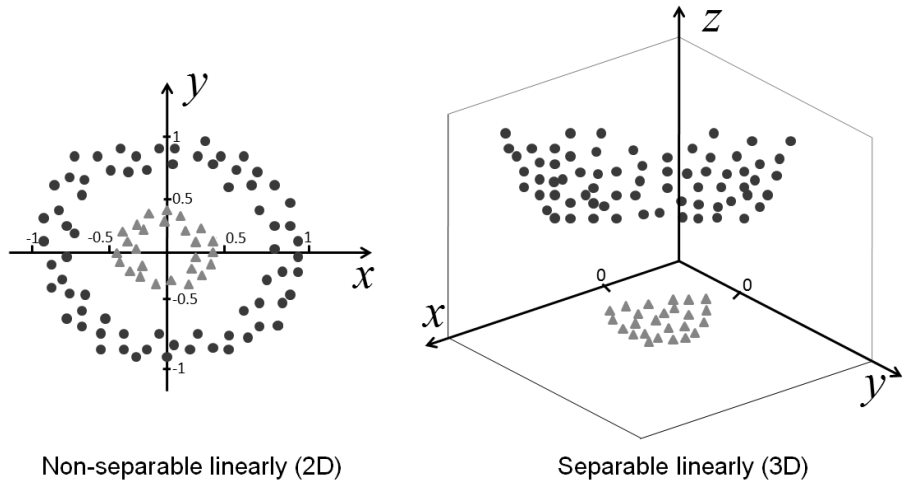


Figure 5.3: There exist samples as $\mathbf{x} = (x, y)$, $\{\mathbf{x} = (x, y)|x^2 + y^2 < 0.5\}$ belong class 1, $\{\mathbf{x} = (x, y)|x^2 + y^2 > 0.5\}$ belong class 2, they are not be separated by a line, when they are transformed into three-dimensional space as $\mathbf{x} = (x, y, x^2 + y^2)$, they can be separated by a plane.

$$\phi : \begin{array}{l} \mathbb{R}^D \to \mathbb{H}^{\mathcal{F}} \\ \mathbf{x}_i \to \phi(\mathbf{x}_i) \end{array} \tag{5.3}$$

A new data set can be obtained in the Hilbert feature space as $\Phi(\mathbf{X}) = \{\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_n)\}$. We do not require that ϕ can be calculated easily; it is enough that inner products between samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are preserved by Φ: $\kappa_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$. The covariance matrix $\overline{\mathbf{C}}$ is defined in the Hilbert feature space as follows:

$$\overline{\mathbf{C}} = \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T \tag{5.4}$$

It satisfies the Eigenvalue equation:

$$\overline{\mathbf{C}}\mathbf{v} = \lambda\mathbf{v}, \tag{5.5}$$

λ and $\mathbf{v}$ are the eigenvalues and eigenvectors of the covariance matrix $\overline{\mathbf{C}}$. In the Hilbert feature space $\mathbf{v}$ can be described in the span of the data set $\Phi(\mathbf{X}) = \{\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_n)\}$:

$$\mathbf{v} = \sum_{i=1}^{n} a_i\phi(\mathbf{x}_i) \tag{5.6}$$

Defining $\mathbf{K} = \Phi(\mathbf{X})^T\Phi(\mathbf{X})$, with elements $\kappa_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$, as shown:

$$\begin{aligned}\mathbf{K} &= \begin{bmatrix}\phi(\mathbf{x}_1)^T \\ \phi(\mathbf{x}_2)^T \\ \vdots \\ \phi(\mathbf{x}_n)^T\end{bmatrix}\begin{bmatrix}\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \cdots, \phi(\mathbf{x}_n)\end{bmatrix} \\ &= \begin{bmatrix}\phi(\mathbf{x}_1)^T\phi(\mathbf{x}_1) & \phi(\mathbf{x}_1)^T\phi(\mathbf{x}_2) & \cdots & \phi(\mathbf{x}_1)^T\phi(\mathbf{x}_n) \\ \phi(\mathbf{x}_2)^T\phi(\mathbf{x}_1) & \phi(\mathbf{x}_2)^T\phi(\mathbf{x}_2) & \cdots & \phi(\mathbf{x}_2)^T\phi(\mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_n)^T\phi(\mathbf{x}_1) & \phi(\mathbf{x}_n)^T\phi(\mathbf{x}_2) & \cdots & \phi(\mathbf{x}_n)^T\phi(\mathbf{x}_n)\end{bmatrix} \\ &= \begin{bmatrix}\kappa(\mathbf{x}_1, \mathbf{x}_1) & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_n) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_n, \mathbf{x}_1) & \kappa(\mathbf{x}_n, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_n, \mathbf{x}_n)\end{bmatrix}\end{aligned} \tag{5.7}$$

Substituting equation (5.4) and (5.6) into (5.5), and we arrive at

$$\frac{1}{n}\mathbf{K}^2\mathbf{a} = \lambda\mathbf{K}\mathbf{a} \tag{5.8}$$

Both sides of the equation (5.8) are divided by $\mathbf{K}$ at the same time will arrive at:

$$\frac{1}{n}\mathbf{K}\mathbf{a} = \lambda\mathbf{a} \tag{5.9}$$

$\mathbf{K}$ is $n \times n$ Gram matrix, λ and $\mathbf{a}$ are the eigenvalues and eigenvectors of $\mathbf{K}$, $\mathbf{a}$ is a column vector as $\mathbf{a} = \{a_1, \cdots, a_n\}$, which can be used in calculating the kernel principle component for a new sample, see equation (5.10).

For kernel principal component extraction, we compute projections of the image of a point $\mathbf{x}_t$ onto the Eigenvectors $\mathbf{v}$ in Hilbert space according to

$$(\mathbf{v}, \phi(\mathbf{x}_t)) = \sum_{i=1}^{n} a_i(\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_t)) = \sum_{i=1}^{n} a_i\kappa(\mathbf{x}_i, \mathbf{x}_t) \tag{5.10}$$

Note that neither equation (5.4) nor equation (5.10) requires the $\phi(\mathbf{x}_i)$ in explicit form-they are only needed in dot products. Therefore, we are able to use kernel functions for computing these dot products without actually performing the map ϕ. For choices of kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j)$, three kernels which have successfully been used in applications include:

1. Polynomial kernel: $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$;
2. RBF (Gaussian) kernel: $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$;

3. Sigmoid kernel: $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\alpha \mathbf{x}_i \cdot \mathbf{x}_j + r)$.

It is assumed that the Gram matrix $\mathbf{K}$ is zero-mean, otherwise, it can be centered as [Scholkopf 05]

$$\overline{\mathbf{K}} = \mathbf{K} - \mathbf{I}_n\mathbf{K} - \mathbf{K}\mathbf{I}_n + \mathbf{I}_n\mathbf{K}\mathbf{I}_n \tag{5.11}$$

where $\mathbf{I}_n = \frac{1}{n}\mathbf{I}_{n\times n}$, and $\mathbf{I}_{n\times n}$ is the identity matrix of size $n \times n$.

5.3 Parallel Version of KPCA on GPU

Due to non-linear methods have computational complexities, it's time consuming if they are executed in CPU with serial implementation. Taking KPCA as an example, if there are n samples in the training dataset (normally in order to extract efficiency kernel principle components, n should be very big (e.g. $n \geq 1000$)), then the size of the Gram matrix is $n \times n$ ($D \times D$ for PCA, $D << n$). The space complexity of storing the Gram matrix is $O(n^2)$, while the time complexity (performing Eigen-decomposition on a $n \times n$ Gram matrix, see equation (5.10)) is $O(n^3)$, which is much more complex than PCA with $O(D^3)$.

According to equation (5.10), it can be known that for a new pixel $\mathbf{x}_t$, if we want to get its kernel principle component (a new coordinate) in the Eigenvector basis $\mathbf{v}$ in Hilbert space, kernel function should be used between $\mathbf{x}_t$ and all training samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ as $\sum_{i=1}^{n} a_i \kappa(\mathbf{x}_i, \mathbf{x}_t)$, thus only for one principle component of one pixel. If there are 1000 samples in training samples set, in order to extract kernel principle component for a hyperspectral image with small size 500×500, a Gram matrix $\mathbf{K}$ with size 250000×1000 should be calculated, thus is a large amount of computation.

From the definition of kernel function and Gram matrix $\mathbf{K}$, the calculation of every element $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ in Gram matrix $\mathbf{K}$ is independent. Therefore, the big Gram matrix $\mathbf{K}$ (as with size 250000×1000) can be executed in GPU in parallel. Suppose the number of training samples set in hyperspectral image is n, the size of hyperspectral image is $N \times N$, then the Gram matrix $\mathbf{K}$ is $N^2 \times n$, as shown in Figure 5.4. If GPU processor has m cores, the each core just needs to calculate $(N^2 \times n)/m$ kernels, thus can save much time than CPU.

The effective utilization of both CPUs and GPUs can provide compelling benefits. GPU are specifically optimized for computational and memory-intensive problems, whereas CPU devotes more resources to caching or control flow operations. Therefore, we present a CPU-GPU heterogeneous framework, Details and steps of the parallel implementation and optimization flow chart is shown in Figure 5.5.

To achieve satisfactory parallel performance, the data throughput is critical in the design of GPU-based parallel algorithms, meaning that enough data should be fed into the GPU to take advantage of the available compute power. Due to the shared-memory architecture, the major bottleneck is memory communication between the host and the device; unnecessary data transfer between host and device should be avoided. In other words, most data computation

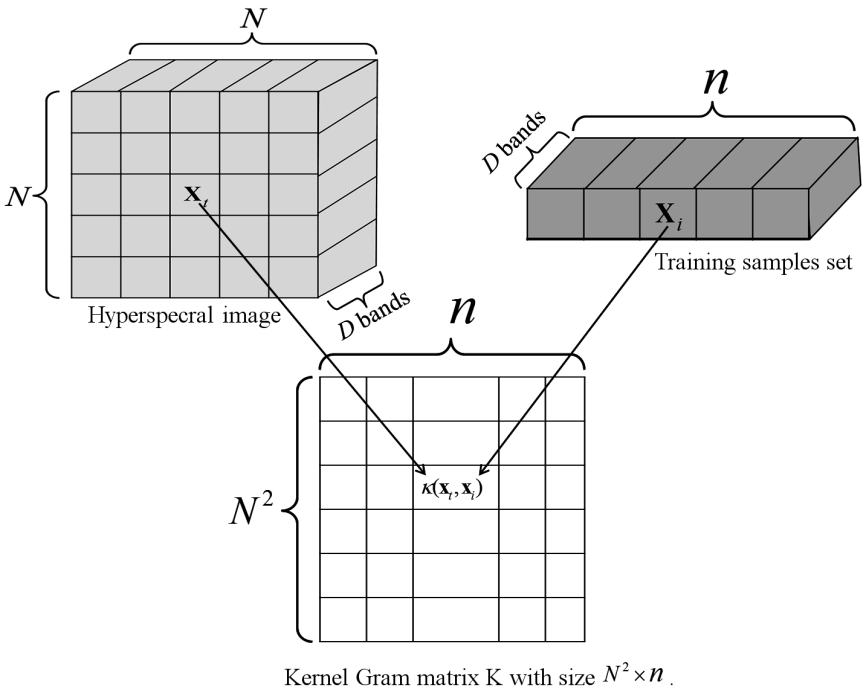


Figure 5.4: The kernel Gram matrix $\mathbf{K}_{N^2 \times n}$ is used for calculating kernel principle components for hyperspectral image, huge $\mathbf{K}_{N^2 \times n}$ can be calculated in parallel in GPU, as every element $\kappa(\mathbf{x}_t, \mathbf{x}_i)$ is independent.

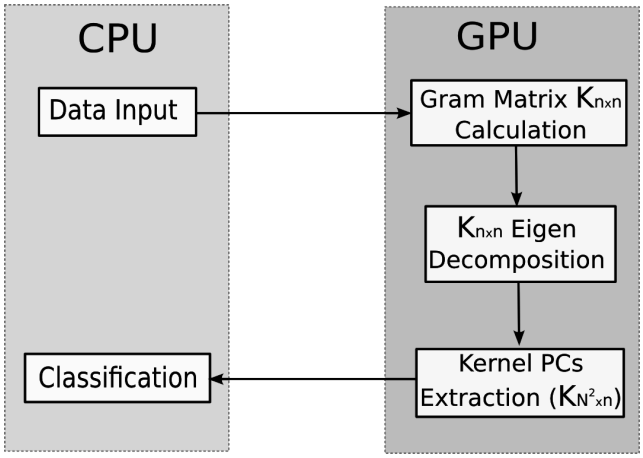


Figure 5.5: CPU-GPU optimization flow chart. In GPU, (1) n training samples are used to construct kernel Gram matrix $\mathbf{K}_{n\times n}$, (2) do Eigen-decomposition for $\mathbf{K}_{n\times n}$ to get the its eigenvectors $\mathbf{a}$, (3) construct construct kernel Gram matrix $\mathbf{K}_{N^2\times n}$ between all pixels in hyperspectral image ($N \times N \times D$, D is number of bands) and training samples set, kernel principle components (PCs)for all pixels can be obtained by combing $\mathbf{a}$ and $\mathbf{K}_{N^2\times n}$, see equation (5.10).

should take place in the GPU without interruption. While data sharing between GPU cores is much easier than in compute clusters, the data-throughput requirement renders current GPUs inappropriate for solving numerous small matrix-operation problems. For the feature extraction of hyperspectral image with KPCA, GPU can be used to accelerate the composite kernel-related computations, and the host CPU can be used to perform other small data computations and most of the control operations. Taking into consideration that the calculation of composite kernels is a dominant part of KPCA that consists of operations on big matrices and high-dimensional vectors, we can pre-compute them and do Eigen-decomposition on the device (GPU), copy and cache the computed results to the host (CPU). The rest of the computations (related with control and small data structures) are computed on the CPU to dramatically reduce the data transfer between the device and the host. In this way, the workload between the GPU and CPU could be well balanced.

5.4 Experiments and Results

The hardware platforms used in our experiment are:
CPU: Intel(R) CoreI7-3630QM 2.4GHZ; memory size: 8GB;
GPU : NVidia's GeForce GT650M that has 384 cores with 2 GB memory;
OS: Windows7 Professional Edition 64;
Development environment: Matlab 2012b, and the parallel algorithms

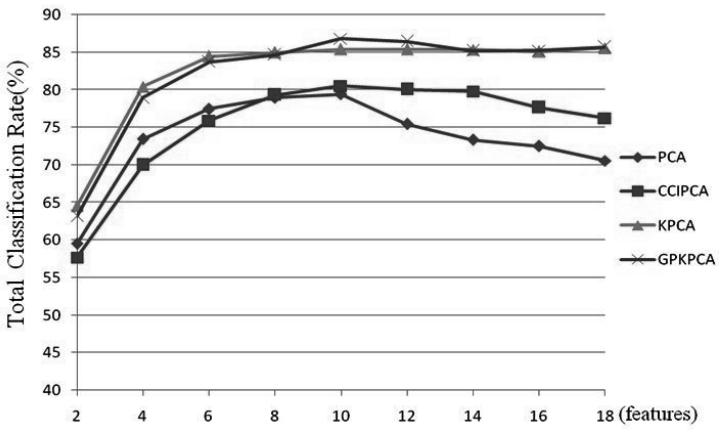


Figure 5.6: Classification accuracies of different methods with increasing number of features on a 50×50 square region.

are implemented in Jacket 2.3 platform developed by AccelerEyes LLC (www.accelereyes.com).

The data used is the AVIRIS Indian Pines image with 220 bands of size 145 lines by 145 samples, and the corresponding spatial resolution is approximately 20 m. From this image, 179 bands are selected by removing water absorption and low signal-to-noise ratio bands. 50×50, 80×80 and 100×100 image sizes are selected from the top left corner of the original image (as the calculation and Eigen-decomposition of Gram matrix need large memory, if whole image 145×145 is used, KPCA method will be out of memory) to compare the execution time and the classification accuracy among PCA, CCIPCA [Weng 03], KPCA and the proposed GPU-based parallel KPCA (GPKPCA). The time measurement is started right after the hyperspectral image file is read to the CPU memory and stopped right after the results of the target/anomaly detection algorithm are obtained and stored in the CPU memory.

For the classifier, we choose support vector machines (SVM) with a linear kernel, the codes used are available in [Chang 01]. 10% samples from each class are chosen randomly from experimental regions as training samples, the remaining labelled samples acts as testing samples.

The classification accuracies for different methods are shown in Figure 5.6. It is first important to emphasize that our parallel version of KPCA (GPKPCA) provides exactly the same results as the serial version of KPCA algorithm. As KPCA runs in central processing units (CPU) with sequential implementation, while GPKPCA runs in graphics processing units (GPU) with parallel implementation, and the calculation resolution and data transmission modes between CPU and GPU are different, the classification accuracies of KPCA and GPKPCA have small difference. What's more, it can be concluded that non-linear

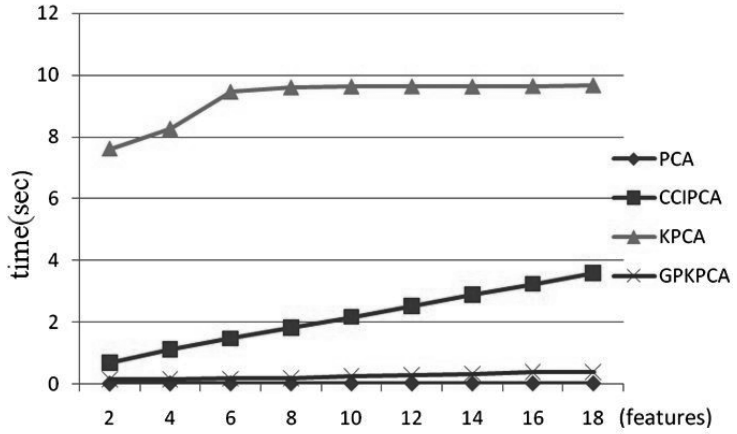


Figure 5.7: The average computing time of extracting different number of features on a 50×50 square region.

feature extraction methods (KPCA and GPKPCA) outperform than linear methods (PCA and CCIPCA), with 5% improvement at least. After the number of extracted features reaches to 10, the classification accuracies of KPCA and GPKPCA almost keep stable, while the results of PCA and CCIPCA decrease with increasing extracted features, one reason for this is that PCA and CCIPCA introduce more noise with more less important features.

The computation time of extracting increasing number of features for 50×50 image size are shown in Figure 5.7. Even using optimized CPU code the KPCA algorithm still requires several seconds for this small region. The GPU however, improves upon this time significantly, consistently achieving speedup of at least 45 times on this 50×50 image size, the overall execution time for the GPU ranges from a few milliseconds up to hundreds milliseconds. It is also clear from Figure 5.7 that the execution time of sequential KPCA are near 10 seconds after 6 features are extracted. The processing time of CCIPCA increase linearly with increasing number of features, nearly 4 seconds corresponding 18 features. As a linear method and with low complexity, PCA performs very fast with only a few milliseconds, even when large number of features are extracted. Although GPKPCA is a non-linear method, it performs faster than linear CCIPCA, its processing time is less than 0.25 second in extracting 18 features.

In order to evaluate the parallel performance of proposed method GPKPCA, we execute GPKPCA and sequential KPCA with different data sizes, as 50×50, 80×80 and 100×100. The speedup times have been shown in Figure 5.8. It is easy to find that GPU is more appropriate for processing huge data sets, the acceleration effect will be much more obvious as larger processed image size. The use of the GPU are very effective in terms of fast feature extraction, when the size of image is 100×100, the GPKPCA produced about 250 times

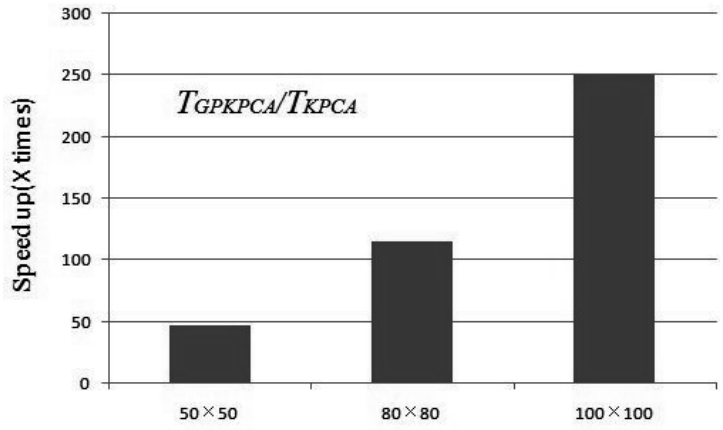


Figure 5.8: The acceleration effect (T_{GPKPCA}/T_{KPCA}) of GPU-based parallel KPCA compared with sequential KPCA with different data sizes, T_{GPKPCA} and T_{KPCA} demote the processing time of GPKPCA and sequential KPCA respectively.

acceleration effect compared to the conventional KPCA in CPU.

5.5 Conclusion

With the ultimate goal of drawing an accelerations of non-linear feature extraction in GPU as high performance computing architectures in the context of remote sensing applications, this chapter described a GPU-based parallel KPCA (GPKPCA) implementation for feature extraction in hyperspectral images. As a parallel implementation, GPKPCA can make full use of powerful GPU architecture to solve complex scientific computing problems. From experimental results on real data set, it can been seen that GPKPCA has obvious acceleration effect, especially for big data. The speedup times can be more than 250 when dealing with bigger dataset.

Although the results obtained with a variety of sizes of hyperspectral image are very encouraging, further experiments should be conducted in order to increase the parallel performance of the proposed algorithms by resolving memory issues and optimizing the parallel design of the algorithms in the GPU-based implementations. Experiments with additional scenes are also highly desirable. Finally, GPU are still far from being exploited in real missions due to power consumption and radiation tolerance issues, the exploration and experiments with GPU devices will be required in order to evaluate the possibility of adapting more parallel algorithms to hardware devices which have been already certified by international agencies and satellite platforms for Earth and planetary observation from space.

The research in this chapter lead to one proceeding as follows:

Luo Renbo, Pi Youguo, "GPU-based parallel kernel PCA feature extraction for hyperspectral images". International Conference on Remote Sensing and Wireless Communications (RSWC 2014). 2014.

6

Conclusions and Future Works

6.1 Conclusions

The latest advances in the remote sensing field have contributed significantly at the broad availability of high quality data or images. Accordingly, the development of efficient and robust algorithms for the analysis of these data is a very important topics. In particular, classification is one of most important tasks, especially for Earth observation. The classification problem, i.e., detecting and identifying the different land-covers that characterize a given geographical area of interest, is a complex process. Among the procedures involved in classification problem, feature extraction and fusion, aiming to extract and analyze all the useful information that different remote sensing data sets contain, is a necessary pre-processed step. As collecting ground-truth is often expensive and time consuming, the number of available training samples is almost always much smaller than the dimensionality of the feature space. This leads to the Hughes phenomenon, i.e. for a limited number of training samples, the classification accuracy decreases as the dimension increases. The present thesis has focused on developing methodologies for feature extraction and fusion of remote sensing data. Specifically the proposed solutions relate to semi-supervised learning and to domain adaptation

Feature extraction and data fusion for classification of remote sensing imagery are challenging topics, and has been intensively investigated for decades. However, the problems have not been solved, particularly due to the difficulty in processing the "Big Remote Sensing Data". Firstly, the high dimensionality of remote sensing data (e.g. hyperspectral images) leads to problems with storage resources and computational load. Secondly, most of the existing techniques focus mainly on performing feature extraction in only the spectral domain, omitting information about spatial structure and correlation. Although many spectral-spatial methods have been proposed in recent years, spectral and spatial information have not been well combined. Finally, different data sources

have different limitations, such as hyperspectral images are easily influenced by cloud and different weather conditions, whereas LiDAR data is difficult to discriminate different objects with similar altitude, fusion of different data sources for better classification is very necessary.

This thesis provided 4 solutions to address the above issues, as discussed below.

6.1.1 Supervised Feature Extraction

In order to extract intersecting features for classification of hyperspectral remote sensing imagery, two supervised feature extraction algorithms, which take into account the label information of samples to infer class separability, were proposed in chapter 2. They both integrated label information into unsupervised feature extraction methods, leading to satisfactory results compared the unsupervised methods in terms of classification accuracy.

By incorporating label information into linear unsupervised neighborhood preserving embedding (NPE), this thesis proposed a Discriminative Supervised Neighborhood Preserving Embedding (DSNPE) method. In this method each data point was represented linearly by its nearest neighboring data points (with the same label). This differed from existing methods which search the whole nearest neighboring data points (including samples with different labels) in NPE [He 05]. Based on the representation introduced above, a correlation matrix (useful for calculating transformation matrix) between the samples was obtained. Furthermore, we defined a new transformation criterion whose aim is to pull the neighboring points with the same class label close to each other, while simultaneously pushing the neighboring points with different labels far away from each other after dimensionality reduction. The results validated on real images demonstrated the effectiveness of the proposed DSNPE algorithm, compared to representative dimensionality reduction algorithms.

By combining principle component analysis (PCA), label information and locality preserving projections (LPP), a PCA-based supervised locality preserving projection (PSLPP) was proposed. Unlike the unsupervised learning scheme LPP, PSLPP used both label and local manifold information to model the similarity of the data and enhance the discriminant power of the data when mapping them into a low-dimensional space. Specifically, the original high-dimensional data was first processed by PCA to remove the noisy and redundancy, then local geometrical structure and label information were used to model the similarity of data points. Experiments on a number of data sets demonstrated that including label information can improve the feature extraction performance.

6.1.2 Semi-supervised Feature Extraction

Semi-supervised feature extraction methods have aroused a great deal of interest in the machine learning community recently, since that manually labelling data sets is time consuming and fairly expensive, while unlabelled samples

could be available in large quantities at very low cost. As a result, chapter 3 focused on exploring semi-supervised feature extraction methods for classification of HS images, and proposed three new semi-supervised methods for feature extraction of hyperspectral remote sensing imagery.

Firstly, we improved the semi-supervised local discriminant analysis (SELD) method proposed in [Liao 13] for feature extraction of hyperspectral image. The proposed improved SELD (ISELD) method aimed to find a projection which can preserve local neighborhood information and maximize the class discrimination of the data. The proposed ISELD included correlations between labelled and unlabelled samples in the within- and between-class scatter matrices which would be used to find optimal projections. This lead to better class discrimination and better local manifold structure preservation than SELD.

Graph-based feature extraction methods rely upon the construction of a graph representation, where the vertices are labelled and unlabelled samples and the edges represent the similarity among samples in the dataset. Chapter 4 built a semi-supervised graph which can better describe the similarities between samples, especially between labelled and unlabelled samples. In our semi-supervised graph, the training samples were divided into labelled and unlabelled sets first, and the labelled samples were connected according to their label information and unlabelled samples were connected by their nearest neighborhood information. By sorting the mean distance between an unlabelled sample and center of each class, we connected the unlabelled sample with all labelled samples belonging to its nearest neighborhood class. Last but not least, the proposed method set weighted edges to connected samples by utilizing distance information between samples, thus better modelled the actual differences and similarities between samples.

Finally, semi-supervised graph learning and multiple feature fusion were coupled in a unified framework for remote sensing classification. In this method, spectral, spatial and label information have been taken into account to construct the semi-supervised fusion graph. For graph construction, samples were connected according to not only their label information, but also their spectral-spatial nearest neighborhood information, so the connected samples were not only very near to each other, but also belong to the same class. By exploiting the fused semi-supervised graph, two transformation matrices were obtained to project high-dimensional hyperspectral image and morphological features to their lower dimensional subspaces. The final classification map is obtained by concentrating the lower-dimensional features together as an input of classifier.

Comparing with some related feature extraction methods on real remote sensing data sets, our proposed semi-supervised methods have better performance and classification accuracies. This is due to combining a small number of labelled samples with a large number of unlabelled samples.

6.1.3 Fusion of Hyperspectral Image and LiDAR Data

As different modalities, such as hyperspectral image and LiDAR data, have different advantages and disadvantages, fusing these data allows more reliable

classification attracts increasing interests but remains challenging. Therefore the topic of fusing data sets from different sensors attracts increasing interest but remains challenging. Chapter 5 proposed a novel framework to fuse hyperspectral and LiDAR data to classify remote sensing scenes mixed with cloud shadow. The proposed method performed classification separately on the cloud-shadow and non-shadow areas. Firstly, the method modelled spatial (from HS image) and elevation (from LiDAR data) information by exploiting attribute profiles, and extracted a cloud-shadow mask by thresholding the attribute profiles of hyperspectral images. This way a remote sensing scene was divided into cloud-shadow and shadow-free regions. The classification result was much more reliable when using single elevation features than when using spectral and/or spatial features alone. Assuming that different feature sources share similar intra-cluster distance relations, new training samples set for cloud-shadow region was generated by searching nearest neighbors of each class center based on both spectral and spatial information. The final classification map was produced by decision fusion of both cloud-shadow and non-shadow maps. This way our proposed fusion method combines the advantages of both hyperspectral and LiDAR data. Experimental results on fusion of hyperspectral and LiDAR data for classification of remote sensing scene mixed with cloud shadow showed the efficiency of the proposed framework, with about 4% and 10% improvements over conventional methods in the shadow-free and cloud-shadow regions, respectively.

6.1.4 GPU-based Non-linear Feature Extraction

Non-linear feature extraction methods, such as kernel principle component analysis (KPCA), is more suitable to describe non-linear, higher-order and complex distributions. However, the relatively large computational complexity of non-linear feature extraction methods make it time consuming to process huge data sets, as hyperspectral images. Therefore, an efficient implementation of non-linear feature extraction algorithm (KPCA) on GPU based on Jacket's MATLAB Toolbox in a parallel strategy was developed in chapter 5. Experiments on hyperspectral data showed that the GPU based parallel KPCA approach was much faster, without sacrificing accuracy. The acceleration effect was more obvious with the increasing size of data set.

6.2 Future Research

Following the work derived from the present thesis, even though the contributions have achieved a certain level of achievements, several challenges still exist and can be potentially improved. As with any new work, there are many open avenues for future research that deserve attention and which will be explored in our future developments. In the following, we list the most relevant of these perspectives for future work:

1. It would be interesting to integrate active learning into our proposed SEGL (in chapter 3) method for unlabelled data sampling (e.g., searching nearest neighborhood samples within a fixed radius, as well as investigating the choice of radius in unlabelled samples selection). What is more, in order to better cope with the problems of multi-modality, we will explore more criterions to connect unlabelled samples with labelled samples instead of only using the mean distance.

2. As indicated by the promising results from preliminary assessments, the performance of the co-training-based framework (in chapter 4) is still under comprehensive investigation. It is foreseeable that the combination of active learning, sparse representation and new criterion such as spectral angle when selecting co-training samples from cloud shadow areas will further advance the classification result.

3. In order to better classify and identify land-cover, future work may be directed towards the fusion of different complementary data sources, such as hyperspectral image, SAR and LiDAR data, and inclusion additional types of features such as texture, border-oriented features, spatial-contextual features, elevation features, and so on.

4. Although the results obtained with big data sets on GPU are very encouraging, GPU is still far from being exploited in real missions due to the power consumption and storage issues. GPU devices certified by international agencies and satellite platforms for Earth and planetary observation from space differ from standard GPUs in the following ways: a) tons of data to be processed; b) limited room and power on satellite platforms, c) long time and uninterrupted operation. Porting the algorithms on such a hardware needs to be investigated. Optimally addressing memory restrictions about GPU is another research topic.

5. Last but not least, we can further improve and adapt the developed techniques for a wide variety of real project applications such as urban management, weather analysis, land-cover mapping and modelling, species identification in forested areas, and so on.

Bibliography

[Agathos 14] A. Agathos, J. L i, D. Petcu & A. Plaza. *Multi-GPU implementation of the minimum volume simplex analysis algorithm for hyperspectral unmixing.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 7, no. 6, pages 2281–2296, Jun. 2014.

[Arunasakthi 14] K. Arunasakthi & L. KamatchiPriya. *A review on linear and non-linear dimensionality reduction techniques.* Machine Learning and Applications: An International Journal, vol. 1, no. 1, pages 65–76, Sep. 2014.

[Arvidson 01] T. Arvidson, J. Gasch & S. N. Goward. *Landsat-7's long-term acquisition plan-An innovative approach to building a global imagery archive.* Remote Sens. Environ., vol. 78, no. 1-2, pages 13–26, Oct. 2001.

[Bandos 06] T. Bandos, D. Zhou & G. Camps-Valls. *Semi-supervised hyperspectral image classification with graphs.* International Geoscience and Remote Sensing Symposium (IGARSS), 2006.

[Bandos 09] T. V. Bandos, L. Bruzzone & G. Camps-Valls. *Classification of hyperspectral images with regularized linear discriminant analysis.* IEEE Trans. Geosci. Remote Sens., vol. 47, no. 3, pages 862–873, Mar. 2009.

[Bao 16] R. Bao, J. Xia, M. Dalla Mura, P. Du, J. Chanussot & J. Ren. *Combining morphological attribute profiles via an ensemble method for hyperspectral image classification.* IEEE Geosci. Remote Sens. Lett., vol. 13, no. 3, pages 359–263, Mar. 2016.

[Baudat 00] G. Baudat & F. Anouar. *Generalized discriminant analysis using a kernel approach.* Neural Comput., vol. 12, pages 2385–2404, 2000.

[Belkin 02] M. Belkin & P. Niyogi. *Laplacian eigenmaps and spectral techniques for embedding and clustering.* Advances in Neural Information Processing Systems 14, MIT Press, British Columbia, pages 585–591, 2002.

[Belkin 03] M. Belkin & P. Niyogi. *Laplacian eigenmaps for dimensionality reduction and data representation.* Neural Computation, vol. 15, no. 6, pages 1373–1396, 2003.

[Bellens 08] R. Bellens, S. Gautama, L. Martinez-Fonte, W. Philips, J.C.W. Chan & F. Canters. *Improved classification of VHR images of urban areas using directional morphological profiles.* IEEE Trans. Geosci. Remote Sens., vol. 46, no. 10, pages 2803–2813, Oct. 2008.

[Benediktsson 05] J. A. Benediktsson, J. A. Palmason & J. R. Sveinsson. *Classification of hyperspectral data from urban areas based on extended morphological profiles.* IEEE Trans. Geosci. Remote Sens., vol. 40, no. 3, pages 480–491, Mar. 2005.

[Bennet 99] K. Bennet & A. Demiriz. *Semi-supervised support vector machines.* Cambridge, 1999. in Advances in Neural Information Processing Systems (NIPS).

[Bernabe 13] S. Bernabe, S. Lopez, A. Plaza & R. Sarmiento. *GPU implementation of an automatic target detection and classification algorithm for hyperspectral image analysis.* IEEE Geosci. Remote Sens. Lett., vol. 10, no. 2, pages 221–225, Mar. 2013.

[Bioucas-Dias 13] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi & J. Chanussot. *Hyperspectral remote sensing data analysis and future challenges.* IEEE Geosci. Remote Sens. Mag., vol. 1, no. 2, pages 6–36, Jun 2013.

[Bruce 13] D. Bruce, A. Lawrence, F. Ross, M. Elizabeth, C. Douglas, T. Joel, G. Jeffrey, J. Kenneth, Vuong Ly & M. Paul. *NASA Goddard's LiDAR, Hyperspectral and Thermal (G-LiHT) Airborne Imager.* Remote Sens., vol. 5, no. 8, pages 4045–4066, Aug 2013.

[Bruzzone 06] L. Bruzzone, M. Chi & M. Marconcini. *A novel transductive SVM for semisupervised classification of remote-sensing images.* IEEE Trans. Geosci. Remote Sens., vol. 44, no. 11, pages 3363–3373, Nov. 2006.

[Bruzzone 09] L. Bruzzone & C. Persello. *A Novel Approach to the Selection of Spatially Invariant Features for the Classification of Hyperspectral Images With Improved Generalization Capability.* IEEE Trans. Geosci. Remote Sens., vol. 47, no. 9, pages 3180–3191, Sep. 2009.

[Cai 07] D. Cai, X. He & J. Han. *Semi-supervised discriminant analysis.* IEEE 11th International Conference on Computer Vision (ICCV07), 2007.

[Campbell 02] J. B. Campbell. Introduction to remote sensing. The Guilford Press, 3rd edition, 2002.

[Camps-Valls 05] G. Camps-Valls & L. Bruzzone. *Kernel-based methods for hyperspectral image classification.* IEEE Trans. Geosci. and Remote Sens., vol. 43, no. 6, pages 1351–1362, Jun. 2005.

[Camps-Valls 06] G. Camps-Valls & L. Bruzzone. *Kernel-based methods for hyperspectral image classification.* IEEE Trans. Geosci. Remote Sens., vol. 43, no. 6, pages 1351–1362, Jun. 2006.

[Camps-Valls 07] G. Camps-Valls, T. Bandos & D. Zhou. *Semisupervised graph-based hyperspectral image classification.* IEEE Trans. Geosci. Remote Sens., vol. 45, no. 10, pages 3044–3054, Oct. 2007.

[Capobianco 09] L. Capobianco, A. Garzelli & G. Camps-Valls. *Target detection with semisupervised kernel orthogonal subspace projection.* IEEE Trans. Geosci. Remote Sens., vol. 47, no. 11, pages 93–97, Nov. 2009.

[Chang] C. Chang & C. Lin. *A Library for Support Vector Machines.* [Online], Available: http://www.csie.ntu.edu.tw/cjlin/libsvm.

[Chang 01] C.C. Chang & C.J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001.

[Chang 03] C.l Chang. Hyperspectral imaging: Techniques for spectral detection and classification, volume 1. Springer Science & Business Media, Jul. 2003.

[Chapelle 05] O. Chapelle & A. Zien. *Semi-supervised classification by low density separation.* pages 57–64. In International Conference on Artificial Intelligence and Statistics (AISTATS), 2005.

[Che 08] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer & K. Skadron. *performance study of general-purpose applications on graphics processors using CUDA*. J. Parallel and Distributed Comput., vol. 68, no. 10, pages 1370–1380, Oct. 2008.

[Chen 11] S. Chen & D. Zhang. *Semi-supervised dimensionality reduction with pairwise constraints for hyperspectral image classification.* IEEE Geosci. Remote Sens. Lett., vol. 8, no. 2, pages 369–373, 2011.

[Chen 14] C. Chen, W. Li, H. Su & K. Liu. *Spectral-spatial classification of hyperspectral image based on kernel extreme learning machine.* Remote Sens., vol. 6, no. 6, pages 5798–5814, Jun. 2014.

[Chouzenoux 14] E. Chouzenoux, M. Legendre, S. Moussaoui & J. Idier. *Fast constrained least squares spectral unmixing using primaldual interior point optimization.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 7, no. 1, pages 59–69, Jan. 2014.

[Christophe 11] E. Christophe, J. Michel & J. Inglada. *Remote sensing processing: From multicore to GPU.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 4, no. 3, pages 643–652, Sep. 2011.

[Coomans 82] D. Coomans & D.L. Massart. *Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules.* Analytica Chimica Acta, vol. 136, pages 15–27, 1982.

[Dalponte 08] M. Dalponte, L. Bruzzone & D. Gianelle. *Fusion of hyperspectral and LIDAR remote sensing data for classification of complex forest areas.* IEEE Trans. Geosci. Remote Sens., vol. 46, no. 5, pages 1416–1427, May 2008.

[Dalponte 10] M. Dalponte. *Analysis of forest areas by advanced remote sensing systems based on hyperspectral and LiDAR data.* PhD thesis, University of Trento, Mar. 2010.

[Debes 14] C. Debes. *Hyperspectral and LiDAR data fusion: outcome of the 2013 GRSS Data Fusion Contest.* IEEE J. Sel.Topics Appl. Earth Observ. Remote Sens., vol. 7, no. 6, pages 2405–2418, Jun. 2014.

[Dempster 77] N. Dempster, A. Laird & D. Rubin. *Maximum likelihood from incomplete data via the EM algorithm.* Journal of the royal statistical society, Series B, vol. 39, no. 1, pages 1–38, 1977.

[Du 13] Q. Du, W. Wei, B. Ma & N. H. Younan. *Hyperspectral image compression and target detection using nonlinear principal component analysis.* Processing IX. SPIE, Processings of Satell. Data Compression Commun, Sep. 2013.

[Eismann 96] M. T. Eismann, C. R. Schwartz, J. N. Cederquist, J. A. Hackwell & R. J. Huppi. *Comparison of infrared imaging*

hyperspectral sensors for military target detection applications. volume 2819, pages 91–101. Proceedings of the SPIE, 1996.

[Elakshe 08] A. F. Elakshe. *Fusion of hyperspectral images and lidar-based DEMs for coastal mapping.* Optics Lasers Eng., vol. 46, pages 493–498, Jul. 2008.

[Fang 14] Y. Fang, H. Li, Y. Ma, K. Liang, Y. Hu, S. Zhang & H. Wang. *Dimensionality reduction of hyperspectral images based on robust spatial information using locally linear embedding.* IEEE Geosci. Remote Sens. Lett., vol. 11, no. 10, pages 1712–1716, Oct. 2014.

[Fauvel 08] M. Fauvel, J. A. Benediktsson, J. Chanussot & J. R. Sveinsson. *Spectral and Spatial Classification of Hyperspectral Data Using SVMs and Morphological Profile.* IEEE Trans. Geosci. Remote Sens., vol. 46, 2008.

[Fauvel 09] M. Fauvel, J. Chanussot & J. A. Benediktsson. *Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas.* J. EURASIP Journal on Advances in Signal Processing, pages 1–14, Mar. 2009.

[Fong 07] M. Fong. *Dimension reduction on hyperspectral images.* University of California, Los Angeles, United States, Report, August 2007.

[Frantz 15] D. Frantz, A. Roder, T. Udelhoven & M. Schmidt. *Enhancing the detectability of clouds and their shadows in multitemporal dryland landsat imagery: extending fmask.* IEEE Geosci. Remote Sens. Lett., vol. 12, no. 6, pages 1242–1246, Jun. 2015.

[Fukunaga 83] K. Fukunaga & J. Mantock. *Nonparametric discriminant analysis.* IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-5, no. 6, pages 671–678, Nov. 1983.

[Fukunaga 90] K. Fukunaga. Introduction to statistical pattern recognition. 2nd. Boston, MA: Academic, 1990.

[Ghamisi 14a] P. Ghamisi, J. Benediktsson & M. Ulfarsson. *Spectral-spatial classification of hyperspectral images based on hidden Markov random fields.* IEEE Trans. Geosci. and Remote Sens., vol. 52, no. 5, pages 2565–2574, May 2014.

[Ghamisi 14b] P. Ghamisi, J. A. Benediktsson & J. R. Sveinsson. *Automatic spectral-spatial classification framework based on*

attribute profiles and supervised feature extraction. IEEE Trans. Geosci. Remote Sens., vol. 52, no. 9, pages 5771–5782, 2014.

[Ghamisi 16] P. Ghamisi, B. Hofle & X. Zhu. *Hyperspectral and LiDAR data fusion using extinction profiles and deep convolutional neural network.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. pp, no. 99, pages 1–14, 2016.

[Green 98] R. 0. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J. Chovit, M. Solis & M. R. Olah. *Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS).* Remote Sensing of Environment, vol. 65, no. 3, pages 227–270, 1998.

[Gu 15] Y. Gu, Q. Wang, X. Jia & J. A. Benediktsson. *A Novel MKL Model of Integrating LiDAR Data and MSI for Urban Area Classification.* IEEE Trans. Geosci. Remote Sens., vol. 53, no. 4, pages 5312–5326, Oct. 2015.

[Harish 07] P. Harish & P. J. Narayanan. *Accelerating large graph algorithms on the GPU using CUDA.* ser. Lecture Notes in Computer Science, Berlin/Heidelberg, 2007. Springer, Proceedings of High Performance Computing-HiPC.

[He 04] X. He & P. Niyogi. *Locality preserving projections.* Advances in Neural Information Processing System 16, MIT Press, British Columbia,, 2004.

[He 05] X. He, D. Cai, S. Yan & H. Zhang. *Neighborhood preserving embedding.* In Tenth IEEE International Conference on Computer Vision 2005, vol. 2, pages 1208–1213, 2005.

[Heesung 05] K. Heesung & N.M. Nasrabadi. *Kernel rx-algorithm: a nonlinear anomaly detector for hyperspectral imagery.* IEEE Trans. Geosci. Remote Sens., vol. 43, no. 2, pages 388–397, Feb. 2005.

[Ho 95] T. K. Ho. *Random Decision Forests.* Montreal, QC, 1995. Proceedings of the 3rd International Conference on Document Analysis and Recognition.

[Hotelling 33] H. Hotelling. *Analysis of a complex of statistical variables into principal components.* Journal of Educational Psychology,, vol. 24, pages 417–441, 1933.

[Huang 10] H. Huang & B. Kuo. *Double nearest proportion feature extraction for hyperspectral-image classification.* IEEE

Trans. Geosci. Remote Sens., vol. 48, no. 11, pages 4034–4046, 2010.

[Huang 13] X. Huang & L. Zhang. *SVM ensemble approach combining spectral, structural, semantic features for the classification of high-resolution remotely sensed imagery.* IEEE Trans. Geosci. Remote Sens., vol. 51, no. 1, pages 257–272, 2013.

[Hughes 68] G. F. Hughes. *On the mean accuracy of statistical pattern recognizers.* IEEE Transactions on Information Theory, vol. 14, no. 1, pages 55–63, 1968.

[Hyp 13] *2013 IEEE GRSS Data Fusion Contest.* http://www.grssieee.org/community/technical-committees/data-fusion/, 2013.

[Hyvarinen 00] A. Hyvarinen & E. Oja. *Independent component analysis: algorithms and applications.* Neural Networks, vol. 13, pages 411–430, 2000.

[Jia 13] X. Jia, B. Kuo & M. M. Crawford. *Feature Mining for Hyperspectral Image Classification.* Proceedings of the IEEE, vol. 101, no. 3, pages 676 – 697, Feb. 2013.

[Jlliffe 86] I. Jlliffe. *Principal Component Analysis.* New York: Springer-Verlag, 1986.

[Joachims 99] T. Joachims. Making large-scale support vector machine learning practical. MIT Press, 1999.

[Jung 14] J. Jung, E. Pasolli, S. Prasad, J. Tilton & M. Crawford. *A framework for land cover classification using discrete return LiDAR data: Adopting pseudo-waveform and hierarchical segmentation.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 7, no. 2, pages 491–502, Feb. 2014.

[Kelman 13] T. Kelman, J. Ren & S. Marshall. *Effective classification of Chinese tea samples in hyperspectral imaging.* Artificial Intelligence Research, vol. 2, no. 4, pages 87–96, Oct. 2013.

[Khodadadzadeh 15] M. Khodadadzadeh, J. Li, S. Prasad & A. Plaza. *Fusion of hyperspectral and LiDAR remote sensing data using multiple feature learning.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 8, no. 6, pages 2971–2983, Jun, 2015.

[Kirk 10] D. Kirk & W.-M. W. Hwu. Programming massively parallel processors: A hands-on approach. New York: Elsevier Science & Technology, 2010.

[kre 15] kret.com. *The history of radar, from aircraft radio detectors to airborne radar*, Apr. 2015.

[Kuo 04] B. Kuo & D. Landgrebe. *Nonparametric weighted feature extraction for classification.* IEEE Trans. Geosci. Remote Sens., vol. 42, no. 5, pages 1096–1105, 2004.

[Kuo 07] B. Kuo & K. Chang. *Feature extractions for small sample size classification problem.* IEEE Trans. Geosci. Remote Sens., vol. 45, no. 3, pages 756–764, Mar. 2007.

[Kuo 09] B. C. Kuo, C. H. Li & J. M. Yang. *Kernel nonparametric weighted feature extraction for hyperspectral image classification.* IEEE Trans. Geosci. Remote Sens., vol. 47, no. 4, pages 1139 – 1155, Apr. 2009.

[Landgrebe 03] D. A. Landgrebe. Signal theory methods in multispectral remote sensing. Hoboken, NJ: Wiley, 2003.

[Lee 11] C. A. Lee, S. D. Gasster & A. Plaza. *Recent developments in high performance computing for remote sensing: A review.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 4, no. 3, pages 508–527, Sep. 2011.

[Liao 12] W. Liao, R. Bellens, A. Pizurica, W. Philips & Y. Pi. *Classification of hyperspectral data over urban areas using directional morphological profiles and semi-supervised feature extraction.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 5, no. 4, pages 1177–1190, Aug. 2012.

[Liao 13] W. Liao, A. Pizurica, W. Philips & Y. Pi. *Semisupervised Local Discriminant Analysis for Feature Extraction in Hyperspectral Images.* IEEE Trans. Geosci. Remote Sens., vol. 51, no. 1, 2013.

[Liao 14] W. Liao, R. Bellens, A. Pizurica, S. Gautama & W. Philips. *Combining feature fusion and decision fusion for classification of hyperspectral and LiDAR data.* pages 1241–1244, Quebec City, Jul. 2014. Proceedings of International Geoscience and Remote Sensing Symposium (IGARSS).

[Liao 15] W. Liao, A. Pizurica, R. Bellens, S. Gautama & W. Philips. *Generalized graph-based fusion of hyperspectral and LiDAR data using morphological features.* IEEE

Geosci. Remote Sens. Lett., vol. 12, no. 3, pages 552–556, Mar. 2015.

[Liao 16] W. Liao, M. Dalla Mura, J. Chanussot & W. Philips R. Bellens. *Morphological Attribute Profiles With Partial Reconstruction.* IEEE Trans. Geosci. Remote Sens., vol. 54, no. 3, pages 1338–1756, 2016.

[LiD 13] *LIDAR-Light Detection and Ranging-is a remote sensing method used to examine the surface of the Earth. Available at::http://www.webcitation.org/6H82i1Gfx*, 2013.

[Liu 07] Z. Liu, J. Yan, D. Zhang & Q. L. Li. *Automated tongue segmentation in hyperspectral images for medicine.* Applied Optics, vol. 46, no. 34, pages 8328–8334, Dec. 2007.

[Lu 14] G. Lu & B. Fei. *Medical hyperspectral imaging: a review.* Journal of Biomedical Optics, vol. 19, no. 1, pages 010 901/1–010 901/23, Jan. 2014.

[Luo 08] Y. Luo, A. P. Trishchenko & K. V. Khlopenkov. *Developing clear-sky, cloud and cloud shadow mask for producing clear-sky composites at 250-meter spatial resolution for the seven MODIS land bands over Canada and North America.* Remote Sens. Environ., vol. 112, no. 12, pages 4167–4185, Dec. 2008.

[Luo 15] R. Luo, W. Liao, W. Philips & Y. Pi. *An improved semi-supervised local discriminant analysis for feature extraction of hyperspectal image.* pages 1–4. Joint Urban Remote Sensing Event (JURSE 2015), 2015.

[Luo 16a] R. Luo, W. Liao, H. Huang, W. Philips & Y. Pi. *Spectral-Spatial Classification of Hyperspectral Images with Semi-Supervised Graph Learning.* volume 10004, pages 1–6, Edinburgh, United Kingdom, Sep. 2016. SPIE.

[Luo 16b] R. Luo, W. Liao, X. Huang, W. Philips & Y. Pi. *Classification of cloudy hyperspectral image and LIDAR data based on feature fusion and decision fusion.* In 2016 IEEE Geoscience and Remote Sensing International Symposium (IGARSS 2016), Beijing, China, Jul. 2016. accepted.

[Luo 16c] R. Luo, W. Liao, X. Huang, W. Philips & Y. Pi. *Feature Extraction of Hyperspectral Images With Semisupervised Graph Learning.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., 2016.

[Marchesi 09] S. Marchesi & L. Bruzzone. *ICA and kernel ICA for change detection in multispectral remote sensing images.* In Proc. Int. Geosci. Remote Sens. Symp., pages 980–983. IEEE, 2009.

[Meer 12] F. V. D. Meer, H. V. D. Werff, F. V. Ruitenbeek, C. A. Hecker, W. H. Bakker, M. F. Noomen, M. V. D. Meijde, E. J. M. Carranza, J. B. D. Smeth & T. Woldai. *Multi- and hyperspectral geologic remote sensing: A review.* International Journal of Applied Earth Observation and Geoinformation, vol. 14, no. 1, pages 112–128, 2012.

[Michiel 01] Hazewinkel Michiel. *Gram matrix*, 2001.

[Montopoli 07] M. Montopoli, P. Tognolatti, F. Marzano, M. Pierdicca & G. Perrotta. *Remote sensing of the Moon sub-surface from a spaceborne microwawe radiometer aboard the European Student Moon Orbiter (ESMO).* pages 4451–4454. IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS), Jul. 2007.

[Mura 10a] M. Dalla Mura, J. Benediktsson, B. Waske & L. Bruzzone. *Morphological attribute profiles for the analysis of very high resolution images.* IEEE Trans. Geosci. Remote Sens., vol. 48, no. 10, pages 3747–3762, 2010.

[Mura 10b] M. Dalla Mura, J. A. Benediktsson, B. Waske & L. Bruzzone. *Extended profiles with morphological attribute filters for the analysis of hyperspectral data.* Int. J. Remote Sens., vol. 31, no. 22, pages 5975–5991, Nov. 2010.

[Mura 11] M. Dalla Mura, A. Villa, J. A. Benediktsson & L. Bruzzone J. Chanussot. *Classification of Hyperspectral Images by Using Extended Morphological Attribute Profiles and Independent Component Analysis.* IEEE Geosci. Remote Sens. Lett., vol. 8, no. 3, pages 541–545, 2011.

[Naidooa 12] L. Naidooa, M. Choa, R. Mathieua & G. Asner. *Classification of savanna tree species, in the greater kruger national park region, by integrating hyperspectral and lidar data in a random forest data mining environment.* ISPRS J. Photogramm. Remote Sens., vol. 69, pages 167–179, Apr. 2012.

[NAS] NASA. *NASA Earth Observatory: Remote Sensing. [Online] Available at: http://earthobservatory.nasa.gov/Features/RemoteSensing/.*

[Nigam 06] K. Nigam, A. McCallum, & T. Mitchell. Semi-supervised text classification using em, chapitre 3, pages 31–51. Cambridge, MIT Press, 2006.

[Olivier 06] C. Olivier, S. Bernhard & Z. Alexander. Semi-supervised learning. Cambridge, MIT Press, 2006.

[Ouzounis 07] G. K. Ouzounis & M. H. F. Wilkinson. *Mask-based second generation connectivity and attribute filters.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pages 990–1004, Jun. 2007.

[Palmason 05] J.A. Palmason, J.A. Benediktsson, J.R. Sveinsson & J. Chanussot. *Classification of hyperspectral data from urban areas using morphological preprocessing and independent component analysis.* In Proc. Int. Geosci. Remote Sens. Symp., volume 1, pages 176–179. IEEE, 2005.

[Pedergnana 12] M. Pedergnana, P. Reddy Marpu, M. Dalla Mura, J. A. Benediktsson & L. Bruzzone. *Classification of Remote Sensing Optical and LiDAR Data Using Extended Attribute Profiles.* IEEE Journals on Selected Topics in Signal Processing, vol. 6, no. 7, pages 856–865, Nov. 2012.

[Pesaresi 01] M. Pesaresi & J. A. Benediktsson. *A new approach for the morphological segmentation of high-resolution satellite imagery.* IEEE Trans. Geosci. Remote Sens., vol. 39, no. 2, pages 309–320, 2001.

[Plaza 05] A. Plaza, P. Martinez, J. Plaza & 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, pages 466–479, 2005.

[Plaza 11] A. Plaza, Q. Du, Y. Chang & R. King. *High performance computing for hyperspectral remote sensing.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 4, no. 3, pages 528–544, Sep. 2011.

[Qiao 15] T. Qiao, J. Ren, C. Craigie, J. Zabalza, C. Maltin & S. Marshall. *Quantitative prediction of beef quality using visible and NIR spectroscopy with large data samples under industry conditions.* Journal of Applied Spectroscopy, vol. 82, no. 1, pages 137–144, Jan. 2015.

[Qu 13] H. Qu, J. Zhang, Z. Lin & H. Chen. *Parallel acceleration of SAM algorithm and performance analysis.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 6, no. 3, pages 1172–1178, Jun. 2013.

[Raudys 91] S. J. Raudys & A. K. Jain. *Small sample size effects in statistical pattern recognition: Recommendations for practitioners.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 13, no. 3, pages 152–164, 1991.

[Richards 06] J. A. Richards & X. Jia. *Remote Sensing Digital Image Analysis.* Springer, 2006.

[Roush 97] T. L. Roush. *Mars: Remote sensing.* Encyclopedia of Planetary Science. Springer, pages 459–461, 1997.

[Roweis 00] S. T. Roweis & L. K. Saul. *Nonlinear dimensionality reduction by locally linear embedding.* Science, vol. 290, no. 5500, pages 2323–2326, Dec. 2000.

[Salembier 09] P. Salembier & M. H. F. Wilkinson. *Connected operators.* IEEE Signal Process. Mag., vol. 26, no. 6, pages 136–157, Nov. 2009.

[Satish 09] N. Satish, M. Harris & M. Garland. *Designing efficient sorting algorithms for many core GPUs.* pages 1–10, Rome, Italy, May 2009. Processing of IEEE Int. Symp. Parallel and Distributed Processing.

[Scholkopf 98] B. Scholkopf, A.J. Smola & K.R. Muller. *Nonlinear component analysis as a kernel eigenvalue problem.* J. Neural Computation, vol. 10, no. 5, pages 1299–1319, Jan. 1998.

[Scholkopf 05] B. Scholkopf, A. Smola & K. Muller. Kernel principal component analysis, volume 1327 of *Artificial Neural Networks-ICANN'97.* Lecture Notes in Computer Science, Jun. 2005.

[Schott 07] J. R. Schott. Remote sensing: the image chain approach. Oxford University Press. p. 1., 2nd edition, 2007.

[Schowengerdt 07] R. A. Schowengerdt. Remote sensing: models and methods for image processing. Academic Press. p. 2., 3rd edition, 2007.

[Serpico 07] S.B. Serpico & G. Moser. *Extractionction of spectral channels from hyperspectral images for classification purposes.* IEEE Trans. Geosci. Remote Sens., vol. 45, no. 2, pages 484–495, Feb. 2007.

[Shimoni 11] M. Shimoni, G. Tolt, C. Perneel & J. Ahlberg. *Detection of vehicles in shadow areas.* pages 1–4. Proceedings of 3rd Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens. (WHISPERS), 2011.

[Soille 03] P. Soille. Morphological image analysis: Principles and applications. 2ed. Springer-Verlag New York, 2003.

[Sugiyama 07] M. Sugiyama. *Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis.* Journal of Machine Learning Research, vol. 8, pages 1027–1061, Mar. 2007.

[Sugiyama 10] M. Sugiyama, T. Ide, S. Nakajima & J. Sese. *Semi-supervised local Fisher discriminant analysis for dimensionality reduction.* Machine Learning, vol. 78, no. 35, pages 35–61, Jan. 2010.

[Sun 10] D. Sun. Hyperspectral imaging for food quality analysis and control. Elsevier, 2010.

[Tuia 16] D. Tuia, C. Persello & L. Bruzzone. *Domain adaptation for the classification of remote sensing data: An overview of recent advances.* IEEE Geosci. Remote Sens. Mag., vol. 4, no. 2, pages 41–47, Jun. 2016.

[Vapnik 98] V.N. Vapnik. Statistical learning theory. Wiley-Interscience, 1998.

[Wang 06] J. Wang & C.I. Chang. *Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis.* IEEE Trans. Geosci. Remote Sens., vol. 44, no. 6, pages 1586–1600, 2006.

[Weng 03] J. Weng, Y. Zhang & W. Huang. *Candid covariance-free incremental principal component analysis.* J. IEEE Trans Pattern Analysis and Machine Intelligence, vol. 25, no. 8, pages 1034–1040, 2003.

[Werff 06] H. Werff. *Knowledge based remote sensing of complex objects: recognition of spectral and spatial patterns resulting from natural hydrocarbon.* Itc dissertation, Utrecht University, 2006.

[Wu 15] Z. Wu, Q. wang & A. Plaza. *Parallel Implementation of sparse representation classifiers for hyperspectral imagery on GPUs.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 8, no. 6, pages 2912–2925, Jun. 2015.

[Yang 10] J. Yang, P. Yu & B. Kuo. *A nonparametric feature extraction and Its application to nearest neighbor classification for hyperspectral image data.* IEEE Trans. Geosci. Remote Sens., vol. 48, no. 3, pages 1279–1293, March 2010.

[Yang 11] H. Yang, Q. Du & G. Chen. *Unsupervised hyperspectral band selection using graphics processing units.* IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 4, no. 3, pages 660–668, Sep. 2011.

[Yokoya 14] N. Yokoya, S. Nakazawa, T. Matsuki & A. Iwasaki. *Fusion of hyperspectral and LiDAR data for landscape visual quality assessment.* IEEE J. Sel. Top. Appl. Earth Observat. Remote Sens., vol. 7, no. 6, pages 2419–2425, Jun. 2014.

[Zelnik 05] L. Zelnik & P. Perona. *Self-tuning spectral clustering.* pages 1601–1608. Advances in Neural Information Processing Systems 17, Cambridge, MA: MIT Press., 2005.

[Zhang 07] D. Zhang, Z. Zhou & S. Chen. *Semi-supervised dimensionality reduction.* pages 629–634, 2007.

[Zhang 10] S. Zhang & G. Yu. *Semi-supervised locality preserving projections with compactness enhancement.* pages 460–464, 2010.

[Zhang 12] L. Zhang, L. Zhang, D. Tao & X. Huang. *On combining multiple features for hyperspectral remote sensing image classification.* IEEE Trans. Geosci. Remote Sens., vol. 50, no. 3, pages 879–893, 2012.

[Zhong 16] Z. Zhong, B. Fan, K. Ding, H. Li, S. Xiang & C. Pan. *Efficient Multiple Feature Fusion With Hashing for Hyperspectral Imagery Classification: A Comparative Study.* IEEE Trans. Geosci. and Remote Sens., vol. 54, no. 8, pages 4461–4478, Aug. 2016.

[Zhou 04] D. Zhou, O. Bousquet, T.N. Lal, J. Weston & B. Scholkopf. *Learning with local and global consistency.* Advances in Neural Information Processing Systems (NIPS), 2004.

[Zhou 15] Y. Zhou, J. Peng & C. P. Chen. *Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification.* IEEE Trans. Geosci. Remote Sens., vol. 53, no. 2, pages 1082–1095, 2015.

[Zhu 08] X. Zhu. *Semi-supervised learning literature survey.* Technical report, Computer Sciences, University of Wisconsin-Madison, 2008.

[Zhu 12] Z. Zhu & C. E. Woodcock. *Object-based cloud and cloud shadow detection in Landsat imagery.* Remote Sens. Environ., vol. 118, no. 15, pages 83–94, Mar. 2012.

[Zhu 14] Z. Zhu & C. E. Woodcock. *cloud, cloud shadow, and snow detection in multitemporal landsat data: An algorithm designed specifically for monitoring land cover change.* Remote Sens. Environ., vol. 152, pages 217–234, Sep. 2014.

[Zubko 07] V. Zubko, Y. J. Kaufman, R. I. Burg & J. V. Martins. *Principal component analysis of remote sensing of aerosols over oceans.* IEEE Trans. Geosci. Remote Sens., vol. 45, no. 3, pages 730–745, March 2007.