

T.C
ISTANBUL AYDIN UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY



Spectral and Spatial Classification of Hyperspectral Images

M.Sc. THESIS
Eng. Emad MOUSELLI

Department of Electrical and Electronics Engineering
Electrical and Electronics Engineering Program

December 2017

T.C
ISTANBUL AYDIN UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY



Spectral and Spatial Classification of Hyperspectral Images

M.Sc. THESIS
Eng. Emad MOUSELLİ
(Y1513.300014)

**Department of Electrical and Electronics Engineering
Electrical and Electronics Engineering Program**

Advisor: Assist. Prof. Dr. Necip Gökhan KASAPOĞLU

December 2017



T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ

Yüksek Lisans Tez Onay Belgesi

Enstitümüz Elektrik- Elektronik Mühendisliği Ana Bilim Dalı Elektrik- Elektronik Mühendisliği (İngilizce) Tezli Yüksek Lisans Programı **Y1513.300014** numaralı öğrencisi **Emad MOUSELLI**' nin "SPECTRAL – SPATIAL CLASSIFICATION OF HYPERSPECTRAL IMAGES" adlı tez çalışması Enstitümüz Yönetim Kurulunun 26.12.2017 tarih ve 2017/31 sayılı kararıyla oluşturulan jüri tarafından **oybirliği** ile Tezli Yüksek Lisans tezi olarak **kabul** edilmiştir.

Öğretim Üyesi Adı Soyadı

İmzası

Tez Savunma Tarihi : 16/01/2018

1) Tez Danışmanı: Yrd. Doç. Dr. Necip Gökhan KASAPOĞLU

N. Gökhan Kasapoğlu

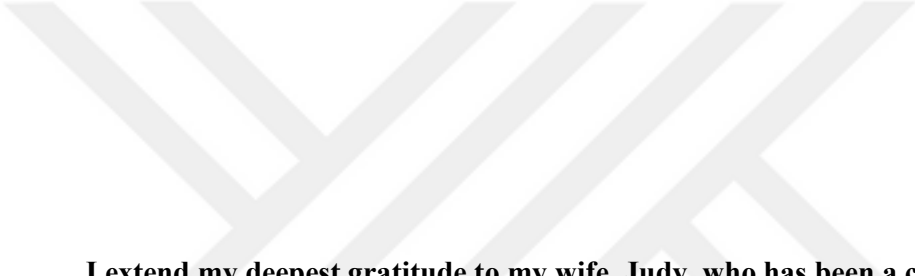
2) Jüri Üyesi : Prof. Dr. Mehmet Emin TACER

Mehmet Emin Tacer

3) Jüri Üyesi : Prof. Dr. Hasan Hüseyin BALIK

Hasan Hüseyin Balık

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.



I extend my deepest gratitude to my wife, Judy, who has been a constant source of support and encouragement during all the challenges in my life. I would like to also thank my father and siblings who always believed in me and Last but not least I would like to dedicate this work to the bright memory of my mother.

FOREWORD

I would first like to thank my thesis advisor Assist. Prof. Dr. Necip Gökhan KASAPOĞLU of the Electric and Electronic Engineering at Istanbul Aydin University. The door to Prof. KASAPOĞLU office was always open whenever I ran into a trouble spot or had a question. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it. I would like also to thank Istanbul Aydin University and its library for providing me with an access to all the books and articles that I needed to finish this work.

December 2017

Emad Mouselli



TABLE OF CONTENT

	<u>Page</u>
TABLE OF CONTENT	iii
LIST OF ABBREVIATIONS	v
LIST OF TABLES	vii
LIST OF FIGURES	ix
ÖZET	xi
ABSTRACT	xiii
1 INTRODUCTION	1
1.1 Remote Sensing and Machine Learning	1
1.2 Hyperspectral Images	2
1.3 Classification of Hyperspectral Images	3
2 FEATURE EXTRACTION METHODS	7
2.1 Principle Component Analysis	7
2.1.1 Principle Component Introduction	7
2.1.2 Principle Component Computation	9
3 STATISTICAL CLASSIFIERS	13
3.1 Statistical Classifier in General	13
3.2 Maximum Likelihood Classifier	13
3.3 Expectation Maximization	15
4 NONPARAMETRIC METHODS	17
4.1 The concept of Nonparametric Methods	17
4.2 Nearest Neighbor Based Classifier	17
4.2.1 Nearest Neighbor Algorithm	18
4.2.2 K-Nearest Neighbor	18
4.3 Kernel Methods and Support Vector Machine	20
4.3.1 Kernel Methods	20
4.3.2 Support Vector Machine	21
5 SPATIAL CLASSIFICATION:	25
5.1 Introduction to Spatial Classification	25
5.2 Watershed:	27
5.2.1 Watershed Segmentation for Hyperspectral Images:	29
5.2.2 Using Segmentation in Classification Scheme	32
5.3 Random Walker:	33
5.3.1 Exposition of the Algorithms:	33
5.3.2 Discrete Dirichlet Problem:	35
5.3.3 Theoretical Properties of the Algorithm	36
5.3.4 Algorithm Summary:	40
5.4 Extended Random Walker:	43
5.4.1 Development of the Algorithm:	44
5.4.2 Label Priors:	44
5.4.3 Algorithm Details:	47
5.4.4 Prior Model:	47
5.4.5 Choosing Weight	48
5.4.6 Numerical Solution	48

5.5	Comparison between ERW and Watershed.....	48
6	MODIFIED ERW	51
6.1	Priority for Large Classes	51
6.2	Priority for Small Classes	51
7	THE DATA USED IN THE EXPERIMENTS	53
7.1	synthetic data:	53
7.2	Indian Pines	53
7.3	Salinas scene:.....	55
8	EXPERIMENTAL RESULTS:.....	59
8.1	Synthetic Data.....	59
8.2	Hyperspectral Image Classification.....	62
8.2.1	Spectral classifiers:	63
8.2.2	KNN:.....	63
8.2.3	SVM.....	65
8.2.4	Feature extraction:	70
8.2.5	Spatial classification:	72
8.2.6	The importance of the Training Samples:	788.3
	Modified ERW Results.....	80
	REFERENCES	83
	RESUME.....	89
8.2.7	Salinas Scene:	79

LIST OF ABBREVIATIONS

NN	Neural Network
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
RW	Random Walker
ERW	Extended Random Walker
PCA	Principle Component Analysis
ML	Maximum Likelihood
EM	Expectation Maximization
MKNN	Modified K-Nearest Neighbor
OAO	One Against One
OAA	One Against All
WHED	Watershed Pixel
ICA	Independent Component Analysis
MNF	Maximum Noisy Fraction
RCMG	Robust Color Morphological Gradient



LIST OF TABLES

	<u>Page</u>
Table 5.1 : Comparison between ERW and Watershed.....	49
Table 7.1 : Indian Pines Groundtruth classes and their respective samples number.	54
Table 7.2 : Salinas Scene Groundtruth classes and their respective samples number.	56
Table 8.1 : K-NN Classification result on Indian Pines.....	63
Table 8.2 : Linear SVM Classification result on Indian Pines.....	67
Table 8.3 : RBF-SVM Classification result on Indian Pines.:	69
Table 8.4 : Watershed-SVM, Spatial-Spectral Classification result on Indian Pines	73
Table 8.5 : EWR Classification Result On Indian Pines Using Only The 1st PC. ...	75
Table 8.6 : ERW Classification Results On Indian Pines Using Full Spectral (200 Bands).	76
Table 8.7 : Confusion Matrix for ERW 17 Classes with seeds and $\lambda = 0.1$	78
Table 8.8 : SVM and EWR results using 4% of the total samples as training Samples:	79
Table 8.9 : SVM and ERW Classification result on 16 Classes Salinas Scene using 6% training Samples.	80
Table 8.10 : ERW results modification by giving different priorities for different classes.....	81



LIST OF FIGURES

Page

Figure 1.1: Hyperspectral Image Cube. 2

Figure 1.2: General Supervised Algorithm for Hyperspectral Images. 5

Figure 2.1: The real coordination axis for scattering the length and the width. 8

Figure 2.2: New transformed coordination axis for scattering the length and the width..... 8

Figure 2.3: Eigenvalues after applying PCA on 200 Bands Hyperspectral Image. ... 11

Figure 4.1: K-NN Numeric Example. 19

Figure 4.2: a) Linearly Inseparable Original Feature Space b) Mapped Feature space via ϕ is Linearly Separable. c) Using Kernel Function Makes Discriminant Function Nonlinear in The Original Space. 21

Figure 4.3: SVM Linear Separation case. 22

Figure 5.1: Topographic representation of a one band image. 27

Figure 5.2: Example of Watershed Transformation in One Dimension. 28

Figure 5.3:Flow Chart Which Shows Strategies of Applying Watershed to Hyperspectral Image. 29

Figure 5.4: Flow Chart of The Proposed Segmentation and Classification Scheme. 32

Figure 5.5: illustrates circuit theory to solve the Dirichlet problem of random walks. 34

Figure 5.6: Random walker Numeric Segmentation Example. 41

Figure 5.7: The Use of Intensity Priors is Equivalent to Using K Labeled Floating Node That Correspond to Each Label and Connected to Each Node. 46

Figure 7.1: A represents the original image with noise, B represents the original noisy image with the location of the seeds, C is the same as B with extra seeds in the 2-separated area..... 53

Figure. 7.2: Indiana Pines ground truth with the position of the random and neighboring samples..... 55

Figure 7.3: Salinas scene Groundtruth and Training Samples..... 57

Figure 8.1:SVM Classification result on Synthetic Data. 60

Figure 8.2: RW segmentation result. a) is the result of using c-seeds group. b) is the result of using b-seeds group..... 60

Figure 8.3:ERW illustrating. on synthetic Data, the 2-floating red and blue points represent the class labels. 61

Figure 8.4: Classification Result using ERW and b-seeds group. 62

Figure 8.5: Flow chart of the diverse way to apply ERW..... 63

Figure 8.6: KNN classification result on Indian pines 17 Classes..... 64

Figure 8.7: KNN classification result on Indian 16 Classes. 64

Figure 8.8: SVM Linear Function 17 classes Classification Results Neighboring Samples. 65

Figure 8.9: SVM Linear Function 16 classes Classification Results Neighboring Samples. 66

Figure 8.10: SVM Radial Basis Function 17 classes Classification Results Neighboring Samples. 68

Figure 8.11: SVM Radial Basis Function 16 classes Classification Results Neighboring Samples. 69

Figure 8.12: First Principle Component From PCA Transformed Indian Pines	71
Figure 8.13: : Watershed segmentation. a) over-segmentation b) under-segmentation c) proper-segmentation.....	72
Figure 8.14: SVM-Watershed, Spectral-spatial classification (17 Classes) a1,b1,c1 represent with-WHEDs classification, a2,b2,c2 represent no-WHEDs classification.....	73
Figure 8.15: SVM-Watershed, Spectral-Spatial Classification (16 Classes) a1,b1,c1 Represent With-WHEDs Classification, a2,b2,c2 Represent No-WHEDs Classification.....	74
Figure 8.16: ERW Classification With Feature Extraction For 16 And 17 Classes With/Without Seeds.	76
Figure 8.17: ERW Classification Without Feature Extraction For 16 And 17 Classes With/Without Seeds.	77
Figure 8.18: SVM and ERW classification results using only 409 training Samples.	79
Figure 8.19: SVM and ERW Classification Result on 16 Classes Salinas Scene using 6% Training Samples.	80

HIPERSPEKTRAL GÖRÜNTÜLER, SPEKTRAL UZAMSAL SINIFLANDIRMA

ÖZET

Son zamanlarda Uzaktan Algılama teknolojisinde çeşitli gelişmeler tanıtılmıştır. Multispektral sensörler yıllardır kullanılmakta ve 10-20 banta kadar çoklu bantlarda görüntüler sağlamaktadırlar. Multispektral görüntülerden çıkartılan bilgiler faydalıdır ve gerçek problemlerde birçok farklı uygulamada yaygın olarak kullanılmakla birlikte, hiperspektral görüntülerin önemli rol oynadığı mineraller ve alt sınıflar arasında ayırım yapmada başarısız olabilirler.

Hiperspektral görüntüler yüzlerce dar banttan oluşmuştur, çoğu durumda bant sayısı 200'e kadar çıkabilir. Bu yüksek düzeyde ayrıntılı spektral bilgilerin olması daha iyi bir ayırma yeteneği sağlar.

Multispektral görüntüler için kullanılan geleneksel sınıflandırma yöntemleri hiperspektral görüntülere birçok nedenden ötürü direk bir biçimde uygulanamaz. Bu nedenle birçok algoritma hiperspektral verileri işleyebilecek şekilde düzenlenmelidir. Örneğin, istatistiksel sınıflandırıcıların hiperspektral görüntüleri işlemede bazı zorluklar vardır. Bunun nedeni istatistiksel parametrelerin yeterli doğrulukta eğitim örneklerinden tahmin edilmesi büyük miktarda veri için kolay bir iş değildir. Ek olarak istatistiksel algoritmalar verilerin gerçek durumuyla örtüşmeyen belirli bir dağılıma sahip olduğunu varsayabilir. Öte yandan, parametrik olmayan sınıflayıcılar hiperspektral veriler için göreceli olarak yüksek ve kabul edilebilir doğrulukta iyi çözümler sağlarlar. Bazı durumlarda bu parametrik olmayan sınıflandırma yöntemleri hiperspektral veriye direk bir şekilde ya da öznitelik çıkarımı uygulandıktan sonra uygulanabilir. K-En yakın Komşuluk (KEK) ya da Destek Vektör Makinaları (DVM) en güçlü ve çok kullanılan parametrik olmayan yöntemlerdir.

Özellikle SVM sınırlı sayıda eğitim örnekleriyle bile hiperspektral veri sınıflandırmasında gürbüz bir yöntem olarak rapor edilmiştir. Gerçek hayatta komşu bölgelerden alınan eğitim örnekleri (pikseller) büyük olasılıkla aynı sınıfa aittirler. Ancak sadece spektral bilgileri kullanan sınıflandırma yöntemleri homojen alanlarda görülen yanlış sınıflandırılmış pikseller gibi bazı sorunlara sahip olabilirler. Son zamanlarda sınıflandırma doğruluğunu geliştirmek ve sınıflandırma haritalarında daha çok homojen alanlar elde etmek için literatürlerde çeşitli yaklaşımlar tanıtılmıştır. Güçlü yaklaşımlardan biri spektral bilgiyi uzamsal bilgi ile bütünleştirmeye dayanmaktadır. Bu tezde Genişletilmiş Rastgele Yürüme (GRY) algoritmasına odaklanılmaktadır. GRY iki adımdan oluşur; birinci adım herhangi bir spektral sınıflandırma algoritması tarafından yapılan spektral sınıflandırmadır. Bu tezde çekirdeğe dayalı metodlardan biri olan DVM ile Radyal Taban Fonksiyonu (RTF) ve doğrusal çekirdek işlevi spektral sınıflandırmada kullanılır. İkinci adım DVM'dan elde edilen sınıflandırma sonuçlarına dayanır ve daha doğru olarak DVM algoritmasından elde edilen her piksele göre olasılıklara dayanır. Bu olasılıklar rastgele yürüme yöntemini bir bölütleme yönteminden, çok sınıflı sınıflandırma yöntemine dönüştürür.

Anahtar kelimeler: hiperspektral görüntüler, spektral uzamsal sınıflandırma.



SPECTRAL AND SPATIAL CLASSIFICATION OF HYPERSPECTRAL IMAGES

ABSTRACT

Recently various developments have been introduced in Remote Sensing Technology. Multispectral sensors have been used for years and provide images with multi bands up to 10-20 bands. The information extracted from the multispectral images are useful and helped in many different applications in the real world, however it may fail in distinguishing between different minerals or sub classes, where hyperspectral images plays an important role.

Hyperspectral images are made of hundreds of narrow bands, in most of the cases it can be up to 200 bands. Having this high level of detailed spectral information gives a better distinguishing capability.

The conventional classification methods used for multispectral images cannot be applied directly on hyperspectral images due to many reasons. Therefore, many algorithms adjusted or introduced to fit the hyperspectral data. For example, statistical classifiers have difficulties with these data, because calculating statistical parameters for such a huge amount of data is not an easy task. Additionally, the statistical algorithms assume that the data have a specific distribution which contradicts the real-world situation. On the other hand, nonparametric classifiers provides good solutions with relatively high and acceptable accuracies. In some cases, these nonparametric algorithms are applied directly on the hyperspectral data or after applying some of the feature extraction methods. K-Nearest Neighbor (KNN) or Support Vector Machines (SVMs) are one of the most widely used and powerful nonparametric methods. Especially SVMs are reported as robust algorithms on hyperspectral data classification even with limited number of training samples.

In real world, pixels/samples from neighboring areas are most likely belong to same class. However, classification algorithms exploiting only spectral information cause some noisy like misclassified samples in homogeneous areas. Various approaches have been introduced recently in the literature to improve the classification accuracy and obtain more homogeneous areas in classification maps. One of the powerful approaches is based on integrating the spatial information with the spectral information. In this thesis, we focus on the extended random walker (ERW) algorithm. ERW consists of two main steps; the first step is the spectral classification which is done by any spectral classification algorithm. In this thesis one of the kernel based methods support vector machine (SVM) with the radial basis function (RBF) and the linear kernel function are used in the spectral classification. The second step relies on the results of the classification obtained by SVM and more accurately it depends on the probabilities for each pixel obtained from the SVM algorithm. These probabilities are used to transfer random walker from a segmentation algorithm into a multi class classifier.

Keywords: Hyperspectral images, Spectral and Spatial Classification



1 INTRODUCTION

1.1 Remote Sensing and Machine Learning

Remote sensing for Earth observation witnessed a lot of development and new approaches. Generally Remote Sensing is divided into 2 main procedures, the first procedure includes capturing images for the surface, the second step is to analyze these captured images.

The data or images can be collected through different type of sensors, these sensors collect the arising electromagnetic energy field, to be more specific the information is included in the three variation of this field which are spatial, spectral and temporal variation, the old sensors focused on collecting one spectral then studying the spatial variation of this energy. The images provided by these sensors weren't neither informative enough nor able to distinguish between different classes. Scientists and researchers tried to improve the resolution of these sensors, but they were confronted by 2 difficulties. The first is producing sensors with a very high accuracy is very expensive. The second difficulty, scanning a very small landscape with high accuracy sensors will produce a huge amount of data and it's very difficult to handle it or process it. To overcome these obstacles a new approach was introduced. New sensors were used to collect both spectral and spatial variation of the electromagnetic field and here is where the multispectral images were originated. Multispectral images consist of ten bands and these bands are relatively wide. Later the demands on more detailed classification increased and these multispectral images were incapable of distinguishing between similar types of land covering materials. As a response to these demands hyperspectral images were introduced to replace multispectral images in the earth observation (EO) application.

Hyperspectral images consist of hundreds of narrow bands covering from the visible to the short-wave infrared region of the electromagnetic field. This new technology wouldn't be useful without finding proper way to handle it and extract information from it. Extracting information and handling data from remote sensing application is done by machine learning algorithms.

Machine learning is a part of artificial intelligence. The general concept of machine learning is to let the machine improve its performance by learning from the available data, learning here means using these data iteratively to optimize the machine performance. Machine learning can be predictive or descriptive. One of the most known predictive machine learning methods is regression where the machine can make a prediction on a specific phenomenon, whereas classification is a famous part in descriptive machine learning. Classification is the method used in most of EO application. Classification in EO applications gives decisions about which area in the land covering image belongs to which class and therefore provide important information in several earth or environment monitoring systems.

Generally, machine learning can be divided into two main methods supervised learning and unsupervised [1]. Supervised learning requires a prior knowledge of the processed data by using labelled training samples. This prior knowledge and labelled training samples are not used in unsupervised learning. However, there is a special kind in of machine learning called semi-supervised [2] which can be considered as a mix between the supervised and unsupervised learning. This can be done by using both labelled and unlabeled training samples.

1.2 Hyperspectral Images

To have a better understanding of hyperspectral images classification, a closer look to hyperspectral images will be introduced. The human's eyes can see only 3 different spectral bands corresponding to the visual primary colors Red Green and Blue, but in hyperspectral both the visible and invisible spectral are taken into consideration. Hyperspectral images include up to several hundred of contiguous spectral bands. Every pixel contains high spectral information which can be used to give precise and detailed classification by using fine wave length resolution and covering a wide range of wave length. Hyperspectral images are used to identify material, finding and detecting objects, certain objects leave a special evidence which are called spectral signature or fingerprints, these signatures are used to detect the objects. Some

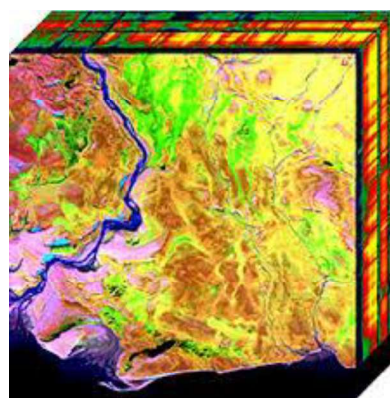


Figure 1.1: Hyperspectral Image Cube.

of the practical used way to collect these data is through using Airborne sensor or satellites. These sensors provide us with cube of data where each layer represent an image with a different wavelength so if we are measuring a 200-different wavelength we will get an Image with 200 band like the image in Figure (1.1). the collected precision can be evaluated according to the width of each band and referred to it as spectral resolution.

What make hyperspectral images different from normal or multispectral images is the spectral resolution. The higher spectral resolution gives the ability to distinguish more substances. The object of interest can become more specific which means even if the size of the observed object is very small, it's still can be detected due the high spectral resolution

1.3 Classification of Hyperspectral Images

In the beginning of hyperspectral images there was a consensus, that classification of multispectral images can be applied directly on hyperspectral images and this consensus came from the fact that hyperspectral were a normal extension of multispectral images with a bigger number in bands. Later, this misconception was removed and proved to be wrong. To illustrate this problem easily, the analysis of real and complex numbers can be taken into consideration. The complex numbers are considered as a normal extension of the real numbers, anyway applying mathematical rules from real numbers directly on complex numbers wouldn't give the required results and for example derivatives in real analysis is totally different from derivatives in complex analysis.

Analysis of hyperspectral image cannot be considered as trivial task and there are many reasons which complicate this task even more, here is some of the main factors which affects the classification in practical. 1) as mention before each class has its own spectral signature, but in real life application these signatures have a large spatial variability. 2) atmospheric effects can cause some noisy in the collected images and a small variation in the collected data. 3) the curse of dimensionality, even scanning a relative small area will give a huge data, due to the fact, that each vector pixel is consist of hundreds of bands. From the supervised learning methods perspective, there are two main inconveniences. 1) the limited number of training samples compared to the number of feature which makes normal statistical methods not applicable on

hyperspectral data, because this limitation of training samples will affect the estimation of the parameters.

2) Hughes phenomenon [3], theoretical the increment in the feature space will lead to a better discriminating ability which mean a better classification accuracy, but in some practical cases the increment of feature space will adversely affect the classification accuracy and this negative effect is Hughes phenomenon. 3) curse of dimensionality it worth to be mentioned here again, as supervised learning requires to study the training samples, creating the classification model and final apply this model on the required data, all this procedures on a huge data set will require more time and more advanced computers to be able to handle this amount of data.

In the literature, many efforts and works were done to overcome this methodological problem, here some of these methods will be mentioned briefly. 1) instead of using the covariance matrix obtained from the training samples directly some regularization was applied on it [4]. 2) the statistic estimation can be enhanced and generalized by including the contribution of the result of classified data in this estimation. [5] 3) dimensionality reduction by using some of the feature selection of feature extraction methods [6]. 4) modeling each class by the analysis of its spectral signature [7].

Nonparametric algorithms gained good reliability credit, due to its ability to function with a very limited number of samples. As earlier mentioned nonparametric methods can be applied directly without making neither distribution estimation nor calculating of the mean values, covariance matrix, etc. Neural network (NN) [8], Support vector machine (SVM) [9] and K-nearest neighbor (KNN) [10] are the commonly used nonparametric algorithms in hyperspectral images. These algorithms can be applied directly on the hyperspectral data without any feature extracting or selection, but also can give a better result when its combined with some feature extraction methods, for instance using principle component analysis with KNN algorithm. One of the deficiency of these algorithms is taking on consideration only the spectral information. Most of the information is included in the spectral variation of the hyperspectral data but still spatial information can be extracted and used to improve the classification results from the spectral classification. This new concept lead to new classification methods called spatial-spectral classification. Many approaches were introduced in the new field using segmentation algorithms, e.g. using watershed algorithms [11] to divide the image into a separate spatial area, then using this area to improve the result

obtained in spectral classification. More methods were used such as spatial feature extraction models, define adaptive neighbor for each pixel by applying some special filtering methods. This integration between spatial and spectral classification had successfully improve the classification accuracy. Figure 1.2 illustrate a general concept of the supervised learning algorithms in hyperspectral image, in fact there is more types and complications of supervised algorithm and this Figure is just the general concept.

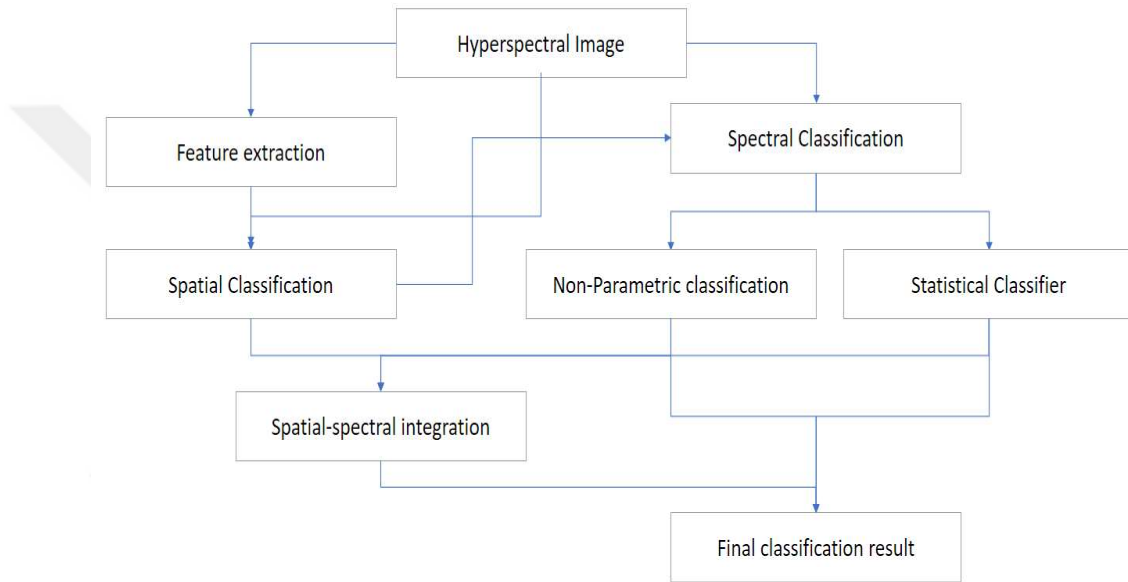


Figure 1.2: General Supervised Algorithm for Hyperspectral Images.

In this thesis SVM is used as a spectral classification method. It can be noticed that the results of spectral algorithms in general like SVM have some misclassified samples, this misclassified samples are distributed as salt and paper noise in the homogenous areas. To reduce this misclassification and increase the accuracy in Extend random walker (ERW) [12] are used to integrate the spatial information into the spectral information. This integration using ERW can be done in many ways, in this approach the different integration methods between SVM and ERW will be taken into consideration in details, there are lot alternative ways which can improve the classification accuracy. In the experiment part, these integration methods are exposed and the results are compared in detail.



2 FEATURE EXTRACTION METHODS

2.1. Principle Component Analysis

2.1.1. Principle Component Introduction

Principle component analysis (PCA) [13] is a statistical procedure which allows to transform of set of possibly correlated variables into a different space where the variables are linearly uncorrelated. For instance, in case we have 2 variables one of them is representing the length and the other one represents the width we can plot these 2 variables into 2- dimensional plane where the first axis represents the length and the second axis represent the width. After plotting these variables, we will get the result shown in Figure 2.1. After scattering the length and width it's more obvious that these 2 variables are having the similar variance and they are highly correlated. We can add a new axis along the biggest change of the data and then adding a new axis perpendicular to the first one to represent the other changing in the data and these two- new axes should pass through the centroid of the data. The data can be represented according to the new coordination axes. In this new feature space, it's obvious that the variances of the data in the first axis is bigger than the variances over the second axis, in the same time the spatial relationships between all the points is kept untouched so the data was represented in a new feature space and saved the spatial relationships as in figure 2.2. We can think of it as the data has been merely rotated.

The new axes are the result of rotating the data and can have many different meanings according to the originals samples for example in this approach we can consider the first axis represent the size measures where the data on the left side consist of small width and small length and moving to the right of the first axis the data will have larger width and length, whereas the second axis can be representing the ratio between the width and length.

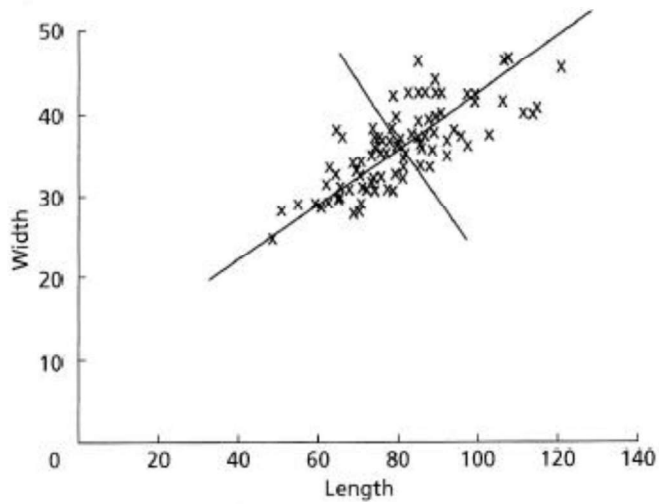


Figure 2.1: The real coordination axis for scattering the length and the width.

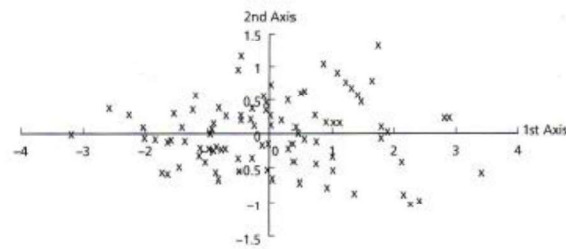


Figure 2.2: New transformed coordination axis for scattering the length and the width.

Dealing with small number of variables, the process of finding the relationships might seems obvious but not when dealing with a larger number. This process helps to find the relationships easier and faster. For some data set the variance of different axis might varies, so the axis corresponding to a higher variance are considered more important or containing more information than the other axis. The axis with low covariance can be ignored this process is called dimensionality reduction where the original d -dimensional space is converted to k -dimensional space where $k < d$.

The main concept of principle component analysis is rotating the data to have successive axes representing the data in way that the covariance is decreasing along

the axes with the first axis having the highest covariance and the last axis having the smallest covariance.

2.1.2. Principle Component Computation

The data being discussed in this computation part consist of P variables and n samples in case of hyperspectral images the variables are the features (bands) and the samples are the number of points (pixels) in the image. Before starting the data should be mean normalized and in some data the feature normalization is also required. This normalization insures that the data is centered on the origin and the spatial relationship and the covariance between the variables are being conserved. the 1st component Y_1 is equal to a linear combination of the variables X_1, X_2, \dots, X_p

$$Y_1 = \alpha_{11}X_1 + \alpha_{12}X_2 + \dots + \alpha_{1p}X_p \quad (2.1)$$

In matrix notation:

$$Y_1 = \alpha_1^T X \quad (2.2)$$

As mentioned before the first axis or the first component represent the greatest variance in the data. To choose a high variance for Y_1 a high value of the weights $\alpha_{11}, \alpha_{12}, \dots, \alpha_{1p}$ can be chosen, when choosing the values of this weights the following constrain should be taken into consideration. The sum of squares of the weights should be equal to 1.

$$\alpha_{11}^2 + \alpha_{12}^2 + \dots + \alpha_{1p}^2 = 1 \quad (2.3)$$

When selection the second principle component another condition should be taking into consideration that the 1st component and the 2nd one are uncorrelated i.e., perpendicular to each other.

$$Y_2 = \alpha_{21}X_1 + \alpha_{22}X_2 + \dots + \alpha_{2p}X_p \quad (2.4)$$

This process will continue till reaching Y_p then the number of the principle component is equal to the number of variables (P). in this stage, the sum of variances to all the principle components should be equal to the sum of the variances to the all variables. Therefore, all the originals information from the data are kept or accounted for in the

principle components. In the matrix notation, we can rewrite the principle components equations collectively as

$$Y = XA \quad (2.5)$$

Calculating the principle component requires a computer to perform the complicated mathematical equations and later there is an example how to use MATLAB to calculate this component. Back to the equation (2.5). The rows of A are called the eigenvectors of matrix S_x , S_x is the variance and covariance matrix of the original data. α_{ij} are called the loading and they are the elements of A the eigenvector matrix. S_y is the variance and covariance matrix of the principle component. The diagonal elements of the matrix S_y are the eigenvalues which represent the varies in the variances between the principle component, as mentioned before variances of the principle component is descending starting with the highest value responding to the first component.

The sample r can be directly calculated on the K^{th} component by using the following equation:

$$Y_{rk} = \alpha_{1k}x_{r1} + \alpha_{2k}x_{r2} + \dots + \alpha_{pk}x_{rp} \quad (2.6)$$

The position of each observation in the new coordinate system is called score.

To have a better understanding of the principle component is good to observe the correlation of the original variables with the principle component as this can be calculated using the following equation.

$$r_{ij} = \sqrt{\alpha_{ij}^2 \text{Var}(Y_j) / s_{ii}} \quad (2.7)$$

The result applying component analysis is a new feature space which have the same number of dimensions as the real data, Figure 2.3 illustrate the result from applying PCA on 200 bands hyperspectral image, it can be noticed that there are 200 Eigenvalue and the high values are concentrated in approximately the first 10 eigenvalues, but the mean idea of PCA is dimension reduction therefore some of the new principle component can be ignored there are many criteria that determine how many PC should be taken into consideration and how many should be ignored. One of this commonly used criterion is to take all the component till reaching a PC that only make a small

increasing in the total variance a second criterion is to take the PCs that represent approximately 90% of the total variance.

In this discussion, we used the first principle component. Computing PCA in MATLAB:

There is a ready function in MATLAB called PCA which returns the principle components coefficient (the loadings) i.e., having X consisting of n samples and P variable, X is n*p matrix we can apply the following code: **Loadings = pca(X)**;

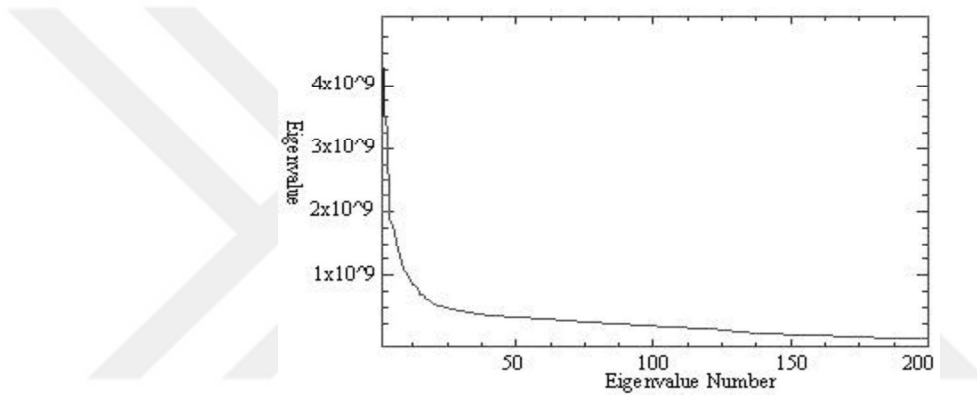


Figure 2.3: Eigenvalues after applying PCA on 200 Bands Hyperspectral Image.

A brief summarization of the PCA algorithm

- 1) Calculating the mean normalization and/or feature normalization

The mean value can be calculated using the following formula $\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^i$

then having $x_j = x_j - \mu_j$

- 2) Compute the covariance matrix: $\Sigma = \frac{1}{m} \sum_{i=1}^m (x^i)(x^i)^T$

- 3) Calculating the eigenvector of Σ matrix

This can be done using the ready MATLAB function (svd) singular value decomposition as follow

[u, s, v] = svd(Σ); u is n*n matrix and it represents the eigenvectors

The dimension reduction into a k-dimensional space can be done by selecting k vector from u then multiple the original data with these new selected vectors **$u_k = u(:, 1:k)$** ;

PC = x * u_k ;



3 STATISTICAL CLASSIFIERS

3.1 Statistical Classifier in General

A brief introduction to statistical classifier will be introduced here as these classifiers can provide an easy and simple explanation of what a classifier is and on the other hand these classifiers have many drawbacks when it comes to complicated classification problems such as hyperspectral images. These methods need to do a pre-calculation to find different parameters in the original data such as the mean, variance, etc. These pre-calculations can become problematic in case of dealing with large number of features and small number of training samples. Here these methods won't be deepened instead only a short review for maximum likelihood (ML) and expectation maximization (EM) [14, 15] are introduced.

3.2 Maximum Likelihood Classifier

This classifier is based on the conditional probability density function and requires a function for each class which means to classify a data with m classes for example this method requires m conditional probability density functions; the general form of these functions is:

$$g_{c_i}(\bar{x}) = p(\bar{x}|C_i), i = 1 \dots m. \quad (3.1)$$

This rule will be applied on each of the classes and the class which gives the highest value (the maximum) will assign its label to the point x

$$w = \arg \max\{g_{c_i}(\bar{x})\}, i = 1 \dots m \Rightarrow \bar{x} \in C_w \quad (3.2)$$

ML [14] is a statistical classifier therefore the solution of this problem is done by calculation some parameters which are related to probability density function (PDF). There are many different probability density functions which can be used in ML and the most commonly used function is Gaussian density function because of its

convenient properties and the fact that it fits many process in the nature. The Gaussian distribution of one-dimensional variable is given by:

$$p(\bar{x}|C_i) = \frac{1}{\sqrt{2\pi\sigma_i}} \exp\left[-\frac{(x - \mu_i)^2}{2\sigma_i}\right] \quad (3.3)$$

The important of Gaussian distribution is confirmed by the central limit theorem which states that, if a random observation is made on a large collection a of number of independent random quantities, the observation will have a Gaussian distribution.

In case of dealing with hyperspectral or multispectral data each variable will be represented as a vector, N-dimensional vector where N represent the number of attribute or bands in this experiment the AVIRIS used dataset consist of 200 bands. The vector form of the Gaussian PDF is

$$p(\bar{x}|C_i) = (2\pi)^{-n/2} |\bar{\Sigma}_i|^{-1/2} \exp\left\{-\frac{1}{2} (\bar{x} - \bar{\mu}_i)^T \bar{\Sigma}_i^{-1} (\bar{x} - \bar{\mu}_i)\right\} \quad (3.4)$$

Where

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_N \end{bmatrix}, \bar{\mu}_i = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \cdot \\ \cdot \\ \mu_N \end{bmatrix}, \bar{\Sigma}_i = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdot & \cdot & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & \cdot & \cdot & \sigma_{2N} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \sigma_{N1} & \sigma_{N2} & \cdot & \cdot & \sigma_{NN} \end{bmatrix} \quad (3.5)$$

\bar{x}_i represent the random variable and for each class there is 2 values need to be calculated the mean and covariance matrix which are noted respectively as $\bar{\mu}_i, \bar{\Sigma}_i$. Having the following training dataset $\{(\bar{x}_1, y_1), (\bar{x}_2, y_2), \dots, (\bar{x}_n, y_n)\}$ where \bar{x}_i represent the training samples, $i = 1, \dots, n$, n is the number of training samples, y_i represent the class labels, $y_i \in \{1, 2, \dots, m\}$, m is the number of classes. There are two variables need to be calculated in this algorithm the mean and covariance matrix for each class, the following formulations explain how to calculate these two parameters respectively.

$$\hat{\mu}_i = \frac{1}{n} \sum_{j=1}^{n_i} \bar{x}_j, \{\bar{x}_j | y_j = i, i = 1, \dots, m\} \quad (3.6)$$

$$\hat{\Sigma}_i = \frac{1}{n_j - 1} \sum_{j=1}^{n_i} (\bar{x}_j - \hat{\mu}_i)(\bar{x}_j - \hat{\mu}_i)^T, \{\bar{x}_j | y_j = i, i = 1, \dots, m\} \quad (3.7)$$

Where n_i is the total number of the training samples of the class i .

3.3 Expectation Maximization

In classifying application specially in remote sensing the training labels are limit and this limitation adversely affects the classification process with a limited number of training samples it is hard or even not possible to obtain high complexity degree of discrimination function which leads to low performance of the classifier, to overcome this drawback the number of training samples used to obtain the parameter can be increased by taken advantage of some of the unlabeled samples, this unlabeled samples will be incorporated with this original training samples to get a better estimation of the parameters, in the following section expectation maximization (EM) [14] will be explained and how it can be used to enhance the estimation of the Gaussian density function. In case of having Z training samples and X unlabeled samples to enhance the mixture density, i will refers to the number of the class, $i = \{1, \dots, m\}$ where m is the total number of classes, K is an index for each individual training sample. In case of training samples, the two indexes will be used i, k to refer to the class and sample index whereas in the labeled samples only the k index will be used.

$$p(\bar{x}|\theta) = \sum_{i=1}^m \alpha_i p_i(\bar{x}), i = 1, \dots, m \quad (3.8)$$

The expectation maximization equations used for approximating maximum likelihood parameters estimation of the mixture density are the following

$$\alpha_i^{t+1} = \frac{\sum_{k=1}^N \frac{\alpha_i^t p_i(\bar{x}_k | \bar{\mu}_i^t, \Sigma_i^t)}{p(\bar{x}_k | \theta^t)}}{N} \quad (3.9)$$

$$\bar{\mu}_i^{t+1} = \frac{\sum_{k=1}^N \frac{\alpha_i^t p_i(\bar{x}_k | \bar{\mu}_i^t, \Sigma_i^t)}{p(\bar{x}_k | \theta^t)} \bar{x}_k + \sum_{k=1}^{n_i} Z_{ik}}{\sum_{k=1}^N \frac{\alpha_i^t p_i(\bar{x}_k | \bar{\mu}_i^t, \Sigma_i^t)}{p(\bar{x}_k | \theta^t)} + n_i} \quad (3.10)$$

$$\sum_i^{t+1} = \frac{\sum_{k=1}^N \frac{\alpha_i^t p_i(\bar{x}_k | \bar{\mu}_i^t, \Sigma_i^t)}{p(\bar{x}_k | \theta^t)} (\bar{x}_k - \bar{\mu}_i^{t+1})(\bar{x}_k - \bar{\mu}_i^{t+1})^T + \sum_{k=1}^{n_i} (\bar{z}_{ik} - \bar{\mu}_i^{t+1})(\bar{z}_{ik} - \bar{\mu}_i^{t+1})^T}{\sum_{k=1}^N \frac{\alpha_i^t p_i(\bar{x}_k | \bar{\mu}_i^t, \Sigma_i^t)}{p(\bar{x}_k | \theta^t)} + n_i} \quad (3.11)$$

Where $\bar{\mu}_i^t$ is the mean vector of class i iteration t and also Σ_i^t is the covariance matrix of class i iteration t. all the prior probabilities, mean vectors and covariance matrices are contained in the parameter θ^t . The parameters estimation in Maximum likelihood are obtained from the training samples and it can be obtained by using different initial values but more reasonable to start from the training samples as initial values after that new parameters can be obtained by iterating the above mentioned EM equations, theoretically using unlabeled samples with the EM equation will always give a better estimation of the parameters which means a better performance for the classifier but unfortunately in practical it's not necessary that unlabeled samples with EM will improve the accuracy sometimes it might leads to undesirable results due to the deviation of the data in the real world Therefore, in case of using supervised, semi-supervised or unsupervised it is a good technique to start with the normal training samples and after evaluating the performance of the classifier some extra unlabeled samples can be used to enhance the statistical estimation of the parameter and in case of unwanted results these unlabeled samples can be abandoned and new samples are taken into consideration.

4 NONPARAMETRIC METHODS

4.1 The concept of Nonparametric Methods

All classification algorithms based on statistics have a mutual disadvantages, in these algorithms the data assumed to have specific distribution which is in most of the case Gaussian distribution but in real-world data this assumption might be incorrect and in addition, these classifiers need to calculate some statistical parameter and sometime it requires to estimate these parameters and this parameter calculating or estimating become problematic and critical when dealing with data that only have a small number of training samples such as hyperspectral data. To overcome of this deficiency a lot of methods were introduced. These complementary methods are used to enhance the estimation of the parameter like EM, but still these methods sometimes fail to achieve the required results therefore, a good alternative solution is nonparametric methods where the main aim of these algorithm is to take full advantage of the available training samples and extract all the information to constrain a proper discriminative rule, for this reason these algorithms are highly used and preferred. Another advantage of these methods that they are more resistant to Hughes effects [3] than parametric methods due to the stabilities of the classification obtained regardless the dimensionality changes. One of the most known and used nonparametric algorithm is K Nearest Neighbor (KNN) [10] and Support Vector Machine (SVM) [9].

4.2 Nearest Neighbor Based Classifier

This algorithm is one of the easiest algorithm in image classification theory and still have an acceptable accuracy. Nearest neighbor algorithm gives decision for each sample based on the class of the nearest neighbor. When having many sample as mentioned in [16] this rule has a probability of error which is less than twice the Bayes probability of error. As all the classification algorithms nearest neighbor needs a training set, this training set is used to classify the patterns in the data. In the Nearest neighbor approach, the algorithm tries to find the similarity between each point of the testing samples and all the training samples.

In this approach, we will discuss Nearest neighbor and K- Nearest neighbor.

4.2.1 Nearest Neighbor Algorithm

This algorithm assigns to a testing pattern the label of its nearest training samples or in other words it assigns the label of the nearest neighbor. Consider having n training samples as follow: $(X_1Y_1), (X_2Y_2), \dots, (X_nY_n)$, Where:

Y_i represent the class label of the i^{th} pattern, i^{th} is the i^{th} training samples P is the testing sample. The decision rule can be written as

$$Y_p = \operatorname{argmin}\{d(P, X_i)\} \quad (4.1)$$

4.2.2 K-Nearest Neighbor

KNN has the same concept as the one used in nearest neighbor but instead of finding only the nearest neighbor here we will find the K nearest neighbor, then the class will be determined according to the majorities of the K nearest neighbors' labels. K is real positive integer chosen by the user.

As K is the only free parameter in this algorithm its value is critical in improving the overall accuracy. Using KNN is more efficient than using 1-NN for a simple reason in case of a noisy data the nearest neighbor might belongs to a different class but in case where more than one of the nearest neighbors is having the same class this means it is more likely that this testing point belongs to that class.

There are many different similarity measurements which are commonly used with KNN, for example Euclidean distance or any other similarity function can be used as Mahalanobis distance. MATLAB provide a ready function for KNN classifiers which is **knnclassify** and MATLAB provide us with several distance measurements which are Euclidean distance, sum of absolute differences, one minus the cosine of the included angle between points, one minus the sample correlation between points (treated as sequences of values) and percentage of bits that differ (suitable only for binary data).

There is still more kind of nearest neighbors' algorithms which MKNN the modified K- nearest neighbor [17], the fuzzy KNN [18] and some other algorithm and as far only KNN will be used in our approach the rest of the algorithms are not going to be explain here. The following KNN numeric example is to illustrate how KNN algorithm

woks. Let's consider having the following training points and the testing point $P = (15, 9, ?)$ as shown in Figure 4.1.

$$\begin{array}{lll}
 X_1 = (4,4,1), & X_2 = (5,5,1), & X_3 = (6,4,1), \\
 X_4 = (4,6,1), & X_5 = (6,6,1), & X_6 = (20,15,2), \\
 X_7 = (19,14,2), & X_8 = (21,14,2), & X_9 = (19,16,2), \\
 X_{10} = (21,16,2), & X_{11} = (16,3,3), & X_{12} = (19,2,3), \\
 X_{13} = (15,9,3), & &
 \end{array}$$

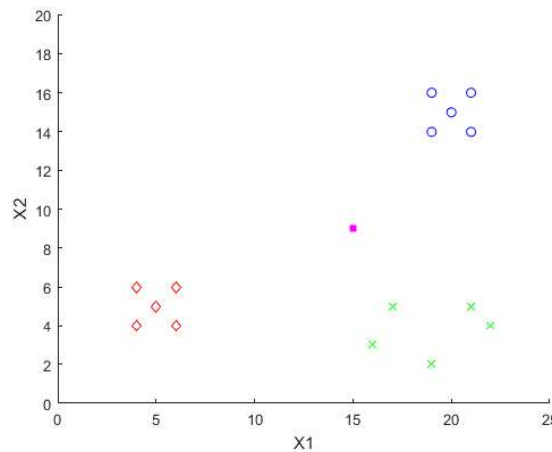


Figure 4.1: K-NN Numeric Example.

Classes 1,2,3 are represented by the red diamonds, blue circle and green 'x' respectively where the magenta square represent the testing point. The first number of each point correspond to the first feature X_1 and the second number correspond to the second feature X_2 and the 3rd number is the label. Applying the Euclidean distant rule on our example

$$\text{Euclidean distance } d(X, P) = \sqrt{(X[1] - P[1])^2 + (X[2] - P[2])^2}$$

After calculating the Euclidean distance between all the training samples and P in case of having $K=3$ the 3 nearest training samples are X_{13}, X_{11}, X_7 with 4.4721, 6.0828 and 6.4031 distance respectively, as we can see X_{13} and X_{11} are responding to class3 and X_7 responding to class 2 therefore the final decision is that P belongs to the class 3.

4.3 Kernel Methods and Support Vector Machine

4.3.1 Kernel Methods

Using linear discriminant functions is well known and easy to understand and apply, due to the simplicity of the mathematical equation representing these functions, but in the real-world applications linear discriminant is not sufficient in most of the cases since most of the real data are not linearly separable. Instead of applying discriminant functions directly on the original feature space Kernel methods transform the feature space into a higher dimensional feature space where applying linear discriminant function is applicable and this characteristic made the kernel methods widely used in many different remote sensing application, due to the significant role of the kernels, a great explanation of the kernels and their application can be found in [19, 20] . Figure 4.2 visualize the way kernel function works. The following sample example can explain the general concept of kernels methods. Suppose that the following empirical data need to be classified. $(x_1, y_1), \dots, (x_n, y_n) \in X \times Y$.

Where X is a set of data and Y is the target or labels of these data $Y = \{1, -1\}$, where n is the index of the sample. Here no more additional assumption will be added to the domain X , X is just a set of data and in order to study these data we need to find a way to generalize these data which means in loosely speak we need to be able to classify any extra point x if it belongs to the class $Y = 1$ or $Y = -1$, to do so similarity measure in X and Y is required, for the former we require the function:

$$k: X \times X \rightarrow \mathbb{R}, (x, x') \rightarrow k(x, x') \quad (4.2)$$

One of the similarity functions which can be used in this example is to find the mean for each class 1 and -1 and then compare the point to these mean values.

Using Kernels allow us to find nonlinear or even sophisticated boundaries. Decision in the real feature space derived from the linear decision boundary in new kernel mapped feature space [13]. One of the well know and widely used kernels based algorithms is Support Vector Machine and it can be used in many different fields, it is a robust tool when it comes to high dimensional feature space and the overall accuracy of the SVM is relatively high compared to other algorithms.

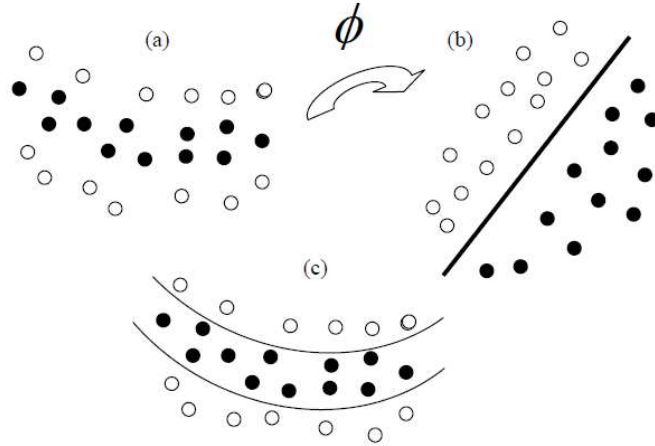


Figure 4.2: a) Linearly Inseparable Original Feature Space b) Mapped Feature space via ϕ is Linearly Separable. c) Using Kernel Function Makes Discriminant Function Nonlinear in The Original Space.

4.3.2 Support Vector Machine

The main approach of support vector machine is to find a hyperplane that separate the data in a way which makes the distance between samples and the hyperplane as big as possible in other words the Idea of this algorithm is to find the optimized separation between classes by selecting a decision boundary with the biggest margin from the other samples. By using geometric margins this distance between the boundaries and sample can be given as $2/\|w\|$ as shown in Figure 4.3 so the generalization of SVM performance is directly related to the concept of the margin if we want to increase the generalization we need to increase the margin so it's proportional relation between generalization and margin. The detailed mathematical information and explanation behind this powerful method can be found in [21, 22]. Here a brief explanation about the mathematic behind the SVM in Linear cases and how we can use the kernel trick to apply SVM on nonlinear cases.

First, let's discuss the linear separable cases if we want to find the optimal hyperplane we need to solve the following quadratic problem.

$$\text{minimize} : \frac{1}{2} \|w^2\| \quad (4.3)$$

$$\text{subject to} : y(\langle \bar{w}, \bar{x}_i \rangle + b) \geq 1 \quad i = 1, 2 \dots m$$

By using Lagrange formulation, we can convert this optimization problem into dual problem

$$\text{maximize : } \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \langle \bar{x}_i, \bar{x}_j \rangle \quad (4.5)$$

$$\text{subject to : } \sum_{i=1}^m \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, i = 1, 2, \dots, m$$

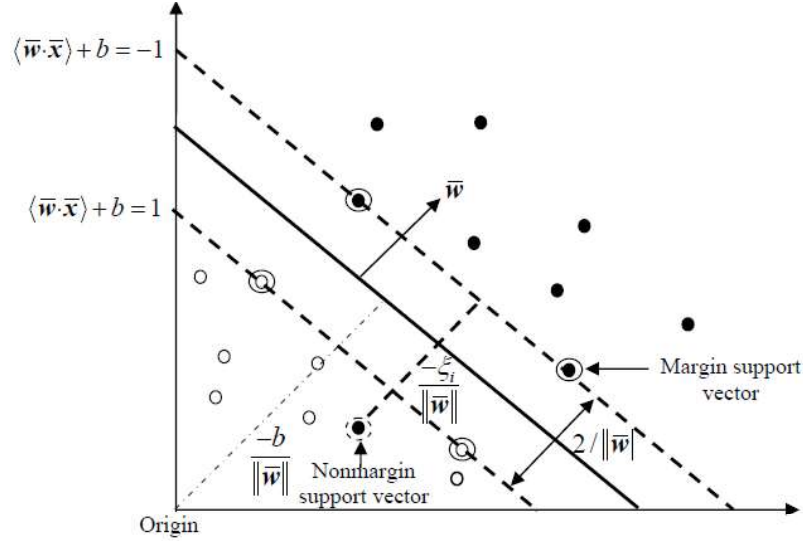


Figure 4.3: SVM Linear Separation case.

The discriminant function which is used to find the optimal plane (with the largest margin) can be written as follows:

$$f(\bar{x}) = \sum_{i \in S} \alpha_i y_i \langle \bar{x}_i, \bar{x} \rangle + b \quad (4.6)$$

We still have one more variables to calculate which are the Lagrange multipliers α_i 's, and those can be estimated using (QP) quadratic programming.

The S in the discriminant function above is equal to the nonzero Lagrange multipliers in the training samples. We have two kinds of samples; one of them effects the decision boundary and the other doesn't, we denote the first one as significant training samples and assign them to a nonzero Lagrange multiplier and assign the others to a zero

Lagrange multiplier. We use the term support vector to denote these samples (nonzero α_i).

A new concept was generalized to find optimal solution for using linear SVM in remote sensing classification where most of the cases are linearly non-separable. The new concept depends on finding an optimized hyperplane using the following cost function

$$\Psi(\bar{w}, \xi) = \frac{1}{2} \|\bar{W}\|^2 + c \sum_{i=1}^m \xi_i \quad (4.7)$$

This cost function used to find the maximization of the margin and in the same time try to keep minimizing the classification error.

We have two variables in this cost function; the first one ξ_i called the slack variable and the second one is C; this variable is used to control the error correction penalty. Its proportional relation between the value of C and the penalty assigned to each misclassification so if we want to increase the penalty we can increase the value of C and vice versa, back to or cost function the minimization has the following constrains:

$$y_i(\langle \bar{w}, \bar{x}_i \rangle + b) \geq 1 - \xi_i, i = 1, 2, \dots, m \quad (4.8)$$

$$\xi_i \geq 0, i = 1, 2, \dots, m \quad (4.9)$$

As it is mentioned before some of the training samples are more important than the others and called support vectors in case of non-separable problem we have two kind of support vectors the first one is the normal support vectors which lie on the margin of the hyperplane therefore they are called margin support vectors the second kind is called non-margin support vectors and these vectors lie on the wrong side of the margin.

To overcome the linearity of SVM in the original feature space we can use kernels. Which can turn the support vector machine into a nonlinear classifier. By using kernel methods, we can use the new transformed feature space $\langle \phi(\bar{x}_i), \phi(\bar{x}_j) \rangle$ instead of the inner product $\langle \bar{x}_i, \bar{x}_j \rangle$. and the advantage of using kernel that satisfied the Mercer's theorem [23] that we don't have to calculate the mapping function instead we can directly calculate the inner product in the

transform space as by using kernels we simplify the solution of the dual problem. The new modified formulations are written as follow:

$$\text{maximize : } \sum_{i=1}^m \alpha_i - 1/2 \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j (\bar{x}_i, \bar{x}_j) \quad (4.10)$$

$$\text{subject to : } \sum_{i=1}^m \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, m$$

By replacing the inner product in the mapping space with kernel function the discriminant function written as

$$f(\bar{x}) = \sum_{i \in S} \alpha_i y_i K(\bar{x}_i, \bar{x}) + b \quad (4.11)$$

There are many types of kernel function and each type effect the classification, therefore it is important to select the appropriate kernel function for each classification problem. One of the widely-used kernel functions is Gaussian radial basis function:

$$K(\bar{x}_i, \bar{x}) = \exp(-\gamma \|\bar{x}_i - \bar{x}\|^2) \quad (4.12)$$

We can control the width of the Gaussian kernel by using the parameter γ which is inversely proportional to the width. Also, polynomial functions can be used as kernels below:

$$K(\bar{x}_i, \bar{x}) = [\bar{x}_i, \bar{x} + 1]^p \quad (4.13)$$

where P is the order of the polynomial function.

SVM is a binary classification algorithm. There are some techniques that can be used to apply SVM on multi classes. The widely used techniques are called One-Against-One (OAO) and One-Against-All (OAA). In our experiments, OAA technique is used in multi-class SVM classification to obtain high classification accuracies.

In case of using linear SVM only complexity (C) parameter is needed to be chosen. and for using RBF-SVM parameters C and γ are chosen. In the experiments grid search method can be used to select proper parameters for SVM classifiers.

5 SPATIAL CLASSIFICATION:

5.1 Introduction to Spatial Classification

The results obtained from spectral classification can be improved by applying complementing spatial algorithms on the results obtained from spectral classifier, therefore the final decision for each pixel can be derived from the spectral features of the pixel and in addition to the relation between this pixel and its neighboring pixels. Some of the algorithms used in spatial classification are morphological filters [24], morphological leveling [25] and Markov random fields (MRF) [26]. These methods have shown the ability to reach high classification accuracy. However, all these algorithms have a common concept, which is all of them take a fixed-window-based of the neighboring pixel into consideration, which leads to scale selection problems and if the image contains small or complex structure this problem becomes more severe.

Alternative approach to the abovementioned methods is image segmentation [27, 28], to obtain high performance a good segmentation of the image is required. This segmentation gives information about each pixel and its neighboring pixels.

In previous study, image segmentation for multispectral image has been thoroughly discussed, where the spectral similarity was mostly used to distinguish between different areas. One of the powerful software which has been used in image segmentation is eCognition, which used bottom-up region margining [29]. Bottom-up methods initially start with the smallest element of the picture, it starts by considering each pixel as a separate region and the next step is to connect these regions according to some criterion in the shape and spectral of these regions. Another hierarchical image segmentation approach was introduced by Tilton [30], in which both region growing and spectral clustering were alternately used. This algorithm has some desired advantages, on the other hand it has a main drawback, to achieve a good segmentation thresholds or homogeneity criterion must be chosen accurately. Segmentation based on mathematical morphology were introduced in [31, 32, 25], which mostly used granulometries or watershed transformation. Since there is no natural way for multivariate pixels total ordering, applying morphological operators is a bit

complicated. Works in [33, 34] provide extensive literature on mathematical morphology for multispectral and color images. Also, some extensive literature about watershed segmentation can be found in [35, 36, 37]

These approaches are not applicable for hyperspectral images, due to the following reasons.

- a) The structure of hyperspectral images which consists of hundreds of bands makes it very hard to apply multivariate data total ordering schemes, for examples bit mixing paradigm [33] can't be applied in hyperspectral images, because it will result a numerous number of value for each pixel.
- b) Perceptual color spaces and polar-based representation were successfully used in color images morphological analysis [35, 38] Unfortunately, these approaches are not suitable for hyperspectral images.

Spectral-spatial classification of multispectral images is investigated in some studies. Linden et al. [39] use the mean vector as the feature for each region. The mean vector for each region are calculated after applying segmentation based on region growing, as a result each region forms the segmentation has its own mean vector, then a spectral classifier such as SVM is applied on the mean vectors for each region. However, the results obtained using these algorithms weren't better than results obtained by applying the spectral classification only. Li and Xiao [40] also introduced spatial spectral classification on 4 bands image by using both watershed segmentation and maximum likelihood for the spectral part the two algorithms are applied separately. A pixel wise approach is used to make decision for the regions, if a region contains more than 50% of its pixel with the same class the whole pixels in this region will be assigned to this class. The results obtain here were ultimately improved compared to the results obtained using only maximum likelihood.

Spatial information is also used in classification problems by Widayati et al. [41]. 4-band IKONOS image was used in the experiments. Merge using moments algorithms are first used to obtain the segmentation map. Then two different methods to integrate the spatial and spectral classifier are used. In the first approach, the mean for each region is calculated and then these regions are classified according to their mean vector as feature. In the second approach, a spectra classifier (Maximum Likelihood) are

applied directly on the image and result are combined with segmentation results by applying the majority voting therefore each region has the class of its majority class.

In this thesis watershed and random walker (RW) [42] algorithms on hyperspectral images are employed for segmentation. And later Extended random walker (ERW) [12] algorithm which is adapted version of random walker to integrate segmentation and spectral classification is employed for hyperspectral data classification. EWR was applied on hyperspectral data and gave good results [43] and later different kind of learning methods integrated to enhance the obtained result [44]

5.2 Watershed:

Watershed was introduced by Beucher and Lantuejoul [11], as a powerful mathematical morphology technique, this powerful method is used for image segmentation.

The watershed deals with topographic images which is a 2D image and one band the value of each pixel represents the elevation of that pixel. Watershed creates lines which divide the image into catchment basins. Figure.5.1 illustrates how the image is divided, each of these basins take one of the minimum in the image. Watershed cannot be applied directly on the image instead first a gradient of the image is calculated and later the watershed is applied on this gradient. The gradient function describes the changes between neighbor pixels if the pixels belong to same region, they have similar values and therefore, the gradient function has minima value, otherwise the gradient function has maxima value. The watershed segmentation can give a meaningful result if the gradient function gave a good description to the border between different areas.

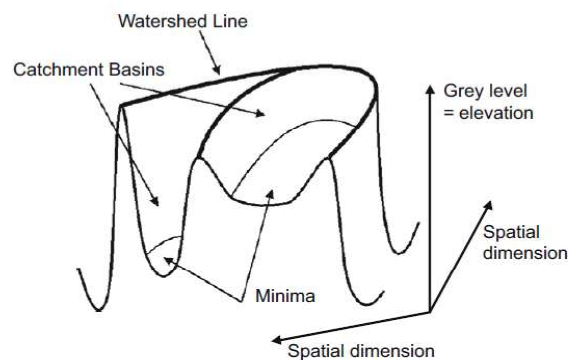


Figure 5.1: Topographic representation of a one band image.

Many works in the literature focused on watershed algorithm e.g. Vencent and Soille in [45] by using the flooding stimulations to apply efficient watershed transformation. Watershed transformation on an image divides the image into small regions, each region is separated from the other regions via watershed pixels (WHED) and each region consists of pixels that all connected to a local minimum. Figure. 5.2 illustrates watershed transformation in one dimension. In Figure. 5.2, there are 2 local maximum points that are the watershed pixels (WHED) and they divide the 1D space into 3 regions each of them is connected to one of the 3 local minimum points.

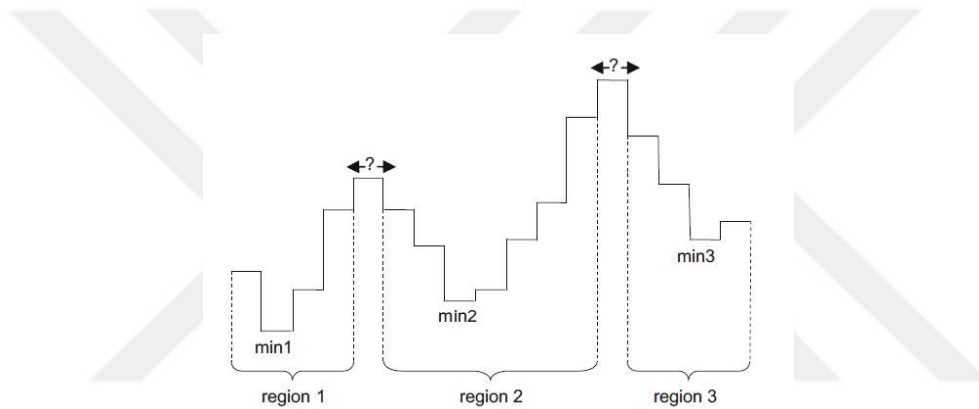


Figure 5.2: Example of Watershed Transformation in One Dimension.

As mentioned before watershed algorithm is applied on gradient function, but practically applying watershed directly on the result of gradient function will lead to over segmentation, over segmentation means that the images is divided into very small regions where each region contain only a local minimum without any of its neighbor. There are many techniques to cope the over segmentation, using marker is one of the good methods more details about using marker can be found in [45], also filtering the image or the gradient function can be a helpful as well. Back to hyperspectral images, these images contains hundreds of bands and watershed is applied on 2-Dimensional one band images therefore some techniques can be used to enable this algorithm on hyperspectral images in the following section these techniques will be discussed in detail.

5.2.1 Watershed Segmentation for Hyperspectral Images:

In the paper [46] watershed transformation was applied on B-band image to get a one-band segmentation map. Let us consider that the image is set of n pixel vectors $X = \{x_j \in \mathbb{R}^B, j = 1, 2, \dots, n\}$ and each image band can be denoted as $X_\lambda, \lambda = 1, 2, \dots, B$. Figure 5.3 illustrates the different ways watershed can be calculated.

Before applying gradient function directly on the original image some feature extraction techniques can be used. The aim of this step is to obtain one band image or multi-band image where most of the spatial information are available to distinguish between different regions and one of the most popular feature extraction techniques is PCA [13], alternative methods are independent component analysis (ICA) and maximum noisy fraction (MNF) [47].

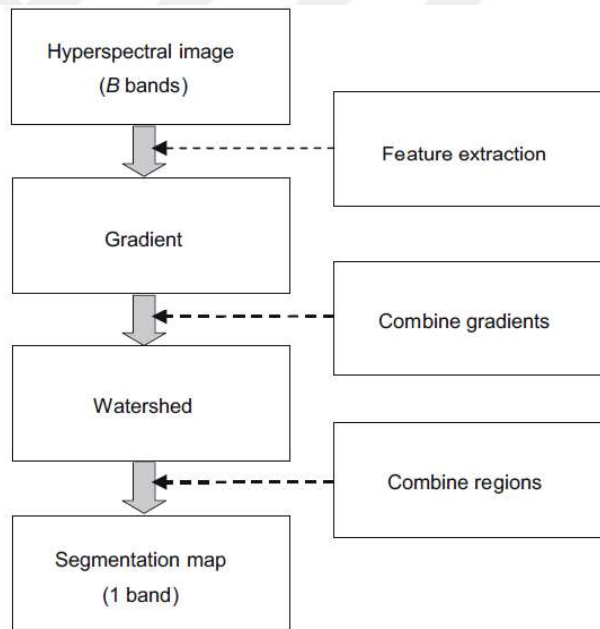


Figure 5.3: Flow Chart Which Shows Strategies of Applying Watershed to Hyperspectral Image.

In case we could obtain one-band image which contains enough spatial information to distinguish between different regions, applying watershed would become an easy straightforward task. A basic morphological gradient can be applied directly on this one-band image the gradient is called Beucher gradient, this gradient basically calculates the difference between the dilation and erosion using the following equation:

$$\rho_E(Y) = \delta_E(Y) - \varepsilon_E(Y) \quad (5.1)$$

In most of the cases obtaining one-band image which preserve most of the spatial information is a hard task and most of the time we need to apply gradient function on a multiband image, there are different ways to apply gradient function on these images these ways can be categorized as the following:

- A) To compute a vectoral gradient;
- B) To compute a multidimensional gradient;
- C) To compute watershed segmentation maps posteriori.

5.2.1.1 Computation of a Vectoral Gradient

Instead of calculating the distance between 2 pixels in the vectoral gradient the distance between to pixels-vectors is calculated and produce a one-band gradient. Many methods were proposed to calculate the metric based gradient in hyperspectral images. To explain these methods let's consider the vector pixel X_p and $\psi = [X_p^1, X_p^2, \dots, X_p^e]$ are the neighboring vector pixels for X_p and e is the number of neighbor vector pixels which can be 4 or 8. The following equation shows how the gradient is calculated according to the difference between the supremum and infimum distances between X_p and its neighbors:

$$\nabla_{\psi,d}^{MB}(X_p) = \sup_{i \in \psi} \{d(X_p, X_p^i)\} - \inf_{j \in \psi} \{d(X_p, X_p^j)\} \quad (5.2)$$

Different distance measurements can be used to calculate the distances e.g. Euclidean, Mahalanobis and chi-squared distance.

Robust color morphological gradient (RCMG) is another vectoral gradient which has been developed by Evans and Liu, and later in [48] the ability to apply RCGM on hyperspectral images are discussed.

5.2.1.2 Multidimensional Gradient Method

Instead of trying to transform our B-band image into a one-band image, the gradient function can be applied on each band from the B-band image by considering each band as a separate image. For B-band image B gradient function can be applied and thus we can get B gradient maps $\rho_e(X_\lambda)$, $\lambda = 1, 2, \dots, B$. These gradient maps can be combined in some linear or nonlinear ways, a sum of the weight function can be used as an

example of the linear operators. Let's denote ω_λ as the weight corresponding to the band λ , where $\lambda = 1, 2, \dots, B$ and the weight function can be given as

$$\nabla_E^+ = \sum_{\lambda=1}^B \omega_\lambda \rho_e(X_\lambda), \quad (5.3)$$

If some bands contain more informative spatial information the weights functions correspond to these bands can be modified to allow these band to have a higher contribution in the final gradient map. Median operator and supremum are examples of nonlinear operators.

5.2.1.3 Combination of Watershed Segmentation Maps:

In this approach, the gradient of each band is calculated independently from the other bands and instead of combining these gradient maps together the watershed transformation is applied on each band. Thus, we will obtain B band segmentation maps and these segmentation maps can be combined to obtain the final segmentation map.

One of the ways to obtain final segmentation map with relevant edges from the B segmentation maps that we have is to add these maps together, each of the segmentation maps is a binary map where the ones represent the edges and zeros represent the segmented region. Let W_λ be the watershed map for the band λ the following equation shows how to sum the watershed maps.

$$W = \sum_{\lambda=1}^B W_\lambda \quad (5.4)$$

To improve the final segmentation W a thresholding can be applied on the image as a result of summing binary maps, some points have the value of zero and this means that these points are not considered as a watershed line in any of the maps, on the other hand some point will have values vary from 1 to B. If the point has high value that means this point is a watershed line in many bands therefore, it can be considered as reliable waterline. One important point should be taken into consideration in this method we lose the information about the regions because adding watershed maps together would change the shape of the obtained region and these new obtained regions are needed to be checked and further closing and region labeling are required.

5.2.2 Using Segmentation in Classification Scheme

The improvement in the classification result is done by integrating the watershed segmentation result with the result obtain from the spectral classifier; in this section, a spectral-spatial scheme is introduced, this scheme is used to enable spectral-spatial classification on hyperspectral images using watershed segmentation.

Figure 5.4 shows a general flow chart of how this combined segmentation classification method can be applied. In the first step B-band image represent the hyperspectral image is subject two methods parallely, the first is pixel-wise classification in our case is done by SVM classifier, the second is segmentation done by watershed and this segmentation can be applied by any of the 3 earlier mentioned technique. This result of the segmentation maps where each pixel has the value of the region it belongs to or the pixel is a watershed pixel and has one value and this value is different from all the other regions. In case of applying watershed in MATLAB the final map will contain integer values where the zeros represent the watershed line and the rest are the indexes of the separated regions.

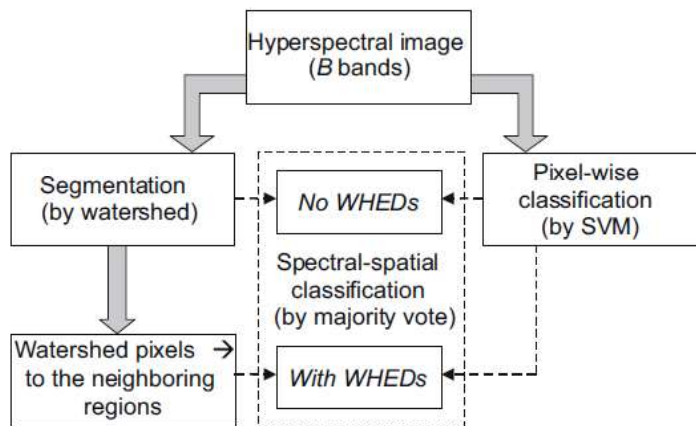


Figure 5.4: Flow Chart of The Proposed Segmentation and Classification Scheme.

There are 2 different ways to combine the spectral and spatial classification the first one is called no-WHEDs, in this method each region from watershed segmentation will have the label of the majority labels obtained from the spectral classification for this region and WHEDs will have their labels reserved. The additional part in with-WHEDs method is checking the labels of WHEDs and assigning each WHED to its nearest pixel,

5.3 Random Walker:

Random walker is a semi-automated algorithm used in image segmentation. In this approach, the image is treated as a graph made of vertices and edges. The edges assigned to a real weighted value represent the likelihood that a random walker standing on the first side of the edge will cross to the next side of the edge. To make it more convenient we can consider our graph as a network each intersection (vertices) represent a pixel from the original image and the lines connecting the intersections are the edges. As a semi-automated algorithm, the user must assign the points as reference points these points are called seeds. The main idea of the random walks is to calculate the probabilities for all points, that a random walker starting from these points will first reach a seed with a specified label. It has been proved in [49] that solving this probabilities issue is equal to the solution of Dirichlet problem [50] and the boundaries are at the location of the seeds to calculate the probability to reach the first seeds (each kind of seeds represent a class) we set these seeds to unity and the rest are to zero. There is a deep connection between the solution of discrete Dirichlet problem and the electrical potential in any circuits where the nodes represent the pixels and the resistor represent the inverse of weights and the seeds are the electrical sources. From this point on circuits theory is used to explain the random walker algorithm. Figure 5.5 illustrates how the circuit theory is applied to solve the Dirichlet problem of random walks. Assuming we have an image of 4 by 4 pixels with 3 different classes and 3 seeds. In 1.1 the image is represented as a graph, L1, L2, L3 are the seeds for class 1, class 2, class 3, respectively. In 1.2 the seeds are replaced with electrical sources and the edges with resistors which are equal to the inverse of weights. Next step is to calculate the electrical potential three times, one for each class. To do that required class seed is set to be the electrical source and make the other seeds as ground and after doing this process for all classes, each node has 3 potential values each represent the probability that a random walker starting from a node v_i will first reach this seed. Finally label of the maximum of the probabilities is assigned to the node v_i .

5.3.1 Exposition of the Algorithms:

In this section, the aspects of the algorithm starting from creating the weighted graph to establishing and solving the system equations are described.

We start with the graph [51] as the data was reassembled into a graph and all the procedure are directly applied on this graph. As mentioned earlier a graph is consist of pairs of vertices (nodes, V) and edges (E) and the graph is noted as $G=(V,E)$, $v \in V$ and $e \in E \subseteq V \times V$. An edge which is denoted as e_{ij} means it is spanning the vertices $v_i v_j$. There are two kinds of graph that are weighted and unweighted graph.

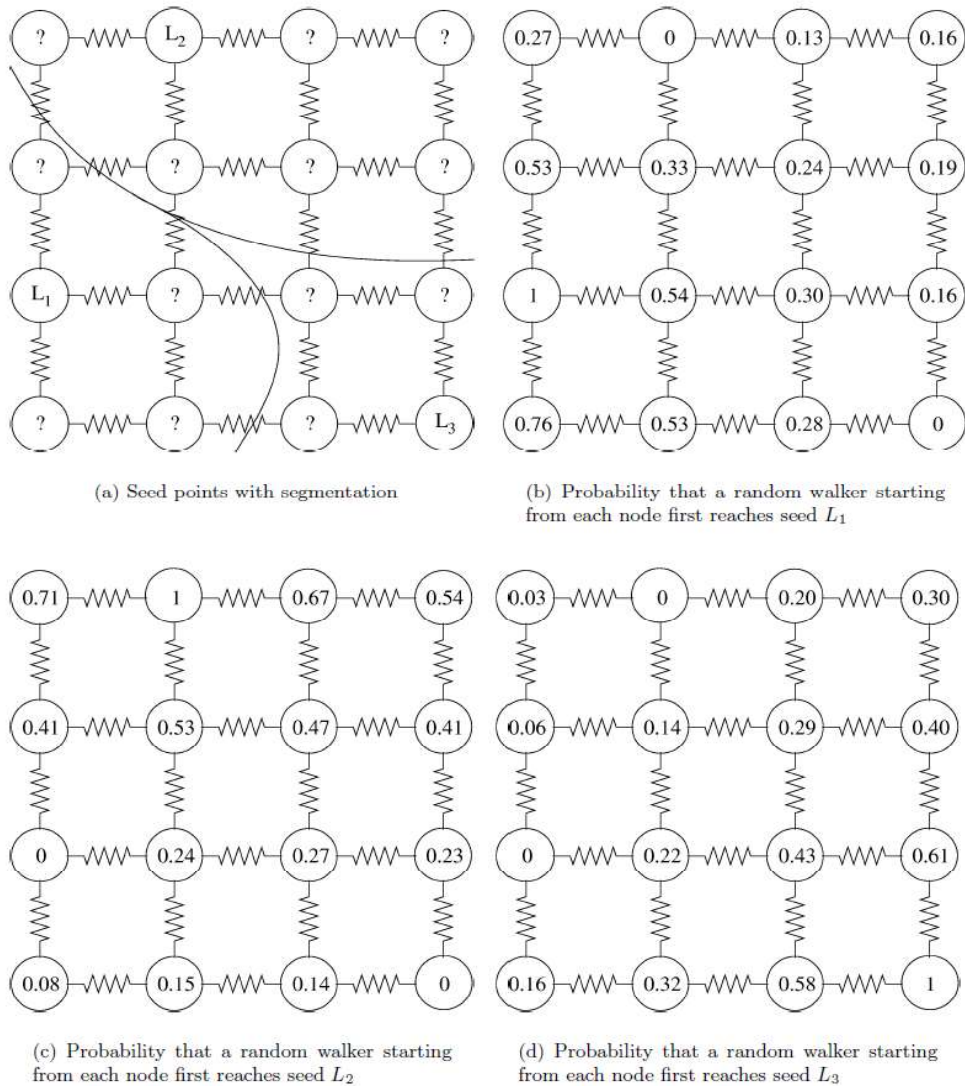


Figure 5.5: illustrates circuit theory to solve the Dirichlet problem of random walks.

In this approach, we are only dealing with weighted graph. Therefore each edge, e_{ij} have value $w(e_{ij})$ which is called weight. The degree of vertices can be defined as follows:

$$d_i = \sum w(e_{ij}) \quad (5.5)$$

which can be said as the degree of each vertices is the sum of the weights for all edges connected to this vertex. To make the weight as bias affecting the direction or steps of a random walkers $w(e_{ij}) > 0$.

Edge weight is a function that represents the relation between pixels of an image and the graph biases. The value of the weight is related to the changes in the image intensities. The idea of using weighted graph in image analysis is not a new concept and many ready weight functions can be found in the literature [52, 42]The most common used weight is in the following Gaussian weight.

$$w_{ij} = \exp(-\beta(g_i - g_j)^2) \quad (5.6)$$

where g_i and g_j are pixel intensities for neighbor pixels i and j , β is a free parameter of random walker algorithm. In this thesis, the Gaussian weighting function is used in random walker algorithm.

5.3.2 Discrete Dirichlet Problem:

Discrete Dirichlet problem can be considered as a complicated problem therefore only the concerned part of this problem is explained and for further details of Dirichlet problem can be found in [53],In [54]a convenient solution to our concerned part are explained. The following part is a review of this solution.

Discrete Laplacian matrix is defined as

$$L_{v_i v_j} = \begin{cases} d_{v_i} & , & \text{if } i = j \\ -w_{ij} & , \text{if } v_i \text{ and } v_j \text{ are adjacent nodes} \\ 0 & , & \text{otherwise} \end{cases} \quad (5.7)$$

L is a n by n square matrix where n is the total number of pixels (vertices) in an image and v_i . and v_j are row and column vertices (indexes) of the matrix, respectively. Discrete Laplacian matrix can be arranged according to the labels of each vertices as follows:

$$L = \begin{bmatrix} L_m & B \\ B^T & L_u \end{bmatrix} \quad (5.8)$$

In this arrangement, the first group is for labeled elements which contains the seed/marked vertices and the second group for the rest of the elements for unmarked vertices. V_m and V_u denote marked and unmarked vertices respectively.

It is noted that $V_m \cap V_u = \emptyset$ and $V_m \cup V_u = V$.

Define x_s as the probabilities that each point V_i belongs to each of the labels s . For instance, 4 class case x_i is the probabilities for V_i which is 4 by 1 vector that each row of it represent the probability of V_i belongs to one of the classes. Define the new function $Q(v_i) = s, \forall v_i \in V_m$ where $s \in Z, 0 < s < K$. Where K is the total number of the classes, Then the marked vector for each label can be defined as follows:

$$m_j^s = \begin{cases} 1 & \text{if } Q(v_i) = s \\ 0 & \text{if } Q(v_i) \neq s \end{cases} \quad (5.9)$$

The solution of combinatorial Dirichlet problem can be given from [54] as:

$$L_u X = -B m^s \quad (5.10)$$

Equation (5.8) is a symmetric sparse positive-definite system of linear equation, $|v_i|, 2|E|$ are the number of equation and the number of nonzero elements, respectively. As mentioned before the graph is a connected graph and for a connected graph, L_U is nonsingular [55] and therefore the solution to our system is granted to be exist and unique. the following system is used to obtain the potentials for all labels,

$$L_u X = -B M \quad (5.11)$$

Each column of M represents m^s and each column of X represent x^s , if K represent the number of labels $K-1$ number of equations are to be solved.

5.3.3 Theoretical Properties of the Algorithm

The properties of the algorithm have already been mentioned in the introduction part and in this section some propositions which have some practical consequences are given. First if interpolation is required to be achieved between the solution of an image

and the neutral solution, and this can be easily solved by adding constant to the weights of the image this situation usually happens when the image is poor. Second in case of independent random noisy at the level of the pixel the ideal weighting function should produce at the weight level multiplicative noise, as a result the expected potentials values in the presence of noisy should be equal to the expected potentials without noisy. Third in case of pure noisy or very close to pure noisy the segmentation obtained with Random Walker is the neutral segmentation. The following two properties are discrete analogues of properties of continuous harmonic functions [50] and they can be found by viewing the solution to the discrete Laplace above mentioned equation with the boundary conditions, by taking into consideration that each unlabeled point should satisfy the following condition

$$x_i^s = \frac{i}{d_i} \sum_{e_{ij} \in E} w(e_{ij}) x_j^s, \quad (5.12)$$

where x_j^s is a vertex and can be unlabeled pixel or seed.

1. Maximum/ minimum principle which states $\forall i, s$ the potential of $x_i^s \in [0,1]$.
2. The mean value theorem: The potential of each unlabeled node assumes the weighted average of its neighboring nodes.

Proposition 1: After the final segmentation, each node assigned to the label S according to the above-mentioned rule is connected through a pass of nodes to at least one of the label S seeds and the points in the path are also assigned to label S. another way to describe this proposition that the connected component through the final segmentation should contain at least one seed and all these connected points should have the label of that seed.

Proof: Any connected subset $P \subseteq V_u$ assigned to the labels must be at least connected to one of the seeds with the label S.

A block of matrix taken from the unlabeled points satisfactory equations can be written as:

$$L_p x_p^s = -R_p x_p^s, \quad (5.13)$$

Where $x^s = [x_p^s, x_p^s]$, and the matrix L as following:

$$L = \begin{bmatrix} L_p & R_p \\ R_{\bar{p}} & L_{\bar{p}} \end{bmatrix} \quad (5.14)$$

\bar{P} is the complement of P in V , in case $\{P = v_i\}$, $L_p = d_i$ and $-R_p x_p^s = \sum_{e_{ij} \in E} w(e_{ij}) x_j^s$,

If $x_p^s > x_p^f \forall f \neq s$ then $x_p^s - x_p^f > 0$ and $L_p^{-1} R_p (x_p^s > x_p^f) > 0$ then by definition of L the entries of R_p are non-positive, $L_p^{-1} R_p$ has nonnegative entries due to the fact the L is a M-Matrix and any diagonal sub-matrix form M-matrix is M-Matrix and the inverse of M-matrix have nonnegative entries therefore, some $x_i^s \in P$ are greater than $x_i^f \in \bar{P}$ and nodes not connected to P , are represented by 0 in R_p therefore to satisfy the above inequality some nodes in \bar{p} must be connected to P .

The rest 4 proposition have a common lemma this lemma is first mentioned (this lemma is referred to as common lemma) and later the rest of the propositions.

Common Lemma: for the following 3 random variables X , A and B such that $X = \frac{A}{B}$, $E[x] = 0$ if $E[A] = 0$ and $B > 0$. by the Hölder inequality it is proved that $E[A] = E[XB] \leq E[X]E[B]$. And $E[X] = E\left[\frac{A}{B}\right] \leq E[A]E\left[\frac{1}{B}\right]$. Therefore, $\frac{E[A]}{E[B]} \leq E[X] \leq E[A]E\left[\frac{1}{B}\right]$.

There is a relation between the potential solved in $L_u x^s = -B M^s$ and weight tree structure of the graph

For a node v_i the potential in the presence of a unit voltage source is given in [6, 26].

$$x_i^s = \frac{\sum_{TT \in TT_i} \prod_{e_{ij} \in TT} w(e_{ij})}{\sum_{TT \in TT_G} \prod_{e_{ij} \in TT} w(e_{ij})} \quad (5.15)$$

In graph theory, a 2-Tree is defined as a tree with one edge removed. TT_i is a set of 2-tree represent in the graph where through this 2-Tree a node is connected to a seed (labeled node), TT_G is the set of all possible 2 trees in the graph. $TT_i \subseteq TT_G \forall V_i$, (5.15) can be restated as the sum over the product of weights over all 2-Trees where the node v_i is connected to a seed and divided over the sum over all the 2-Tree in the graph the results is the potential obtained from solving (5.15) and it's neither practical

nor helpful to use the equation (5.11) to solve (5.15) due to the enormous number of 2-Tree in any image graph but it's helpful to prove some of the behavior of x_i^s with the usage of different weight functions. In the neutral case the potentials can be given as:

$$x_i^s = \frac{|TT_i|}{|TT_g|} \quad (5.16)$$

Now the 4 other propositions will be proved regarding x^s in different conditions. It is noted before that all the weights can be multiplied by a constant i.e. k and it wouldn't affect the result. It can be proved easily from (8.10) that both the numerator and dominator are divided on the same number K .

Proposition 2. In case of identical random distributed positive weights ($w_{ij} > 0$) the segmentation results are equal to the neutral segmentation results.

Proof. This is prove as mentioned earlier using the common lemma. New variable will be donated n_i^s, TT_c where n_i^s the neutral potential that the node v_i belongs to the class S and TT_c is the complement of TT_i in TT_G which means $TT_c \cup TT_i = TT_g$ and $TT_c \cap TT_i = \emptyset$. For brevity S_{TT_i} will be used to donate $\sum_{TT \in TT_i} \prod_{e_{ij} \in TT} w(e_{ij})$.

$$E[x_i^s - n_i^s] = E \left[\frac{S_{TT_i}}{S_{TT_i} + S_{TT_c}} - \frac{|TT_i|}{|TT_i| + |TT_c|} \right] \quad (5.17)$$

Since each of 2-Tree have the same number of edges which is $(n-2)$ and all the weights in this case are identical distrusted the sum of $|TT_i|$ are contained in S_{TT_i} . Let μ donate the mean of new variable distribution is this case the numerator of (8.12) can be written

$$\begin{aligned} E[S_{TT_i}(|TT_i| + |TT_c|) - |TT_i|(S_{TT_i} + S_{TT_c})] = \\ \mu|TT_i|(|TT_i| + |TT_c|) - |TT_i|(\mu|TT_i| + \mu|TT_i|) = 0 \end{aligned} \quad (5.18)$$

And due to the fact that all the weights are positive the dominator is strictly positive. The condition of the common lemma can be satisfied if the left-hand side equal to zero and subsequently $E[x_i^s] = n_i^s$.

Proposition 3. $E[x_i^s]$ equals the potential obtain by setting the weights to be equal the corresponding means, in case the weights were uncorrelated (not necessary independent)

Proposition 4. the same $E[x_i^s]$ can be obtained by replacing the weight w_{ij} with constants K_{ij} (not necessary that these constants are equal) as long as $w_{ij} = K_{ij}y_{ij}$ and y_{ij} are identically distributed random variables where $y_{ij} > 0$

Proposition 5. If $w_{ij} = k_{ij} + r$, k_{ij} are not necessarily equal and r is equal constant,
 $\lim_{r \rightarrow \infty} x_i^s = n_i^s$

5.3.4 Algorithm Summary:

Random walker can be applied by using the following steps:

1. Define the set of seeds (labeled nodes), if the training samples are available those can be used as seeds with K classes or the seeds should be chosen manually
2. Calculate the weights using the image intensity
3. Solve (8.4) for each label expect the final one which can be calculated from the following formulation $x_i^f = 1 - \sum_{s < f} x_i^s$ or potential for all classes can be solved directly from (8.5)

The final segmentation can be obtained by assigning the class with the highest potentials to the nodes or an alternative methods K -dimensional clustering technique can be applied on the potential vectors on each node

A Segmentation Example using Random Walker

The following example Figure 5.6 explains how the random walker algorithm works. In this example, the algorithm is used to apply segmentation on small synthetic data made up 9 pixels and 2 classes represented by 2 seeds (Seed 1 and seed 2). Other 7 pixels in the synthetic data are free pixels and they are needed to be assigned to one of the 2 groups. The network between the pixels is to help in clarifying the idea of the graph

The numeric solution of this problem consist of following steps:

- 1) Finding the weight between the adjacent points.
- 2) Writing the linear equation system.
- 3) Finding the solution of the linear equation system according to each seed.

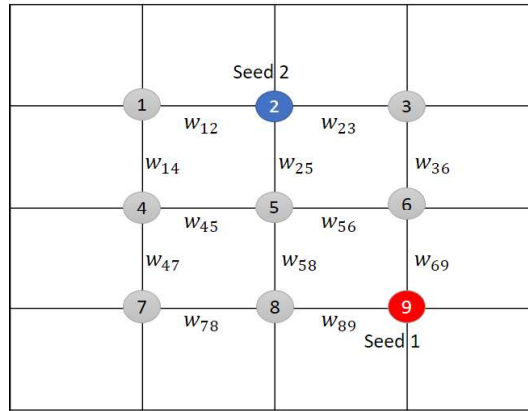


Figure 5.6: Random walker Numeric Segmentation Example.

The weight is a function that describe the gradient in intensities between adjacent seeds here it will be referred to the weight between pixel 1 and pixel 2 as w_{12} , these weights can be calculated using the Gaussian form (8.1) as follows: $w_{ij} = \exp(-\beta(I_i - I_j)^2)$ where I_i and I_j are intensities of pixels 1 and 2, respectively. As it can be noticed from this equation if I_1 and I_2 have close intensity values then w_{12} is almost one. If the values were close enough to each other these values show the probabilities that a random walker standing at pixel number i will move toward pixel number j having a bigger difference in intensity will reduce the probabilities that a random walker will move to this direction and this is what is called a biased graph. In this example, some random values are given to the weight function and this value must vary between 0 and 1.

A linear equation is written to each unlabeled pixel but here only the equation for pixel number 1 is explained. U is used as a function for all unlabeled pixels and L is the function for labeled pixels, then to obtain the equation for the first pixel $U(p_1)$ all the labeled and unlabeled pixels connected to p_1 must be written with their weight functions.

$$U(p_1) = w_{12}L(p_2) + w_{14}U(p_4)$$

By applying the same rules on all the other unlabeled pixels:

$$U(p_1) = w_{12}L(p_2) + w_{14}U(p_4)$$

$$U(p_3) = w_{23}L(p_2) + w_{36}U(p_6)$$

$$\begin{aligned}
U(p_4) &= w_{14}U(p_1) + w_{45}U(p_5) + w_{47}U(p_7) \\
U(p_5) &= w_{45}U(p_4) + w_{56}U(p_6) + w_{25}L(p_2) + w_{58}U(p_8) \\
U(p_6) &= w_{69}L(p_9) + w_{56}U(p_5) + w_{36}U(p_3) \\
U(p_7) &= w_{47}U(p_4) + w_{78}U(p_8) \\
U(p_8) &= w_{78}U(p_7) + w_{89}L(p_9) + w_{58}U(p_5)
\end{aligned}$$

By moving all the values of U to the left side of the equality and dividing equations by the weight of the seeds, these equations can be rewritten as follows:

$$\begin{aligned}
\frac{1}{w_{12}}U(p_1) - \frac{w_{14}}{w_{12}}U(p_4) &= L(p_2) \\
\frac{1}{w_{23}}U(p_3) - \frac{w_{36}}{w_{23}}U(p_6) &= L(p_2) \\
U(p_4) - w_{14}U(p_1) - w_{45}U(p_5) - w_{47}U(p_7) &= 0 \\
\frac{1}{w_{25}}U(p_5) - \frac{w_{45}}{w_{25}}U(p_4) - \frac{w_{56}}{w_{25}}U(p_6) - \frac{w_{58}}{w_{25}}U(p_8) &= L(p_2) \\
\frac{1}{w_{69}}U(p_6) - \frac{w_{56}}{w_{69}}U(p_5) - \frac{w_{36}}{w_{69}}U(p_3) &= L(p_9) \\
U(p_7) - w_{47}U(p_4) - w_{78}U(p_8) &= 0 \\
\frac{1}{w_{89}}U(p_8) - \frac{w_{78}}{w_{89}}U(p_7) - \frac{w_{58}}{w_{89}}U(p_5) &= L(p_9)
\end{aligned}$$

To solve these linear equations L functions for seed 1 and seed 2 are replaced with a value to get 7 variables and 7 equations which make it possible to be solved. First the probabilities of reaching the seed 1 are solved. To do that 1 is substituted in $L(p_2)$ and 0 is substituted in $L(p_9)$. For solving probabilities of reaching the seed 2, 1 is substituted in $L(p_9)$ and 0 is substituted in $L(p_2)$. By this concept the discrete Dirichlet problem are used to solve the equations system as follow:

$$\begin{bmatrix}
\frac{1}{w_{12}} & 0 & -\frac{w_{14}}{w_{12}} & 0 & 0 & 0 & 0 \\
0 & \frac{1}{w_{23}} & 0 & 0 & -\frac{w_{36}}{w_{23}} & 0 & 0 \\
-w_{14} & 0 & 1 & -w_{45} & 0 & 0 & 0 \\
0 & 0 & -\frac{w_{45}}{w_{25}} & \frac{1}{w_{25}} & -\frac{w_{56}}{w_{25}} & 0 & 0 - \frac{w_{58}}{w_{25}} \\
0 & -\frac{w_{36}}{w_{69}} & 0 & -\frac{w_{56}}{w_{69}} & \frac{1}{w_{69}} & 0 & 0 \\
0 & 0 & -w_{47} & 0 & 0 & 1 & -w_{78} \\
0 & 0 & 0 & -\frac{w_{58}}{w_{89}} & 0 & -\frac{w_{78}}{w_{89}} & \frac{1}{w_{89}}
\end{bmatrix}
\begin{bmatrix}
U(p_1) \\
U(p_3) \\
U(p_4) \\
U(p_5) \\
U(p_6) \\
U(p_7) \\
U(p_8)
\end{bmatrix}
=
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 1 \\
0 & 0 \\
0 & 1
\end{bmatrix}$$

A solution of the matrix equation above is 7 by 2 matrix and the first and second columns of the matrix are the probabilities that each unlabeled pixel belong to the group of seed 1 and seed 2 respectively. Each unlabeled pixel is assigned a label of (column index) the highest probabilities.

5.4 Extended Random Walker:

The Random Walker segmentation [42] has shown a great performance in many different fields including medical images and it has desirable theoretical properties. Random Walker is generally made as a semi-automated algorithm or in other words an interactive segmentation tool, that the algorithm cannot proceed without the interaction of the user. The user must select a few number of pixels from the processed image and assigns them to specific labels then the algorithm calculates as mentioned above the probabilities that a random walker start from each pixel will first reach one of the preselected pixels. Random walker algorithm has many desirable properties which are outlined in [42].

1. The solution of the probabilities is unique
2. The expected value of the probabilities for an image of pure noise, given by identically distributed (not necessarily independent) random variables, is equal to those obtained in uniform image
3. The expected value of the probabilities in the presence of random, uncorrelated weights is equal to the probabilities obtained by using weights equal to the mean of each random variable.

Despite these powerful and desirable properties. Random walker algorithm has some disadvantages as mentioned in [43].

1. There must be a seed in each segment.
2. The absolute intensities are not well employed instead only the intensity gradients were used.
3. The algorithm can proceed without the user intervene to select seeds.

These 3 disadvantages are considered as desirable properties in some segmentation tasks such as ignoring the absolute intensity can be helpful in some case where employing only the gradients can increase the robustness to quantization and decrease

the classification error in the same homogenous area in addition to prevent noisy caused by inversed and shifted intensity. However, it can become really impractical when it comes to images containing many disconnected pieces. In such case the user has to select seeds inside each disconnected piece and this is one of the main incentives to come up with the new Extended Random Walker Algorithm where instead of using user defined seeds for each disconnected area, the intensity model of an image can be obtained and this model is used instead of the user input. This intensity model can be calculated in different ways also can be calculated priori. In the explanation of the algorithm in [43] for simplicity and clarity they used image with only one channel and user seeds but this concept only to make it easier to explain but the algorithm can be applied in multi-channels and without user intervene as it's applied in this thesis.

The mixing between the spatial information and statistical information is not a quite new approach in the computer vision literature. And this is usually performed by adding new energy term to the total energy and applying minimization on the new energy function [56]. Some spatial algorithms are considered as conservative algorithm where it is not easy to mix between these algorithms and the density estimation priors such as watershed transformation [46]. The new achievement in Extended Random Walker algorithm is the ability of employing image priors to the affective old spatial algorithm Random Walker to have a new algorithm which can classify the disconnected area without the need of the user intervene.

5.4.1 Development of the Algorithm:

As the Random Walker algorithm, Extended Random Walker is also formulated on a weighted graph, and all the definition of the graph such as edges, degree and Laplacian matrix are valid for the Extended Random Walker algorithm therefore no need to mention these definitions in this section again.

5.4.2 Label Priors:

The probability density λ_i^s represent that the density at the node, v_i belongs to the intensity distribution of the classes g^s , and these probability densities are considered as nodewise priors, if we want to calculate the probability x_i^s for the node v_i belongs to g^s , and after assuming that the likely of all the nodes are equal then x_i^s can be written as follows:

$$x_i^s = \frac{\lambda_i^s}{\sum_{q=1}^k \lambda_q^s} \quad (5.19)$$

by using vector notation this equation. can be written as follows:

$$\left(\sum_{q=1}^k \Lambda^q \right) x^s = \lambda^s \quad (5.20)$$

Note Λ^s is a diagonal matrix where the values of λ^s on the diagonal and the rest of the elements are zero.

The following formulation can be used to calculate the minimum energy distribution for our new aspatial space.

$$E_{aspatial}^s(x^s) = \sum_{q=1, q \neq s}^k x^{qT} \Lambda^q x^q + (x^s - 1)^T \Lambda^s (x^s - 1) \quad (5.21)$$

A total energy function can be written by combining the spatial energy function and the aspatial energy function by using the free parameter γ .

$$E_{Total}^s = E_{Spatial}^s + \gamma E_{Aspatial}^s \quad (5.22)$$

If we considered the case where we don't have any seeds, which means all the nodes are label free (all x_i are free nodes) we can calculate x^s which satisfy the minimum of the total energy function as:

$$\left(L + \gamma \sum_{r=1}^k \Lambda^r \right) x^s = \gamma \lambda^s \quad (5.23)$$

The Laplacian matrix in Random Walker is a singular matrix and therefore it cannot be solved without having the user seeds but in the case adding the diagonal matrix which is strictly positive definite to the original Laplacian matrix will guarantees that the new combined matrix is positive definite. In this way, we can circumvent the semi-automated Random Walker algorithm into an automated algorithm, however the user seeds (if desired) can be used in this algorithm by solving the following system:

$$\left(L_u + \gamma \sum_{r=1}^k \Lambda_u^r \right) x_u^s = \gamma \lambda_u^s - B f^s \quad (5.24)$$

If we compare the new lattice we obtained with the one obtained from the Random Walker we can easily see that the main different represented in having extra nodes. These are the label nodes therefore there are extra one node for each label in the image and these nodes are referred as floating nodes.

Each floating node is connected to all the other nodes and instead of using the weight function on the new edges between the floating nodes and the normal nodes as it's mentioned in [12] the values of $\gamma \lambda_i^s$ are used. The weight of each new edge is equal to the relevant $\gamma \lambda_i^s$. The new lattice is depicted in Figure 5.7.

By making comparison between random walker and extended random walker we can figure out that they are the same with $B f^s = \lambda^s$ and L_u is simply the Laplacian matrix of the new lattice with the addition to the diagonal matrix $(\gamma \sum_{r=1}^k \Lambda_u^r)$. We obtain the same results in case we apply the extended random walker with the incorporation of priors or if we directly applied the Random walker on augmented graph. It's more convenient to consider the extended random walker as augmented graph since we can treat it in the same frame work as the random walks and all the proofs given in [42] can be considered applicable on the extended version therefore the robustness and the behavior of the Random Walker also apply in the extended version (when the priors are applied) [12].

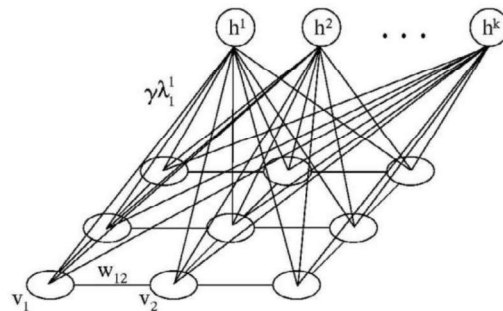


Figure 5.7: The Use of Intensity Priors is Equivalent to Using K Labeled Floating Node That Correspond to Each Label and Connected to Each Node.

5.4.3 Algorithm Details:

This algorithm can be described in the following steps:

1. A prior model describes the label intensities in some cases it can be available for certain images or if not, it can be obtained with a user interaction via estimation, and from this prior model the probabilities λ_i^s describes that each node v_i belongs the class S. This is the general first step in case of using SVM and these probabilities can be obtained directly from the classification maps.
2. The second step is to apply random walker algorithm on the image without calculating the segmentation which means only the graph from the original image is created and the weights are calculated
3. Solving equation. (5.24) in case of very large image can be difficult. Checking the array-size limit in case of using MATLAB, Octave etc. can be useful. This calculation needs to be done to each class g^s and we can use the unity sum condition to calculate only k-1 equations and the last one can be obtained as follow $x_i^k = 1 - \sum_{s < k} x_i^s$.
4. Each node v_i have k probabilities representing that this node belongs to the class S. The simplest rule and the most used one is to take the highest probabilities and assign its label to the node. another way to assign labels is apply some clustering technique on all nodes probabilities.

5.4.4 Prior Model:

In this approach, the prior model is easily obtained using training data and it is used to obtain probabilities maps. The used SVM library in the experiments is LIBSVM [57] which have many options. The user can choose to get a probabilities map as classification result. In the general case, this prior need to be estimated and there is many different ways to obtain priors. In [58]many helpful methods have been provided. alternatively, Gaussian kernels can be calculated for each class and a normalized histogram can be created and the probabilities can be found simply for each intensity values of an image.

5.4.5 Choosing Weight

There are many available functions to create weights between image intensities. In [59] various weight functions and their proper use are mentioned. Apart from well known a Gaussian weight function ubiquitous function is very useful.

$$w_{ij} = e^{-\beta(I_i - I_j)^2} \quad (5.25)$$

In practice, two variables can be added and the ubiquitous function becomes as follow:

$$w_{ij} = e^{\frac{\beta}{\rho}(I_i - I_j)^2} + \epsilon \quad (5.26)$$

where ϵ is a small constant and the value of it might be around 10^{-6} , ρ is a normalization function and it is equal to the maximum difference between the intensity in the image. With this adjustment to the ubiquitous function we make sure that none of the weights are exactly equal to zero instead the minimum weight is equal to ϵ . Another advantage is to keep β relevant to images with different contrast and quantization.

5.4.6 Numerical Solution

This algorithm does not differ much from the original random walker in context of computational hurdle. It has larger sparse, symmetric and positive definite system of linear equations. This equation can be solved using direct methods but it may include high memory consumption and in case of large images this cannot be the best way to use. Instead, iterative methods can be used some of these methods are mentioned in [60] such as preconditioned conjugate gradient is more appropriate to solve the linear system of a large image due to its lower memory consumption and parallelization capability.

5.5 Comparison between ERW and Watershed

The following table provide a brief comparison between the two proposed spatial algorithms.

Table 5.1: Comparison between ERW and Watershed.

EWR	Watershed
No need for any image filtering procedures	Without filtering over-segmentation will occur
Seeds or labeled samples in the image can be used to enhance the result	no seeds or labeled samples can be integrated to the algorithm
The relation between the spatial and spectral classification can be calibrated using the free parameter γ , it's very easy and simple to control this relation	To calibrate the relation between the spatial and the spectral new filtering procedure should be conducted on the image and its very complicated and hard to control this relation
Can be applied directly or by using feature extraction	Can be applied directly or by using feature extraction
The spectral classification part should provide probability or probability density for each class	The spectral classification part can provide only the labels for all samples
The spatial classification cannot be applied without the integration of the classification result	Spatial and spectral classification can be applied separately and later the results can be integrated together



6 MODIFIED ERW

Two different approaches were checked here to improve ERW performance the validity of these methods was checked only by doing limited number of experiments, but it still worth to mention these methods here. 3 different ways will be mentioned here, all of them focus on the probabilities maps obtained from SVM.

6.1 Priority for Large Classes

This method is helpful when bigger classes have higher priority, to give a higher priority to bigger classes. first of all, SVM is applied on the data then the classification result from SVM will be used to calculated occurrence of each class to the total number of classes as follow:

$$H_{c_i} = \frac{\text{number of } c_i \text{ sampels obtained from SVM}}{\text{Total number of Samples}} \quad (6.1)$$

Where H_{c_i} is the percentage of the total occurrence of the class i . Then the probabilities map will be modified as follow: each column of the SVM probabilities map represent a class, each column will be multiplied by the respective H_{c_i} . So, if the class i appeared a lot in the result H_{c_i} will have a high value, therefor it will increase the effects of this class in the spectral-spatial classification

6.2 Priority for Small Classes

The method is the opposite of the above-mentioned method, we can use this one when small classes have higher priority in the experiments, here also need to calculate H_{c_i} the percentage of the total occurrence for each class and instead of multiplying it directly by probabilities map, instead a new value will be calculated as follow:

$$\bar{H}_{c_i} = 1 - H_{c_i} \quad (6.2)$$

By using equation (6.2) high values will be related to small classes and then each column of the SVM probabilities map will be multiplied by the relevant \bar{H}_{c_i}



7 THE DATA USED IN THE EXPERIMENTS

- synthetic data
- Indian Pines
- Salinas

7.1 synthetic data:

this is a grayscale image with Gaussian noisy, this image will first be used instead of hyperspectral, because this image consists of only one band it's easier to apply EWR approach on it. Dealing with such image usually doesn't require complicated algorithm and a high accuracy can be reached by using simple algorithms such as maximum distance to mean. This image consists of 2 classes, Figure 7.1 shows this image and two alternative training samples or seeds.

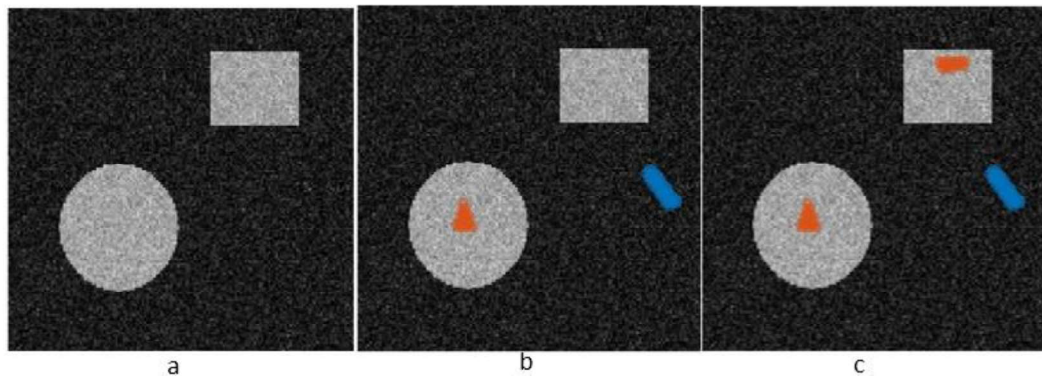


Figure 7.1: A represents the original image with noise, B represents the original noisy image with the location of the seeds, C is the same as B with extra seeds in the 2-separated area.

7.2 Indian Pines

This data was collected using AVIRIS sensor scanning the Indiana pines in the north-western Indiana, the collected data is 145×145 with 224 spectral bands with the wavelength range $0.4 - 2.5 \times 10^{-6}$ meters in this experiment we are using the corrected

version of Indiana pines which consist of 145*145 pixel and only 200 bands, the excluded bands are bands covering the region of water absorption. This data is available through Pursue's univeristy MultiSpec site.

This data consists of 16 class and background the total number of samples is 21025, the number of background samples is 10776.

The following table contain the ground truth table of the 16 classes and their respective number

Table 7.1: Indian Pines Groundtruth classes and their respective samples number.

#	Class	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265

15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93

In this experiment, there are 2 kind of training samples randomly selected samples which represent around 4% of the original Image and neighboring pixels.

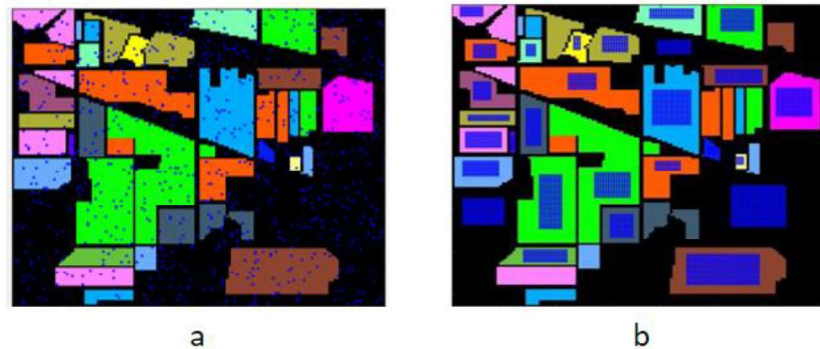


Figure. 7.2 Indiana Pines ground truth with the position of the random and neighboring samples

In the real-life application, it's hard and expensive to collect randomly distributed samples but these randomly selected samples can give a better performance because they cover a bigger range from each class. the random distributed training samples and the real-life training samples will be referred to as random samples and real samples respectively. Figure 7.2 shows the distribution or the position of the 2 kinds of samples.

7.3 Salinas scene:

This scene was collected as well with AVIRIS sensor over Salinas valley, California USA. Like the Indiana pines this scene contain 224 bands and 20 water absorption were discarded, in this case bands: [108-112], [154-167], 224. It includes vegetables, bare soils, and vineyard fields. Salinas groundtruth contains 16 classes and background occupies 56975 pixels.

Table 7.2: Salinas Scene Groundtruth classes and their respective samples number.

#	Class	Samples
1	Brocoli_green_weeds_1	2009
2	Brocoli_green_weeds_2	3726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807

Figure 7.3 shows the ground truth Salinas scene with the training samples, these samples were randomly selected and they represent approximately 4 percent of the total number of samples in the image and these samples are not randomly distributed instead they have neighboring relation among each class More information about this

scene, the size of this image is 512*217 pixel and the real number of bands is 224 but here the corrected version will be used, the corrected version has only 204 bands.

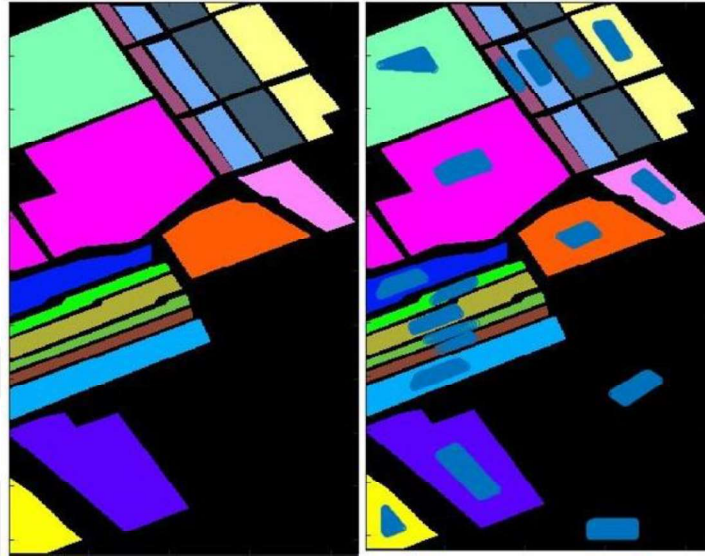


Figure 7.3: Salinas scene Groundtruth and Training Samples.



8 EXPERIMENTAL RESULTS:

In this part, Indian Pines and Salinas datasets were used to evaluate the robustness and the reliability of the proposed algorithms, where the synthetic data will be used only to illustrate the way these algorithms works. This synthetic data is a grayscale image; however, the proposed algorithms can handle it as well. To evaluate the results the overall accuracy, average accuracy, kappa coefficient and confusion matrix will be calculated. These measurements are well known and used in most of the literature work in this field. A brief definition will be introduced here. Overall accuracy is simply the ratio between correct classified pixels and the total number of pixel in the image, it's similar to the average accuracy, but in the later mentioned the correct classified pixels for each class will be divided on the number of pixels in each class then the average will be calculated, Kappa coefficient is a measurement of agreement between two variable, which compares the observed accuracy with the expected accuracy it's a good static to evaluate the classifier itself and to compare between different classifier. Covariance matrix compares the obtained result with the ground truth and provides a detailed information about each class, for example the number of misclassified pixel, number of correct classified and further information about which class misclassified data belongs to.

8.1 Synthetic Data

As this data is a one band image not a hyperspectral image, not all the above-mentioned statics need to be calculated. First a spectral classifier will be applied on this data, the training samples marked in Figure 7.1.b will be used. SVM is basically a binary classification algorithm so, it can be applied directly on this gray scale 2-classes image. As it can be shown in Figure 8.1. the result obtain by applying SVM is quiet good result with an accuracy up to 98%, but the noise affects is obvious in the background class the misclassification was caused by the noise in the real image. SVM gives classification results upon the pixel intensities only despite any other factors which can help in identifying whether this pixel belong to a specific class or not. 98% overall accuracy can be considered as a very high accuracy but in this case, we are dealing

with a simple syntactic data so such a result can be obtained easily by using any kind of simple classifier.

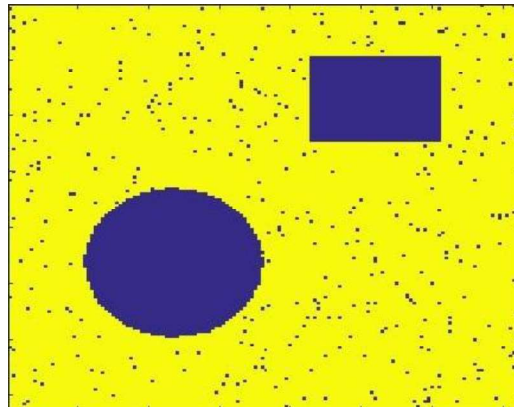


Figure 8.1: SVM Classification result on Synthetic Data.

To show the differences between RW and ERW, segmentation using RW and two different training samples or as they called in the RW algorithms two different seeds groups will be used. In Figure 7.1, there are two different seeds group, group b contains smaller amount of seeds and doesn't have seeds in all separated areas, while group c has a bigger number of seeds and these seeds are distributed in all the separated areas.

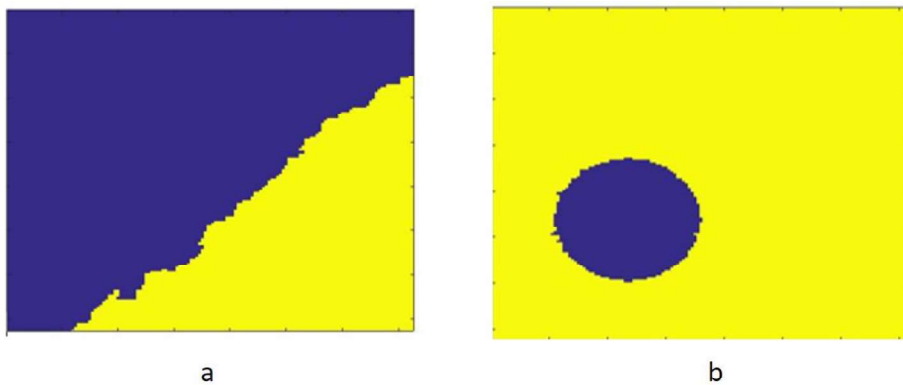


Figure 8.2: RW segmentation result. a) is the result of using c-seeds group. b) is the result of using b-seeds group.

In Figure 8.2-b, the segmentation result identified only the circular part of class 2 because there is no seeds in the other part of group 2. This is what makes RW not applicable to real life images where there are many separated parts belongs to the same class and in order to solve this problem a seed in each separated class is required. In

Figure 8.2-a seeds were distributed in all separated areas, even though RW failed to give a good segmentation, the main reason behind this failure is, that this image is a gray scale image with a gaussian noisy and the intensity between the 2 classes are close to each other, which makes it hard to separate between the classes using only spatial algorithm. In ERW we need to apply both spectral intensity classifier and spatial classification. The presented SVM results will be used to make classification map for this image.

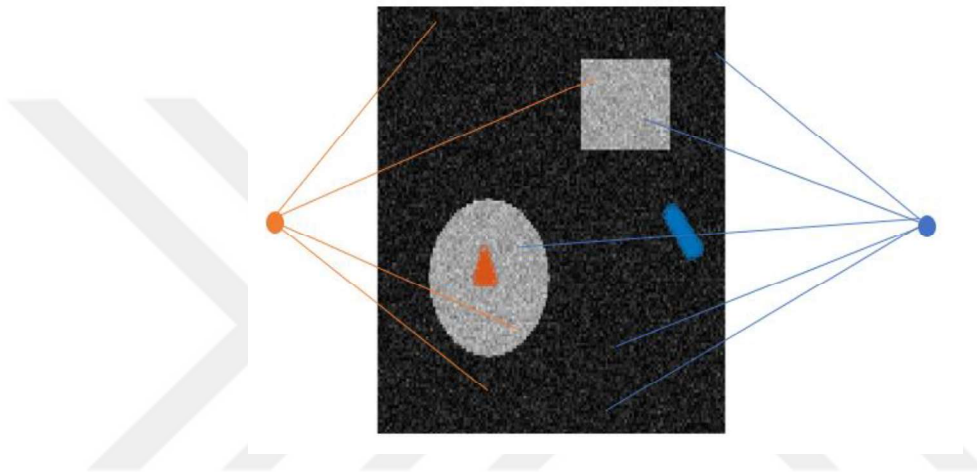


Figure 8.3: ERW illustrating. on synthetic Data, the 2-floating red and blue points represent the class labels.

For this syntactic data, we will get 2 maps, one for each class and the number of points in each map is equal to the total number of pixels (21025).

These maps are used later in the ERW. Each map will be used as an extra seed. To be represented in an easier way Figure 8.3. illustrate EWR in this image.

In RW algorithm the intensity gradient between neighboring pixels are calculated and used as weights in the Laplacian matrix. In ERW in addition to these weights there are extra points represent the classes, like the blue and red points in Figure 8.3. These points are connected to all the pixel in the image, since these points are not pixels, therefore the gradient intensity cannot be calculated among these points and the pixels of the image, instead the probability obtained by the spectral classifier will be used here as already explained in EWR algorithm, the result of SVM will be used in cooperation with RW to give better classification result.

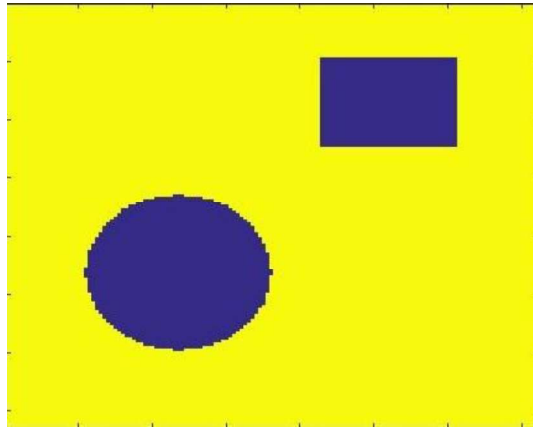


Figure 8.4: Classification Result using ERW and b-seeds group.

By using both spectral and spatial classification we can overcome some of the misclassified points in the homogeneous area, as it can be seen in Figure 8.4 the classification results are more accurate than the results obtained via SVM. What make ERW overcome RW is the ability of identifying pieces from the same class without the necessity of having seeds inside of each separated piece. By comparing the results obtain via SVM and ERW we can see that ERW with a good spectral classification method can give a quit good result and overcome misclassification in homogeneous areas.

8.2 Hyperspectral Image Classification

In this part, a lot of different experiments were applied on Indian pines data set and to verify the generality of these algorithms, the results of some experiments on Salinas scene will be briefly mentioned.

Indian Pines data set consist of 16 classes and background, so we divided our work into 2 different parts, first part 17 classes including background, second part the background were neglected and only 16 classes were studied. Figure 8.5 shows the diverse options to apply these algorithms.

As it can be seen in Figure 8.5 spatial-spectral classification can be applied directly on the hyperspectral images or it can be applied after doing some feature extraction such as PCA.

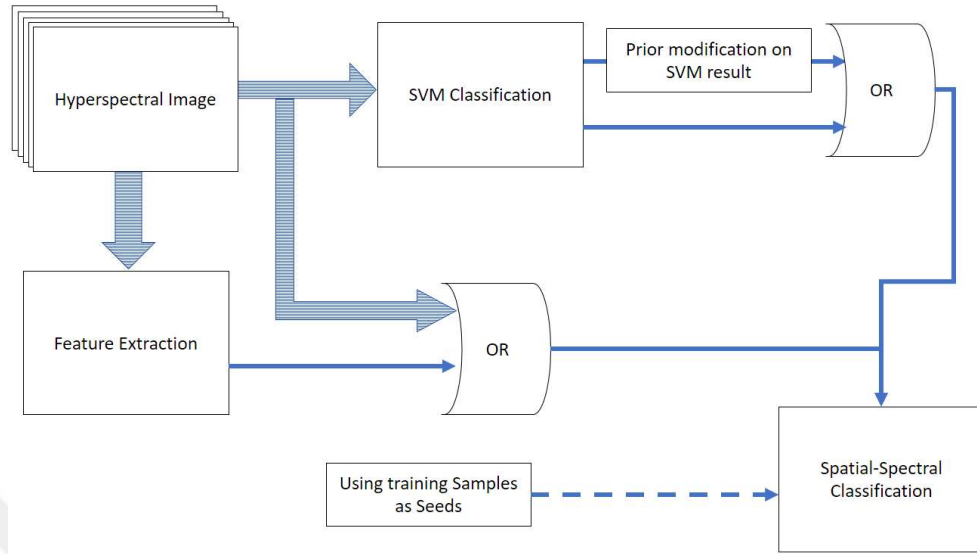


Figure 8.5: Flow chart of the diverse way to apply ERW.

8.2.1 Spectral classifiers:

In these following sections, all the experiments are done by using Indian pines real life training samples, unless the opposite is mentioned.

8.2.2 KNN:

As KNN considered one of the easiest spectral classification algorithms it will be used first on hyperspectral data to compare the results with the one obtained using SVM. For KNN experiment 4 different values for K will be used, $K = \{1,3,5,7\}$; here KNN will be applied directly on hyperspectral data.

Table 8.1: K-NN Classification result on Indian Pines.

17 Classes			16 Classes		
K	OA	Kappa	K	OA	Kappa
1	55.94%	0.465	1	68.87%	0.644
3	53.96%	0.443	3	66.78%	0.619
5	53.45%	0.437	5	65.87%	0.608

7	52.88%	0.430	7	65.39%	0.603
---	--------	-------	---	--------	-------

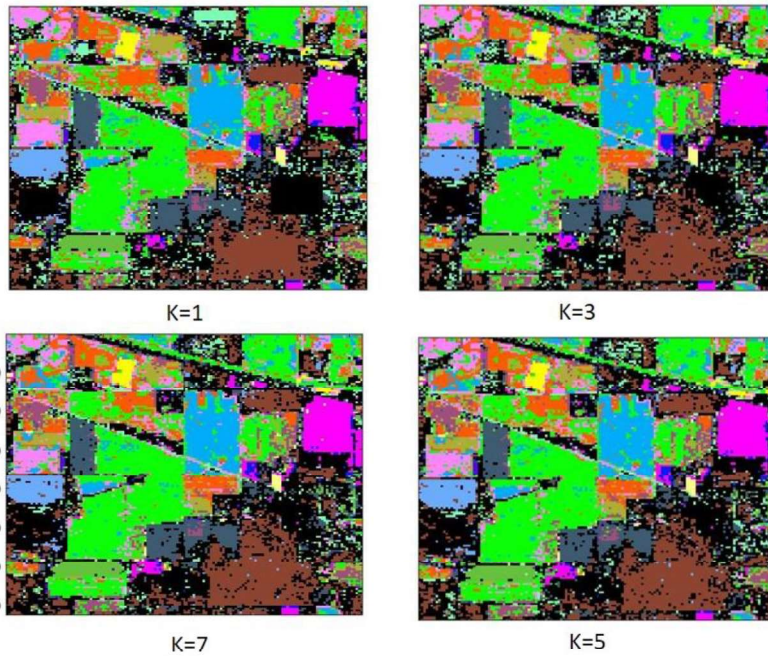


Figure 8.6: KNN classification result on Indian pines 17 Classes.

there is some literature about using feature extraction technique to improve the result of the KNN classifier, but here only the direct application of KNN on Hyperspectral data will be taken into consideration.

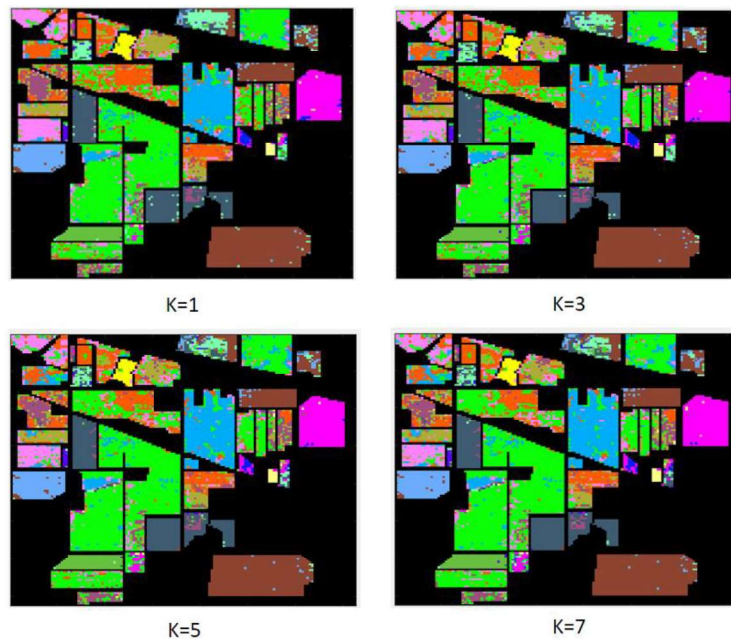


Figure 8.7: KNN classification result on Indian 16 Classes.

By comparing between Figure 8.6 and Figure 8.7 it can be noticed that a lot of misclassification is caused by the background class (the black colored class). the results obtain here are not going to be used later in the Spectral-Spatial classification algorithm and it's just to emphasis on the robust SVM algorithm by comparing the 2 results.

8.2.3 SVM

Before applying SVM on hyperspectral data, the data need to be Normalized in order to get a better classification results, here the data is normalized between $[-1, +1]$. LibSVM is used in all SVM experiments done in this work here only the neighboring samples will be taking into consideration and later the result of random training samples for spectral-spatial classification will be introduced.

Two different Kernel Functions will be used here. First one is Linear SVM, second one is radial basis function. Linear SVM has only one parameter which can be adjusted, which is complexity, while radial basis function has two parameters complexity and gamma.

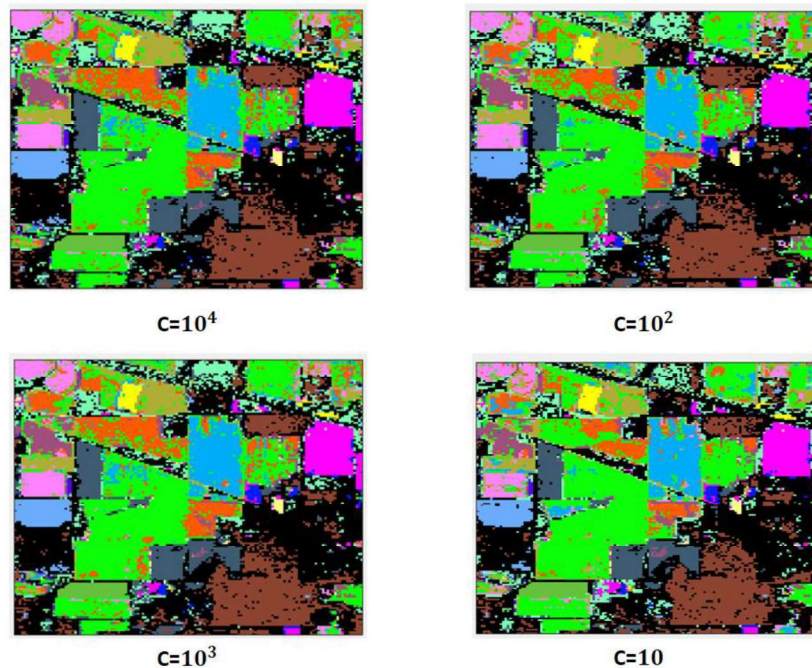


Figure 8.8: SVM Linear Function 17 classes Classification Results Neighboring Samples.

to obtain a better evaluating of the SVM classifier performance SVM was not only applied on the full hyperspectral data, but it was applied on the training data in order to get the training accuracy and statics, then SVM classifier was applied on the testing samples. In these experiments, all the sample from the real image excluding the training samples were taken as testing samples. From the result in table 8.2, it can be seen that, the parameter C in the linear SVM plays a major role. The parameter C or as it called the penalty factor, can be used to control the trade-off between how complex the decision boundary or decision rule is supposed to be and between the error frequency.

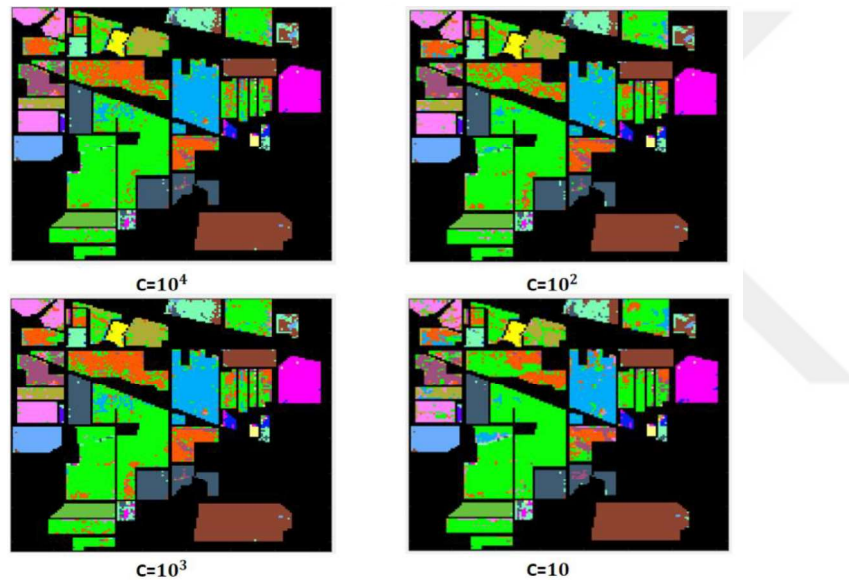


Figure 8.9: SVM Linear Function 16 classes Classification Results Neighboring Samples.

By comparing classification results for 16 classes and classification result for 17 classes, it's obvious that the background can be considered as problematic class and because it occupies a large space of the image, even more than 50% percent of the image is background. This should be taken into consideration while comparing the results between 16 and 17 classes case and not only the overall accuracy but the kappa and average accuracy as well. In this case for example if the whole image was classified as background we will get an overall accuracy equal to $10776/21025$ (number of background pixels / total number of pixel) $\approx 51,25$, but average accuracy will be equal to $1/17$. Table 8.2. show the accuracies obtained using different C, complexity variable in Linear SVM for both 16 and 17 classes and by using the

neighboring training group which has 3403 training samples in 17 classes case and 2648 training samples in 16 classes case Figure 8.8 and Figure 8.9 show the classification result for 17 and 16 classes respectively.

Table 8.2: Linear SVM Classification result on Indian Pines.

17 classes case									
	Training Result			Testing Result			Full Image result		
C	Overall Acc.	Average Acc.	kappa	Overall Acc.	Average Acc.	kappa	Overall Acc.	Average Acc.	kappa
10	88.77 %	88.92 %	0.87 2	53.79 %	62.73 %	0.41 6	59.46 %	70.48 %	0.50 3
10 ²	94.65 %	96.52 %	0.62 3	56.24 %	66.23 %	0.56 0	62.45 %	75.29 %	0.53 7
10 ³	97.20 %	98.70 %	0.63 2	56.85 %	66.92 %	0.57 0	63.38 %	76.34 %	0.54 8
10 ⁴	98.54 %	99.53 %	0.63 7	57.11 %	66.71 %	0.57 5	63.80 %	76.39 %	0.55 3
16 classes case									
	Training Result			Testing Result			Full Image Result		
C	Overall Acc.	Average Acc.	kappa	Overall Acc.	Average Acc.	kappa	Overall Acc.	Average Acc.	kappa
10	93.55 %	93.36 %	0.92 6	64.07 %	67.51 %	0.57 9	71.79 %	75.85 %	0.58 4
10 ²	97.76 %	97.95 %	0.97 4	68.53 %	69.60 %	0.63 2	62.45 %	75.29 %	0.53 7

10^3	97.20 %	98.70 %	0.63 2	56.85 %	66.92 %	0.57 0	63.38 %	76.34 %	0.54 8
10^4	100%	100%	1	70.08 %	70.52 %	0.65 1	77.91 %	79.81 %	0.65 5

Now the radial basis kernel function will be used. Radial basis SVM has two parameters which can be tuned to get a better classification result. These parameters are complexity C and Gamma. The C has the same effect as the one mentioned in linear kernel. Gamma defines the influence of the training samples; low values means that training samples have far influence and high value means a close influence.

Figure 8.10 and 8.11 shows the result obtained from RBF SVM for 17 and 16 classes respectively. Both 16 and 17 classes results in RBF are better than the results obtained earlier using Linear SVM. 4 alternative values for the parameters C and γ were applied.

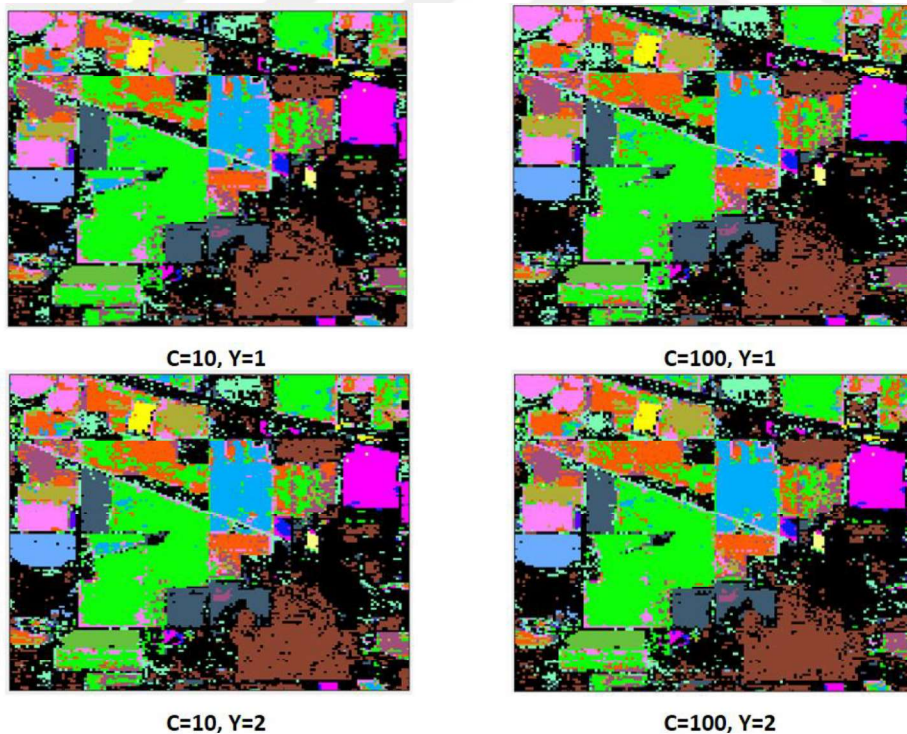


Figure 8.10: SVM Radial Basis Function 17 classes Classification Results Neighboring Samples.

Table 8.3. show the accuracy obtained from Radial Basis Kernel SVM. By comparing between the different result for both linear and Radial Basis function, the Radial Basis SVM is more reliable and has a better classification accuracy, however there are more kind of Kernel SVM for example polynomial, sigmoid, or by using pre-computed kernels. a lot of studied focused on how to improve this accuracies by finding different method focusing only on the spectral classification. Later the concept of cooperating the spatial and spectral classification was introduced.

Here you will consider the highest accuracy reached in SVM as the optimal spectral classification result and this result will be used in the spatial classification.

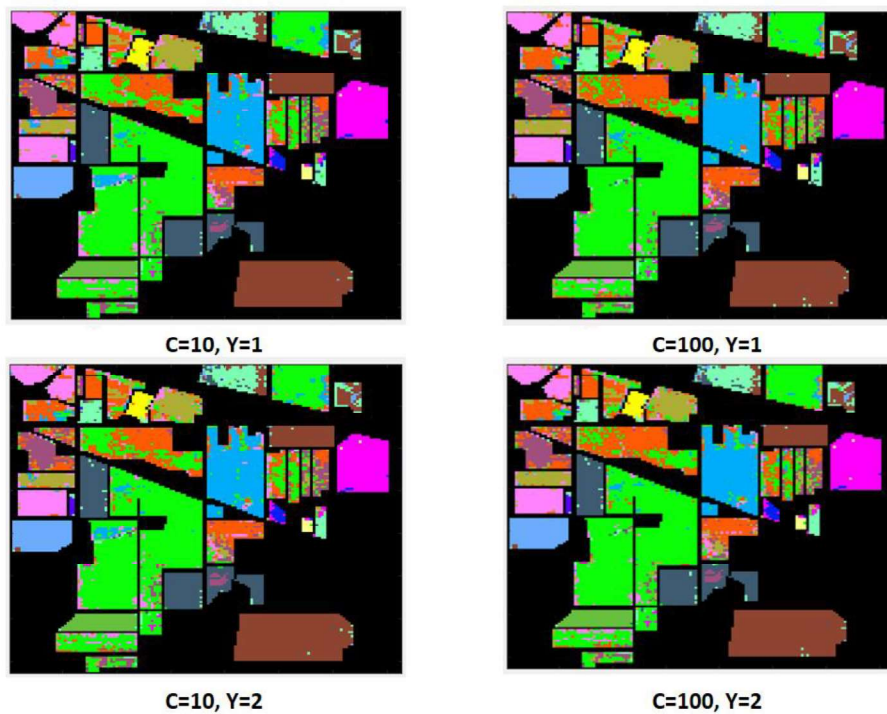


Figure 8.11: SVM Radial Basis Function 16 classes Classification Results Neighboring Samples.

Table 8.3: RBF-SVM Classification result on Indian Pines.

17 classes									
	Training Result			Testing Result			Full Image Result		
C, λ	OA	AA	K	OA	AA	K	OA	AA	K

10,1	93.32%	95.36%	0.625	56.68%	61.95%	0.556	62.61%	72.02%	0.534
10 ² , 1	98.56%	99.47%	0.647	58.38%	64.11%	0.585	64.88%	74.71%	0.564
10,2	95.76%	97.50%	0.630	56.87%	60.94%	0.563	63.17%	71.90%	0.540
10 ² , 2	99.61%	99.89%	0.642	57.53%	62.13%	0.578	64.34%	73.35%	0.556
16 classes									
	Training Result			Testing Result			Full Image Result		
C,λ	Overall Acc.	Average Acc.	kappa	Overall Acc.	Average Acc.	kappa	Overall Acc.	Average Acc.	kappa
10,1	97.35%	97.49%	0.966	65.20%	67.04%	0.597	73.62%	76.62%	0.601
10 ² , 1	99.96%	99.94%	0.999	69.68%	68.84%	0.649	77.61%	78.61%	0.652
10 ³ , 0.1	99.77%	99.81%	0.997	73.27%	72.32%	0.690	80.21%	81.23%	0.693
10 ⁴ , 0.1	100%	100%	1	73.69%	72.39%	0.695	80.58%	81.34%	0.698

All the results obtained via different kind of SVM are acceptable, but it's obvious that, there is a lot misclassification inside each separate homogenous area this misclassification caused like the salt and paper noisy effect. This noisy alike effect wasn't that sever intense but in hyperspectral images, this affect is more visible. This misclassification cannot be solved completely for all the images still the result can be enhanced and as mentioned earlier one of the methods to enhance the quality of a classification obtained from a spectral classifier is to integrate a spatial information through a spatial classifier into the result.

8.2.4 Feature extraction:

Before applying spatial classifier on the spectral result PCA will be applied on our image as a feature extraction method. PCA is usually used in image compression application. In hyperspectral image, each band doesn't contain a reliable spatial information in contrast if one layer was shown it will appear like very noisy image. Figure 8.12 shows the first principle Indian pines component after applying principle

component analysis. One of the most basic definition of hyperspectral image is transformation methods used to make the data easier to visualize and explore by emphasizing on the variation to extract the strong patterns in the image. Later, PCA result will be used in the coming experiment. Only the first 6 PCA component are area of interest specially the first PCA component. The following shortened forms will be used to refer to the first PCA component respectively 1st PC, 2nd PC, 3rd PC, etc.



Figure 8.12: First Principle Component from PCA Transformed Indian Pines

Feature extraction can be used as well with the spectral classification to overcome the dimensionality curse. Transferring hyperspectral data from a high dimensional space into a smaller space, while reserving most of the information in the data is a required advantage, because it can help in dealing with a very large data with limited computer resources. The studied images in this experiment are relatively small in the spatial size, not spectral size, the spectral is huge for example the Indian pine is 145*145 pixel is the spatial size and 200 band is the spectral size, so even with limited computer resources these data can be handled efficiently, but feature extraction is used here not only to make the data easier to handle but to make the spatial feature easier to extract.

In the next section the spectral and spatial classification will be applied with using feature extraction and without. it

8.2.5 Spatial classification:

8.2.5.1 Watershed:

Here the spatial classification will be used to improve the spectral classification. First, Watershed algorithm will be applied. Watershed is an easy algorithm to use, but in hyperspectral images it has some drawbacks. These drawbacks are mainly related to extracting the spatial texture.

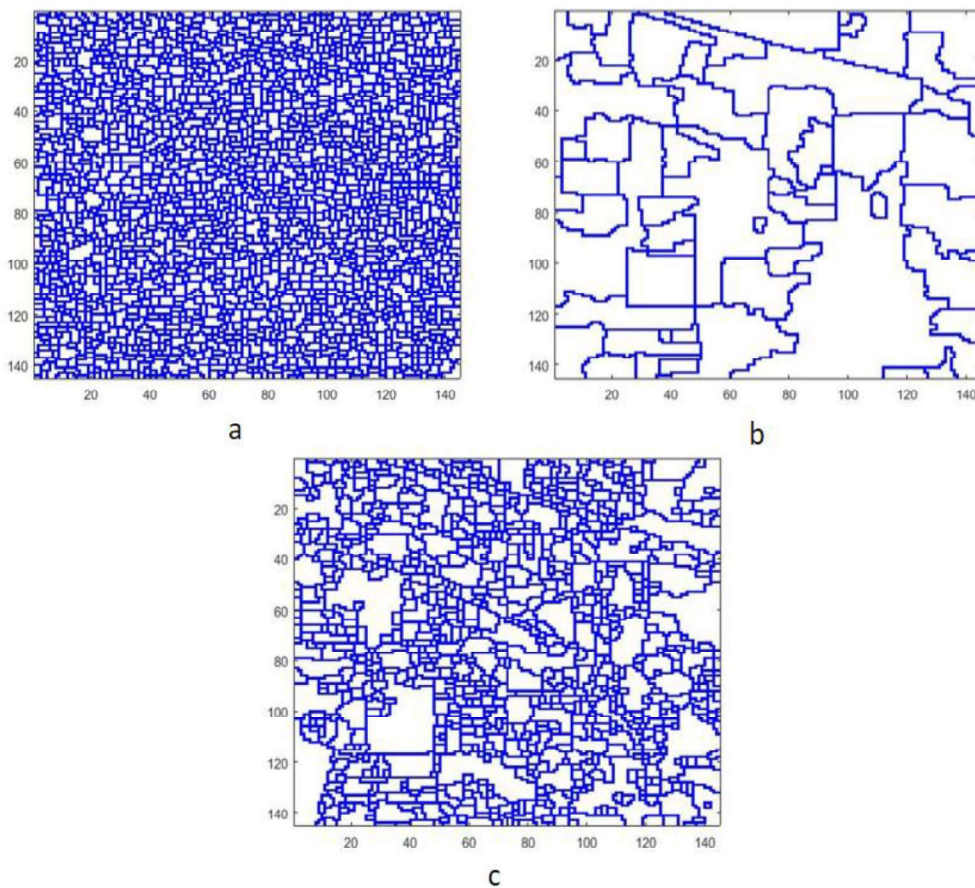


Figure 8.13: Watershed segmentation. a) over-segmentation b) under-segmentation
c) proper-segmentation.

As explain in watershed algorithm there are main alternative solutions to apply watershed on hyperspectral image, one of the powerful methods is to apply feature extraction on the hyperspectral data and then use the most informative principle component. applying watershed on an image directly will cause over-segmentation, therefore some image filtering and edge detection is advised to be used. On the other

hand, if the image was over filtered, the segmentation under fits the image which means having big areas considered as separated areas. In Figure 8.13, a) represent over segmentation where the watershed was applied directly on the first principle component. b) represent watershed applied with some excessive unsuitable filtering technique. c) represent the midpoint between a and b where the segmentation fit the spatial texture and helps to improve the classification result by applying watersheds on the first 6 PCA components. These three segmentation results will be used to integrate the spatial information with the spectral classification result. The classification results can be shown in Figure 8.14 and Figure 8.15.

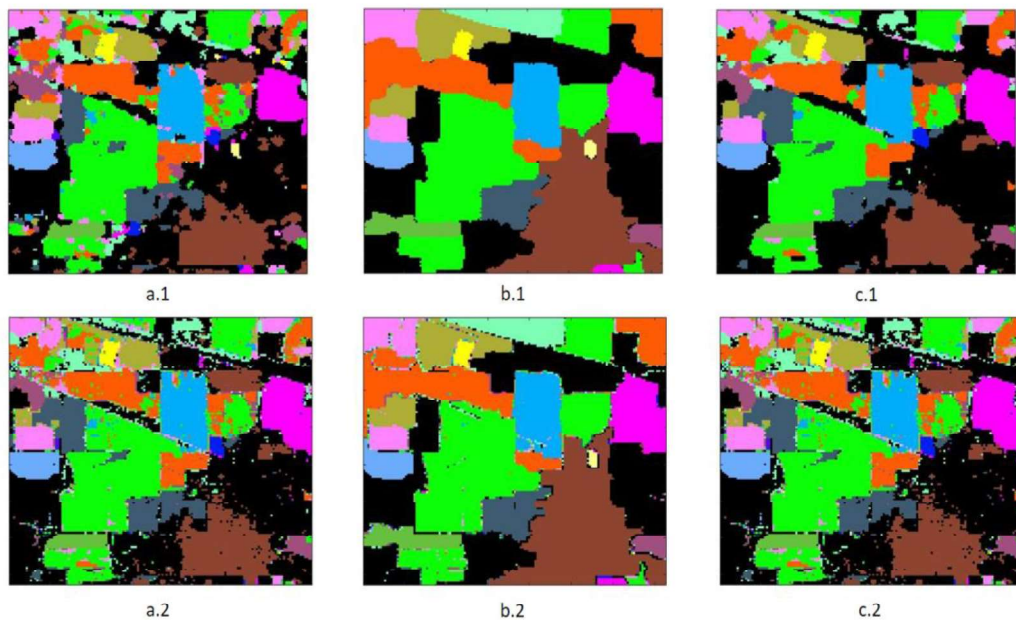


Figure 8.14: SVM-Watershed, Spectral-spatial classification (17 Classes) a1,b1,c1 represent with-WHEDs classification, a2,b2,c2 represent no-WHEDs classification.

Table 8.4: Watershed-SVM, Spatial-Spectral Classification result on Indian Pines

SVM-Watershed Spectral-Spatial classification (No WHEDs)						
	17 Classes			16 Classes		
	Overall acc.	Average acc.	Kappa	Overall acc.	Average acc.	Kappa
Seg (a)	67.16%	77.95%	0.592	79.83%	79.83%	0.769

Seg (b)	57.14%	62.88%	0.480	74.75%	64.28%	0.710
Seg (c)	69.04%	72.06%	0.609	80.45%	72.93%	0.776
SVM-Watershed Spectral-Spatial classification (with WHEDs)						
	17 Classes			16 Classes		
	Overall acc.	Average acc.	Kappa	Overall acc.	Average acc.	Kappa
Seg (a)	71.56%	77.43%	0.638	81.55%	78.39	0.788
Seg (b)	57.37%	62.92%	0.481	74.96%	63.32%	0.712
Seg (c)	72.53%	73.26%	0.647	82.07%	73.86%	0.794

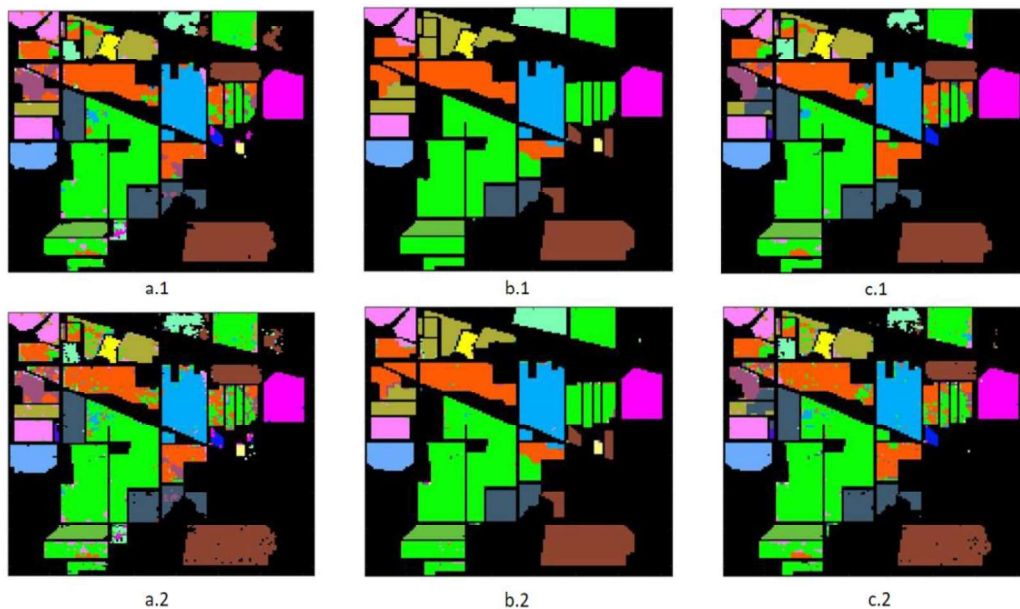


Figure 8.15: SVM-Watershed, Spectral-Spatial Classification (16 Classes) a1,b1,c1 Represent With-WHEDs Classification, a2,b2,c2 Represent No-WHEDs Classification

By comparing the results between No WHEDs and With WHEDs it can be seen the latter algorithm gives a better overall accuracy but the average accuracy is decreased which means the improvement of the classification wasn't equally distributed on all the classes.

8.2.5.2 ERW:

In the part, a lot of experiments were applied and developed. ERW basically can be used in 2 different ways, one of them is to use some the pixels in the classified image as seeds by solving equation (5.24) and in case there are no available seeds in the image, equation (5.23) can be used to solve the ERW by using only the probabilities obtained via SVM without any seeds, In addition to that, the image used in the spatial part has also different options in this experiment we tried all the following choices a) only the 1st PC. b) full spectral by using 200 bands

1st principle component experiment:

Table 8.5: EWR Classification Result On Indian Pines Using Only The 1st PC.

SVM-ERW 1 st PC Spectral-Spatial classification (No Seeds)								
	17 Classes				16 Classes			
	Overall acc.	Average acc.	Kappa	*	Overall acc.	Average acc.	Kappa	*
$\lambda = 0.001$	68.52%	46.35%	0.563	a	80.17%	61.33%	0.770	G
$\lambda = 0.01$	75.68%	69.16%	0.689	b	86.90%	77.44%	0.849	H
$\lambda = 0.1$	75.57%	76.32%	0.690	c	86.85%	85.81%	0.849	I
SVM-ERW 1 st PC Spectral-Spatial classification (With Seeds)								
	17 Classes				16 Classes			
	Overall acc.	Average acc.	Kappa	*	Overall acc.	Average acc.	Kappa	*
$\lambda = 0.001$	64.52%	83.58%	0.578	d	89.56%	92.38%	0.878	J
$\lambda = 0.01$	77.04%	87.74%	0.708	e	89.48%	92.28%	0.879	K
$\lambda = 0.1$	76.81%	87.11%	0.706	f	87.98%	90.81%	0.861	L

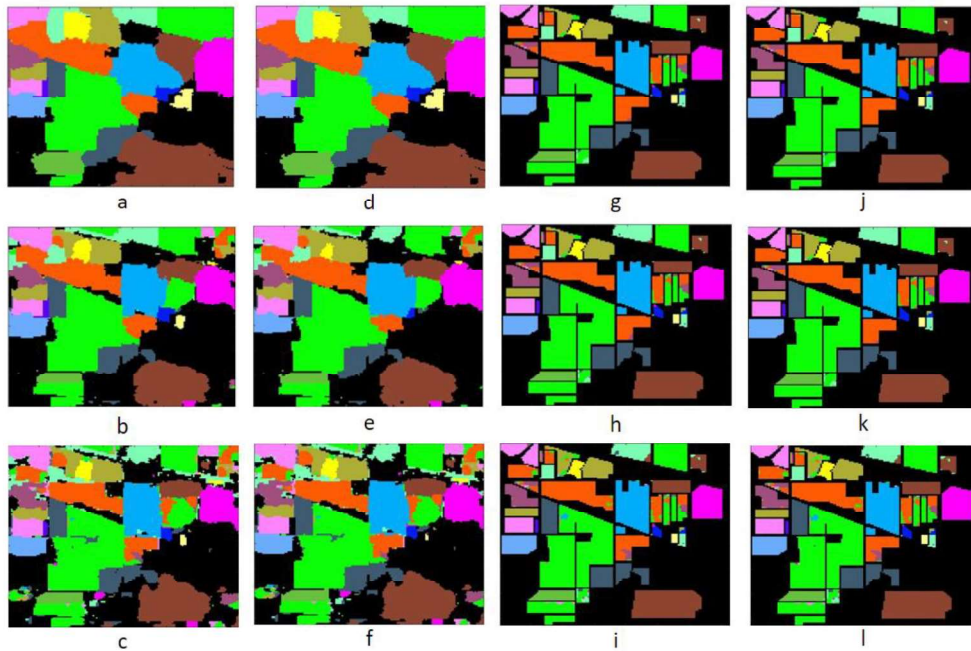


Figure 8.16: ERW Classification with Feature Extraction For 16 And 17 Classes with/without Seeds.

Table 8.6: ERW Classification Results On Indian Pines Using Full Spectral

SVM-ERW Full Spectral – (no Feature Extraction) Spectral-Spatial classification (No Seeds)								
	17 Classes				16 Classes			
	OA	AA	K	*	OA	AA	K	*
$\lambda = 0.001$	75.11%	69.57	0.682	a	88.64%	90.22%	0.869	g
$\lambda = 0.01$	76.31%	78.72%	0.698	b	86.77%	87.86%	0.847	h
$\lambda = 0.1$	72.08%	80.92%	0.652	c	83.51%	84.14%	0.810	i
SVM-ERW Full Spectral – (no Feature Extraction) Spectral-Spatial classification (With Seeds)								
	17 Classes				16 Classes			
	OA	AA	K	*	OA	AA	K	*
$\lambda = 0.001$	77.47%	86.23%	0.712	d	88.81%	91.67%	0.871	j

$\lambda = 0.01$	76.93%	83.69%	0.706	e	86.37%	88.68%	0.843	k
$\lambda = 0.1$	71.91%	80.46%	0.804	f	83.22%	83.94%	0.807	l

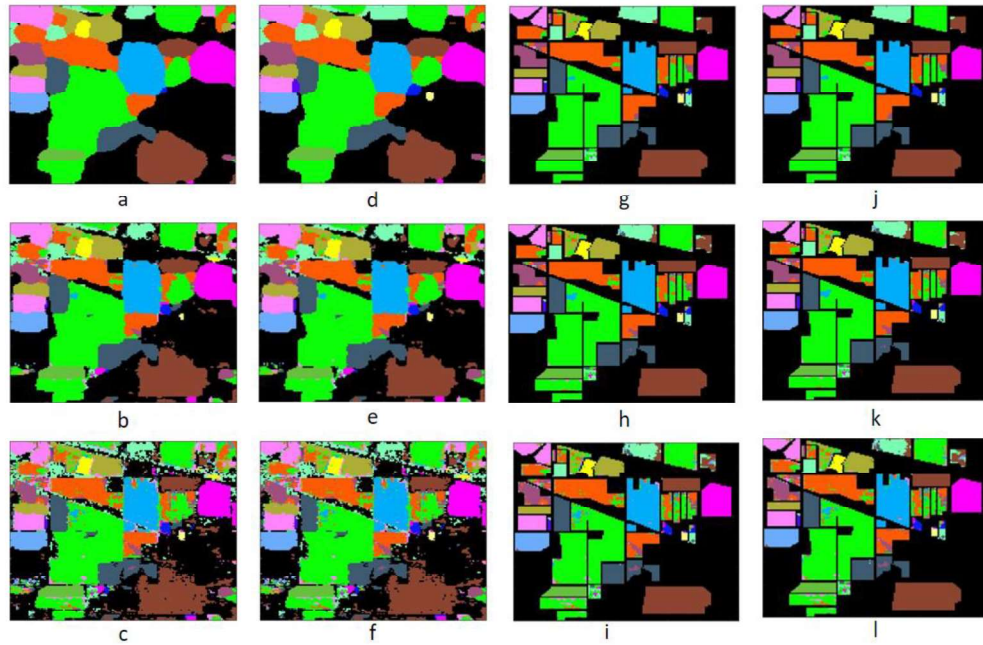


Figure 8.17: ERW Classification Without Feature Extraction For 16 And 17 Classes With/Without Seeds.

(*) is the Figure number for each result, 16 classes results are high than 17 classes, the highest OA for 16 classes was obtained by using Seeds, 1st component and $\lambda = 0.001$. for 17 classes case, the highest OA was obtained by using Full spectral information, to give a better understanding of the parameter λ role it's easier to check the 17 classes results, for example if we checked Figure 8.16 and Figure 8.17 and compared between the classification map obtained in a and the classification map obtained in c, in (a) $\lambda = 0.001$ and therefore the spatial information has a higher effect on the final classification map whereas in (c) $\lambda = 0.1$ and the contribution of the spectral classification in the final classification map is higher. From equation (5.24) and (9.5) it can be seen that λ is multiplied by the probabilities obtained from spectral classifier, which means lower λ values will cause smaller contribution of the spectral values and vice versa. To give a better overview of the results obtained from ERW the confusion matrix will be calculated only for the 17 classes, 1st PC, with seeds.

Table 8.7: Confusion Matrix for ERW 17 Classes with seeds and $\lambda = 0.1$

		Predicted Classes																
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
Actual Classes:	C1	6627	18	334	332	179	43	381	5	230	14	184	883	260	51	564	607	64
	C2	0	43	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0
	C3	0	0	1130	4	27	0	2	0	0	0	21	232	8	1	3	0	0
	C4	0	0	17	497	49	0	0	0	0	0	0	266	1	0	0	0	0
	C5	0	0	3	0	226	0	0	0	0	0	0	0	8	0	0	0	0
	C6	120	0	0	11	3	312	6	0	10	0	0	0	5	0	0	16	0
	C7	42	0	0	0	0	0	688	0	0	0	0	0	0	0	0	0	0
	C8	1	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0
	C9	0	0	0	0	0	0	0	0	478	0	0	0	0	0	0	0	0
	C10	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0
	C11	5	0	59	0	1	0	1	0	0	0	735	166	1	0	3	1	0
	C12	1	0	42	2	9	0	3	0	3	0	37	2344	13	0	1	0	0
	C13	0	0	6	2	0	0	0	0	0	0	0	16	562	0	0	3	4
	C14	0	0	0	0	0	0	0	0	0	0	0	1	0	204	0	0	0
	C15	93	0	0	0	0	0	0	0	0	0	0	1	0	0	1169	2	0
	C16	76	0	1	0	0	0	5	0	0	0	0	0	0	0	1	303	0
	C17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In Table 8.7. C1 to C17 are the Classes, if we check the small classes we have C2,C8 and C10, and for example in C2, 43 samples were correctly classified out of 46, and 18 samples from Class C1 were classified as C2. And for the big Classes we can check C12, 2344 samples out of 2455 samples were correctly classified and 883 samples from C1 were classified as C12.a further study of the confusion matrix is important to diagnose the performance of the algorithm in detail and to check which classes are causing problem and for example different training sample can be chosen for these classes.

8.2.6 The importance of the Training Samples:

In this section, the randomly distributed samples in Indian Pines from Figure 7.2 will be used to emphasis on the importance of the Training Samples, only a few experiments will be done here as in real life application it's hard to obtain randomly distributed samples. SVM on the 16 classes case will be applied using 4% of the total samples as training samples in total 409 Training Samples after discarding the background samples. Then ERW will be applied on the result obtained from SVM Figure 8.18 and Table 8.8. show the result of using this training samples

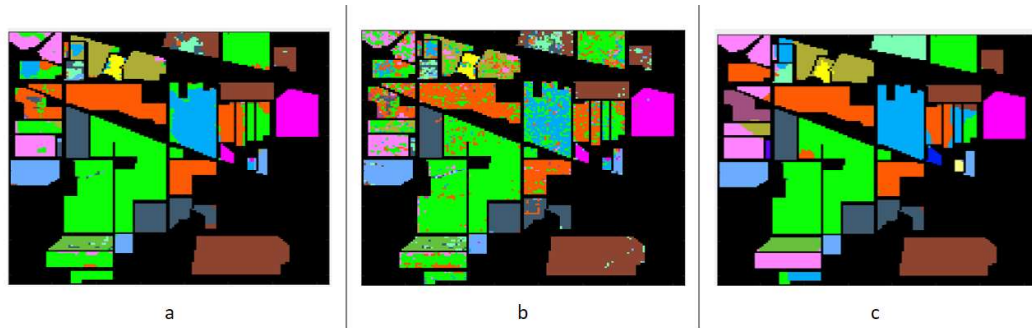


Figure 8.18: SVM and ERW classification results using only 409 training Samples.

In Figure 8.18 a) represent the result from ERW without seeds, b) represent the result obtained via RBF-SVM and c) represent the result of ERW with using all the training samples as seeds.

Table 8.8: SVM and EWR results using 4% of the total samples as training Samples:

Method	Parameters	OA
RBF-SVM	$c = 10^4, g = 0.1$	74.10%
ERW no Seeds	$\lambda = 0.1$	82.22%
ERW with Seeds	$\lambda = 0.01$	96.42%

The OA accuracy obtained using only %4 of the total samples as training samples is very is very high in ERW with seeds (96. 42%), the main reason of reaching this high accuracy is the distribution of the training samples, for ERW which is diverted from segmentation algorithm (RW) the spatial distribution of the samples plays a major role in the final result. And form Figure 7.2 (a) it can be seen how the training samples are distributed in all the separated areas, on the contrary of the training Samples in Figure 7.2 (b).

8.2.7 Salinas Scene:

Another data set is used to prove the generality of ERW algorithm, will take into consideration only the 16 classes, the Training Samples shown in Figure 7.3 are going to be used discarding the background samples. The total number of 16 classes training samples in 3416 which is equal to 6% of the total 16 classes sample in the image. As all the other experiment, SVM will be applied and the result of SVM will be integrated with ERW, also here ERW will be applied twice, with seeds and without seeds. Figure

8.19 shows the results for Salinas Scene and Table 8.9 contains the overall accuracies and used parameters.

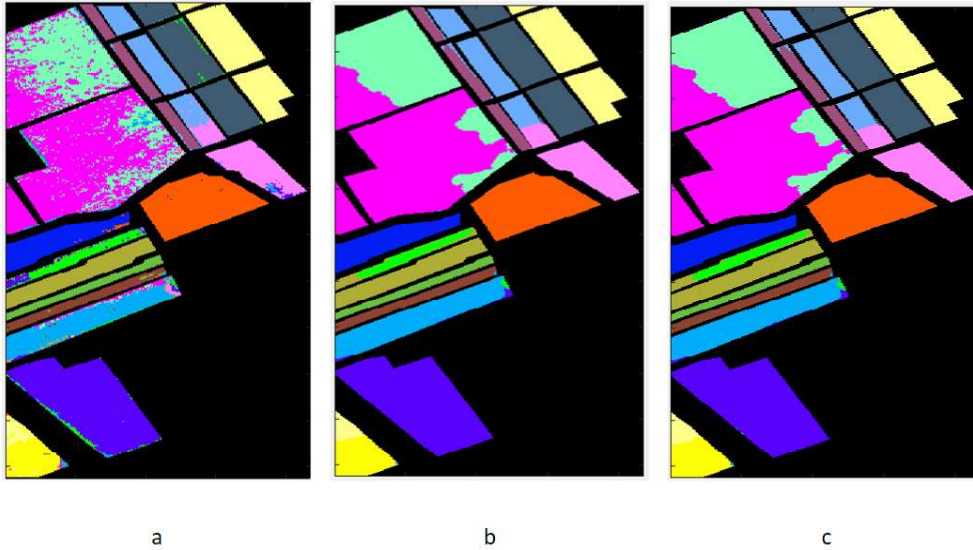


Figure 8.19: SVM and ERW Classification Result on 16 Classes Salinas Scene using 6% Training Samples.

From comparing the results obtained between ERW with seed and ERW without seeds in Table 8.9 we can find that OA for EW without seeds is higher than the OA for ERW with seeds, but if the seeds were well distributed in the image the accuracy of latter algorithm will be higher.

Table 8.9: SVM and ERW Classification result on 16 Classes Salinas Scene using 6% training Samples.

Method	Parameters	OA
RBF-SVM	$c = 10^4, g = 1$	86.16%
ERW no Seeds	$\lambda = 0.05$	91.75%
ERW with Seeds	$\lambda = 0.05$	91.61%

8.3 Modified ERW Results

Many different approaches were tried to improve the accuracy obtained from ERW, for example trying to find extra seeds or training samples. In most of the cases only small enhancement in the accuracy was obtained, but one technique which worth to be mentioned is adjusting the probabilities maps obtained from the spectral classification.

As mentioned before there 2 possible ways to adjust the probabilities obtained from SVM. 1) priority for small classes. 2) priority for large classes the first one will be used with Indian pines 16 classes and the second one with Indian pines 17 classes. The following table shows the improvement of the result by applying these 2 methods. To apply these methods, the number of each class after applying SVM need to be calculated then these numbers are normalized then multiplied with the relevant column of the probability map.

Table 8.10: ERW results modification by giving different priorities for different classes.

17 classes No seeds			16 Classes No seeds		
Method	OA	AA	Method	OA	AA
ERW	75.59%	81.65%	ERW	86.38%	86.79
ERW Priority for	73.46%	83.63%	ERW Priority for	87.38%	86.69%
ERW Priority for	76.93%	53.56%	ERW Priority for	80.17%	61.74%

By checking the results in Table 8.10. We see that these methods can improve the accuracy in some cases, but still not final or a general method to improve ERW and further studies might be applied later to find better methods to improve the accuracy.



REFERENCES

- [1] **G. Camps-Valls and L. Bruzzone**, Kernel methods for remote sensing data analysis, John Wiley & Sons, Ltd, 2009.
- [2] **M. Guillaumin, J. Verbeek and C. Schmid**, "Multimodal semi-supervised learning for image classification," in *Computer Vision and Pattern Recognition (CVPR) 2010 IEEE Conference on (pp. 902-909)*. IEEE., 2010.
- [3] **G. Hughes**, "On the mean accuracy of statistical pattern recognizers," *IEEE transactions on information theory*, vol. 14, p. 55–63, 1968.
- [4] **J. P. Hoffbeck and D. A. Landgrebe**, "Covariance matrix estimation and classification with limited training data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 763-767, 1996.
- [5] **B. M. Shahshahani and D. A. Landgrebe**, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon," *IEEE Transactions on Geoscience and remote sensing*, vol. 32, no. 5, pp. 1087-1095, 1994.
- [6] **L. O. Jimenez and D. A. Landgrebe**, "Hyperspectral data analysis and supervised feature reduction via projection pursuit.," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 6, pp. 2653-2667, 1999.
- [7] **F. Tsai and W. D. Philpot**, "A derivative-aided hyperspectral image analysis system for land-cover classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 2, pp. 416-425, 2002.
- [8] **S. Subramanian, N. Ga, S. Michael, J. Barhen and N. Toomarian**, "Methodology for hyperspectral image classification using novel neural network.," *PROCEEDINGS-SPIE THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING*, pp. 128-137, 1997.
- [9] **S. R. Gunn**, "Support vector machines for classification and regression.," ISIS technical report 14, 1998.
- [10] **L. E. Peterson**, "K-nearest neighbor," *Scholarpedia* , vol. 4, no. 2, 2009.
- [11] **S. Beucher and C. Lantuéjoul**, "Use of watersheds in contour detection," 1979.

- [12] **L. Grady**, "Multilabel random walker image segmentation using prior models.," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 763-770, 2005.
- [13] **S. M. Holland**, "Principal components analysis (PCA).," Department of Geology, University of Georgia, Athens, GA (2008), 2008.
- [14] **D. A. Landgrebe**, *Signal theory methods in multispectral remote sensing*, vol. 29, John Wiley & Sons, 2005.
- [15] **D. A. Landgrebe**, "Hyperspectral image data analysis," *IEEE Signal Processing Magazine*, vol. 19, pp. 17-28, 2002.
- [16] **T. Cover and P. Hart**, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21-27, 1967.
- [17] **H. Parvin, H. Alizadeh and B. Minaei-Bidgoli**, *MKNN: Modified k-nearest neighbor*, vol. 1, Citeseer, 2008.
- [18] **J. M. Keller, M. R. Gray and J. A. Givens**, "A fuzzy k-nearest neighbor algorithm," *IEEE transactions on systems, man, and cybernetics*, no. 4, pp. 580-585, 1985.
- [19] **R. Herbrich**, *Learning Kernel Classifiers*, MIT Press, 2016.
- [20] **T. Hofmann, B. Schölkopf and A. J. Smola**, "A review of kernel methods in machine learning," *Technical Report 156*, 2006.
- [21] **C. J. Burges**, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121-167, 1998.
- [22] **N. Cristianini and J. Shawe-Taylor**, *An introduction to support vector machines and other kernel-based learning methods*, Cambridge university press, 2000.
- [23] **J. Mercer**, "Functions of positive and negative type, and their connection with the theory of integral equations," *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, vol. 209, pp. 415-446, 1909.
- [24] **M. Fauvel**, "Spectral and spatial methods for the classification of urban remote sensing data," 2007.
- [25] **M. Pesaresi and J. A. Benediktsson**, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE transactions on Geoscience and Remote Sensing*, vol. 39, no. 2, pp. 309-320, 2001.

- [26] **A. A. Farag, R. M. Mohamed and A. El-Baz**, "A unified framework for map estimation in remote sensing image segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 7, pp. 1617-1634, 2005.
- [27] **K.-S. Fu and J. Mui**, "A survey on image segmentation," *Pattern recognition*, vol. 13, no. 1, pp. 3-16, 1981.
- [28] **Y. Tarabalka, J. Chanussot, J. A. Benediktsson, J. Angulo and M. Fauvel**, "Segmentation and classification of hyperspectral data using watershed," *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, vol. 3, pp. III-652, 2008.
- [29] **A. Darwish, K. Leukert and W. Reinhardt**, "Image segmentation for the purpose of object-based classification," in *Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International*, vol. 3, IEEE, 2003, pp. 2039-2041.
- [30] **J. C. Tilton**, "Analysis of hierarchically related image segmentations," in *Advances in Techniques for Analysis of Remotely Sensed Data, 2003 IEEE Workshop on*, IEEE, 2003, pp. 60-69.
- [31] **P. Soille**, "Morphological partitioning of multispectral images," *Journal of Electronic Imaging*, vol. 5, no. 3, pp. 252-266, 1996.
- [32] **G. Noyel, J. Angulo and D. Jeulin**, "Morphological segmentation of hyperspectral images," *Image Analysis and Stereology*, vol. 26, pp. 101-109, 2009.
- [33] **J. Chanussot and P. Lambert**, "Bit mixing paradigm for multivalued morphological filters," *Image Processing and Its Applications, 1997., Sixth International Conference on*, vol. 2, pp. 804-808, 1997.
- [34] **P. Lambert and J. Chanussot**, "Extending mathematical morphology to color image processing," *Proc. CGIP*, pp. 158-163, 2000.
- [35] **L. Shafarenko, M. Petrou and J. Kittler**, "Automatic watershed segmentation of randomly textured color images," *IEEE transactions on Image Processing*, vol. 6, pp. 1530-1544, 1997.
- [36] **J. Angulo and J. Serra**, "Mathematical morphology in color spaces applied to the analysis of cartographic images," *Proceedings of GEOPRO*, vol. 3, pp. 59-66, 2003.
- [37] **J. Chanussot and P. Lambert**, "Watershed approaches for color image segmentation.," in *NSIP*, vol. 99, 1999, pp. 129-133.

- [38] **J. Angulo**, "Unified morphological color processing framework in a lum/sat/hue representation," *Mathematical Morphology: 40 Years On*, pp. 387-396, 2005.
- [39] **S. Van der Linden, A. Janz, B. Waske, M. Eiden and P. Hostert**, "Classifying segmented hyperspectral data from a heterogeneous urban environment using support vector machines," *Journal of Applied Remote Sensing*, vol. 1, p. 013543.
- [40] **P. Li and X. Xiao**, "Evaluation of multiscale morphological segmentation of multispectral imagery for land cover classification," in *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, vol. 4, IEEE, 2004, pp. 2676-2679.
- [41] **A. Widayati, B. Verbist and A. Meijerink**, "Application of combined pixel-based and spatial-based approaches for improved mixed vegetation classification using IKONOS," in *Proc. 23rd Asian Conf. Remote Sens*, 2002, p. 8.
- [42] **L. Grady and G. Funka-Lea**, "Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials," in *ECCV Workshops CVAMIA and MMBIA*, vol. 3117, Springer, 2004, pp. 230-245.
- [43] **X. Kang, S. Li, M. Li and J. A. Benediktsson**, "Extended random walkers for hyperspectral image classification," in *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, IEEE, 2014, pp. 1520-1523.
- [44] **B. Sun, X. Kang, S. Li and J. A. Benediktsson**, "Random-walker-based collaborative learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, pp. 212-222, 2017.
- [45] **L. Vincent and P. Soille**, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp. 583-598, 1991.
- [46] **Y. Tarabalka, J. Chanussot and J. A. Benediktsson**, "Segmentation and classification of hyperspectral images using watershed transformation," *Pattern Recognition*, vol. 43, pp. 2367-2379, 2010.
- [47] **A. A. Green, M. Berman, P. Switzer and M. D. Craig**, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Transactions on geoscience and remote sensing*, vol. 26, pp. 65-74, 1988.
- [48] **A. N. Evans and X. U. Liu**, "A morphological gradient approach to color edge detection," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1454-1463, 2006.

- [49] **P. G. Doyle and J. L. Snell**, Random walks and electric networks, Mathematical Association of America, 1984.
- [50] **R. Courant and D. Hilbert**, Methods of Mathematical Physics, Volume 2: Differential Equations, John Wiley & Sons, 2008.
- [51] **F. Harary**, Graph theory., Addison-Wesley, MA, 1969.
- [52] **J. Shi and J. Malik**, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888-905, 2000.
- [53] **N. Biggs**, "Algebraic potential theory on graphs," *Bulletin of the London Mathematical Society*, vol. 29, pp. 641-682, 1997.
- [54] **L. Grady and E. Schwartz**, "Anisotropic interpolation on graphs: The combinatorial Dirichlet problem," 2003.
- [55] **N. Biggs**, "Algebraic Graph Theory, volume 67 of Cambridge Tracts in Mathematics," Cambridge University Press, 1974.
- [56] **J. Morel and S. Solimini**, "Variational Methods in Image Segmentation: With Seven Image Processing Experiments (Progress in Nonlinear Differential Equations and Their Applications)," 1994.
- [57] **C.-C. Chang and C.-J. Lin**, "LIBSVM: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, p. 27, 2011.
- [58] **C. M. Bishop**, Neural networks for pattern recognition, Oxford university press, 1995.
- [59] **M. J. Black, G. Sapiro, D. H. Marimont and D. Heeger**, "Robust anisotropic diffusion," *IEEE Transactions on image processing*, vol. 7, no. 3, pp. 421-432, 1998.
- [60] **J. J. Dongarra, I. S. Duff, D. C. Sorensen and H. A. Van der Vorst**, Solving linear systems on vector and shared memory computers, vol. 10, Society for Industrial and Applied Mathematics Philadelphia, 1991.



RESUME

Name Surname: Emad Mouselli

Place and Date of birth: 01.04.19189 Jeddah / Saudi Arabia

Education:

- **Bachelor:** 2013, Aleppo University Faculty of Electric and Electronic Engineering

Presentations and researchs:

- Support Vector Machine algorithm for image classification at **IAU**
- Internet of things advantages and future at **IAU**
- Artificial intelligence to guide a robot car at **Aleppo University**