İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

BORDER FEATURE DETECTION AND ADAPTATION: A NEW ALGORITHM FOR CLASSIFICATION OF REMOTE SENSING IMAGES

Ph.D. Thesis by

N. Gökhan KASAPOĞLU, M.Sc.

Department : Electronics and Communication Engineering

Programme: Electronics and Communication Engineering

MAY 2007

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

BORDER FEATURE DETECTION AND ADAPTATION: A NEW ALGORITHM FOR CLASSIFICATION OF REMOTE SENSING IMAGES

Ph.D. Thesis by

N. Gökhan KASAPOĞLU, M.Sc.

(504002207)

Date of submission : 8 February 2007
Date of defence examination : 25 May 2007

Supervisors (Chairmen):	**Prof. Dr. Bingül YAZGAN**
	Prof. Dr. Okan K. ERSOY (Purdue)
Members of the Examining Committee	**Prof. Dr. Ahmet H. KAYRAN (İTÜ)**
	Prof. Dr. Osman N. UÇAN (İÜ)
	Prof. Dr. Aydın AKAN (İÜ)
	Prof. Dr. Metin YÜCEL (YTÜ)
	Assoc. Prof. Dr. Sedef KENT (İTÜ)

MAY 2007

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SINIR ÖZNİTELİKLERİNİN BELİRLENMESİ VE ADAPTASYONU: UZAKTAN ALGILAMA GÖRÜNTÜLERİNİN SINIFLANDIRILMASI İÇİN YENİ BİR ALGORİTMA

DOKTORA TEZİ

Y. Müh. N. Gökhan KASAPOĞLU

(504002207)

Tezin Enstitüye Verildiği Tarih : 8 Şubat 2007
Tezin Savunulduğu Tarih : 25 Mayıs 2007

Tez Danışmanları: **Prof. Dr. Bingül YAZGAN**
Prof. Dr. Okan K. ERSOY (Purdue)

Diğer Jüri Üyeleri **Prof. Dr. Ahmet H. KAYRAN (İTÜ)**
Prof. Dr. Osman N. UÇAN (İÜ)
Prof. Dr. Aydın AKAN (İÜ)
Prof. Dr. Metin YÜCEL (YTÜ)
Doç. Dr. Sedef KENT (İTÜ)

MAYIS 2007

PREFACE

I would like to express my gratitude to my advisor Prof. Dr. Okan K. Ersoy for his guidance and reviewing my thesis. Without his constant support this thesis would not materialize.

I would like to express my special thanks to my advisor, Prof. Dr. Bingül Yazgan and steering committee members; Prof. Dr. Ahmet H. Kayran and Prof. Dr. Osman N. Uçan for their support.

AVIRIS and Landsat data used in chapter 5 were obtained from Purdue LARS (Laboratory for Applications of Remote Sensing) and GLCF (Global Land Cover Facility) respectively. Therefore, I would like to thank both LARS and GLCF for sharing their data resources.

My family has always encouraged me during my Ph.D. study. I appreciate to my mother Nimet Kasapoğlu, my sister Nurgül Kınacı and my nephew, Yarkın Kınacı for their support and patience. At last, I wish to dedicate this thesis to my father, Mehmet Kasapoğlu.

February, 2007 N. Gökhan KASAPOĞLU

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AVIRIS : Airborne Visible/Infrared Imaging Spectrometer
BFDA : Border Feature Detection and Adaptation
DAFE : Discriminate Analysis Feature Extraction
DBFE : Decision Boundary Feature Extraction
EM : Expectation Maximization
EO : Earth Observation
FLD : Fisher Linear Discriminate
GAL : Grow and Learn
GML : Gaussian Maximum Likelihood
K : Kappa
KNN : K-Nearest Neighbor
LVQ : Learning Vector Quantization
MAP : Maximum a Posteriori
ML : Maximum Likelihood
NN : Neural Network
OAA : One-Against-All
OAO : One-Against-One
PCA : Principle Component Analysis
PP : Projection Pursuit
PSHNN : Parallel, Self Organizing Hierarchical Neural Network
RBF-SVM : Radial Basis Function- Support Vector Machine
SAR : Synthetic Aperture Radar
SNR : Signal to Noise Ratio
SOM : Self Organizing Map
SVM : Support Vector Machine
SWIR : Short Wave Infrared
TIR : Thermal Infrared
TM : Thematic Mapper
VNIR : Visible Near Infrared

LIST OF TABLES

Page Number

LIST OF FIGURES

Page Number

SINIR ÖZNİTELİKLERİNİN BELİRLENMESİ VE ADAPTASYONU: UZAKTAN ALGILAMA GÖRÜNTÜLERİNİN SINIFLANDIRILMASI İÇİN YENİ BİR ALGORİTMA

ÖZET

Çeşitli sensörler dünya yüzeyinden çok miktarda data toplarlar. Toplanan bu dataların karakteristlikleri, kullanılan sensörün sahip olduğu görüntüleme geometrisine bağlıdır. Normalde, görüntü işleme tekniklerinin direk olarak uzaktan algılamaya uygulanması, sadece multispektral datalar için geçerli olabilir ki; bu datalar da göreceli olarak daha düşük sayıda öznitelik vektörüne sahiplerdir. Bu nedenle, 100-200 civarinda öznitelik vektörlerine (spektral band) sahip hiperspektral dataların analizi için gelişmiş algoritmalara ihtiyaç vardır.

Denetimli öğrenmede, eğitim işlemi çok önemlidir ve sınıflayıcının genelleme kabiliyetini belirler. Bu yüzden, yeterli sayıda eğitim örneği, düzgün bir sınıflama yapmak için istenir. Uzaktan algılamada, eğitim örneklerinin toplanması zor ve masraflı bir işlemdir. Bu yüzden, uygulamada sıklıkla karşılaşılan sınırlı sayıda eğitim örneğinin olmasıdır.

Geleneksel istatistiksel sınıflayıcılar, datanın belirli bir dağılıma sahip olduğunu kabul ederler. Gerçek veriler için bu tür bir yaklaşım geçerli olmayabilir. Ek olarak, hiperspektral datalarda doğru parametre tahmini oldukça zordur. Normalde sınıflandırma işleminde kullanılan band sayısı arttığı zaman, sınıfların ayrıntılı ve doğru olarak belirlenmesi beklenir. Yüksek boyutlu öznitelik uzayı için, yeni bir öznitelik dataya eklendiği zaman, sınıflandırma hatası azalır, fakat bunun yanı sıra sınıflandırma hatasının yanlılığı artar. Eğer sınıflandırma hatasının yanlılığındaki artış, sınıflandırma hatasındaki azalmadan daha büyük olur ise eklenen yeni özniteliğin kullanımı karar kuralının performansını düşürür. Bu etki Hughes etkisi olarak adlandırılır ve hiperspektral datada multispektral datadan daha zararlı olabilir.

Bu tezde bizim amacımız, istatistiksel dağılıma bağlı olmayan, sadece eldeki eğitim elemanlarını dayanan bir algoritma geliştirerek yukarıda özetlenen genel sınıflandırma problemlerinin üstesinden gelmektir. Bizim önerdiğimiz sınır özniteliklerinin belirlenmesi ve adaptasyonu (SÖBA) algoritması, karar yüzeylerine yakın sınır öznitelik vektörlerini kullanır ve bu sınır öznitelik vektörleri, maksimum marjin prensibini sağlayacak şekilde adapte edilerek, öznitelik uzayında doğru bölütlemenin yapılmasını sağlar.

Uzaktan algılama görüntülerinin sınıflandırılması için çok uygun olan SÖBA algoritması sınır özniteliği vektörlerinin eğitim kümesi elemanlarından seçilmesi ve eğitim kümesi elemanları yardımıyla adapte edilmesine dayanan yeni bir yaklaşımla geliştirilmiştir. Bu yaklaşım, özellikle enformasyon kaynağının sınırlı sayıda örnekle temsil edilmesi durumuyla karşılaşıldığında ve dağılımın gauss olmaması durumunda belirli öncül kabuller kullanmadığı için geleneksel istatistiksel sınıflayıcılara göre daha iyi sonuçlar üretir. Sınıflayıcılar, sınıf karar sınırlarına yakın olan eğitim örnekleri için hatalı karar vermeye eğilimlidirler. Bu yüzden, önemli öznitelik vektörleri sınıflandırma hatasını azaltmak için kullanılır. Önerilen sınıflandırma algoritması, hataya sebep olan eğitim örneklerini özel bir şekilde araştırarak, sınır öznitelik vektörlerini üretmek için adapte eder ve etiketli öznitelik vektörleri olarak sınıflandırmada kullanır.

SÖBA algoritması iki bölüme ayrılabilir. İlk kısım, sınıf merkezleri ve hatalı karar verilen eğitim örnekleri kullanılarak sınır öznitelik vektörlerinin başlangıç değerlerinin belirlenmesinden ibarettir. Bu yaklaşımla yönetilebilir sayıda sınır öznitelik vektörleri elde edilir. Algoritmanın ikinci bölümünde sınır öznitelik vektörlerinin adaptasyonu, learning vector quantization (LVQ) algoritmasıyla benzerlikler gösteren bir teknik kullanılarak gerçekleştirilir.Bu adaptasyon işleminde sınır öznitelik vektörleri, sınıf merkezleriyle olan mesafelerini uygun olarak sağlamak, farklı sınıflara ait komşu sınır öznitelik vektörleri arasındaki mesafeyi arttırmak için adaptif olarak güncellenir. Adaptasyon işlemi esnasında, sınıf merkezleri de aynı zamanda güncellenir. Sonraki sınıflandırma işlemi etiketli sınır öznitelik vektörlerine ve sınıf merkezlerine dayanır. Bu yaklaşımla herbir sınıf için uygun sayıda öznitelik vektörü algoritma tarafından atanır.

Denetimli öğrenmede eğitim süreci daha iyi sonuçlara ulaşabilmek için yansız olmalıdır. SÖBA algoritmasında başarım sınır öznitelik vektörlerinin başlangıç değerlerinin atanmasına ve eğitim örneklerinin eğitimde kullanılma sırasına bağlıdır. Bu bağımlılık sınıflayıcıyı nisbeten yanlı karar verici haline getirir. Konsensüs stratejisi ve çapraz sağlama birlikte kullanılarak, bu bağımlılıklar azaltılabilir.

Bu tezde, başlıca performans analizi ve karşılaştırmalar, AVIRIS datası kullanılarak yapılmışır. AVIRIS datası hiperspektral datadır ve sıklıkla litaratürde sınıflayıcıların performansını göstermek amacıyla kullanılır. Elde edilen ortalama eğitim, test başarımı ve kappa istatistiği Tablo.1 'de gösterilmiştir. AVIRIS data kümesi 17 sınıf içerir. Data kümeleri 1 ve 2 için elde edilen sonuçlar 9 ve 190 bandlı durumlar için, SÖBA' nın multispektral ve hipersipektral datadaki başarımını karşılaştırmak amacıyla verilmiştir. SÖBA' nın başarımı, maximum likelihood, Fisher linear likelihood, correlation, matched filtering gibi çeşitli istatistiksel sınıflayıcı teknikleri ve destek vektör makinalarını (SVMs) içerecek şekilde verilmiştir. SÖBA' nın diğer sınıflandırma teknikleriyle olan karşılaştırmasında sadece spektral öznitelikler dikkate alınmıştır.

Tablo 1: Ortalama Eğitim, Test Başarımları ve Kappa İstatistiği

DATA	METOD	EĞITIM		TEST	
		BAŞARIM %	K	BAŞARIM %	K
1	MAXIMUM LIKELIHOOD	84.83	0.82	67.56	0.63
	FISHER LINEAR LIKELIHOOD	63.7	0.59	47.3	0.42
	CORRELATION	48.4	0.43	37.2	0.31
	MATCHED FILTER	32.8	0.24	36.1	0.29
	KNN [K=5]	89.01	0.87	68.06	0.63
	LINEAR SVM [C=40]	82.40	0.81	69.01	0.64
	RBF SVM [γ=1, C=20]	86.10	0.83	71.73	0.67
	SÖBA	94.05	0.89	70.82	0.66
	KONSENSÜS SÖBA	96.41	0.95	73.36	0.69
2	KNN [K=5]	90.71	0.89	70.01	0.65
	LINEAR SVM [C=10]	83.84	0.81	74.00	0.73
	RBF SVM [γ=0.1, C=10]	87.74	0.86	77.64	0.74
	SÖBA	99.46	0.99	76.40	0.73
	KONSENSÜS SÖBA	100	1	78.71	0.75

SÖBA algoritmasıyla hem multispektral hemde hiperspektral datalar için tatminkar sonuçlar elde ettik. SÖBA, Hughes etkisi karşısında gürbüz bir algoritmadır. Bundan dolayı hem multispektral hem de hiperspektral datalar için uygundur. Ek olarak azınlık sınıf üyeleri, SÖBA algoritması tarafından geleneksel sınıflayıcıları gözönüne aldığımızda daha iyi bir şekilde korunur.

BORDER FEATURE DETECTION AND ADAPTATION: A NEW ALGORITHM FOR CLASSIFICATION OF REMOTE SENSING IMAGES

SUMMARY

Various types of sensors gather very large amounts of data from the earth surface. The characteristics of the data are related to sensor type which has its own imaging geometry. Therefore, sensor types affect processing techniques used in remote sensing. Normally, image processing techniques used directly in remote sensing are usually valid for multispectral data which is relatively in a low dimensional feature space. Therefore, advanced algorithms are needed for hyperspectral data which has at least 100-200 features (attributes/bands).

In supervised learning, the training process is very important and affects the generalization capability of a classifier. Therefore, enough number of training samples is required to make proper classification. In remote sensing, collecting training samples is difficult and costly. Consequently, a limited number of training samples is often available in practice.

Conventional statistical classifiers assume that the data has a specific distribution. For real world data, these kinds of assumption may not be valid. Additionally, proper parameter estimation is difficult, especially for hyperspectral data. Normally, when the number of bands used in the classification process increases, precise detailed class determination is expected. For high dimensional feature space, when a new feature is added to the data, classification error decreases, but at the same time, the bias of the classification error increases. If the increment of the bias of the classification error is more than the reduction in classification error, then the use of the additional feature decreases the performance of the decision rule. This phenomenon is called the Hughes effect, and it may be much more harmful with hyperspectral data than with multispectral data.

Our motivation in this thesis is to overcome some of these general classification problems by developing a classification algorithm which is directly based on the available training data rather than on the underlying statistical data distribution. Our proposed algorithm, border feature detection and adaptation (BFDA), uses border feature vectors near the decision boundaries which are adapted to make a precise partitioning in the feature space by using maximum margin principle.

The BFDA algorithm well suited for classification of remote sensing images is developed with a new approach to choosing and adapting border feature vectors with the training data. This approach is especially effective when the information source has a limited amount of data samples, and the distribution of the data is not necessarily Gaussian. Training samples closer to class borders are more prone to generate misclassification, and therefore are significant feature vectors to be used to reduce classification errors. The proposed classification algorithm searches for such error-causing training samples in a special way, and adapts them to generate border feature vectors to be used as labeled feature vectors for classification.

The BFDA algorithm can be considered in two parts. The first part of the algorithm consists of defining initial border feature vectors using class centers and misclassified training vectors. With this approach, a manageable number of border feature vectors are achieved. The second part of the algorithm is adaptation of border feature vectors by using a technique which has some similarity with the learning vector quantization (LVQ) algorithm. In this adaptation process, the border feature vectors are adaptively modified to support proper distances between them and the class centers, and to increase the margins between neighboring border features with different class labels. The class centers are also adapted during this process. Subsequent classification is based on labeled border feature vectors and class centers. With this approach, a proper number of feature vectors for each class is generated by the algorithm.

In supervised learning, the training process should be unbiased to reach more accurate results in testing. In the BFDA, accuracy is related to the initialization of the border feature vectors and the input ordering of the training samples. These dependencies make the classifier a biased decision maker. Consensus strategy can be applied with cross validation to reduce these dependencies.

In this thesis, major performance analysis and comparisons were made by using the AVIRIS data. The AVIRIS data is a hyperspectral data set and is often used in the literature to demonstrate performancec of classifiers. Average training, testing accuracies and kappa statistics are given in Table.1. The AVIRIS data set contains 17 classes. The results were obtained for data sets 1 and 2 for 9 and 190 features respectively to make proper comparison of the BFDA with multispectral and hyperspectral data. The performance of the BFDA was compared with other classification algorithms including support vector machines and several statistical classification techniques such as maximum likelihood, Fisher linear likelihood, correlation and matched filtering algorithms. Only spectral features were taken into account in the comparison of BFDA with other classification techniques.

Table 1: Average Training ,Testing Accuracies and Kappa Statistics

DATA SET	METHOD	TRAINING		TESTING	
		ACCURACY %	K	ACCURACY %	K
1	MAXIMUM LIKELIHOOD	84.83	0.82	67.56	0.63
	FISHER LINEAR LIKELIHOOD	63.7	0.59	47.3	0.42
	CORRELATION	48.4	0.43	37.2	0.31
	MATCHED FILTER	32.8	0.24	36.1	0.29
	KNN [K=5]	89.01	0.87	68.06	0.63
	LINEAR SVM [C=40]	82.40	0.81	69.01	0.64
	RBF SVM [γ=1, C=20]	86.10	0.83	71.73	0.67
	BFDA	94.05	0.89	70.82	0.66
	CONSENSUAL BFDA	96.41	0.95	73.36	0.69
2	KNN [K=5]	90.71	0.89	70.01	0.65
	LINEAR SVM [C=10]	83.84	0.81	74.00	0.73
	RBF SVM [γ=0.1, C=10]	87.74	0.86	77.64	0.74
	BFDA	99.46	0.99	76.40	0.73
	CONSENSUAL BFDA	100	1	78.71	0.75

Using the BFDA, we obtained satisfactory results with both multispectral and hyperspectal data sets. The BFDA is a robust algorithm with the Hughes effect. Therefore it is well-suited for both multispectral and hyperspectral data. Additionally, rare class members are more accurately classified by the BFDA as compared to conventional statistical methods.

1. INTRODUCTION

Electromagnetic radiation from visible to microwave regions reflected from the earth's surface can be measured by passive and active sensors. These measurements can be taken in to account as spectral feature vectors (attributes) for classification problems. Both the sensor types employed for gathering information, and the size of feature vectors (total number of bands) designate the design considerations of classification algorithms for multispectral and hyperspectral remote sensing.

1.1 Multispectral and Hyperspectral Data Structure

The multispectral sensors collect data in a small number of bands (features) from the different regions of the electromagnetic spectrum. Remote sensing images acquired by multispectral sensors, such as the widely used Landsat Thematic Mapper (TM) sensor, have shown their usefulness in numerous earth observation (EO) operations. In general, relatively small number of acquisition channels that characterizes multispectral sensors may be sufficient to discriminate among different land-cover classes (e.g., forestry, water, crops, urban areas, etc). However, their discrimination capability is very limited when different types (or conditions) of the same species (e.g., different types of forest) are to be recognized. For a specific band in multispectral data, measured value is averaged through the band with typically 100-200 nm in width. Therefore, narrow spectral features masked by stronger proximal features may not be readily discriminated across the spectral sampling range [1]. As an example, 17 spectral signatures for 17 classes have been depicted in Figure 1.1. As we can see from the Figure 1.1, discriminating these 17 classes from each other is a very complex classification problem and only using multispectral sensors can not be sufficient to support precise discrimination, for especially detailed class identification for the same species. Therefore, making individual measurements in a narrow band to detect instantaneous variations of specific target response is required.

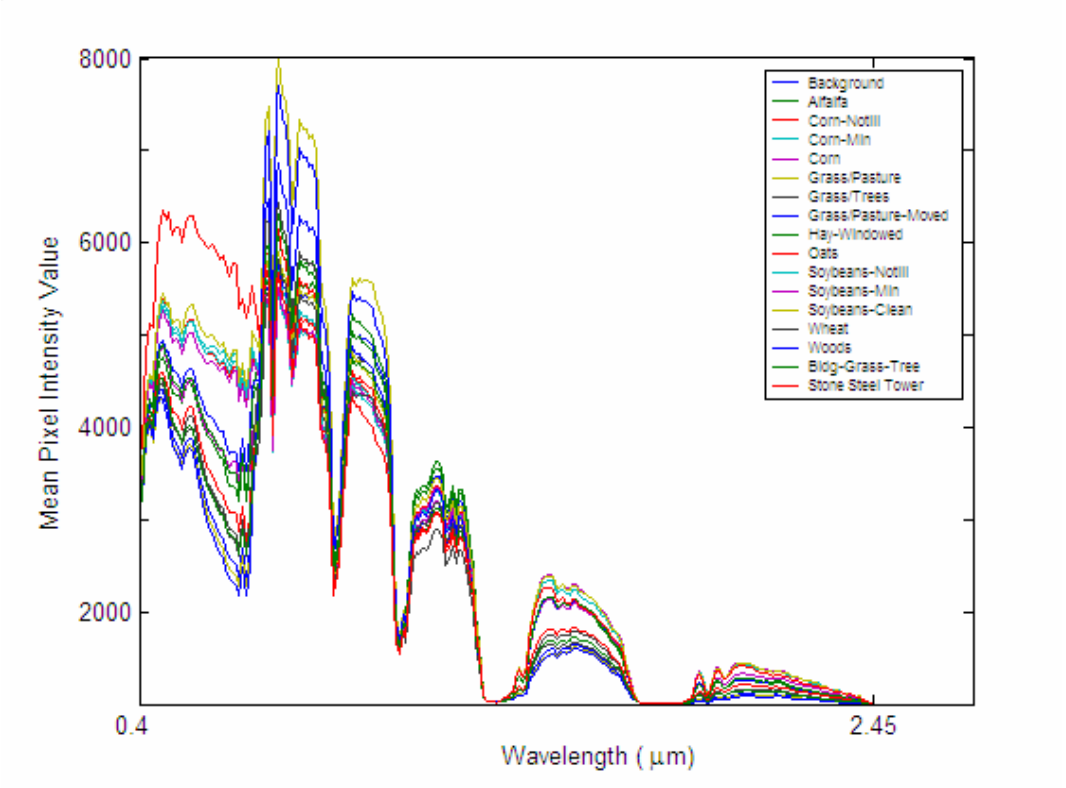


Figure 1.1: Spectral Signature of 17 Classes

Hyperspectral sensors can be used to deal with this problem. Hyperspectral sensors collect data using hundreds of narrow (2-20 nm in width) contiguous wavelength intervals over visible, near infrared (VNIR), short wave infrared (SWIR) and the thermal infrared (TIR) wavelength regions. Hyperspectral imaging spectrometers were subsequently able to retrieve reflectance spectra such that the data associated with each pixel approximated the true spectral signature of a target material, with sufficiently high signal-to-noise ratio (SNR) across the full contiguous wavelength range (normally 400-2500 nm). This collected data is represented as a hyperspectral image cube as depicted in Figure 1.2 [2]. In this cube, x and y axes specify the size of the images (spatial coordinates), whereas the z axis denotes the number of bands (features) in the hyperspectral data. The detailed spectral response of a pixel assists in providing accurate and precise extraction of information than is obtained from multispectral imaging. It is also possible to address various additional applications requiring very high discrimination capabilities in the spectral domain. From a methodological viewpoint, the automatic analysis of hyperspectral data is not a trivial task. In particular, it is made complex by many factors, such as the large

spatial variability of the hyperspectral signature of each land cover class, atmospheric effects and the curse of dimensionality.

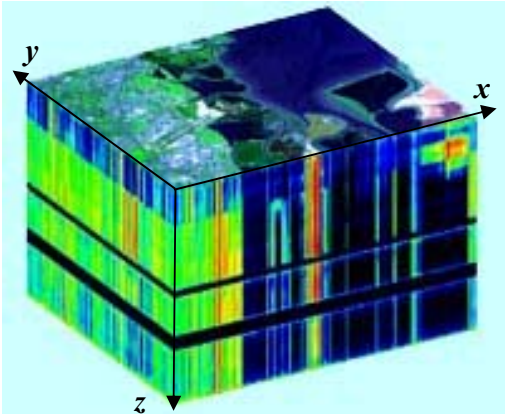


Figure 1.2: The Hyperspectral Cube

The processing of hyperspectral data remains a challenge since it is quite different from multispectral processing. Cost effective and computationally efficient procedures are required to process hundreds of bands (spectral resolution) consisting of 10-bit to 16-bit data (radiometric resolution).

The data gathered by The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) was used in the experiments in this thesis. This sensor was the first to acquire image data in continuous narrow bands simultaneously in the visible to shortwave infrared (SWIR) wavelength regions. The original AVIRIS data has 224 bands and spectral range of the data is 400-2450 (nm). For a 12-bit reflectance data, the number of discrete points in the 220-dimentional space is $(2^{12})^{220}$. One enormous advantage of hyperspectral imaging is the concept of "Spectral Signature". A spectral signature refers to the one-dimensional plot of brightness values of a pixel in the spectral domain, which is related to the characteristics of the observed material on the Earth surface, at a specific location. Each individual material has its own spectral signature.

Data analysis is aimed at extracting meaningful information from remotely sensed data. A limited number of image analysis algorithms have been developed to exploit the extensive information contained in hyperspectral imagery in many applications such as military target detection, mineral mapping, pixel and sub-pixel level land cover classification, etc. Most of these algorithms have originated from the ones used

for analysis of multispectral data, and thus have limitations. A novel classification algorithm called border feature detection and adaptation (BFDA) is proposed in this thesis for both multispectral and hyperspectral data classification to help reduce some of these problems.

1.2 General Classification Problem

Two main classification types can be considered. They are supervised and unsupervised methods. In this thesis, a supervised classification algorithm was introduced for both multispectral and hyperspectral images. For supervised learning, we have two different sets, one for training and the other one for testing. Sets of training and testing samples have features with their belongings labels. Ground truth refers to the reference set used for selecting samples to generate training and testing sets.

The classification problem occurs in its simplest form as the two class problem (binary case classification problem). It involves two partially disjoint finite sets X and Y, and an object $z \notin XUY$ is to be classified as a member of X or Y. The multi-class problem occurs when there are additional sets corresponding to other classes. The main goal of the classification problem is to find a classifier that can predict the label of new unseen data samples correctly. This can be achieved by learning from the given labeled data (training set). The test set correctness of classification is the main criterion used to evaluate a given classifier.

1.3 Problem Description and Aims of This Thesis

In supervised learning, a selected set of labeled training data is used during learning. The performance of a classification algorithm is closely related to how the labeled training data set is correlated with the unlabeled testing data set. Errors are more difficult to control in the case of detection of rare class members. Especially in hyperspectral data classification, there is a large number of spectral bands, and a relatively low number of labeled training samples [3]. Therefore, one of the main difficulties is related to the small ratio between the number of available training samples and the number of features. This may cause unsatisfactory estimates of the

class-conditional probability density functions used in standard statistical classifiers. As a consequence, when the number of features given as input to the classifier over a given number of training samples is increased, the classification accuracy decreases. This behavior is known as the Hughes phenomenon [3]. Our motivation in this study is to overcome some of these general classification problems, by developing a classification algorithm which is directly based on the available training data rather than on the underlying statistical data distribution.

In the literature, four main approaches can be identified to make statistical classification methods applicable for hyperspectral data classification problem. These approaches are:

1) Regularization of sample covariance matrix by using sample and common covariance matrices together [4,5].

2) Adaptive statistical estimation by the exploitation of the classified (semi-labeled) samples (e.g., Expectation maximization algorithm, EM) [6,7].

3) Preprocessing techniques based on feature selection/extraction, aimed at reducing/transforming the original feature space into another space of a lower dimensionality (e.g., Fisher Linear Discriminate (FLD), Discriminate Analysis Feature Extraction (DAFE), Decision Boundary Feature Extraction (DBFE), Projection Pursued (PP), etc.) [8-10].

4) Analysis of the spectral signature to model the classes [11,12].

Many supervised classification techniques have been used for multispectral and hyperspectral data classification, such as the maximum-likelihood classification, neural networks and support vector machines. Practical implementational issues and computational load are additional factors to evaluate classification algorithms. Statistical classification algorithms are fast and reliable, but they assume that the data has a specific distribution. For real world data, these kinds of assumptions may not be sufficiently accurate, especially for low probability classes. The k-nearest neighborhood algorithm is another simple and effective classification method.

In recent years, kernel methods such as support vector machines (SVMs) have demonstrated good performance in hyperspectral data classification [13]. Some of

the drawbacks of SVMs are the necessity of choosing an appropriate kernel function and time-intensive optimization. In addition, the assumptions made in the presence of samples which are not linearly separable are not necessarily optimal. Parallel, self-organizing hierarchical neural networks (PSHNNs) also achieve high classification accuracy [14]. By using parallel stages of neural network modules, hard vectors are rejected to be processed in the succeeding stage modules, and this rejection scheme is effective in enhancing classification accuracy. Consensual classifiers are related to PSHNNs, and also reach high classification accuracies [15,16].

Combining different classification algorithms to get high classification accuracy is a reliable approach [17]. It is also possible to combine the outputs of classifiers which use the same classification algorithm but are differently structured to make the decisions of the individual classifiers sufficiently independent from each other [18]. For example, this can be done by changing the input order of training samples.

In this thesis, a new classification algorithm well suited for classification of remote sensing images is developed with a new approach to choosing and adapting border feature vectors with the training data. This approach is especially effective when the information source has a limited amount of data samples, and the distribution of the data is not necessarily Gaussian. Training samples closer to class borders are more prone to generate misclassification, and therefore are significant feature vectors to reduce classification errors. The proposed classification algorithm searches for such error-causing training samples in a special way, and adapts them to generate border feature vectors to be used as labeled feature vectors for classification.

The BFDA algorithm can be considered in two parts. The first part of the algorithm consists of defining initial border feature vectors using class centers and misclassified training vectors. With this approach, a manageable number of border feature vectors are achieved. The second part of the algorithm is adaptation of border feature vectors by using a technique similar to the learning vector quantization (LVQ) algorithm [19]. In this adaptation process, the border feature vectors are adaptively modified to support proper distances between them and the class centers, and to increase the margins between neighboring border features with different class labels. The class centers are also adapted during this process. Subsequent classification is based on labeled border feature vectors and class centers. With this

approach, a proper number of feature vectors for each class is generated by the algorithm.

1.4 Organization of the Thesis

This thesis is organized in six chapters. Feature extraction or dimensionality reduction is needed for classification of hyperspectral data when classification algorithms based on statistics such as maximum likelihood is used. In chapter 2, a brief discussion is given about feature extraction, and the method called projection pursued (PP) [9] is mentioned. Using spatial features in addition to spectral features improves classification accuracy. A spatial feature extraction method [20] is also discussed in chapter 2. We categorized classification techniques in three parts and explained some important ones in chapter 3 to make precise comparison with our proposed algorithm, the BFDA [21]. These categories are parametric, non-parametric and kernel methods. Methods based on statistics such as maximum likelihood (ML) and expectation maximization (EM) [7] are parametric methods, whereas k-nearest neighbor (KNN), grow and learn (GAL) [22], and self-organizing map (SOM) [19] are non-parametric methods. In addition, kernel methods such as support vector machines (SVMs) [13,23] are also explained in chapter 3 as a relatively new generation of techniques for classification and regression problem. Our proposed algorithm border feature detection and adaptation (BFDA) [21] is introduced in chapter 4. Additionally, to reach better classification accuracies, usage of the BFDA as an individual classifier in a consensual scheme and a safe rejection scheme for the BFDA are also provided. Descriptions of the data set and experiments designed are introduced, and detailed comparison of methods is discussed in chapter 5. Conclusions and future work are given in chapter 6.

2. FEATURE EXTRACTION

In this section, feature dimension reduction for increasing class discrimination, and spatial feature extraction from conventional spectral features are discussed. In general, basic remote sensing classification systems include a module of feature extraction. This module is necessary in hyperspectral data classification for dimension reduction especially when parametric classifier based on density estimation is used. The basic classification flow graph for remote sensing including feature extraction is depicted in Figure 2.1.

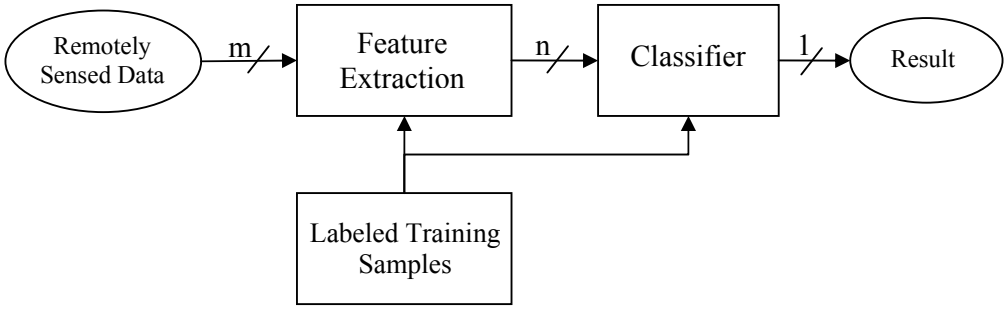


Figure 2.1: The Basic Classification Flow Graph

The aim of feature extraction is to reduce dimensionality to support proper density estimation and to increase class separability at the same time. To make a proper comparison between parametric classifiers and our proposed algorithm, the BFDA, concept of feature extraction for dimensionality reduction for increasing class discrimination is discussed, and some important methods used in the experiments are introduced in this chapter. We first explain feature extraction for dimensionality reduction. Then, we discuss how spatial features are extracted from spectral ones. The effects on classification accuracy are shown with experiments. In addition, it is also possible to apply dimensionality reduction after extraction of spatial features.

2.1 Methods for Feature Dimension Reduction

High-Dimensional space characteristics are a major issue in design considerations of classifiers for hyperspectral data classification. Therefore, it is useful to understand high dimensional feature space characteristics. It has been proven that as the number of dimensions increases, volume of a hypercube (whole feature space) concentrates in the corners, and the volume of a hyperellipsoid concentrates in an outside shell in the feature space [24]. These characteristics have two important consequences for high dimensional data: 1) High-dimensional data is mostly empty, which implies that multivariate data is in a low dimensional structure. 2) Normally distributed data will have a tendency to concentrate in the tails, while uniformly distributed data will be more likely to be collected in the corners. Together, these consequences make density estimation more difficult in the high-dimensional feature space. Under these circumstances, it would be difficult to obtain accurate results with most density estimation procedures.

The required number of labeled samples for supervised classification increases as a function of dimensionality. The required number of training samples is linearly related to the dimensionality for a linear classifier and to the square of the dimensionality for a quadratic classifier. It has been estimated that as the number of dimensions increases, the sample size needs to increase exponentially in order to obtain an effective estimate of multivariate densities [25,26].

The second-order statistics is often important in the process of discriminating among classes. In hyperspectral data, neighbor bands are usually highly correlated. Therefore, most of the data is distributed a long a few major components producing a hyperelipsoid-shaped data distribution characterized by second order statistics [27].

It is to be expected that high-dimensional data contains more information. At the same time, the above characteristics tell us that it is difficult to extract such information with techniques which are based on density estimation since these are usually based on computations at full dimensionality requiring a substantial number of labeled data. Hughes proved that with a limited number of training samples, there is a penalty in classification accuracy as the number of features increases beyond some point [3].

From classification viewpoint, especially for classification algorithms based on statistics, lower dimensional feature vectors are needed in order to make proper density estimation. Some widely used feature extraction methods for dimensionality reduction are principle component analysis (PCA), discriminate analysis feature extraction (DAFE), decision boundary feature extraction (DBFA), and projection pursuit (PP). It is also very useful to mention the difference between dimensionality reduction for data compression and classification. For data compression, most important aim is to keep most informative components but for data classification most important aim is to keep most discriminative components.

Principle component analysis assumes that the distribution takes the form of a single hyperellipsoid, such that its shape and dimensionality can be determined by mean-vector and covariance matrix of the distribution. A problem with this method is that it treats the data as if it is a single distribution. Principle components analysis is more appropriate for data compression than for class discrimination [25].

DAFE is a method that reduces the dimensionality, optimizing the Fisher ratio [28]. If the total number of classes is *c* than the final dimension will be *c-1* after DAFE. It performs the computations at full dimensionality, requiring a large number of labeled samples in order to accurately estimate parameters.

DBFE is an algorithm based directly on decision boundaries [8]. This method also predicts the number of features necessary to achieve the same classification accuracy as in the original space. DBFE has the advantage of finding the necessary feature vectors. One disadvantage of this method is that it demands a high number of training samples in a high-dimensional space. This occurs because it computes the class statistical parameters at full dimensionality.

When there are only a limited number of training samples, method of projection pursuit (PP) can be used [9,29,30]. This method performs the computation in a lower dimensional subspace that is a result of a linear projection from the original high dimensional space. This dimension reduction increases the ratio of labeled samples per feature, resulting in better parameter estimation and better classification accuracy.

2.1.1 Dimension Reduction via Projection Pursuit (PP)

Feature extraction for dimensionality reduction is needed for parametric classifiers, especially in high-dimensional feature space. Parametric classifiers use first and second order statistics whose parameters are estimated by using only labeled training samples. From the nature of the classification problem for hyperspectral data, these labeled training samples are not sufficient to make proper estimation of these parameters. Therefore dimensionality reduction is needed in hyperspectral data classification especially for parametric classifiers.

The basic dimensionality reduction scheme is depicted in Figure 2.2.

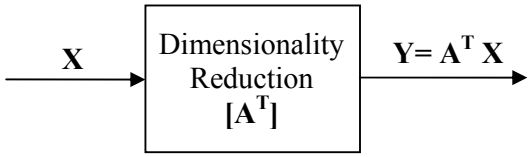


Figure 2.2: Dimensionality Reduction by Projection Pursuit

X is a multivariate data set and it is *dxN* dimensional matrix, **Y** is the resulting dimensionality reduced projected data which is *mxN* dimensional matrix and **A** is the transform matrix which is a *dxm* dimensional matrix. Dimension reduction also desired to include improvement of discrimination of classes. Therefore the algorithm should optimize the projection index $\mathbf{I(A^TX)}$ to increase class discrimination. In general, the projection index is related to first and second order statistics such as mean and covariance matrix of the training samples as in Bhattacharyya distance index which is widely used for discrimination measurement. The PP uses Bhattacharyya distance between two classes as the projection index because of its relationship with Bayes-classification accuracy, and its use of both first order and second order statistics [31].

$$\mathbf{I(A^TX)} = \frac{1}{8}(\mathbf{M}_{2Y} - \mathbf{M}_{1Y})^T \left(\frac{\Sigma_{1Y} + \Sigma_{2Y}}{2} \right)^{-1} (\mathrm{M}_{2Y}\text{-}\mathrm{M}_{1Y}) + \frac{1}{2} \ln \left(\frac{\left| \frac{\Sigma_{1Y} + \Sigma_{2Y}}{2} \right|}{\sqrt{|\Sigma_{1Y}||\Sigma_{2Y}|}} \right) \qquad \textbf{(2.1)}$$

where M_{jY} and $\sum_{jY}$ are the mean vector and covariance matrix, respectively, of the j^{th} class in the projected subspace **Y**. In the case of more than two classes, the minimum Bhattacharyya distance among the classes can be used after the Bhattacharyya distances are calculated for all combinations of two classes. Then, the minimum of the Bhattacharyya distance is chosen as

$$\mathbf{I}(\mathbf{A}^{\mathrm{T}}\mathbf{X}) = \min_{i \in C} \left\{ \frac{1}{8}(\mathbf{M}_{2Y}^{i} - \mathbf{M}_{1Y}^{i})^{T} \left(\frac{\sum_{1Y}^{i} + \sum_{2Y}^{i}}{2} \right)^{-1} (\mathrm{M}_{2Y}^{i} - \mathrm{M}_{1Y}^{i}) + \frac{1}{2} \ln \left(\frac{\left| \frac{\sum_{1Y}^{i} + \sum_{2Y}^{i}}{2} \right|}{\sqrt{\left| \sum_{1Y}^{i} \right| \left| \sum_{2Y}^{i} \right|}} \right) \right\} \quad \textbf{(2.2)}$$

C is the number of combinations of pair of two classes. Assuming that total number of classes is L, C is given by

$$C = \frac{L!}{2!(L-2)!} \quad \textbf{(2.3)}$$

The main advantage of PP is that of calculating the projection index in a low dimensional space. In addition, nearest spectral responses are correlated with each other for hyperspectral data. Therefore band grouping is applied for dimensionality reduction as a preprocessing in the PP. First and second order statistics are calculated in this low dimensional space much more accurately.

Thus, the global projection index to be maximized is the minimum Bhattacharyya distance among the classes. A sequential aspect of this algorithm is that it projects groups of neighboring bands while maximizing the minimum Bhattacharyya distance in the projected subspace. Maximization can be done with a known numerical optimization algorithm.

As explained above, projection indices for optimizing discrimination are parametric, and estimation of these parameters is carried out in a lower dimensional space. The computations at a lower-dimensional space enable PP to better handle the problem of small numbers of samples. In Figure 2.3, band grouping in projection pursuit is depicted. An iterative procedure to estimate a_i's is described in the following steps [30]:

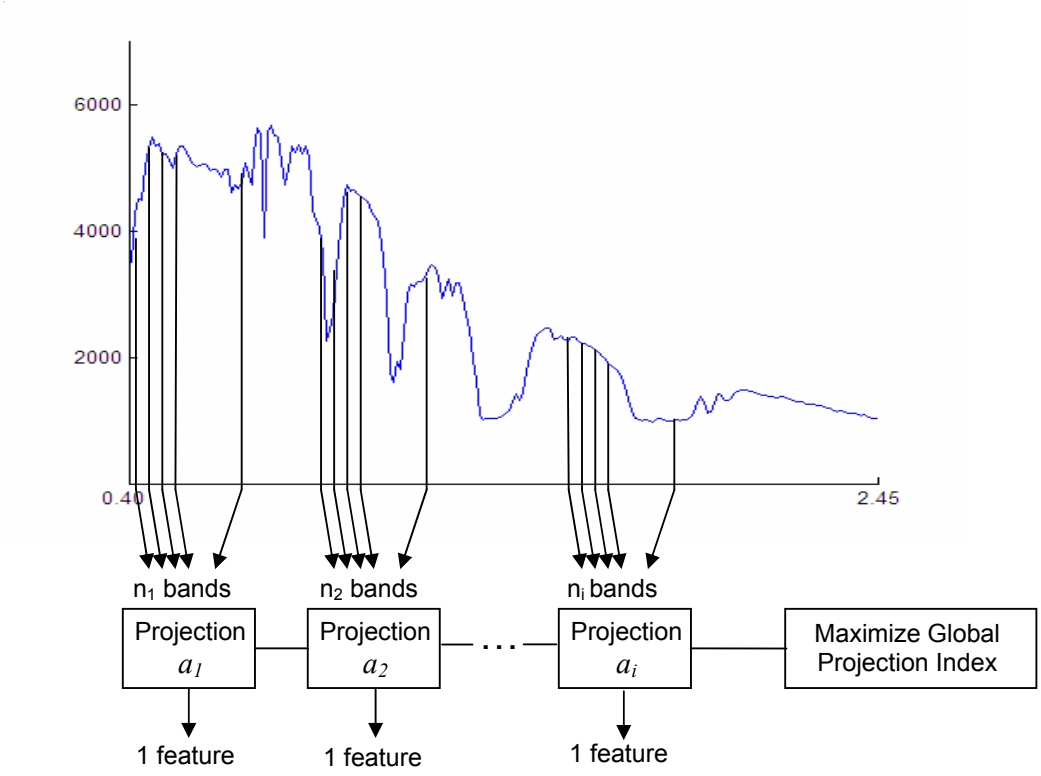


Figure 2.3: Band Grouping in Projection Pursuit

1) Make an initial guess for every a_i for each group of adjacent bands.

2) Maintaining the rest of a_i's constant, compute the a_1 (the vector that projects the first group of adjacent bands) that maximizes the minimum Bhattacharyya distance.

3) Repeat the procedure for each of the groups while a_i's, for $i \neq j$, remain constant.

4) Once the last group of adjacent bands is projected, repeat the process starting from step 2 (compute all a_j's sequentially) until convergence.

In Figure 2.4 projection pursuit is used as a preprocessing technique for dimensionality reduction by optimizing a projection index. After the processing described above has been applied, a scheme of feature extraction such as discriminate analysis feature extraction (DAFE) or decision boundary feature extraction (DBFA) can be used in the lower dimensional feature space. As a

consequence, the most discriminative features and lower dimensional space *(m<n<o)* can be achieved.

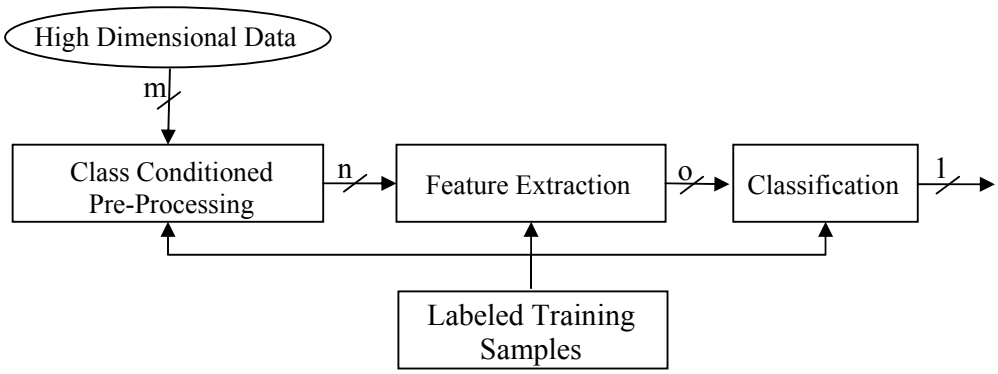


Figure 2.4: PP Based Preprocessing Technique Used for Dimensionality Reduction

It is obvious that there are a variety of projection indices which can be used in the PP algorithm. Such a projection index related to correlation between features is as follows. Highly correlated features are combined with each other to form a group. The adjacent features of the data exhibit high correlation. Therefore, the hyperspectral subspace is partitioned into subspaces based on correlation existing between adjacent features. The correlation ρ for bands i and j is given by

$$\rho_{ij} = \frac{\sum_{ij}}{\sqrt{\sum_{ii} \sum_{jj}}} \tag{2.4}$$

where $\sum_{ij}$ is the element of the covariance matrix for band i versus band j and $\sum_{ii}$ and $\sum_{jj}$ are the variances of the i^{th} and j^{th} features of the data [31]. The parameter ρ_{ij} indicates the covariance between bands i and j. The variables i and j vary from 1 to d, where d is the dimensionality of the subspace. The correlation measure C of the hyperspectral subspace quantifies the correlation between two bands, i.e. C gives the minimum of all the correlations between every pair of bands in the subspace. Therefore,

$$C_n = min(\rho_{ij})_n \tag{2.5}$$

where C_n represents correlation of the n^{th} subspace .

Supervised and automatic selection procedures can be applied for feature subset selection procedure. It is also possible to select feature size fixed or adaptively chosen.

2.2 Spatial Feature Extraction

Spatial variations of the spectral features in a predefined sub-image with appropriate sub-image size can be used as effective features in remote sensing applications [32,33]. Features based on spatial variations are called texture features as well. Texture features are robust features on noisy remote sensing data such as the data acquired by synthetic aperture radar (SAR). Especially for SAR data classification, noise is an important concern to deal with in order to achieve sufficient classification accuracy. The noise called speckle has its origin in collecting data by using active sensors in microwave frequencies. Therefore, using spatial variations instead of spectral response from individual pixels is necessary to make proper SAR classification. Gray level co-occurrence matrix statistical parameters can be used as texture features [33].

There are three different texture categories. They are course texture (neighboring points similar), fine texture (neighboring points different) and directional texture (courser in one direction). Because of speckle noise in SAR data, fine texture properties are typical. For hyperspectral data, texture category is typically course texture. Therefore, we expect to find more homogeneous areas in hyperspectral data classification.

Spatial filtering can be used to generate more homogenous regions and thereby improve classification performance in hyperspectral data classification [20]. The spatial filter can be a simple mean filter, which uses standard deviation as a homogeneity criterion. Using a homogeneity criterion, sub-image size (window size) changes adaptively to achieve more homogeneous regions in the spatial domain. If homogeneity test passes, then the mean value of the pixels in the window is assigned to the center pixel of the sub-image.

Formally, given an image *I(m,n),* the median filter can be shown by

$$I_{median}(m,n) = median\{I(m-k,n-l)\}, \quad (k,l) \in A \tag{2.6}$$

Where A is the neighborhood over which the median is applied. Median filters are most useful in mitigating the effects of salt and pepper noise that arises typically due to isolated pixels incorrectly switching to opposite intensity.

In this thesis, we extracted spatial features such as mean and variance for sub-image size (window size) from 3x3 to 9x9, and obtained combined classification results which are based on individual spatial and spectral features by using a consensual rule to reach better classification accuracy. In this way, we showed the use of spatial features together with spectral features on hyperspectral data classification.

3. TYPES OF CLASSIFIERS

The aim of classifiers is to partition the feature space into an exhaustive set of nonoverlapping regions to reach high classification accuracy by using some rules related to discrimination of the classes. These discrimination rules can be based on statistical theory or computational methods such as neural networks. Decision boundaries can be determined by a threshold function obtained by equalization of the neighbor class discrimination rules. In this chapter, a brief summary is given on classifiers used in the experiments to make a detailed comparison between the proposed classification algorithm, the BFDA, and other conventional classification methods used.

We can categorize classifiers into three types. They are parametric, non-parametric and kernel methods. For parametric methods, maximum likelihood (ML) and expectation maximization (EM) are described [35,36]; k-nearest neighbor (KNN) [37], grow and learn (GAL) [22] and self organizing map (SOM) [19] are discussed as examples of non-parametric methods. In recent years, use of support vector machines for classification and regression problems has been increasing rapidly. Support vector machines (SVMs) are discussed as an example of kernel methods [13]. SVM is initially a binary classifier. Therefore, proper hierarchical methods are needed to combine binary classifiers outputs to generate multi-class classification results.

An important performance criterion is overall classification accuracy for classifiers. Additionally, detection of rare class members is a desirable specification. Kappa statistics was used to measure reliability of decisions made [34]. In addition, practical implementation issues and computational load are important design concerns to make a proper comparison of the classifiers.

3.1 Statistical Classifiers

Classifiers based on statistics are widely used, especially in low dimensional feature spaces. In general, they are parametric classification methods, and their drawbacks are proper parameter estimation needs and pre-assumptions on their distributions made before classification. Especially in high dimensional feature space, difficulties of proper parameters estimation can be reduced by using dimension reduction methods for increasing class discrimination such as DAFE, DBFA and PP [35]. It is also difficult to detect rare class members with statistical methods. In this section, a brief summary is given on statistical methods such as maximum likelihood (ML) and expectation maximization (EM) [36].

3.1.1 Maximum Likelihood Classifier

Conditional probability density function (pdf) is used as discrimination rule in the maximum likelihood (ML) classifier. If the number of classes is *m,* then there are *m* discrimination functions that can be defined by using conditional probability density function as follows:

$$g_{C_i}(\overline{x}) = p\left(\overline{x}\middle|C_i\right),\ i = 1...m\,. \tag{3.1}$$

The label of the class which makes the discrimination rule maximum is assigned as the class of $\overline{x}$:

$$w = \arg\ \max\left\{g_{C_i}(\overline{x})\right\},\ i = 1..m \quad \Rightarrow \overline{x} \in C_w \tag{3.2}$$

In this approach, the classification problem is reduced to estimate some parameters which are related to probability density function (pdf). The Gaussian density function is widely used for classification problems because it has convenient properties and fits many processes in nature. The Central Limit Theorem states that if a random observation is made on a collection a large of number of independent random quantities, the observation will have a Gaussian distribution. If the random variable is one-dimensional, then the Gaussian density function is given by

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

In remote sensing data classification problems random variable is a vector. Especially for hyperspectral data classification, the dimension of the random variable may be larger than 100. The AVIRIS data set which is used in the experiments can be cited as an example of hyperspectral data. The original dimension (total number of bands/attributes) of the AVIRIS data set is 224. For multispectral data such as Landsat and Spot, the numbers of dimensions are 7 and 4, respectively. In this case, assuming N dimensions, the pdf can be written in the vector form as

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

where

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

where $\bar{\boldsymbol{x}}_{\boldsymbol{i}}$ is random variable, $\bar{\mu}_i$ the mean vector of the i^{th} class, and $\bar{\bar{\Sigma}}_i$ is the covariance matrix of the i^{th} class, respectively. The Gaussian pdf is also called normal distribution and is depicted by $N(\mu_i,\Sigma_i)$. The unbiased estimaties of the multidimensional Gaussian pdf parameters are calculated as follows: assuming a labeled training data set $\{(\bar{\boldsymbol{x}}_{\boldsymbol{1}},y_1),(\bar{\boldsymbol{x}}_{\boldsymbol{2}},y_2),\ \cdot\ \cdot\ \cdot\ ,(\bar{\boldsymbol{x}}_{\boldsymbol{n}},y_n)\}$ where the training vectors are $\bar{\boldsymbol{x}}_{\boldsymbol{i}}\in\mathbb{R}^N, i=1,...,n$, the class labels are $y_i\in\{1,2,\ \cdot\ \cdot\ \cdot\ ,m\}$, n is the total number of training samples, and m is the number of classes, class means are estimated as

$$\hat{\bar{\mu}}_i=\frac{1}{n_i}\sum_{j=1}^{n_i}\bar{x}_j,\{\bar{x}_j\mid y_j=i,\quad i=1,\ \cdot\ \cdot\ \cdot\ ,m\} \tag{3.6}$$

where n_i is the total number of training samples for class *i*. The covariance matrix estimate of the i^{th} class is given by

$$\hat{\bar{\Sigma}}_i = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (\bar{\boldsymbol{x}}_{\boldsymbol{j}} - \hat{\bar{\mu}}_i)(\bar{\boldsymbol{x}}_{\boldsymbol{j}} - \hat{\bar{\mu}}_i)^T, \{\bar{\boldsymbol{x}}_{\boldsymbol{j}} \mid y_j = i, \quad i = 1, \cdots, m\} \tag{3.7}$$

Logarithmic version of the pdf is widely used as a discrimination function as follows:

$$g_{C_i}(\bar{x}) = \ln\left(p\left(\bar{x} \middle| C_i\right)\right) = (1/2)\ln\left|\Sigma_i\right| + (1/2)(\bar{x} - \mu_i)^T \Sigma_i^{-1}(\bar{x} - \mu_i), \; i = 1..m \tag{3.8}$$

which is a quadratic function and is commonly used in the Gaussian maximum likelihood (GML) classifier [38].

The minimum expected error that can be achieved in performing classification is referred to as the Bayes' error. A decision rule that assigns a sample to the class with highest a posteriori probability (the MAP classifier) achieves the Bayes' error [31]. A posteriori probability can be written as follows by using Bayes' rule:

$$p\left(C_i \middle| \bar{x}\right) = \frac{p\left(\bar{x} \middle| C_i\right) p(C_i)}{p(\bar{x})} = \frac{p\left(\bar{x}, C_i\right)}{p(\bar{x})} \tag{3.9}$$

If the prior probabilities of classes are known, and they are used to multiply with the class density functions the resulting algorithms are called minimum error classifiers, because they result in the theoretically minimum overall error:

$$g_{C_i}(\bar{x}) = p\left(\bar{x}, C_i\right) = p\left(\bar{x} \middle| C_i\right) p(C_i), \; i = 1..m \tag{3.10}$$

In practice, the prior class probabilities are often not known need to be estimated. Class conditional density functions (pdf's) also need to be estimated from a set of training samples. For a high dimensional feature space, when a new feature is added to the data, the Bayes error decreases, but at the same time the bias of the classification error increases. The reason of this increase is that more parameters need to be estimated from the same number of training samples. If the increase in the

bias of the classification error is more than the decrease in the Bayes error, then the use of the additional feature decreases the performance of the decision rule. This phenomenon is called the Hughes effect [3]. The larger the number of the parameters that need to be estimated, the more severe the Hughes phenomenon can be. Therefore, when the dimensionality of data and the complexity of the decision rule increase, the Hughes effect can become more severe. It is obvious that, linear classifiers such as minimum distance to mean (minimum Euclidean distance) are less affected by the Hughes effect than the quadratic classifiers such as the Gaussian maximum likelihood (GML) classifier [36,38]. The discriminatant function for the minimum distance to mean classifier is given by

$$g_{C_i}(\bar{x}) = (\bar{\boldsymbol{x}} - \bar{\mu}_i)(\bar{\boldsymbol{x}} - \bar{\mu}_i)^T,\ i = 1...m. \quad \textbf{(3.11)}$$

In addition, Fisher's linear discriminant classifier assumes that each class has the same covariance matrix called the common covariance matrix which can be calculated by using all available labeled samples (training samples) [36]. The Fisher's linear discriminant classifier is given by

$$g_{C_i}(\bar{x}) = (\bar{\boldsymbol{x}} - \bar{\mu}_i)\sum^{-1}(\bar{\boldsymbol{x}} - \bar{\mu}_i)^T,\ i = 1...m. \quad \textbf{(3.12)}$$

3.1.2 Expectation Maximization (EM)

Performance of a classifier is usually related to the degree of discrimination function complexity. More complex classifiers need much more labeled training samples to make a proper estimation of parameters used in the discrimination function. Especially in remote sensing, labeled samples are limited. This drawback affects classification accuracy in a negative way especially when the feature vector size increases. Parametric classifiers such as quadratic ones are much more affected by limited training samples. In order to enhance estimation of parameters, unlabeled samples can be incorporated together with limited labeled ones. In the following discussion, enhancement of Gaussian density function parameters (prior probabilities, mean vectors and covariance matrices) is achieved via expectation maximization (EM) algorithm [38].

When individual classes are multivariate Gaussian, the ML estimates of the parameters of the mixture density consisting of the *m* normal classes are considered. We assume i^{th} class n_i labeled training samples are available. We will denote these training samples by z_{ik} where *i* indicates the class *(i=1,...,m),* and *k* is the index of each particular sample. In addition, we assume that *N* unlabeled samples denoted by x_k are available to enhance the mixture density given by

$$p(\bar{\boldsymbol{x}}|\theta)=\sum_{i=1}^{m}\alpha_i p_i(\bar{\boldsymbol{x}}),\ i=1...m \tag{3.13}$$

The EM equations for approximating the ML estimates of the parameters of the mixture density are the following [39]:

$$\alpha_i^{t+1}=\frac{\sum_{k=1}^{N}\dfrac{\alpha_i^t p_i(\bar{x}_k|\bar{\mu}_i^t,\Sigma_i^t)}{p(\bar{x}_k|\theta^t)}}{N} \tag{3.14}$$

$$\bar{\mu}_i^{t+1}=\frac{\sum_{k=1}^{N}\dfrac{\alpha_i^t p_i(\bar{x}_k|\bar{\mu}_i^t,\Sigma_i^t)}{p(\bar{x}_k|\theta^t)}\bar{x}_k+\sum_{k=1}^{n_i}z_{ik}}{\sum_{k=1}^{N}\dfrac{\alpha_i^t p_i(\bar{x}_k|\bar{\mu}_i^t,\Sigma_i^t)}{p(\bar{x}_k|\theta^t)}+n_i} \tag{3.15}$$

$$\Sigma_i^{t+1}=\frac{\sum_{k=1}^{N}\dfrac{\alpha_i^t p_i(\bar{x}_k|\bar{\mu}_i^t,\Sigma_i^t)}{p(\bar{x}_k|\theta^t)}\left(\bar{\boldsymbol{x}}_{\boldsymbol{k}}-\bar{\mu}_i^{t+1}\right)\left(\bar{\boldsymbol{x}}_{\boldsymbol{k}}-\bar{\mu}_i^{t+1}\right)^T+\sum_{k=1}^{n_i}\left(\bar{z}_{ik}-\bar{\mu}_i^{t+1}\right)\left(\bar{z}_{ik}-\bar{\mu}_i^{t+1}\right)^T}{\sum_{k=1}^{N}\dfrac{\alpha_i^t p_i\left(\bar{x}_k|\bar{\mu}_i^t,\Sigma_i^t\right)}{p(\bar{x}_k|\theta^t)}+n_i} \tag{3.16}$$

where $\bar{\mu}_i^t$ and Σ_i^t are the mean vector and the covariance matrix of class *i* at iteration *t*. The parameter set θ^t contains all the prior probabilities, mean vectors and covariance matrices. The ML estimates are obtained by starting from an initial point in the parameter space and iterating through the above equations. Reasonable initial values are obtained by using the training samples alone. An important practical point is that, although in theory additional unlabeled samples should always improve the

classification accuracy, this might not always be the case in practice. The reason for this is the deviation of the real world situations from the models that are assumed. Therefore, additional care must be taken when supervised-unsupervised learning is used in practice. Designing a classifier by using the training sample alone and then trying to improve classification accuracy by enhancing statistics via incorporating unlabeled samples with labeled ones is an efficient indicator to show the contribution of enhancement of statistics. If the performance of the classifier is not satisfactory, then a new set of unlabeled samples is selected and used to enhance the statistics to reach more accurate classification results.

3.2 Nonparametric Methods

Classification algorithms based on statistics assume that data has a specific distribution, typically a Gaussian distribution. For real world data, such assumptions may not be valid. Additionally, statistical classifiers are parametric classifiers, and proper estimation of parameters is needed. Especially with limited number of labeled training samples, which is very common situation in remote sensing, there are additional difficulties involved in proper parameter estimation of class distributions. These difficulties get harder in high dimensional feature space as compared to low dimensional feature space. Therefore, some complementary methods such as dimensionality reduction and enhancing estimation of parameters are needed for parametric methods to get satisfactory results. In addition, increasing classification accuracy is not guaranteed by using methods for enhancing estimation of parameters. Therefore, nonparametric methods are widely used to overcome these classification problems summarized above. Main aim of nonparametric methods is to extract maximum information from limited number of labeled samples to make an appropriate decision. Directly using training samples is an important specification of the nonparametric methods to describe feature space. Another advantage of nonparametric methods is stability of obtained classification accuracies with dimensionality changes. Therefore, the Hughes effect is less harmful for nonparametric methods than parametric ones. Training process often takes more time for nonparametric methods, and that could be a disadvantage. Our proposed algorithm the BFDA is also a nonparametric classifier. Therefore, in this section we explain some nonparametric methods such as the k-Nearest Neighbor (KNN), grow

and learn (GAL) and self organizing map (SOM), which have some similarity with our proposed algorithm, the BFDA.

3.2.1 K-Nearest Neighbor (KNN)

The k-nearest neighbor rule is a technique of nonparametric pattern recognition that does not need knowledge about distribution of the patterns [40]. It is one of the simple and precise classification methods. The obtained error by the KNN algorithm often converges to the Bayes error. However, heavy computational load that is proportional to the number of samples, and the number of dimensions of the feature space is an important disadvantage of the algorithm. The original k-nearest neighbor algorithm does not need any training phase to make a decision, all available training samples are used for making decision. It is also called lazy classification method. Methods based on branch and bound methods have been proposed to define k-neighbors in a fast way [41]. There are also some methods which have been developed to decrease the number of training samples that are needed for distance calculation by dividing the space [42]. There are some methods for fast recognition using the KNN rule. These methods can be classified in space to two types. In some methods, the number of samples for distance calculation is limited, and in the other methods, the search space is limited. The first type of methods reduces computation time and space complexity, but the latter reduces only time complexity. To limit the number of samples for distance calculation, an effective subset is calculated from the training data set [43], or a new set is reconstructed for classification [44]. In recent years, some fast algorithms derived from KNN are widely used for giving proper responses to queries made to extract required information from databases [46]. In the following, we give a brief description of the KNN algorithm as follows. Given a point $\bar{\boldsymbol{x}}'$ in the N-dimensional feature space, an ordering function $\boldsymbol{f}_{\bar{\boldsymbol{x}}'} : \Re^N \to \Re$, is defined. A typical ordering function is based on the Euclidean metric:

$$\boldsymbol{f}_{\bar{\boldsymbol{x}}'}(\bar{\boldsymbol{x}}_j) = \boldsymbol{D}_j(\bar{\boldsymbol{x}}', \bar{\boldsymbol{x}}_j) = \left\| \bar{\boldsymbol{x}}' - \bar{\boldsymbol{x}}_j \right\| = \sqrt{\sum_{d=1}^{N} \left(x'(d) - x_j(d) \right)^2}, \; j = 1 \ldots n \; . \tag{3.17}$$

By means of an ordering function, it is possible to order the entire set of training samples $\bar{\boldsymbol{x}}_j$, $(j = 1, \ldots, n)$, with respect to $\bar{\boldsymbol{x}}'$. This corresponds to define a function

$r_{\bar{x}'} : \{1,...,n\} \to \{1,...,n\}$ that reorders the indexes of n training points of the feature space. This function can be defining recursively as follows:

$$r_{\bar{x}'}(k=1) = \arg\min_j f_{\bar{x}'}(\bar{x}_j) \; with \; j \in \{1,...,n\} \tag{3.18}$$

$$r_{\bar{x}'}(k=k') = \arg\min_j f_{\bar{x}'}(\bar{x}_j) \; with \; j \in \{1,...,n\} \; and \; j \neq r_{\bar{x}'}(1),...,j \neq r_{\bar{x}'}(k'-1) \; for \; k' = 2,...,n \tag{3.19}$$

In this way, $\bar{x}_{r_{\bar{x}'}(k)}$ is the point of the training set in the k^{th} position in terms of distance from $\bar{x}'$, namely the k^{th} nearest neighbor, and its distance from $\bar{x}'$ is written as

$$f_{\bar{x}'}(\bar{x}_{r_{\bar{x}'}(k)}) = D_j(\bar{x}', \bar{x}_{r_{\bar{x}'}(k)}) = \left\| \bar{x}' - \bar{x}_{r_{\bar{x}'}(k)} \right\| \tag{3.20}$$

where $y_{r_{\bar{x}'}(k)}$ is its class label.

Given the above definition, the decision rule of the KNN classifier for binary classification problem is defined by

$$kNN(\bar{x}') = sign\left(\sum_{j=1}^{k} y_{r_{\bar{x}'}(j)} \right). \tag{3.21}$$

Additionally, there are some basic issues with Euclidian distance which are important to make proper decisions:

1) Scaling of values: Distances should be relative, not absolute. Since each numeric attribute (features/bands) may be measured in different units, they should be standardized to have a mean value of 0 and variance 1.

2) Weighting of attributes:

- Manual weighting: Weights may be suggested by experts.

- Automatic weighting: Weights may be computed based on discriminatory power or other statistics.

Therefore, not all attributes are equally important, so some weighting of attributes may be appropriate. Taking w_d as the weight for feature (attribute/band) d, our distance metric becomes:

$$\boldsymbol{f}_{\bar{\boldsymbol{x}}'}(\bar{\boldsymbol{x}}_j) = \boldsymbol{D}_j(\bar{\boldsymbol{x}}', \bar{\boldsymbol{x}}_j) = \left\| \bar{\boldsymbol{x}}' - \bar{\boldsymbol{x}}_j \right\| = \sqrt{\sum_{d=1}^{N} w_d \left(x'(d) - x_j(d) \right)^2}, \; j = 1..n. \qquad \textbf{(3.22)}$$

3.2.2 Grow and Learn (GAL)

The Grow and Learn algorithm (GAL) can be thought as a variation of the KNN algorithm [22]. Instead of using all the training samples as nodes (prototypes), a subset of the training sample set is used as nodes in the GAL algorithm. In the learning phase, the members of the subset which are used as nodes are chosen from the whole training set. After the learning phase, some redundant nodes may occur, and a pruning procedure is applied to discard redundant nodes. Incremental style learning is used in the GAL algorithm [47]. As seen in Figure 3.1, the first layer of the network is the input layer. The total number of input units in the input layer is $\boldsymbol{N}$, which is equal to the size of the feature (attribute) vector. In the second layer, the prototypes are stored by the algorithm. During the training phase, new nodes can be added as new prototypes in the second layer to reach required training accuracy. For initialization of the network, randomly selected training samples for each class can be chosen and assigned as prototypes in the second layer. When the accuracy of the training reaches to a required value, some nodes in the second layer can be discarded by a pruning algorithm called forgetting in GAL. The weight vector corresponding to the unit $\boldsymbol{e}$ in the second layer is depicted as $\bar{\boldsymbol{w}}_{\boldsymbol{e}}$, and the connection between the input layer to the unit $\boldsymbol{e}$ in the second layer is depicted as $\boldsymbol{T}_{\boldsymbol{ec}}$. When $\bar{\boldsymbol{x}}$ is the input vector, the activation of a unit $\boldsymbol{e}$ in the second layer, $\boldsymbol{A}_{\boldsymbol{e}}$ involves the computation of the distance between $\bar{\boldsymbol{x}}$ and the weight vector of the unit $\boldsymbol{e}$, $\bar{\boldsymbol{w}}_{\boldsymbol{e}}$.

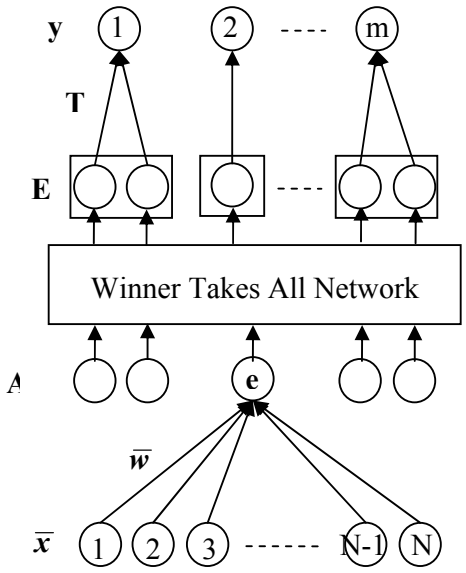


Figure 3.1: The GAL Network Structure

The Euclidian distance used as similarity measure to calculate the activation function for unit ***e*** and is given by

$$\boldsymbol{A_e = D(\bar{x}, \bar{w}_e)} = \|\bar{x} - \bar{w}_e\| = \sqrt{\sum_{d=1}^{N} \left(x(d) - w_e(d)\right)^2} \,. \qquad \textbf{(3.23)}$$

A winner-take-all type network chooses the closest node called the winner node to the input vector, and the label of the winner node is assigned. Mathematical description of the decision process can be shown as follows:

$$\forall \boldsymbol{e},\ \boldsymbol{A_e = D(\bar{x}, \bar{w}_e)} \qquad \textbf{(3.24)}$$

$$E_e = \begin{cases} 1, \ if \ A_e = \min_i(A_i) \\ 0, \ otherwise. \end{cases} \qquad \textbf{(3.25)}$$

$$T_{ec} = \begin{cases} 1, & if \ e \ is \ a \ prototype \ of \ the \ class \ c; \\ 0, & otherwise; \end{cases} \qquad \textbf{(3.26)}$$

$$C_c = \sum_e E_e \cdot T_{ec} . \quad \textbf{(3.27)}$$

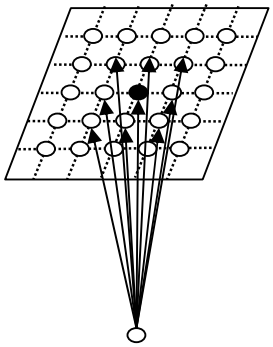
This structure is a neural network version of the KNN algorithm for k=1 (nearest neighbor). Only a part of the training set is used in the GAL algorithm. Selection of the subset procedure used in GAL is called learning in GAL, and discarding procedure for redundant members from a selected subset is called forgetting in GAL. A randomly selected training sample is assigned as a node (prototype, unit) in the second layer for each class as an initialization process. Then, samples from the training set are randomly selected, if the current network causes wrong decision according to rules described above by equations (3.24) thru (3.27), a randomly selected training vector assigned as an additional node (unit vector, prototype) in to layer 2 [48]. The procedure described above is applied iteratively until reaching a desired training accuracy or pre-defined iteration number. Therefore, during the learning process, the number of prototypes or nodes (units) in the second layer increases. The learning process is an online process. After the learning process, a pruning procedure is applied to discard redundant nodes from the second layer. This pruning procedure is called forgetting in GAL, and this process is an off-line process. For forgetting in GAL, one node is randomly selected from the second layer, and applied to the network as an input. Then, the decision of the network is obtained with temporarily forgetting this randomly selected node. If the network gives right decision, then this forgetting node is discarded permanently from the second layer. Otherwise, this node is kept as a necessary node and used in the final decision process.

3.2.3 Self Organizing Map (SOM)

Research on the cerebral cortex leads to decision makers called self organizing maps (SOMs). There are different sensory inputs that are mapped on to corresponding areas of the cerebral cortex with huge number of neurons. Therefore, a part of cerebral cortex which has a pre-defined task can be simulated as a self organizing map.

Self organizing maps are based on competitive learning; therefore only one output neuron is activated by the algorithm at a time and the activated neuron is called the

winning neuron. In a self organizing map, the neurons are placed at the nodes of a grid which may be one or two dimensional. In this way, a self organizing map (SOM) is characterized by the formation of a topographic map. Kohonen's self organizing map is the first such artificial neural network [19]. In Figure 3.2, Kohonen's feature-mapping model is depicted for a rectangular grid which shows the topographic characterization of the network as a second layer in the network. A group of neurons is located on to a two-dimensional grid in Figure 3.2. This grid could be formed with different geometric structure and could be of different dimensionality. Mostly, two dimensional grids are used. A group of neurons was placed on to a hexagonal grid by Kohonen, motivated by shape similarity with real biological structures [49]. In this section, we assume that the grid is a two-dimensional rectangular grid as seen in Figure 3.2.



$\bar{x}$: Input feature vector

Figure 3.2: Kohonen's Feature-Mapping Model

Randomly selected training vectors are used as input to the SOM. Let N be the dimension of the input vector $\bar{x}$. The synaptic weight vector of each neuron in the network depicted as a circle in Figure 3.2 has the same dimension as the input training sample. Therefore, all neurons have N dimensional weights as well. The synaptic weight vector of neuron i can be written as follows:

$$\bar{w}_i = \begin{bmatrix} w_i(1) & w_i(2) & \cdots & w_i(N) \end{bmatrix}, \ (i = 1...n) \qquad \textbf{(3.28)}$$

where n is the total number of neurons placed on the grid. To find a most similar neuron in the network with input training sample $\bar{x}$, the Euclidian distance can be used:

$$D_j = D(\bar{x}, \bar{w}_j) = \left\| \bar{x} - \bar{w}_j \right\| = \sqrt{\sum_{d=1}^{N} \left(x(d) - w_j(d) \right)^2}, \quad (j = 1, ..., n). \tag{3.29}$$

Let w be the index of the weight vector which corresponds to the best matching neuron on the grid called winning neuron:

$$w = \arg\min \left\{ D_j \right\} \tag{3.30}$$

Not only winning neuron but also neurons which are topologically neighbor to the winning neuron are adapted during the process. Therefore, a neighborhood function is needed to describe the area in which the adaptation is applied. In Figure 3.2, adaptation is applied on one-neighborhood of the winning neuron in addition to the winning neuron itself. One desired specification of the neighborhood function is to be a decreasing function when the distance between the winning neuron and the neuron which is in the neighborhood of the winning neuron, is increasing. Another desired specification for the neighborhood function is being decreasing function when the iteration number is increased. A neighborhood function that covers the requirements listed above is depicted by

$$h_{j,w}(t) = exp\left(- \frac{d_{j,w}^2}{2\sigma^2(t)} \right), \quad (t = 0, 1, ...) \tag{3.31}$$

where t denotes the iteration number, $d_{j,w}$ is the distance between winning neuron and neuron j in the grid, and for a two-dimensional grid, $d_{j,w}^2$ can be calculated by

$$d_{j,w}^2 = \left\| \bar{r}_j - \bar{r}_w \right\|^2 \tag{3.32}$$

where the discrete vector $\bar{r}_j$ defines the position of the neuron j, and $\bar{r}_w$ defines the position of the winning neuron. $\sigma(t)$ is the width of the topological neighborhood function $h_{j,w}$ which makes the neighborhood function a decreasing function by time (iteration), and can be chosen as an exponential function given by

$$\sigma(t) = \sigma_0 exp\left(-\frac{t}{\tau_1}\right), \quad (t = 0,1,...) \tag{3.33}$$

where σ_0 is the initial value of the σ and τ_1 is the time constant. Adaptation expression during learning can be written as follows:

$$\bar{w}_j(t+1) = \bar{w}_j(t) + \eta(t) \cdot h_{jw}(t) \cdot (\bar{x} - \bar{w}_j(t)) \tag{3.34}$$

which is applied the neurons in the topological neighborhood of the winning neuron. $\eta(t)$ is a descending function of time and is called the learning rate. A good choice for it is given by

$$\eta(t) = \eta_0 e^{-t/\tau_2}, \quad (t = 0,1,2,...) \tag{3.35}$$

where τ_2 is another time constant of the SOM algorithm.

It is useful to mention some values chosen for the parameters in practical implementations. Some important hints are given in Kohonen's paper for numerical examples [19]. It is also useful to separate adaptation process in to two phases. They are ordering and convergence phases. At the beginning, randomly chosen training samples can be assigned to the neurons which lie on the grid. One important issue in initialization is to choose training samples different from each other. Therefore, every neuron should take a unique value at the beginning. Another suggestion is to assign small values to the neurons as initial values. The ordering phase is the first phase of the adaptation process. It can be chosen as many as 1000 iterations or possibly more. In conclusion, learning rate and neighborhood function are important considerations for satisfactory convergence. The learning rate $\eta(t)$ should begin with a value close to 0.1, and should decrease during the learning, but should remain

above 0.01. To support these considerations related to parameters, the initial values of the learning rate and the time constant can be chosen as $\eta_0 = 0.1$ and $\tau_2 = 1000$, respectively. The neighborhood function $h_{j,w}(t)$ should cover all neurons which lie on the grid, then shrink as the iterations increase. At the end of the ordering phase, the neighborhood function should cover only a couple of neurons near the winning neuron. The time constant τ_2 can be chosen as

$$\tau_2 = \frac{1000}{\log \sigma_0}. \quad \textbf{(3.36)}$$

The final statistical accuracy of the mapping depends on the number of iterations used in the convergence phase, which should be reasonably large. Therefore, the number of iterations should be at least 500 times the number of neurons used in the network. Typically 100,000 iterations can be used but for fast learning 10,000 iterations may be enough. Additionally, learning rate should be maintained on the order of 0.01 during the convergence phase, and should not be decreased to 0.

Cross validation can also be used to specify appropriate parameters. In addition, a pattern search algorithm is helpful to find out proper values of parameters.

3.3 Kernel Methods

In recent years, kernel methods are widely used in remote sensing applications, because of their advantages in high dimensional feature spaces [13,51-54]. Linear discriminant functions are well known. Their simple mathematical description makes the linear functions attractive but using linear functions in the original feature space is often not satisfactory when classification is not linearly separable in the original feature space. Kernel methods map the original feature space into a higher dimensional feature space, and classification problem in this new high dimensional feature space can be linearly separable. This is visualized in Figure 3.3.

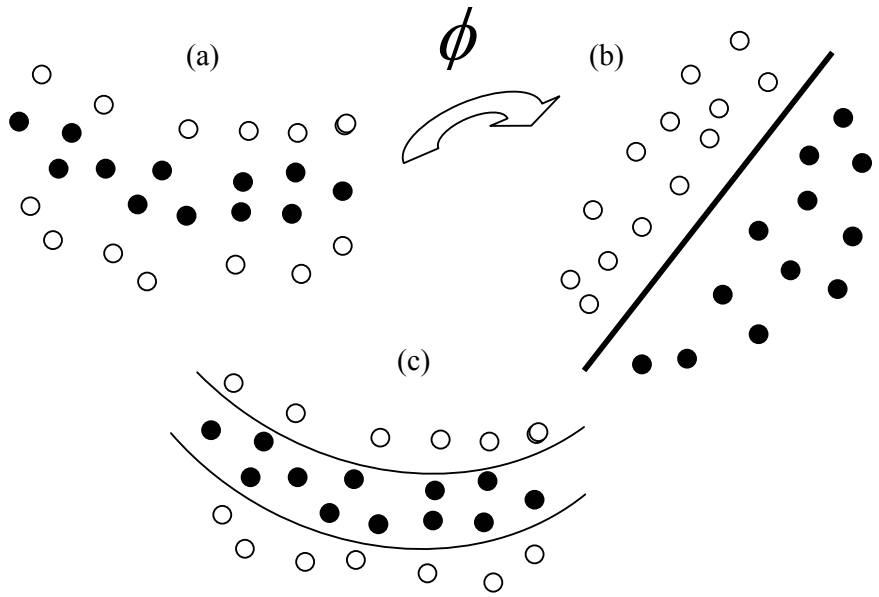


Figure 3.3: a) Linearly Inseparable Original Feature Space b) Mapped Feature Space Via $\phi(.)$ is Linearly Separable, c) Using Kernel Functions Makes Discriminant Function Nonlinear in the Original Space

Replacing inner products by Mercer kernels is a major idea of the kernel methods. In this way, the linear discriminant functions produce nonlinear decision boundaries in the original feature space [50]. In this section, a brief summary of kernel methods whose best known type is the support vector machine (SVM) is given. SVMs have become very popular classification tools in remote sensing applications, especially due to their satisfactory results in high dimensional feature spaces. Classification accuracies obtained by SVMs often give the highest results. Therefore, comparing classification results between our proposed algorithm, the BFDA, and SVMs is a valuable approach [21].

Assume that X is a set of input feature vectors in a N dimensional feature space, and Y is a label set of the corresponding input feature vectors. Classification can be considered as a functional transformation described by $f : X \subseteq \Re^N \to \Re$. Assume a training input is $\bar{\boldsymbol{x}} \in \boldsymbol{X}$ and its possible class labels are $y \in \{-1, +1\}$. This is a binary classification problem. $\bar{\boldsymbol{x}}$ can be assigned to the positive class if $f(\bar{\boldsymbol{x}}) \geq 0$, and to the negative class otherwise. If the discriminant function $f(\bar{\boldsymbol{x}})$ is considered as a linear function, it can be written by

$$f(\bar{x}) = \langle \bar{w} \cdot \bar{x} \rangle + b = = \sum_{i=1}^{N} w_i x_i + b \quad \textbf{(3.37)}$$

where the inner products of vectors $\bar{w}$ and $\bar{x}$ is depicted as $\langle \bar{w} \cdot \bar{x} \rangle$. Functional margin with respect to a hyperplane ($\bar{w}$,b) for an input $\bar{x}_i$ can be defined as follows:

$$\gamma_i = y_i \left(\langle \bar{w} \cdot \bar{x}_i \rangle + b \right) \quad \textbf{(3.38)}$$

$\gamma_i > 0$ implies correct classification. Additionally, when we use normalized linear discriminant function $\left(\frac{\bar{w}}{\|\bar{w}\|}, \frac{b}{\|\bar{w}\|} \right)$, then geometric margin can be defined instead of functional margin [23]. The physical meaning of the geometric margin of the hyperplane is related to the Euclidean distances of the feature vectors from the decision boundary in the input space. During learning, if $\gamma_i \leq 0$, then the weight vector is adapted as in the primal form of the perceptron learning algorithm:

$$\bar{w}_{k+1} = \bar{w}_k + \eta y_i \bar{x}_i, \; i = 1, ..., m \quad \textbf{(3.39)}$$

where *k* denotes the iteration, *i* is the index of the training sample and *m* is the total number of training samples. The dual form of the decision function in (3.40) can be derived by substituting (3.39) in (3.37):

$$f(\bar{x}) = \sum_{i=1 \ldots m} \alpha_i y_i \langle \bar{x}_i \cdot \bar{x} \rangle + b \quad \textbf{(3.40)}$$

The dual form of the discriminant function is important, and is used in kernel methods. The main aim of the kernel methods is to partition the nonlinearly separable feature space by using linear discriminant functions in a higher dimensional feature space. Assume that ϕ is a function which maps the original input feature space to a higher dimensional feature space where the classification problem is probably linearly separable. The discriminant function for the mapped space can be written as

$$f(\bar{x}) = \sum_{i=1\ldots m} \alpha_i y_i \langle \phi(\bar{x}_i) \cdot \phi(\bar{x}) \rangle + b \,. \tag{3.41}$$

The kernel trick is defined by

$$K(\bar{x}_i, \bar{x}) = \langle \phi(\bar{x}_i) \cdot \phi(\bar{x}) \rangle \tag{3.42}$$

Therefore $K(\bar{x}_i, \bar{x})$ is used instead of $\langle \phi(\bar{x}_i) \cdot \phi(\bar{x}) \rangle$. Kernels must verify the Mercer condition to be valid kernels.

3.3.1 Support Vector Machines (SVMs)

The SVM approach consists of finding the optimal hyperplane that maximizes the distance between the closest training sample and the separating hyperplane. This distance is given by $2/\|\bar{w}\|$ by using geometric margin. The generalization capability of the SVM approach is strictly related to the concept of margin. The larger the margin is the higher is the expected generalization [50].

The optimal hyperplane can be determined as the solution of the following quadratic programming problem for a linearly separable case:

$$\begin{aligned} &minimize: \ \frac{1}{2}\|\bar{w}\|^2 \\ &subject\ to: \ y_i\left(\langle \bar{w} \cdot \bar{x}_i \rangle + b\right) \geq 1, \qquad i = 1, 2, \ldots, m \end{aligned} \tag{3.43}$$

This optimization problem can be converted into the dual problem by using a Lagrangian formulation:

$$\begin{aligned} &maximize: \ \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \bar{x}_i \cdot \bar{x}_j \rangle \\ &subject\ to: \ \sum_{i=1}^{m} \alpha_i y_i = 0 \ and \ \alpha_i \geq 0, \ i = 1, 2, \ldots, m \end{aligned} \tag{3.44}$$

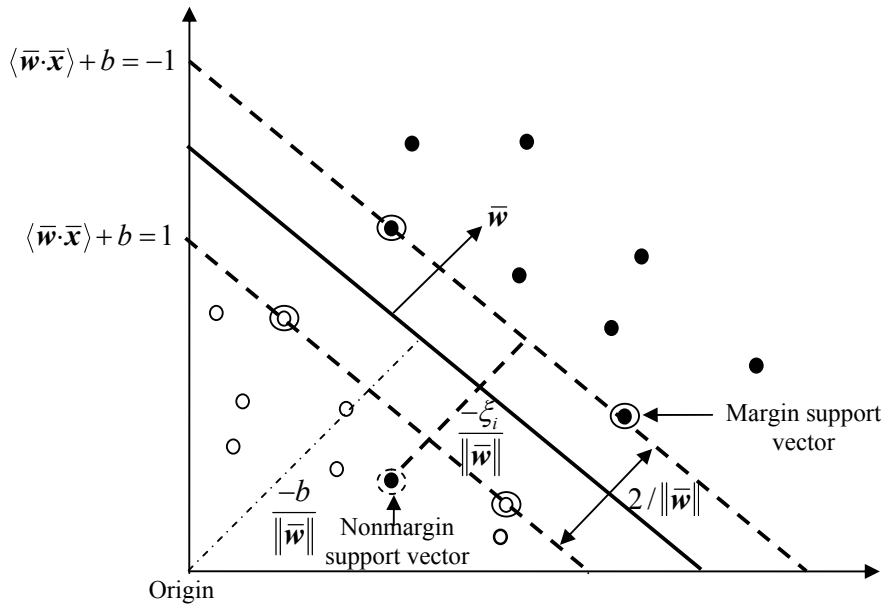


Figure 3.4: Optimal Separating Hyperplane in SVM for a Linearly Nonseparable Case

The Lagrange multipliers α_i's can be estimated using quadratic programming (QP) techniques [50]. The discriminant function which specifies the optimal hyperplane can be written as follows:

$$f(\bar{x}) = \sum_{i \in S} \alpha_i y_i \langle \bar{x}_i \cdot \bar{x} \rangle + b \qquad \textbf{(3.45)}$$

where S is the subset of training samples corresponding to nonzero α_i's. Nonzero Lagrange multipliers are thus indicators of the significant training samples which determine the discriminant function. The training samples with nonzero are called support vectors.

The linear SVM can be used in the nonseparable case as well. The classification problem in remote sensing is generally nonseparable. Therefore, the concept of optimal separating hyperplane has been generalized as the solution that minimizes a cost function that support both margin maximization and error minimization. The new cost function is defined by

$$\Psi(\bar{w}, \xi) f(\bar{x}) = \frac{1}{2} \|\bar{w}\| + C \sum_{i=1}^{m} \xi_i \qquad \textbf{(3.46)}$$

where ξ_i's are called slack variables and C controls the penalty assigned to errors. Larger the C value is, the higher is the penalty associated to misclassified samples. The minimization of the cost function is subject to the following constraints:

$$y_i\left(\left\langle \bar{\boldsymbol{w}} \cdot \bar{\boldsymbol{x}}_{\boldsymbol{i}} \right\rangle + b\right) \geq 1 - \xi_i, \quad i = 1, 2, ..., m \tag{3.47}$$

$$\xi_i \geq 0, \quad i = 1, 2, ..., m \tag{3.48}$$

In the nonseparable case, there are two types of support vectors: margin support vectors that lie on the hyperplane margin and nonmargin support vectors that fall on the wrong side of the margin.

Using kernel functions makes SVM a nonlinear classifier in the original input feature space. This can be achieved by replacing the inner product in the original space $\left\langle \bar{\boldsymbol{x}}_{\boldsymbol{i}} \cdot \bar{\boldsymbol{x}}_{\boldsymbol{j}} \right\rangle$ with the inner product in the transformed space $\left\langle \boldsymbol{\phi}(\bar{\boldsymbol{x}}_{\boldsymbol{i}}) \cdot \boldsymbol{\phi}(\bar{\boldsymbol{x}}_{\boldsymbol{j}}) \right\rangle$ as explained at the beginning of the Kernel Methods section. A kernel function that satisfies the Mercel's theorem allows calculation of inner products without calculation of mapping function. Using kernel function allows simplifying the solution of the dual problem; The optimization formulation can be written as follows:

$$\begin{aligned} &maximize: \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K\left(\bar{\boldsymbol{x}}_{\boldsymbol{i}} \cdot \bar{\boldsymbol{x}}_{\boldsymbol{j}}\right) \\ &subject\ to: \sum_{i=1}^{m} \alpha_i y_i = 0 \ and\ 0 \leq \alpha_i \leq C, \quad i = 1, 2, ..., m \end{aligned} \tag{3.49}$$

Using kernel function instead of inner product in mapping space allows the discriminant function in the original input feature space be written as

$$f(\bar{\boldsymbol{x}}) = \sum_{i \in S} \alpha_i y_i K\left(\bar{\boldsymbol{x}}_{\boldsymbol{i}} \cdot \bar{\boldsymbol{x}}\right) + b \tag{3.50}$$

Type of kernel function affects the discriminant function. A common example of kernel type is the Gaussian radial basis function given by

$$K\left(\bar{x}_i \cdot \bar{x}\right) = \exp\left(-\gamma \left\|\bar{x}_i - \bar{x}\right\|^2\right) \quad \textbf{(3.51)}$$

where γ is a parameter inversely proportional the width of the Gaussian kernel. Additionally, polynomial function of order p can be used as a kernel function as follows:

$$K\left(\bar{x}_i \cdot \bar{x}\right) = \left[\bar{x}_i \cdot \bar{x} + 1\right]^p. \quad \textbf{(3.52)}$$

In this thesis, Linear SVM and Gaussian Radial Basis Function SVM (RBF-SVM) are used as challenging classifiers to compare with our proposed algorithm, the BFDA. SVM formulation has been taken from the literature directly without any contribution. Proper parameters discovered for linear SVM (C) and RBF-SVM (C, γ) in the experiments would be useful for both multispectral and hyperspectral data classification [51]. We applied both ten-fold cross validation technique and pattern search technique to find out proper parameters for SVM classifiers.

The SVM classifier is initially a binary classifier. Therefore some methods are needed to extend SVM in a multi-class problem. To achieve this, one simple but valuable method is based on combining binary classification results with a proper consensual rule such as majority voting. In the literature, there are two techniques widely used. They are One-Against-All (OAA) Strategy and One- Against-One (OAO) Strategy [13,51]. It is possible to construct hierarchical tree based structures as well. In the literature, OAO strategy takes more computational time than OAA Strategy that has been reported [13]. Additionally, more classification accuracies are usually obtained by using OAA strategy. Therefore OAO strategy is chosen in our experiments to get highest results obtained by SVM classifiers.

4. BORDER FEATURE DETECTION AND ADAPTATION (BFDA)

Performance of a classifier is strictly related to training samples in supervised learning [52,53]. A desirable classifier is expected to achieve sufficient classification accuracy while keeping rare class members correctly classified in the same process. Achieving this aim is not a trivial task, especially when the training samples are limited in number. Lack of sufficient number of training samples decreases generalization performance of a classifier. Especially in remote sensing, collecting training samples is a costly and difficult process. Therefore, a limited number of training samples is obtained in practice. A heuristic metric is that the number of training samples for each class should be at least 10-30 times the number of attributes (features/bands) [54,55]. It is true that this may be achieved for multispectral data classification. However, for hyperspectral data which has at least 100-200 bands, sufficient number of training samples can not be collected. Normally, when the number of bands used in the classification process increase, precise detailed class determination is expected. For high dimensional feature space, when a new feature is added to the data, classification error decreases, but at the same time the bias of the classification error increases [31]. If the increment of the bias of the classification error is more than the reduction in classification error, then the use of the additional feature degreases the performance of the decision rule. This phenomenon is called the Hughes effect [3], and it may be much more harmful with hyperspectral data than multispectral data.

Additional effort can be focused upon determining efficient samples which are much more effective to use for forming the decision boundary [56]. Structure of discriminant functions used by classifiers can give some important clues about the positions of the effective samples in the feature space. The training samples near the decision boundaries can be considered effective samples. The problem would be to specify the positions of these samples in the image. In crop mapping applications, some samples near to parcel borders (spatial boundary in the image) are assumed to

be samples with mixed spectral responses. Samples compromising mixed spectral responses can be taken into consideration to determine effective samples. Therefore, same classification accuracy can be achieved by using lower number of effective samples than samples collected from pure pixels [57]. One of the design considerations of the classifier should be benefiting from training samples which are near the decision boundaries [50].

It is obvious that the training stage is very important in supervised learning and affects generalization capability of the classification algorithms. In some cases, not all training samples are useful; some of them can even be detrimental to classification [58]. Therefore some samples are discarded from training set (noisy samples) or their intensity values can be fine tuned (noise reduction) by using appropriate spatial filtering operations (such as mean filter) to enhance generalization capability of the classification algorithm [20]. This kind of special filtering with small window size (1x2) is also applied to parcel borders in agricultural areas to find appropriate intensity values of the spectral mixture type pixels [57].

The training process should not be biased. Equal number of training samples should be selected for each class if possible. In practice, this may not be possible. In addition, for neural network classifiers, the training process can be related to the order of the input training samples. To reduce these dependencies for making final decision unbiased, a consensual rule [17,18] can be applied to combine results obtained from a pool of classifiers. This process can also be combined with cross validation to improve generalization capability of the classifier.

Our motivation in this thesis is to overcome some of these general classification problems, by developing a classification algorithm which is directly based on the available training data rather than on the underlying statistical data distribution. Our proposed algorithm, the BFDA, uses border feature vectors near the decision boundaries which are adapted to make a precise partitioning in the feature space by using maximum margin principle.

Many supervised classification techniques have been used for multispectral and hyperspectral data classification, such as the maximum-likelihood classification (MLC), neural networks (NNs) and support vector machines (SVMs). Practical

implementational issues and computational load are additional factors used to evaluate classification algorithms.

Statistical classification algorithms are fast and reliable, but they assume that the data has a specific distribution. For real world data, these kinds of assumptions may not be sufficiently accurate, especially for low probability classes. For high dimensional feature space, first and second order statistics (mean and covariance matrix) could not be accurately estimated. The total number of parameters in the covariance matrix is equal to the square of the feature size. Therefore, proper estimation of covariance matrix is a difficult challenge. To overcome proper parameter estimation problem, some valuable methods are introduced in the literature. Covariance matrix regularization is one of the methods that can be applied to estimate more accurate covariance matrix. In this method, sample and common covariance matrices are combined in some way to make proper covariance matrix estimation [4,5]. Enhancing statistics by using unlabeled samples iteratively is another method to reduce the effects of poor statistics. The expectation maximization (EM) algorithm can be used to enhance statistics [7]. In hyperspectral data, neighbor bands are usually highly correlated. Methods such as defusing effects of Hughes phenomena in hyperspectral data, dimensionality reduction methods for increasing class discrimination such as discriminate analysis feature extraction (DAFE) [31], and decision boundary feature extraction (DBFE) [8] can be applied. Working in high dimensional feature space directly is also problematic for these two methods. Therefore, subset feature selection via band grouping such as projection pursuit (PP) [9] can be used before DAFE and DBFE.

Non-parametric classification methods are robust with both multispectral and hyperspectral data. Therefore, Hughes effect is less harmful for nonparametric methods than parametric ones. The K-nearest neighbor rule is one of the simple and efective classification techniques in nonparametric pattern recognition that does not need knowledge of distribution of the patterns [40], but it is also sensitive to the presence of noise in the data. Neural networks are widely used in the analysis of remotely sensed data. There is a variety of network types used in remote sensing such as multilayer perceptron or feed forward networks trained with the backpropagation algorithm [52]. There are also some additional classification schemes to improve classification performance of neural networks to simplify the

complex classification problem by accepting or rejecting samples such as parallel, self-organizing hierarchical neural networks (PSHNNs) [14]. By using parallel stages of neural network modules, hard vectors are rejected to be processed in the succeeding stage modules, and this rejection scheme is effective in enhancing classification accuracy. Consensual classifiers are related to PSHNNs, and also reach high classification accuracies [15-18].

In recent years, kernel methods such as support vector machines (SVMs) have demonstrated good performance in multispectral and hyperspectral data classification [13,51,59]. Some of the drawbacks of SVMs are the necessity of choosing an appropriate kernel function and time-intensive optimization. In addition, the assumptions made in the presence of samples which are not linearly separable are not necessarily optimal. It is also possible to use composite kernels for remote sensing image classification [59] to reach higher classification accuracies.

In this thesis, a new classification algorithm well suited for classification of remote sensing images is developed with a new approach to choosing and adapting border feature vectors with the training data. This approach is especially effective when the information source has a limited amount of data samples, and the distribution of the data is not necessarily Gaussian. Training samples closer to class borders are more prone to generate misclassification, and therefore are significant feature vectors to be used to reduce classification errors. The proposed classification algorithm searches for such error-causing training samples in a special way, and adapts them to generate border feature vectors to be used as labeled feature vectors for classification [21].

The BFDA algorithm can be considered in two parts. The first part of the algorithm consists of defining initial border feature vectors using class centers and misclassified training vectors. With this approach, a manageable number of border feature vectors are achieved. The second part of the algorithm is adaptation of border feature vectors by using a technique similar to the learning vector quantization (LVQ) algorithm [19]. In this adaptation process, the border feature vectors are adaptively modified to support proper distances between them and the class centers, and to increase the margins between neighboring border features with different class labels. The class centers are also adapted during this process. Subsequent classification is based on labeled border feature vectors and class centers. With this

approach, a proper number of feature vectors for each class is generated by the algorithm. The flow graph of the BFDA is depicted in Figure 4.1.

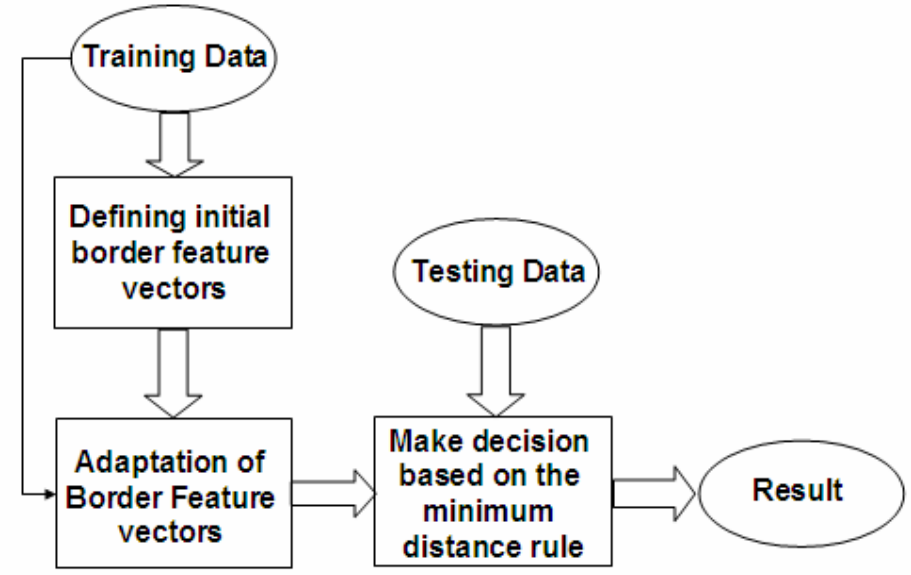


Figure 4.1: Flow Graph of the BFDA Algorithm

Partitioning feature space by using some selected reference vectors from a training set is a well-known approach in pattern recognition [22]. In general, there is an optimal number of reference vectors which can be used. More number of reference vectors above the optimal number may cause reduction of generalization performance. To avoid performance reduction, additional efforts should be taken to discard redundant reference vectors. An example of such a procedure is given in the grow and learn algorithm (GAL) [48].

We propose a new approach to reference vector selection called border feature detection. In developing such an approach, the selected reference vectors are required to satisfy certain geometric considerations. For example, a major property of SVMs is to optimize the margin between the hyperplanes characterizing different classes [50]. The training vectors on the hyperplanes are called support vectors. In the proposed algorithm, the same type of consideration leads to the positions of the reference vectors selected from the training set to be adapted so that they become closer to the decision boundaries while the reference vectors from different classes are as far away from each other as possible. These adapted reference vectors are called border feature vectors.

4.1 Border Feature Detection

The border feature detection algorithm is developed by considering the following requirements:

1. Border feature vectors should be adapted so that they are as close as possible to the decision boundaries.
2. The initial selection procedure is desired to be automatic, with a reasonable number of initial border feature vectors.
3. Every class is represented with an appropriate number of border feature vectors to properly represent the class.

In order to choose the initial border feature vectors, the class centers are used. A particular class center is defined as the nearest vector to its class mean. Using class center instead of class mean is a precaution for some classes which are spread in a concave form in the feature space.

Assuming a labeled training data set $\{(\bar{\boldsymbol{x}}_1, y_1), (\bar{\boldsymbol{x}}_2, y_2), \cdots, (\bar{\boldsymbol{x}}_n, y_n)\}$ where the training vectors are $\bar{\boldsymbol{x}}_i \in \mathbb{R}^N, i = 1, ..., n$, the class labels are $y_i \in \{1, 2, \cdots, m\}$, n is the total number of training samples, and m is the number of classes, the class means are calculated as follows:

$$\bar{\boldsymbol{m}}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \bar{\boldsymbol{x}}_j, \{\bar{\boldsymbol{x}}_j \mid y_j = i, \quad i = 1, \cdots, m\} \tag{4.1}$$

where n_i is the total number of training samples for class *i*. The class center $\bar{\boldsymbol{c}}_i$ for class *i* is defined as follows:

$$\bar{\boldsymbol{c}}_i = \bar{\boldsymbol{x}}_k, \left\{ \begin{array}{l} k = \quad \arg\min \left\{ \boldsymbol{D}_j \right\} \\ \qquad (1 \le i \le m), (1 \le j \le n) \end{array}, \boldsymbol{D}_j(\bar{\boldsymbol{m}}_i, \bar{\boldsymbol{x}}_j) = \left\| \bar{\boldsymbol{m}}_i - \bar{\boldsymbol{x}}_j \right\| = \sum_{d=1}^{N} \sqrt{\left(m_i(d) - x_j(d) \right)^2}, \left\{ \bar{\boldsymbol{x}}_j \,\middle|\, y_j = i \right\} \right\} \tag{4.2}$$

Let $\mathbf{B}^t$ be a set of border feature vectors in the feature space. For t=0, $\mathbf{B}^0$ is the set of initial border feature vectors chosen as a combination of some initial border feature vector sets $\mathbf{B}_i$:

$$\mathbf{B}^0 = \bigcup_{0 \le i \le m} \mathbf{B}_i \tag{4.3}$$

$\mathbf{B}_0$ is chosen as the set of initial class centers. They can be written together with their class labels as

$$\mathbf{B}_0 = \{(\bar{\boldsymbol{c}}_1, y_1), (\bar{\boldsymbol{c}}_2, y_2), \cdots, (\bar{\boldsymbol{c}}_m, y_m)\} = \{(\bar{\boldsymbol{b}}_1, y_1), (\bar{\boldsymbol{b}}_2, y_2), \cdots, (\bar{\boldsymbol{b}}_m, y_m)\}. \tag{4.4}$$

The number of members for the set $\mathbf{B}_0$ is $m_0 = m$. Additionally, $\mathbf{B}_i$, $i = 1 \cdots m$ is chosen as a set of initial border feature vectors detected for class *i* as discussed next. Assume that the total number of detected border feature vectors is m_i for class *i*. In this assignment procedure, $\mathbf{R}_i = \mathbf{B}_0 \cup \mathbf{B}_i$ is called the reference set for class *i,* and the number of members for the reference set is m_0+m_i . At the beginning of the detection procedure for every class, $\mathbf{B}_i(t=0)=\varnothing$, $_s(\mathbf{B}_i)=m_i=0$, $_s(\mathbf{R}_i)=m$ and therefore, $\mathbf{R}_i(t=0) = \mathbf{B}_0 \cup \mathbf{B}_i = \mathbf{B}_0$. During the detection process for class i=q, every member of the training samples belonging to class q is randomly selected only once as an input. Assume that $(\bar{\boldsymbol{x}}_k, y_k = q)$ is selected. Then, the Euclidean distances calculated between this sample and the current reference set members are given by

$$\boldsymbol{D}_j(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{b}}_j) = \left\| \bar{\boldsymbol{x}}_k - \bar{\boldsymbol{b}}_j \right\|, \; j = 1..(m_0 + m_q) \tag{4.5}$$

The winning border feature vector is chosen by

$$w = \arg\min \left\{ D_j \right\} \tag{4.6}$$

If the label of the winning border feature vector $\bar{\boldsymbol{b}}_w$ is $y_w \neq y_k = q$, then $(\bar{\boldsymbol{x}}_k, y_k = q)$ is chosen as a new reference vector for class q and added to the reference vector set.

This can be written as $\mathbf{R}_{i=q}(t) = \mathbf{R}_{i=q}(t-1) \cup \{(\bar{x}_k, y_k = q)\}$. This procedure is somewhat similar to the ART1 algorithm [61]. The procedure for selecting border feature vectors is applied with all the classes.

We define b as the total number of border features, and $m_i, i = 1,...,m$ as the number of detected border feature vectors for class i, with $m_0 = m$ being the number of classes. Then, the following is true:

$$b = \sum_{i=0}^{m} m_i = m + \sum_{i=1}^{m} m_i \qquad \textbf{(4.7)}$$

As an example, a binary classification problem in a two-dimensional feature space is depicted in Figure 4.2. In this figure, the training samples shown with symbols + and x are for classes 1 and 2, respectively. The samples detected as initial border feature vectors are shown as circles. The initial decision boundary based on only the class centers, $\mathbf{B}_0$, is shown as a line. The border feature vectors other than the class centers are selected from the misclassified samples, as seen in Figure 4.2.

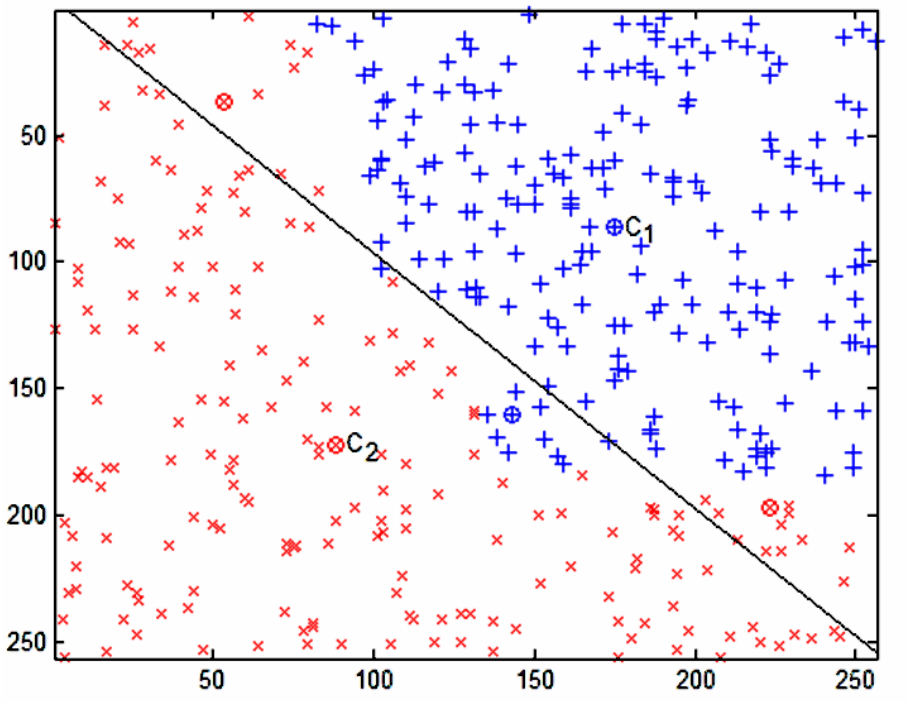


Figure 4.2: Binary Classification Problem: Class Centers and Selected Initial Border Features Depicted as Circles, and the Initial Border Line between Classes when the Decision is Made Based on Only Class Centers

In Figure 4.3, all the feature vectors, $\mathbf{B}^0$, are used to partition the feature space. The next step is to adapt the border feature vectors so that they more accurately represent the class boundaries. Additionally in the adaptation procedure, if any new border feature requirement occurs, additional border feature vectors are added to the border feature vector set.

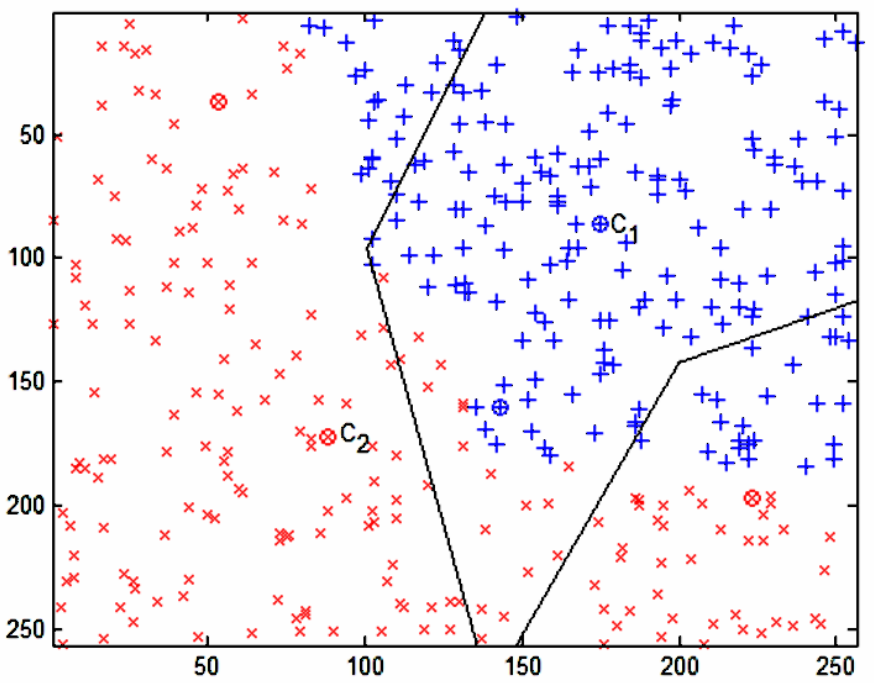


Figure 4.3: Partitioning of the Two-Dimensional Feature Space by Using Initial Border Feature Vectors Obtained at the end of the Border Feature Selection Procedure

4.2 Adaptation Procedure

In the adaptation process, competitive learning principles are applied as follows: The initial border feature vectors, $\mathbf{B}^0$ are adaptively modified to support maximum distance between the border feature vectors and their means, and to increase the margins between neighboring border features with different class labels. The means of border feature vectors to be used during adaptation are given by

$$\bar{\boldsymbol{m}}_i = \frac{1}{m_i + 1} \sum_{j=1}^{b} \bar{\boldsymbol{b}}_j, \{\bar{\boldsymbol{b}}_j \mid y_j = i, \quad i = 1, \cdots, m\} \tag{4.8}$$

$$\mathbf{M}^0 = \{(\bar{\boldsymbol{m}}_1, y_1), (\bar{\boldsymbol{m}}_2, y_2), \cdots, (\bar{\boldsymbol{m}}_m, y_m)\} \tag{4.9}$$

The means of border feature vectors are not taken in to account in the final decision process. At the end of the adaptation process, the means of border feature vectors are redundant. During the adaptation process, they are used to decide whether new border feature vectors should be generated. They are also adapted during learning due to the changes of border feature vectors.

The strategy of adaptation can be explained as follows: a nearest border feature vector $\bar{\boldsymbol{b}}_{\boldsymbol{w}}(t)$ which causes wrong decision should be farther away from the current training vector. On the other hand, the nearest border feature vector $\bar{\boldsymbol{b}}_{\boldsymbol{l}}(t)$ with the correct class label should be closer to the current training vector. The corresponding adaptation process used has some similarity with the LVQ algorithm [19]. The adaptation procedure is depicted as a flow graph in Figure 4.4.

Let $\bar{\boldsymbol{x}}_j$ be one of the training samples with label y_j. Assume that $\bar{\boldsymbol{b}}_w(t)$ is the nearest border feature to $\bar{\boldsymbol{x}}_j$ with label y_{b_w}.If $y_j \neq y_{b_w}$, then the adaptation is applied as follows:

$$\bar{\boldsymbol{b}}_{\boldsymbol{w}}(t+1) = \bar{\boldsymbol{b}}_{\boldsymbol{w}}(t) - \eta(t) \cdot (\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{b}}_{\boldsymbol{w}}(t)) \tag{4.10}$$

$$\bar{\boldsymbol{m}}_{\boldsymbol{y}_{b_w}}(t+1) = \left(m_{y_{b_w}} \cdot \bar{\boldsymbol{m}}_{\boldsymbol{y}_{b_w}}(t) - \eta(t) \cdot \left(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{b}}_{\boldsymbol{w}}(t) \right) \right) \Big/ m_{y_{b_w}} \tag{4.11}$$

$$\eta(t) = \eta_0 e^{-t/\tau} \tag{4.12}$$

During training, after a predefined number of iterations, t', the combination of $\mathbf{M}^{\mathbf{t}}$ and $\mathbf{B}^{\mathbf{t}}$ are used as reference nodes to classify input training vectors. If the nearest node to a selected training vector $\bar{\boldsymbol{x}}_j$ with label y_j is one of the means of the border feature vectors $\bar{\boldsymbol{m}}_{\boldsymbol{w}}(t > t')$ with label y_{m_w} and if $y_j \neq y_{m_w}$, then the wrongly classified training sample $\bar{\boldsymbol{x}}_j$ is added as an additional border feature vector:

$$\mathbf{B}^{\mathbf{t+1}} = \mathbf{B}^{\mathbf{t}} \cup \{(\bar{\boldsymbol{x}}_{\boldsymbol{j}}, y_j)\}, \quad (t > t') \tag{4.13}$$

The corresponding mean vector is also adapted as follows:

$$\bar{\boldsymbol{m}}_{y_j}(t+1) = \left(m_{y_j}(t) \cdot \bar{\boldsymbol{m}}_{y_j}(t) + \bar{\boldsymbol{x}}_j\right) \Big/ (m_{y_j}(t)+1) \qquad \textbf{(4.14)}$$

where $m_{y_j}(t)$ is the number of border feature vectors belonging to class y_j at iteration *t*. Therefore $m_{y_j}(t+1)$ is the number of border feature vectors in class y_j after the addition of the new border feature vector.

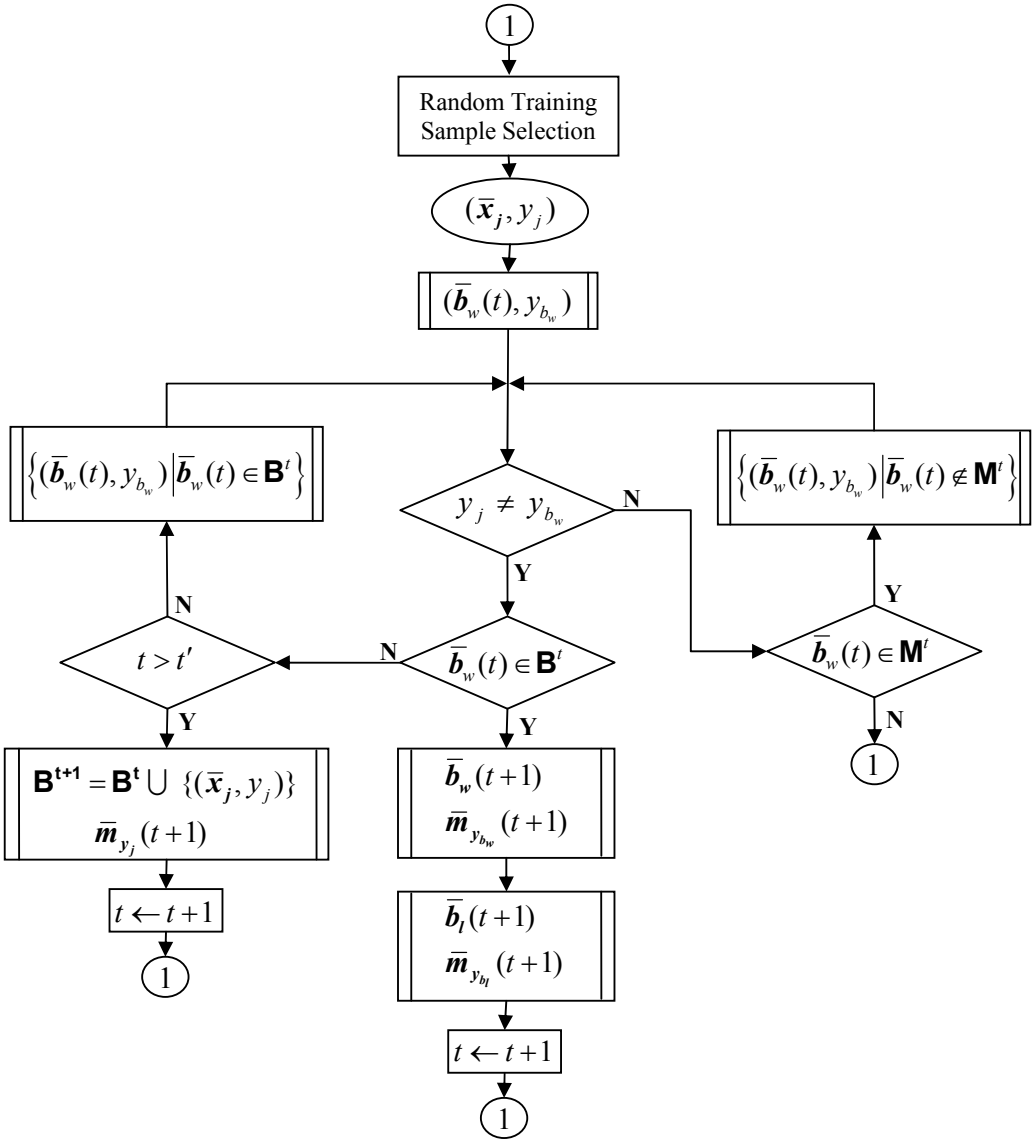


Figure 4.4: Flow Graph of the Adaptation Stage of the BFDA

The synthetic data result for the binary classification in the two-dimensional space is depicted in Figure 4.5. After the adaptation process, the final border feature vectors shown as circles and the final decision boundary as combination of partial lines are observed in Figure 4.5.

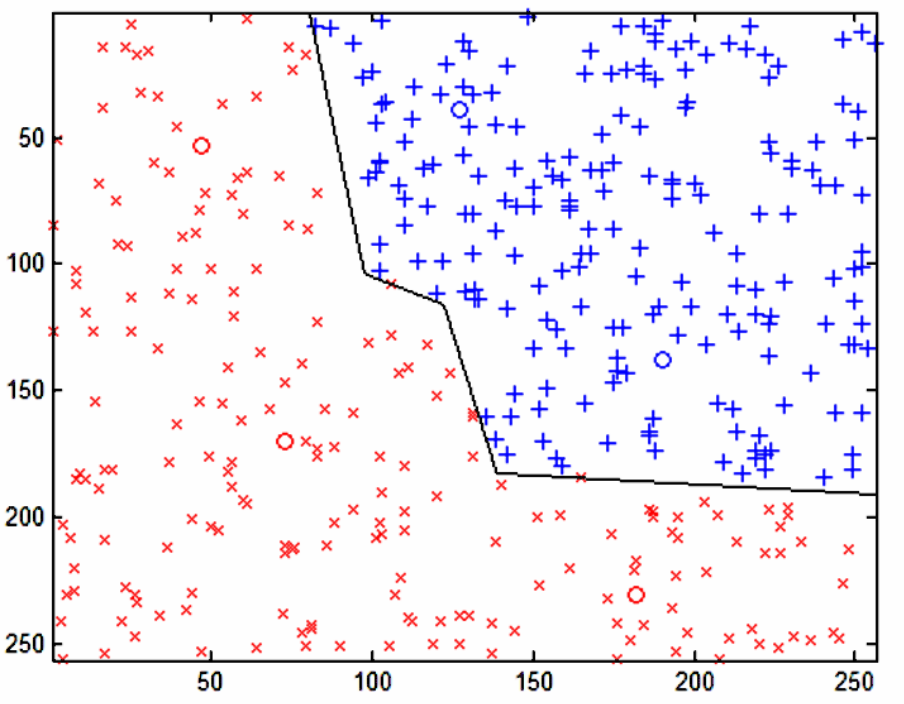


Figure 4.5: Partitioning of the Two-Dimensional Feature Space by Using the Final Border Feature Vectors Obtained at the end of the Adaptation Procedure

During testing with the testing data set, classification is based on the 1-nearest neighbor algorithm with the border feature vectors determined at the end of the adaptation procedure. This can be generalized. For example, the K-nearest neighbor algorithm can be used.

4.3 Additional Methods for Accuracy Enhancement in the BFDA

Additional methods can be used in the BFDA to obtain higher classification accuracies.

4.3.1 Consensus Strategy with Cross Validation

In supervised learning the training process should be unbiased to reach more accurate results in testing. In the BFDA, accuracy is related to the initialization of the border

feature vectors and the input ordering of the training samples. These dependencies make the classifier a biased decision maker. Consensus strategy can be applied with cross validation to reduce these dependencies. The cross validation fold number, *f* should be chosen big enough with a limited number of training samples. The block scheme of consensus strategy with *k* fold cross validation is depicted in Figure 4.6.

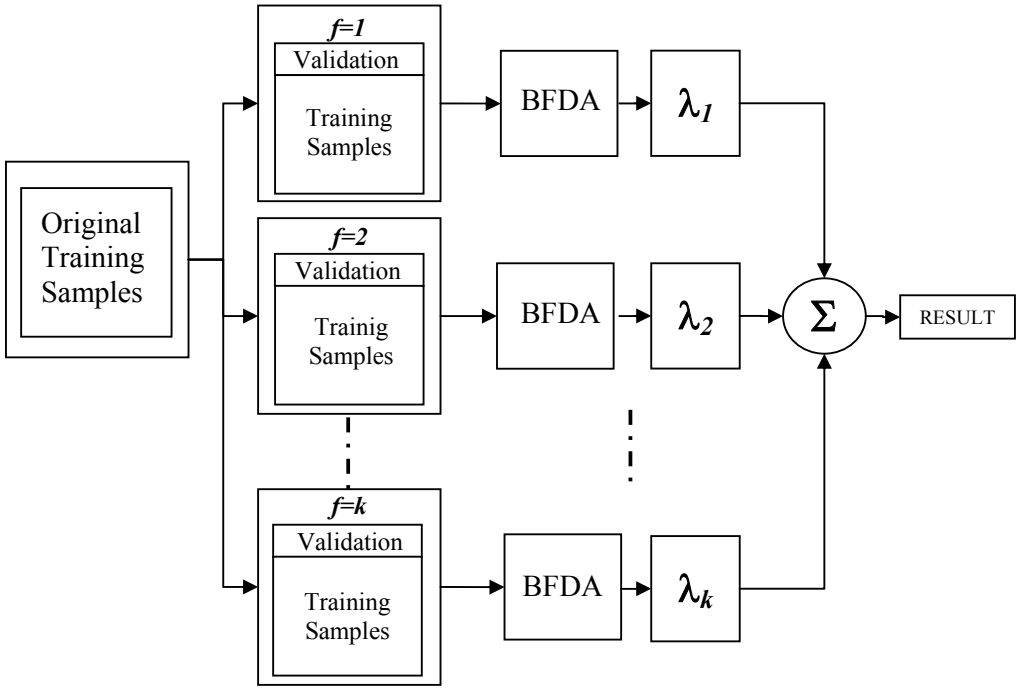


Figure 4.6: Block Scheme of Consensus Strategy with *K* Fold Cross Validation

There are a variety of consensual rules that can be applied to combine *k* individual results to obtain improved classification. The reliability factor of the classification results is depicted as a weight λ_k for the k^{th} BFDA classifier in Figure 4.6. This reliability factor can be specified by the consensual rule applied. For majority voting (MV) rule, weights can be equally chosen, and the majority label is taken as the final label. It is also possible to use non-equal voting structure (Qualified Majority Voting, QMV) based on training accuracies. By using cross validation as a part of the consensual strategy, part of the training samples are used for cross validation, and reliability factors can be assigned more precisely based on validation. Once the reliability factors are determined, consensual classification results can be obtained by applying a maximum rule with reliability factors. Additionally, obtaining optimal

reliability factors (weights, λ_k) can be done by least squares analysis [17]. Suppose the training results of single BFDA classifiers are represented by

$$\mathbf{Y}=\left[\bar{Y}_1\ \bar{Y}_2\ \ldots\ \bar{Y}_k\right]=\begin{bmatrix} y_{11} & y_{21} & \cdots & y_{k1} \\ y_{12} & y_{22} & \cdots & y_{k2} \\ y_{13} & y_{23} & \cdots & y_{k3} \\ . & . & \cdots & . \\ . & . & \cdots & . \\ y_{1n} & y_{2n} & \cdots & y_{kn} \end{bmatrix} . \qquad \textbf{(4.15)}$$

$\bar{Y}_i$ is column vector containing the output of a single classifier. **Y** is a nxk matrix where n is the number of validation vectors which are chosen from the original training set for cross validation. k is the number of the BFDA classifiers to be combined. Then, the optimal weights can be found by solving the following equation:

$$\begin{bmatrix} y_{11} & y_{21} & \cdots & y_{k1} \\ y_{12} & y_{22} & \cdots & y_{k2} \\ y_{13} & y_{23} & \cdots & y_{k3} \\ . & . & \cdots & . \\ . & . & \cdots & . \\ y_{1n} & y_{2n} & \cdots & y_{kn} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ . \\ . \\ . \\ . \\ \lambda_k \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \\ . \\ . \\ . \\ L_n \end{bmatrix} . \qquad \textbf{(4.16)}$$

The optimal weights are obtained by minimizing the square error:

$$\boldsymbol{\lambda}_{\mathbf{opt}} = \min_{\boldsymbol{\lambda}} \left\| \mathbf{Y}\boldsymbol{\lambda} - \mathbf{L} \right\|^2 \qquad \textbf{(4.17)}$$

$\boldsymbol{\lambda}_{\mathbf{opt}}$ is calculated as follows by using the pseudo inverse of **Y**:

$$\boldsymbol{\lambda}_{\mathbf{opt}} = (\mathbf{Y}^{\mathbf{T}}\mathbf{Y})^{-1}\mathbf{Y}^{\mathbf{T}}\mathbf{L} \qquad \textbf{(4.18)}$$

4.3.2 Refinement of Training Samples

Noisy training samples cause performance reduction of classification algorithms. Refinement of training samples can be used to improve classification. For the BFDA, selection of noisy training samples as initial values of border feature vectors should be avoided. To achieve this, the BFDA is run once, and wrongly decided training samples are specified. At the second run of the BFDA, the border feature detection procedure is applied on all the training samples except the wrongly decided training samples at the previous run. In the adaptation stage, the whole training set can be used.

4.3.3 Spatial Feature Extraction

It is also desirable to combine decisions of both spectral and spatial features together even if they are extracted from the same data source. Both spectral and spatial features can be used in order to reach high classification accuracies. As spatial features, mean and standard deviation of the neighborhood pixels are extracted for a pixel which is in the middle of a predefined window. The window size could be varied between 3x3 and 9x9 pixels. All extracted spectral and spatial features are classified individually via BFDA and is based on qualified majority voting (QMV).

5. EXPERIMENTS

Reliable data sets well-known in the literature are more convenient for performance analysis of the proposed algorithm BFDA than data sets which are not tested before. Two well known data sets which are widely encountered in the literature were used in the experiments to support validity of the results obtained [62,65]. Additionally, one synthetic data set was used to demonstrate the classification mechanism of the proposed BFDA and to expose differences with some other popular classification methods. The synthetic data is in a two-dimensional feature space, which makes it possible to visually display the decision boundaries and to help to understand the classification behaviour of the classifiers. One additional data set from Turkey [67] was also used to make proper comparison, and to show the robustness of the proposed algorithm. As a consequence, four different data sets, one of them having six different combinations of input vectors and corresponding classes, were used in the experiments to demonstrate case-independent results obtained by the BFDA. We were able to show that the overall classification accuracies obtained with the BFDA are satisfactory. Additionally, we were able to present rare class members more precisely than other conventional classification methods, especially in high dimensional feature spaces. Kappa statistics [34] was used to show the reliability of the results in the experiments. Another goal of the experiments was to show the Hughes effect [3] is less harmful with the BFDA than other conventional statistical methods. This meant that the performance of the BFDA with a limited number of training samples is generally higher than conventional classifiers.

5.1 Data Sets Used in the Experiments

Four different data sets were used in the experiments. Their names assigned and brief introduction about data sets are listed below.

1. Synthetic Data Set: It involves a binary classification problem in two-dimensional feature space. There are 200 randomly selected training samples that are used to partition the feature space. This data set is helpful to demonstrate decision boundaries obtained in the feature space.

2. AVIRIS Data set: The AVIRIS image taken from the northwest Indiana's Pine site in June 1992 [62] was used in the experiments. This is a well known test image and has been often used for validating hyperspectral image classification techniques [63,64]. We derived 6 different data sets from the AVIRIS data set by using combinations of different numbers of classes and feature sizes. Number of classes and feature vector sizes also influence the complexity of the classification. Therefore, this data set also demonstrates the classification performance as related to the complexity of classification. Detailed comparisons were made by using the AVIRIS data set in this thesis.

3. Satimage Data set: This data set is a part of the Landsat MSS data and contains six different classes. 4435 training samples and 2000 testing samples were obtained from statlog web site with their labels [65]. 4 spectral bands were used with one neighboring feature extraction method to extract features. Therefore 4x9=36 features were assigned to a pixel.

4. Karacabey Data set: This Landsat 7 ETM+ image was taken from northwest Turkey, Karacabey region in Bursa in July 2000 [66]. Six visible infrared bands (Band 1-5 & 7) having 30 m resolution were used with spectral features. Previous works were used as auxiliary information for extraction of the ground reference data [67].

5.2 Experiments

Four different experiments were designed. The names of the experiments are the same as the names of the data sets described above.

5.2.1 Experiment 1: Synthetic Data

The reference feature space with randomly selected training samples is depicted in Figure 5.1. This is a linearly non-separable binary classification problem. The feature space contains 250x250 points. 200 samples were randomly selected from each class. The experiments were performed with the Linear-SVM, the RBF-SVM, the BFDA and the Consensual-BFDA. In the literature, results obtained by Linear-SVM and RBF-SVM were very satisfactory [13,57]. Therefore, in the experiment, these classifiers were selected for reliable comparison. This experiment was designed understand mainly to the partitioning mechanisms of the classifiers used.

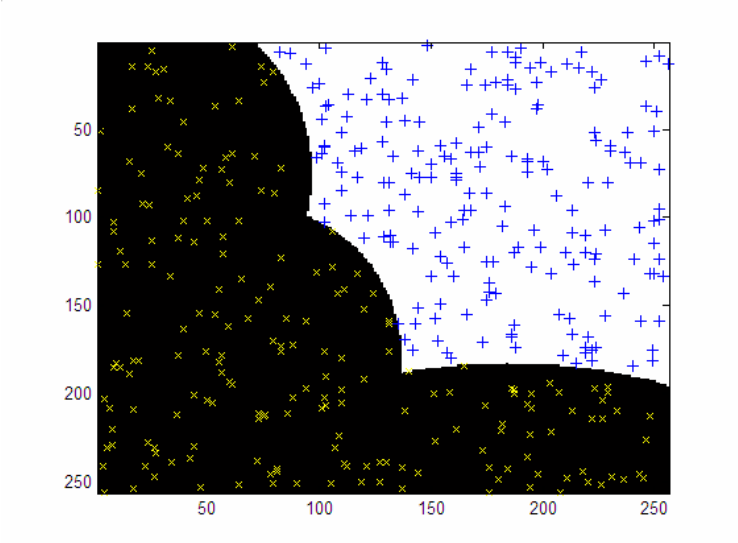


Figure 5.1: Reference Feature Space with Randomly Selected Training Samples

The BFDA result is depicted in Figure 5.2 with the final border feature vectors. In this figure the border feature vectors depicted as circles and final decision boundary consists of a combination of partial linear boundaries. The locations of the border feature vectors were obtained with the adaptation procedure. The number of border feature vectors is specified by the algorithm automatically and is also related to the problem complexity. In this example, the numbers of border feature vectors assigned to each class were 2 and 3, respectively. During the adaptation procedure, if the requirement of the border feature vector occurs, then a new border feature vector from the training set can be added to the network. Excessive number of border feature vectors reduces the adaptation procedure performance. Then, the generalization capability diminishes. The accuracy obtained from the BFDA is

related to the initially chosen border feature vectors, and the input orders of the training samples. Therefore, these dependencies make the algorithm biased. To reduce these dependencies, consensual strategy with cross validation can be applied.

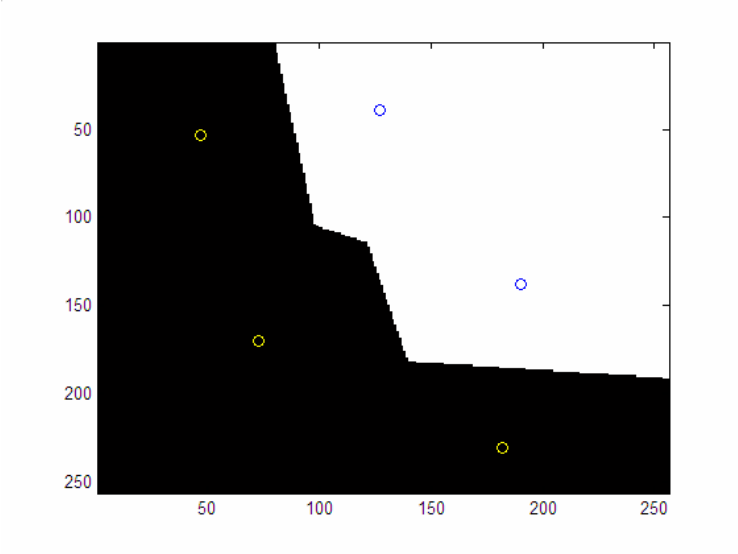


Figure 5.2: The BFDA Result

In Figure 5.3 the consensual-BFDA result is depicted. Using consensual strategy with cross validation makes the partial decision boundaries more nonlinear.

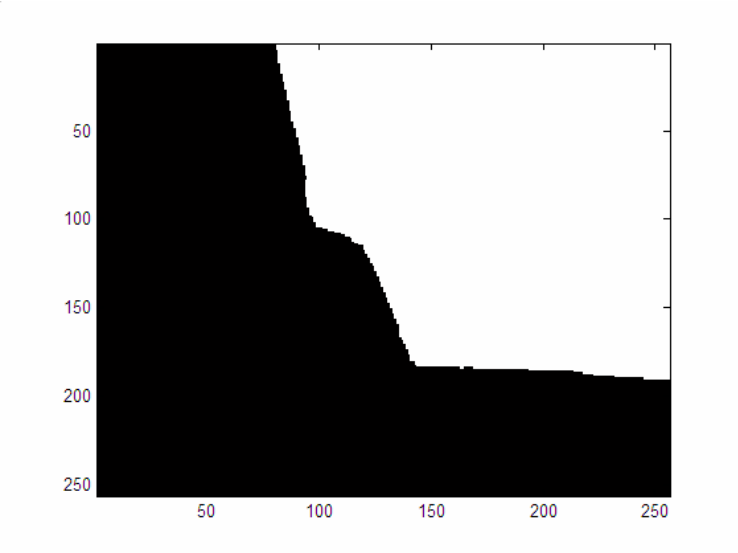


Figure 5.3: The Consensual-BFDA Result

In recent years, kernel methods such as support vector machines are widely used to improve classification accuracy. Maximum margin principle is applied by the SVM classifiers [50]. In this experiment, two different types of SVM classifier were used

to compare with the BFDA and the consensual-BFDA results. The linear SVM and RBF-SVM results are depicted in Figures 5.4 and 5.5, respectively. One linear decision boundary occurs for Linear-SVM as shown in .

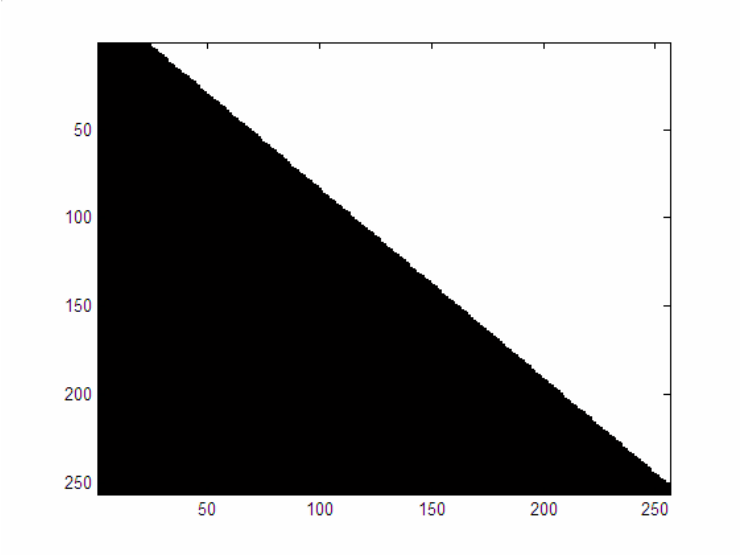


Figure 5.4: Linear SVM Result [C=2]

In this experiment, kernel parameters C for linear SVM, C and γ for RBF-SVM were obtained by using a pattern search algorithm to reach higher classification accuracy.

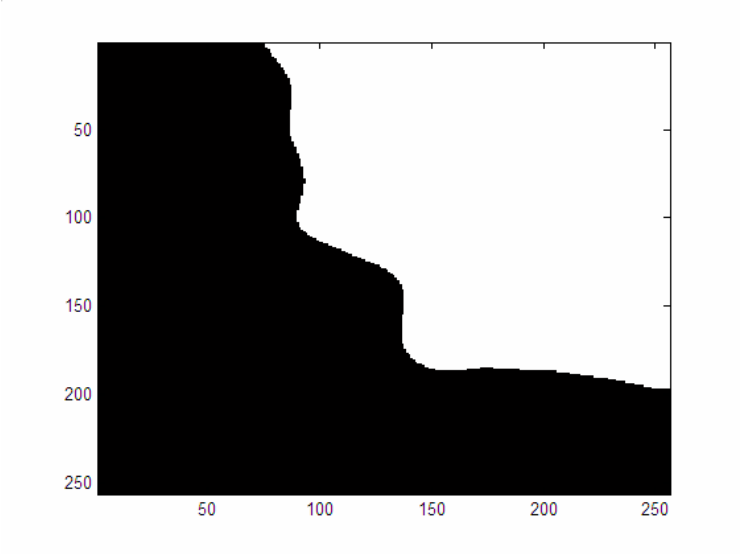


Figure 5.5: RBF-SVM Result [C=2, γ=32]

Results are shown in Table 5.1. As we can see from the table, the results for the BFDA, the consensual-BFDA and the RBF-SVM are almost the same. Lower classification accuracy is obtained by the linear SVM. We got the highest

classification accuracy with the consensual-BFDA. Thus, the best matching decision boundary was achieved with the consensual-BFDA. The BFDA produces a satisfactory simulation of decision boundary by using three linear partial boundaries. In the table, the classification accuracy as well as the kappa statistics (**κ**) are shown. Kappa statistics is a good indicator, showing not only classification accuracy but also reliability of the decisions made for all the classes [34].

Table 5.1: Classification Accuracies for the Synthetic Data Set

FIGURES	METHOD	ACCURACY %	κ
FIGURE 5.2	BFDA	98.40	0.965
FIGURE 5.3	CONSENSUAL-BFDA	98.98	0.979
FIGURE 5.4	LINEER SVM [C=2]	89.54	0.787
FIGURE 5.5	RBF-SVM [C=2, γ=32]	98.13	0.962

5.2.2 Experiment 2: AVIRIS Data

In this thesis, major performance analysis and comparisons were made by using the AVIRIS data. The AVIRIS data is a hyperspectral data and often used in the literature to demonstrate performance of the classifiers [62,64]. The AVIRIS data used in the experiment is shown for a color composite of the bands 50, 27 and 17 in Figure 5.6.

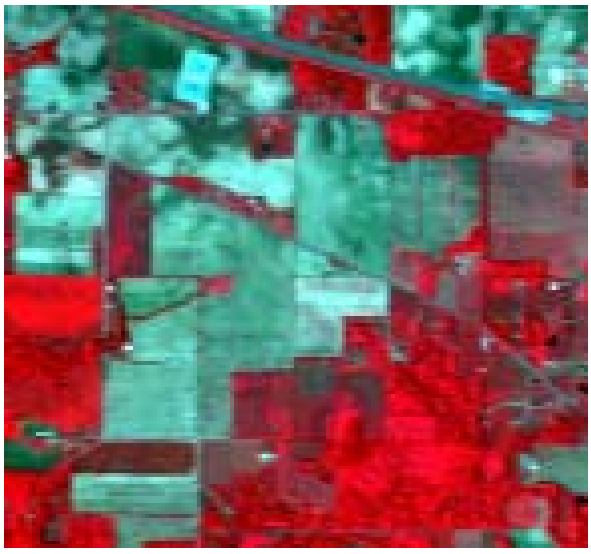


Figure 5.6: AVIRIS Data for the Bands 50, 27 and 17

We used the whole scene consisting of the full 145 x 145 pixels with three different class combinations, and two different spectral band combinations. The training sample sets with 17 classes (pixels with class labels of mixture type were considered for classification), 16 classes (whole class types apart from background) and 9 classes (more significant classes from the statistical viewpoint) were generated with different combinations of 9 (to illustrate multispectral data classification performance) and 190 spectral bands (30 channels discarded from the original 220 spectral channels because of atmospheric problems). Table 5.2 shows the number of training and testing samples for 17 and 16 class sets which were used in the experiments. Data sets 1 and 2 contain background class which is of mixture type. Therefore, these two classification experiments involved more complex classification problems than the other data sets. The large number of classes to be discriminated also increases the complexity of classification. There is also a trade-off between complexity and feature size, especially for classes which has a limited number of training samples (alfalfa, oats, etc). In such situations, lower classification performance with rare class members is expected, especially in a high dimensional feature space (data set 2) even if the all classification accuracy is increased.

Table 5.2: Numbers of Training and Testing Samples Used in Experiments

	17-CLASS DATA SET-1/2 (9 / 190 FEATURES)			16-CLASS DATA SET 3/4 (9/190 FEATURES)		
LABEL	CLASS	TRAINIG	TESTING	CLASS	TRAINING	TESTING
BACKGROUND	ω1	719	2627	-	-	-
ALFALFA	ω2	16	39	ω1	16	39
CORN-NOTILL	ω3	201	720	ω2	201	720
CORN-MIN	ω4	157	498	ω3	157	498
CORN	ω5	63	117	ω4	63	117
GRASS/PASTURE	ω6	112	265	ω5	112	265
GRASS/TREES	ω7	207	409	ω6	207	409
GRASS/PASTURE MOVED	ω8	12	24	ω7	12	24
HAY-WINDOWED	ω9	196	374	ω8	196	374
OATS	ω10	14	16	ω9	14	16
SOYBEANS-NOTILL	ω11	255	519	ω10	255	519
SOYBEANS-MIN	ω12	545	1302	ω11	545	1302
SOYBEANS-CLEAN	ω13	128	310	ω12	128	310
WHEAT	ω14	102	132	ω13	102	132
WOODS	ω15	546	870	ω14	546	870
BLDG-GRASS-TREE	ω16	109	229	ω15	109	229
STONE STEEL TOWERS	ω17	21	44	ω16	21	44
TOTAL NUMBER OF SAMPLES		3403	8495		2684	5868
WHOLE SCENE		21065			10366	

The background class which is of mixture type was discarded for data sets 3 and 4. Comparison made between data sets 1-2 and data sets 3-4 demonstrates the robustness of classification algorithms on data which contains mixture type. Numbers of training and testing samples in data sets 5 and 6 is depicted in Table 5.3.

Table 5.3: Numbers of Training and Testing Samples Used in the Experiments

	9-CLASS DATA SET-5/6 (9 / 190 FEATURES)		
LABEL	CLASS	TRAINIG	TESTING
CORN-NOTILL	ω1	288	288
CORN-MIN	ω2	200	200
GRASS/PASTURE	ω3	197	197
GRASS/TREES	ω4	200	200
HAY-WINDOWED	ω5	209	209
SOYBEANS-NOTILL	ω6	193	193
SOYBEANS-MIN	ω7	493	493
SOYBEANS-CLEAN	ω8	199	199
WOODS	ω9	258	258
	TOTAL NUMBER OF SAMPLES	2237	5809
	WHOLE SCENE	9345	

Statistical meaningful classes were chosen account for the data sets 5 and 6. Therefore, these data sets are more convenient for the conventional statistical classifiers. The data set 5 has sufficient number of training samples. Therefore, conventional statistical classifiers are expected to yield maximum accuracy for the data set 5. Additionally, data sets 5 and 6 are convenient to demonstrate the Hughes effects with the conventional statistical classifiers.

Average training, testing accuracies and kappa statistics are given in Table 5.4 for Data sets 1-6. The performance of the BFDA was compared with other classification algorithms including support vector machines (SVMs) [13,57] and several statistical classification techniques such as maximum likelihood, Fisher linear likelihood, correlation and matched filtering algorithms [63]. The data analysis software called Multispec [62] was used to perform the four statistical classification methods. Linear SVM and SVM with a radial basis kernel function were implemented in MATLAB using SVMlight [68], and its MATLAB interface by Schwaighofer [69]. A one-against-one multiclassification scheme was adopted in the experiments to compare SVMs performance to BFDA's. The parameters of the RBF-SVM (gamma and C) and Linear-SVM (C) methods could be selected by a pattern search algorithm with

cross validation. Only spectral features were taken into account in the comparison of BFDA with other classification techniques.

Table 5.4: Average Training ,Testing Accuracies and Kappa Statistics

DATA SET	METHOD	TRAINING		TESTING	
		ACCURACY %	K	ACCURACY %	K
1	MAXIMUM LIKELIHOOD	84.83	0.82	67.56	0.63
	FISHER LINEAR LIKELIHOOD	63.7	0.59	47.3	0.42
	CORRELATION	48.4	0.43	37.2	0.31
	MATCHED FILTER	32.8	0.24	36.1	0.29
	KNN [K=5]	89.01	0.87	68.06	0.63
	LINEAR SVM [C=40]	82.40	0.81	69.01	0.64
	RBF SVM [γ=1, C=20]	86.10	0.83	71.73	0.67
	BFDA	94.05	0.89	70.82	0.66
	CONSENSUAL BFDA	96.41	0.95	73.36	0.69
2	KNN [K=5]	90.71	0.89	70.01	0.65
	LINEAR SVM [C=10]	83.84	0.81	74.00	0.73
	RBF SVM [γ=0.1, C=10]	87.74	0.86	77.64	0.74
	BFDA	99.46	0.99	76.40	0.73
	CONSENSUAL BFDA	100	1	78.71	0.75
3	LINEAR SVM [C=40]	90.50	0.89	75.07	0.72
	RBF SVM [γ=1, C=40]	95.64	0.95	80.16	0.77
	BFDA	99.32	0.99	80.31	0.77
	CONSENSUAL BFDA	100	1	82.42	0.79
4	LINEAR SVM [C=1]	94.85	0.94	79.43	0.77
	RBF SVM [γ=1, C=20]	98.21	0.97	83.34	0.81
	BFDA	99.21	0.99	83.01	0.80
	CONSENSUAL BFDA	100	1	85.30	0.82
5	MAXIMUM LIKELIHOOD	86.99	0.85	77.07	0.74
	KNN [K=5]	93.69	0.92	83.04	0.80
	LINEAR SVM [C=20]	83.24	0.81	78.65	0.74
	RBF SVM [γ=1, C=20]	90.93	0.89	84.75	0.81
	BFDA	99.15	0.99	84.98	0.82
	CONSENSUAL BFDA	99.68	0.99	87.98	0.86
6	MAXIMUM LIKELIHOOD	100	1	67.00	0.57
	FISHER LINEAR LIKELIHOOD	91.3	0.90	81.8	0.78
	CORRELATION	45.4	0.39	47.7	0.40
	MATCHED FILTER	78.1	0.75	72.6	0.67
	KNN [K=5]	95.08	0.94	84.31	0.81
	LINEAR SVM [C=10]	96.24	0.95	88.36	0.86
	RBF SVM [γ=1, C=10]	100	1	91.39	0.89
	BFDA	100	1	88.58	0.86
	CONSENSUAL BFDA	100	1	90.18	0.88

Parameters choosen for the BFDA is also important concern. Two parameters needs to be assigned. These parameters are the learning rate η and the time constantτ. For fast convergence, η=0.1 and τ=1000 were found satisfactory. Faster training process is also suitable for less complex classification problems. For complex classification problem, fine tuning can be necessary, and η=0.2 and τ=6750 can be chosen. Parameter selection for the BFDA has also some similarity with the SOM [19]. Cross

validation can also be used for specifying an appropriate learning rate and time constant. Additionally, during the training process, validation set can be used to avoid overfitting. Then, early stopping can be applied.

Determination of the proper parameters are also an important concern for SVM classifiers. The accuracy obtained by SVM is dependent on the magnitude of the parameters C and γ.The large value of C and γ cause poor generalization of the classifier due to the overfitting of training data. SVM is a binary classifier and One-Against-One (OAO) strategy was used to enhance SVM classifier for multi-class classification in this thesis. For One-against-one strategy, C and γ should be obtained for every binary class combination. We assigned common parameters for each binary SVM classifier by using pattern search with cross validation [71] in this thesis. It is possible to use a multi class SVM classifier by reducing the classification to a single optimization problem. This approach may also require fewer support vectors than a multi-class classification based on combined use of many binary SVMs [72,73].

For the KNN classifier, the choice of K is related to the generalization performance of the classifier. Choosing a small number of K causes reduction of generalization of the KNN classifier. It is also obvious that, K=1 is most sensitive for noisy samples. Therefore K=5 was chosen in the experiments.

With all the data sets we obtained satisfactory results with the proposed algorithm the BFDA, and commonly the highest accuracy with the Consensual-BFDA. The RBF-SVM results were also very good.

The Hughes effect is less harmful for the BFDA than the maximum likelihood classifier (MLC) as expected. As we can see from Table 5.4, the accuracy obtained by the MLC is almost 10 % less for data set 6 than data set 5. Additionally, for data set 6, Kappa statistics is almost 10 % less than the testing accuracy. As a consequence, the results obtained with the MLC are not highly reliable in high dimensional feature space. With the BFDA, it is obvious that accuracy obtained in a high dimensional feature space is also very satisfactory. We also observe in Table 5.5, when the number of features increases, the overall classification accuracy increases. However, the accuracy of rare class members decreases. This reduction was observed with rare class members as related to the Hughes effect with the BFDA

algorithm. As a result a lower dimensional feature space is more convenient for detection of rare class members. Another important result is observed in Table 5.4 with the Fisher Linear Likelihood classifier. The Fisher Linear likelihood classifier uses class centers and the common covariance matrix for parameters. The accuracy obtained with the Fisher Linear Likelihood Classifier for data set 6 was the best in the statistical classifier category. The reason of this relatively high classification accuracy obtained by the Fisher Linear Likelihood classifier as compared to other statistical classifiers such as MLC is related to proper parameter estimation. Use of common covariance matrix instead of sample covariance matrix supports this result. In a high dimensional feature space, much more number of training samples is needed to make proper parameter estimation especially for covariance matrix estimation.

Table 5.5: Class by Class Accuracies Obtained by the Proposed Algorithm BFDA

LABEL	ACCURICIES %					
	DATA SETS					
	1	2	3	4	5	6
BACKGROUND	58.39	68.51	-	-	-	-
ALFALFA	87.17	80.48	89.74	84.61	-	
CORN-NOTILL	62.08	69.08	68.05	73.19	72.10	78.95
CORN-MIN	53.01	52.40	50.20	52.61	88.39	86.16
CORN	69.23	70.23	70.94	67.52	-	
GRASS/PASTURE	63.39	66.28	65.66	66.41	96.08	97.15
GRASS/TREES	94.13	92.73	97.79	94.62	94.57	95.70
GRASS/PASTURE MOVED	91.66	84.33	91.66	91.66	-	
HAY-WINDOWED	97.05	99.66	96.25	99.46	98.87	99.15
OATS	100	96.75	87.5	100	-	
SOYBEANS-NOTILL	77.26	79.91	74.18	78.42	81.07	80.90
SOYBEANS-MIN	84.33	86.63	84.17	88.40	77.57	85.82
SOYBEANS-CLEAN	76.77	73.90	73.87	81.29	89.86	91.95
WHEAT	97.72	99.48	99.24	100	-	
WOODS	75.17	89.39	95.28	97.35	98.86	99.11
BLDG-GRASS-TREE	61.57	64.88	75.10	72.05	-	
STONE STEEL TOWERS	93.18	91.90	97.72	97.72	-	
OVERALL	70.82	76.40	80.31	83.01	84.98	88.58

The ground reference data image for 17 classes [63] used in the experiment is depicted in Figure 5.7.

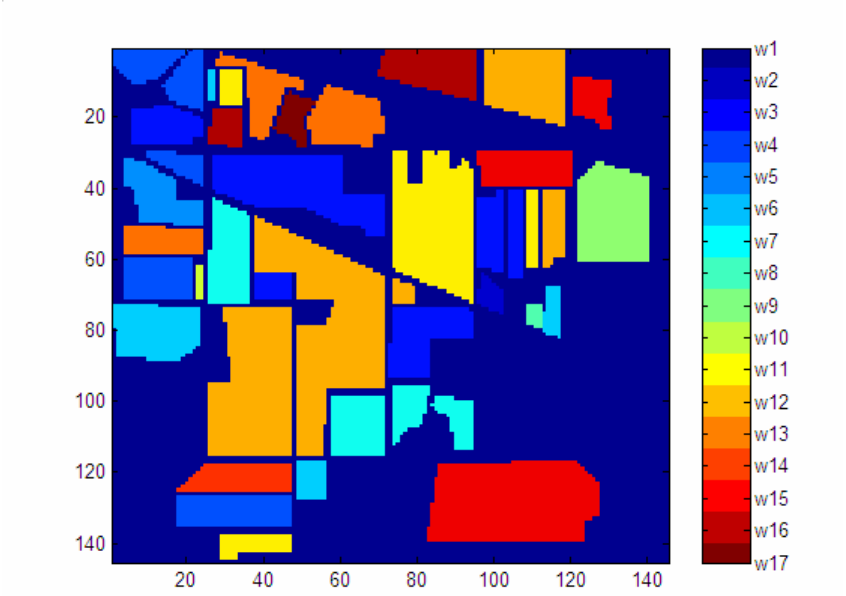


Figure 5.7: The Ground Truth of the AVIRIS Data Set for 17 Classes

The thematic map of the BFDA result for data set 1 is depicted in Figure 5.8. This data set has mixture type class (background), and this makes classification complex. Results obtained with spectral features are presented here, but it is obvious that using spatial features can improve classification accuracy [18,20].

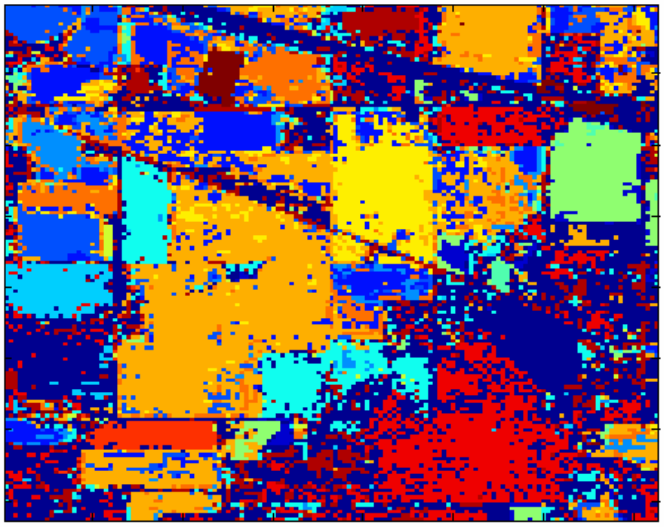


Figure 5.8: The Thematic Map of the BFDA Result for Data Set 1

The thematic map of the consensual BFDA result for data set 2 is depicted in Figure 5.9. Data set 2 has a mixture type class in a high dimensional feature space. There are also rare classes in data set 2. As we observe in Figure 5.9, the result obtained in the high dimensional feature space representing complex classification problem is satisfactory.

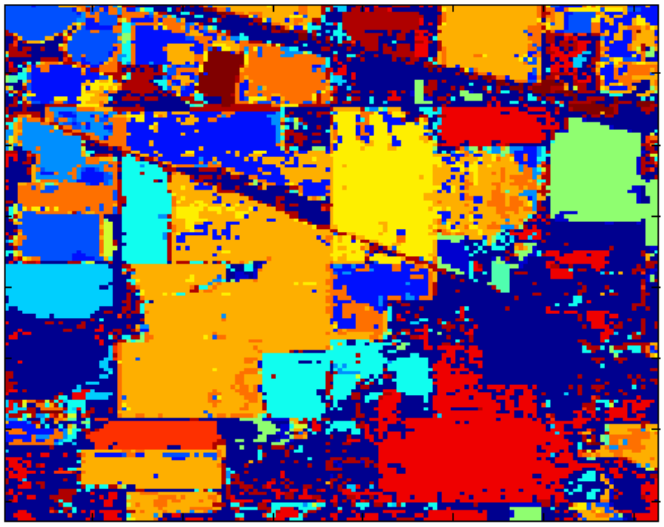


Figure 5.9: The Thematic Map Obtained with the Consensual BFDA and Data Set 2

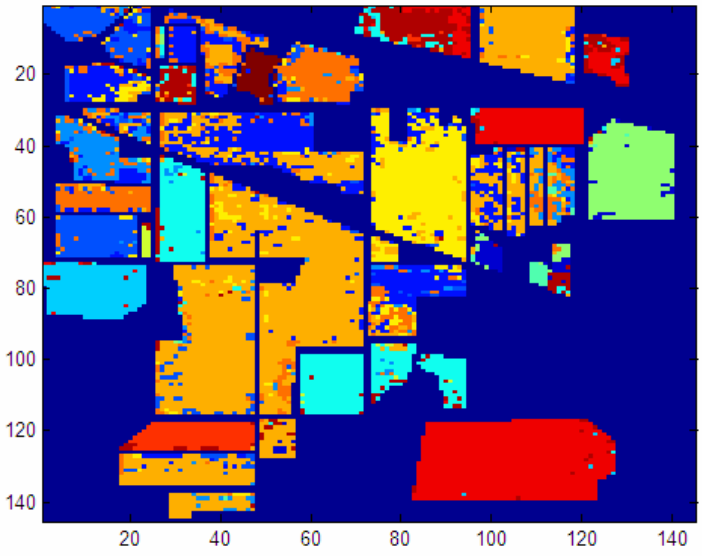


Figure 5.10: The Thematic Map Obtained with the BFDA and Data Set 3

Data sets 3 and 4 consist of pure classes. Therefore, this experiment is less complex than the experiments data sets 1 and 2. However, data sets 3 and 4 contain rare class members. A detailed class discrimination investigation (number of classes is 16) were made in these experiments. The thematic maps obtained with the BFDA the consensual BFDA are depicted in Figures 5.10 and 5.11, respectively. The BFDA satisfies high classification accuracy while performing well with the rare class members, as observed in Table 5.5.

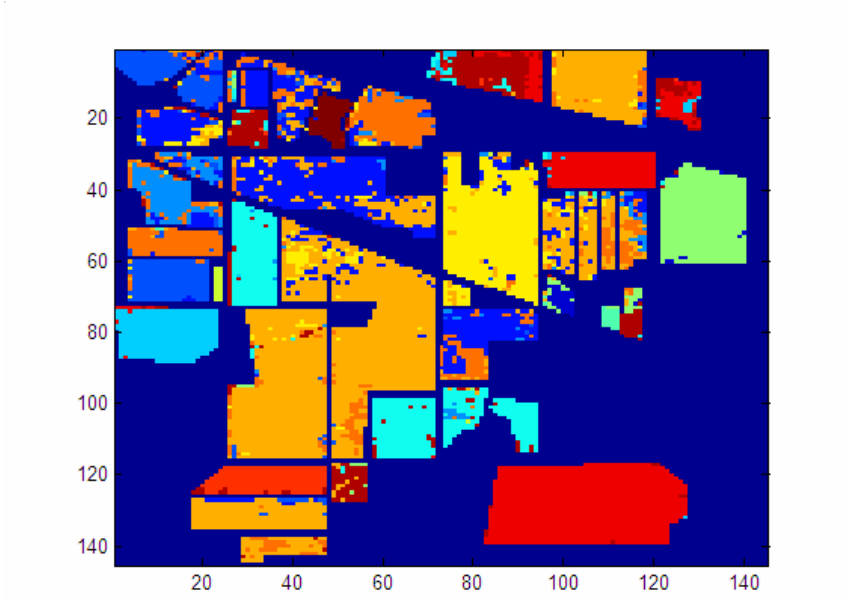


Figure 5.11: The Thematic Map Obtained with the Consensual BFDA and Data Set 4

Data sets 5 and 6 contain statistically meaningful classes. Especially for data set 5, the number of training samples is convenient for statistical classifies to make proper classification. The same number of training samples in a high dimensional feature space also characterizes data set 6. When the feature vector size increases, requirement of more number of training samples occurs with the MLC. The thematic map observed with the BFDA and consensual BFDA result depicted in Figures 5.12 and 5.13 for data sets 5 and 6 respectively. The BFDA satisfies high classification accuracy but we expected to reach a higher classification accuracy than we obtained for statistically meaningful data sets 5 and 6. The reason may be overfitting or detecting excessive number of border feature vectors. In this case, during the adaptation process, a redundant border feature reduction procedure can be applied to

have higher classification accuracy than what is obtained with the original BFDA algorithm.

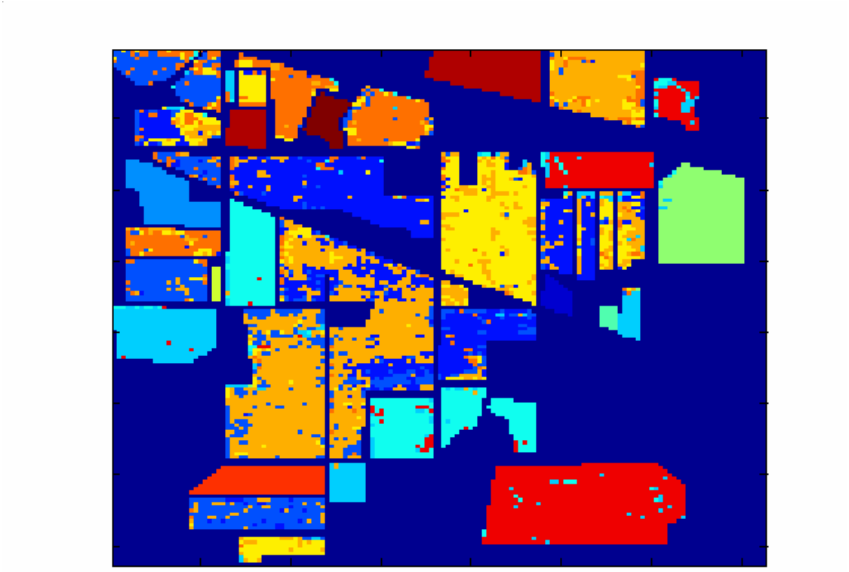


Figure 5.12: The Thematic Map Obtained with the BFDA for Data Set 5

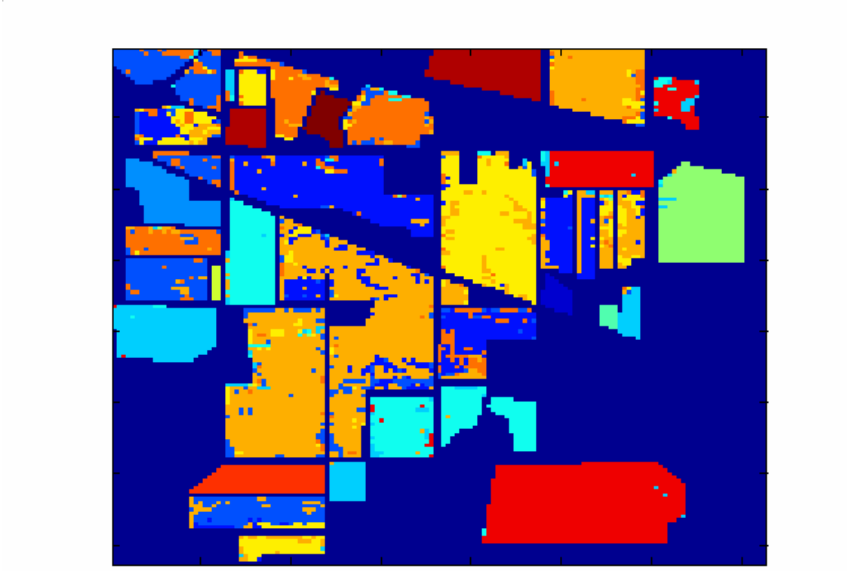


Figure 5.13: The Thematic Map Obtained with the Consensual BFDA for Data Set 6

Thus, the number of border features detected by the algorithm affects the generalization performance of the BFDA algorithm. The average numbers of border feature vectors detected by the algorithm are depicted in Table 5.6. As we observe in the table, the number of detected border feature vectors is related to the complexity

of the problem. Therefore, less complex problems need less number of border feature vectors to avoid overfitting.

Table 5.6: Average Number of Border Feature Vectors Obtained with the BFDA

DATA SETS	1	2	3	4	5	6
AVERAGE NUMBER OF BORDER FEATURES VECTORS	189	184	143	136	93	95

5.2.3 Experiment 3: Satimage Data

Satimage data was obtained from the statlog website [65]. This website serves variety of data sets which are for various types of applications. Satimage data set is a Landsat MSS imagery. One frame of the Landsat MSS imagery consists of four digital images of the same scene in different spectral bands. Two of these are in the visible region (corresponding approximately to green and red regions of the visible spectrum), and two are in the (near) infra-red. Each pixel is a 8-bit binary word, with 0 corresponding to black and 255 to white. The spatial resolution of a pixel is about 80m x 80m. The data set is a sub-area of a scene, consisting of 82 x 100 pixels. Each line of data corresponds to a 3x3 square neighborhood of pixels completely contained within the 82x100 sub-area. The total numbers of training and testing samples used in the experiment are depicted in Table 5.7.

Table 5.7: Numbers of Training and Testing Samples Used in the Satimage Data Set

	6-CLASS SATIMAGE DATA SET (36 FEATURES PER PIXEL)		
LABEL	CLASS	TRAINIG	TESTING
RED SOIL	ω1	1072	461
COTTON CROP	ω2	479	224
GREY SOIL	ω3	961	397
DAMP GREY SOIL	ω4	415	200
SOIL WITH VEGETATION STUBBLE	ω5	470	211
VERY DAMP GREY SOIL	ω6	1038	470
	TOTAL NUMBER OF SAMPLES	4435	2000

Highest accuracy in previous works with this data set was obtained with the SVM [72]. In this experiment, the RBF-SVM classifier and the MLC were used to make comparisons with the BFDA. Additionally, the results obtained with the binary version of the BFDA algorithm using one-against-one (OAO) binary classification strategy with the SVM and a neural network with backpropagation learning also using OAO strategy are provided. The aim of this experiment is to demonstrate the

robustness of the results obtained by the BFDA, and illustrate the performance of the BFDA on additional types of remotely sensed data.

Table 5.8: Classification Results for the Satimage Data Set

METHOD	TRAINING ACCURACY %	TESTING ACCURACY %
MAXIMUM LIKELIHOOD	89.69	85.69
NN - BACKPROPAGATION	90.48	87.80
RBF SVM [C=6,γ=1.5]	98.92	91.75
BFDA	98.42	89.90
BINARY BFDA	97.65	89.45
CONSENSUAL BFDA	99.47	91.95

The classification accuracy of the RBF-SVM (C=16, γ=1) classifier with one-against -one strategy was reported 91.3 % for satimage testing data set in reference [72]. In comparison, the results obtained with the BFDA are satisfactory for the satimage data set. Matlab' s Neural Network toolbox was used to obtain the result of the neural network with backpropagation learning [74]. 20 neurons in one hidden layer was chosen with learning rate 0.01 as network parameters. Activation function was also chosen as a sigmoid function in this experiment.

5.2.4 Experiment 4: Karacabey Data

Karacabey Data set is a part of the Landsat 7 ETM+ image acquired in July 2000 [66]. Six visible infrared bands (Band 1-5 & 7) having 30 m resolution were used for analysis. The area is located in Karacabey, Bursa which is in the North-West of Turkey. A sub-image which consists of 150x200 pixels was used in the experiment. A color composite of the sub-image is depicted in Figure 5.14.

The ground truth data extracted from previous work which is related to parcel-based crop mapping was used in this experiment [67] The previous work covers a wider agricultural area than the part of the scene used in the experiment. Registration of the ground truth map to sub-spatial scenes was made by using Envi [75]. The ground truth map used in our experiment is depicted in Figure 5.15.

Figure 5.14: Color Composite Image of Karacabey Data Set for Bands 4, 3 and 2

9 classes were utilized while background and parcel boundaries were discarded from evaluation. The description of the classes and the numbers of class samples used for training, testing and whole scene are depicted in Table 5.9.

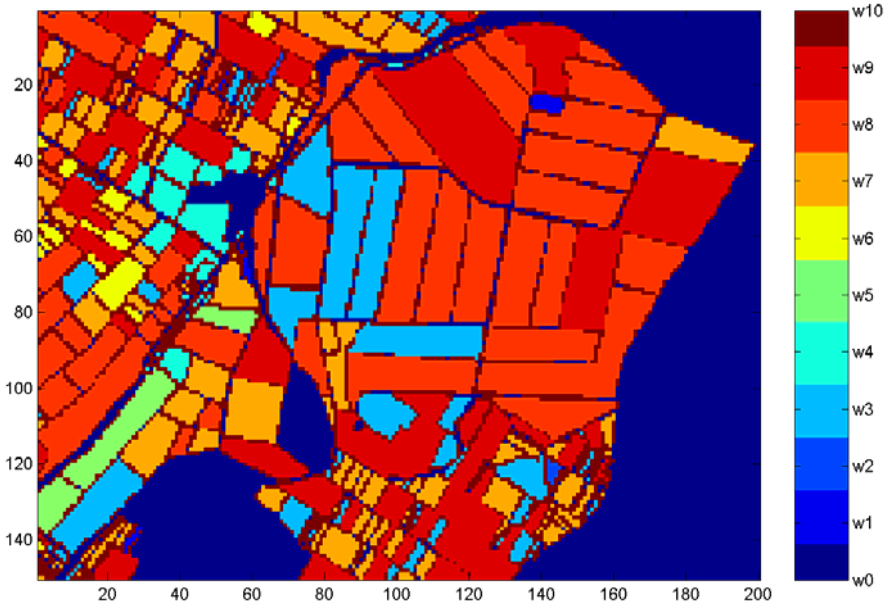


Figure 5.15: The Ground Truth of the Karacabey Data Set with 9 Classes

Our goal was to demonstrate whether the BFDA is robust and performs well, in general. In this experiment, we compare the BFDA with the SVMs classifiers and the

MLC. In Table 5.10 training and testing accuracy as well as accuracy obtained with the whole scene is depicted.

Table 5.9: Number of Samples for Training Testing and Whole Scene

	9-CLASS SATIMAGE DATA SET (6 FEATURES PER PIXEL)			
LABEL	CLASS	TRAINIG	TESTING	WHOLE SCENE
BARE SOIL	ω1	10	10	66
WATERMELON	ω2	10	10	27
PEPPER	ω3	60	60	2110
PASTURE	ω4	60	60	508
CLOVER	ω5	60	60	442
SUGAR BEET	ω6	60	60	300
TOMATO	ω7	60	60	2694
RESIDU	ω8	60	60	6846
CORN	ω9	60	60	4752
	TOTAL NUMBER OF SAMPLES	440	440	17737

As we observe in Table 5.10, the result obtained with the BFDA is satisfactory in comparison to other results. Overall classification accuracies are less than 70 %. Using only one multispectral data is not sufficient for discriminating detailed class types. In the previous work, three different scenes acquired in approximately one month period were used for classification. Therefore, multitemporal data classification can be used to improve classification. In this experiment, we obtained highest accuracy with the SVM classifier and the Consensual BFDA in the experiment they give almost equal accuracy. The thematic map of the BFDA result for the Karacabey data set is depicted in Figure 5.16.

Table 5.10: Classification Results for the Karacabey Data Set

METHOD	ACCURACY OF TRAINING %	ACCURACY OF TESTING %	ACCURACY OF WHOLE SCENE %
MAXIMUM LIKELIHOOD	73.86	65.90	63.80
LINEER SVM [C=10]	82.30	67.90	65.80
RBF SVM [C=1,γ=0.1]	85.20	70.24	69.20
BFDA	95.40	68.80	67.41
CONSENSUAL BFDA	99.20	70.02	68.80

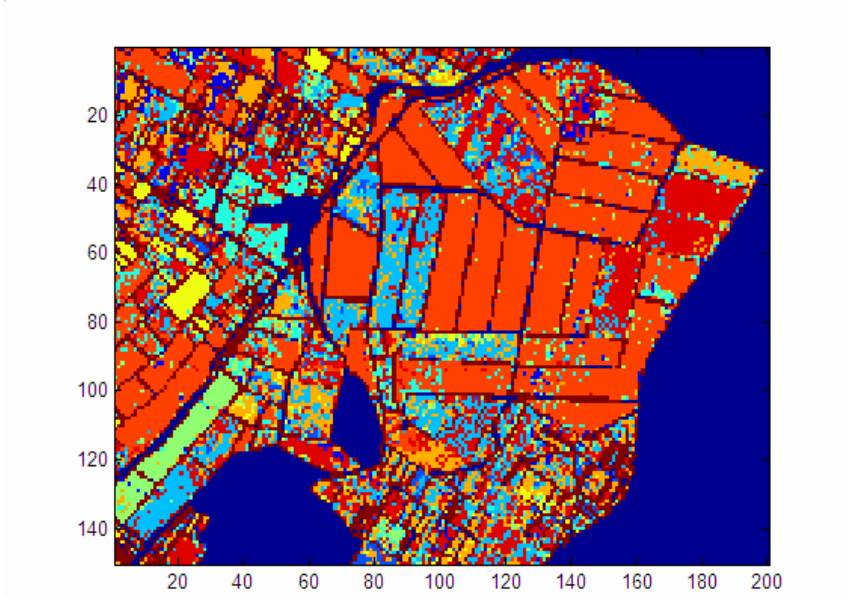


Figure 5.16: The Thematic Map Obtained with the BFDA and the Karacabey Data Set

6. SUMMARY, CONCLUSIONS AND FUTURE WORK

In recent years, the sensor technology has progressed rapidly, and the remote sensing community has gathered huge amount of data collected from the earth surface. Using data obtained with different kinds of sensors (such as multispectral, hyperspectral, radar and lidar) is a big challenge to produce value added products. Different kinds of sensors have different imaging mechanisms and the collected data has specific characteristics depending on the sensor types. From the view point of the end user, it is also challenging to develop appropriate algorithm to be used with different types of remote sensing data. Thus the development of robust pattern recognition methods especially suited to each type of data is necessary, especially for the automatic target recognition (ATR) task.

6.1 Summary and Conclusions

In this thesis, we developed a new algorithm for classification of remote sensing images. The method first makes use of border feature vectors as part of an adaptation process aimed at better describing the classes, and then uses nearest neighbor algorithm with the final border feature vectors for classification. In chapter 3, principle classifiers are discussed. We especially focused on SVM, SOM, KNN and GAL algorithms, which are important for comparison with the proposed algorithm BFDA. In chapter 4, the BFDA is discussed in detail. The concept of border feature vectors proposed in this thesis has some similarity with support vectors in SVM classifiers. However, the procedure of the initialization of border feature vectors, and subsequent adaptation process to find final border feature vectors is completely different. The competitive learning principle is applied during the adaptation procedure. In this sense, the adaptation algorithm used has some similarity with the LVQ algorithm. Two rules, 1) a border feature vector which causes wrong decision should be far away from the input training sample, and 2) the nearest border feature vector which has the same label with the input training sample should be closer to the

input training sample are applied during adaptation. The reason for this adaptation first strategy is to satisfy the maximum margin principle adaptively. The BFDA algorithm first chooses a subset of the training samples as initial border feature vectors by using the proposed border feature detection procedure. The proposed border feature detection technique is novel.

It can be useful to mention some classification algorithms which have some similarity with the BFDA to make a proper comparison. The GAL algorithm randomly chooses a subset of training samples to satisfy predefined training accuracy until reaching predefined iteration number without any geometric consideration. The KNN algorithm uses the whole training set. This makes the obtained results very sensitive to noise. Another drawback of the KNN is processing time which is very high for classification of large data sets. In the BFDA algorithm, a small number of border feature vectors are chosen especially in comparison to the number of reference vectors used in KNN algorithm and the number of support vectors used in the SVM classifiers. As a result, the processing time for testing is approximately 95% less with the BFDA than with the KNN algorithm. Using the BFDA, we obtained satisfactory results with both multispectral and hyperspectal data sets as discussed in chapter 5. The BFDA is a nonparametric classifier, robust against the Hughes effect, and well-suited for remote sensing applications.

6.2 Future Work

The BFDA has been applied so far full feature space. Initially band grouping can be applied to produce lower dimensional feature spaces. Then, the BFDA applied in the lower dimensional feature spaces can be combined by using consensual rule. This procedure may be called band grouping and fusion. Additionally, appropriate safe rejection schemes [14] can be applied to the BFDA to reach higher classification accuracies. In spatial space, there are also variety of applications suitable for processing with the BFDA, such as target detection (bridge detection in SAR images), and contour specification (detection of sea-land contours). In conclusion, the BFDA can be applied in various appropriate applications in remote sensing, image processing, and other classification applications.

REFERENCES

[1] **Kumar, L., Schmidt, K., Dury, S. and Skidmore, A.**, 2001. Imaging spectrometry: basic principles and prospective applications, pp. 111-155, Kluwer Academic Publishers, Dordrecht, The Netherlands.

[2] AVIRIS hyperspectral data cube is available online at website: http://popo.jpl.nasa.gov/html/aviris.cube.html.

[3] **Hughes, G.F.**, 1968. On the mean accuracy of statistical pattern recognizers, *IEEE Trans. Inform. Theory,* **IT-14,** 55-63.

[4] **Hoffbeck, J.P. and Landgrebe, D.A.**, 1996. Covariance matrix estimation and classification with limited training data, *IEEE Trans. Pattern Anal. Machine Intell.,* **18,** 763-767.

[5] **Tadjudin, S. and Landgrebe, D.A.**, 1999. Covarience estimation with limited training samples, *IEEE Trans. Geosci. Remote Sensing,* **37,** 2113-2118.

[6] **Dempster, A.P. Laird, N.M. and Rubin, D.B.**, 1977. Maximum likelihood from incomplete data via the EM algorithm, *J. R. Dtatist. Soc.,* **19,** 1-38.

[7] **Moon, T.K.**, 1996. The expectation-maximization algorithm, *Signal Process. Mag.,* **13,** 47-60.

[8] **Lee, C. and Landgrabe, D.A.**, 1993. Feature extraction based on decision boundries, *IEEE Trans. Pattern Anal. Machine Intell.,* **15,** 388-400.

[9] **Jimenez, L.O. and Landgrebe, D.A.**, 1999. Hyperspectral data analysis and feature reduction via projection pursuit, *IEEE Trans. Geosci. Remote Sensing,* **37,** 2653-2667.

[10] **Kumar, S., Ghosh J. and Crawford, M.M.**, 2001. Best-bases feature extraction algorithms for classification of hyperspectral data, *IEEE Trans. Geosci. Remote Sensing,* **39,** 1368-1379.

[11] **Hoffbeck, J.P. and Landgrebe, D.A.**, 1996. Classification of remote sensing images having high-spectral resolution, *Remote Sens. Environ.,* **57,** 119-126.

[12] **Tsai, F. and Philpot, W.D.**, 2002. A derivative-aided hyperspectral image analysis system for land-cover classification, *IEEE Trans. Geosci. Remote Sensing,* **40,** 416-425.

[13] **Melgani, F. and Bruzzone, L.**, 2004. Classification of hyperspectral remote sensing images with support vector machines, *IEEE Trans. Geosci. Remote Sensing,* **42,** 1778-1790.

[14] **Cho, S., Ersoy, O.K. and Lehto, M.R.**, Parallel, self-organizing, hierarchical neural networks with competitive learning and safe rejection schemes, *IEEE Trans. Circuits and Systems,* **40,** 556-567.

[15] **Benediktsson, J.A., Sveinsson, J.R. and Swain, P.H.**, Hybrid consensus theoretic classification, *IEEE Trans. Geosci. Remote Sensing,* **35,** 833-843.

[16] **Chee, H., and Ersoy, O.K.**, 2005. A statistical self-organizing learning system for remote sensing classification, *IEEE Trans. Geosci. Remote Sensing,* **432,** 1890-1900.

[17] **Benediktsson, J.A., Sveinsson, J.R., Ersoy, O.K. and Swain, P.H.**, 1997. Parallel consensual neural networks, *IEEE Trans. Geosci. Remote Sensing,* **8,** 54-64.

[18] **Lee, J. and Ersoy, O.**, 2006. Consensual and hierarchical classification of remotely sensed multispectral images, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS'06),* Denver, CO, USA, 31 July-04 August.

[19] **Kohonen, T.,** 1990. The self-organizing map, *Proceedings of the IEEE,* **78.**

[20] **Karakahya, H., Yazgan, B. and Ersoy, O.K.**, 2003. A spectral-spatial classification Algorithm for multispectral remote sensing data, *in Proc. The 13th Int'l Conf. Artificial Neural Networks*, Istanbul, Turkey, June 2003, 1011-1017.

[21] **Kasapoglu, N.G., Ersoy, O.K. and Yazgan, B.**, 2006. Border feature detection and adaptation for classification of remote sensing images, *IEEE*

International Geoscience and Remote Sensing Symposium (IGARSS'06), Denver, CO, USA, 31 July-04 August.

[22] **Alpaydin, E.**, 1991. GAL: networks that grow when they learn and shrink when they forget, *International Computer Science Institute,* **TR 91-032,** *Berkeley,* CA.

[23] **Cristianini, N. and Taylor, J.S.**, 2000.Support Vector Machines, Cambridge University Press,

[24] **Jimenez, L.O. and Landgrebe, D.A.**, 1998. Supervised classification in high-dimensional space: Geometrical, statistical, and asymptotical properties of multivariate data, *IEEE Trans. Syst. Man. Cybern.*, **28,** 39-54.

[25] **Scott, D.W.**, 1992. Multivariate Density Estimation, pp. 208-212, Wiley-Interscience, New York.

[26] **Hwang, J., Lay, S. and Lippman, A.**, 1994. Nonparametric multivariate density estimation: A comparative study, *IEEE Trans. Signal Processing,* **42,** 2795-2810.

[27] **Lee, C. and Landgrebe, D.A.**, 1993. Analyzing high dimensional multispectral data, *IEEE Trans. Geoscience Remote Sensing,* **31,** 792-800.

[28] **Richards, J.A.**, 1993. Remote Sensing Digital Image Analysis, An Introduction, Springer-Verlag, New York.

[29] **Jimenez, L. and Landgrebe, D.A.**, 1995. Projection pursuit for high dimensional feature reduction: parallel and sequential approaches, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS'95),* Florence, Italy, July 10-14.

[30] **Jimenez, L.O., Morel, A.M. and Creus, A.**, 1999. Classification of hyperdimentional data based on feature and decision fusion approaches using projection pursuit, majority voting, and neural networks, *IEEE Trans. Geosci. Remote Sensing,* **37,** 1360-1366.

[31] **Fukunaga, K.**, 1990. Introduction to Statistical Pattern Recognition, pp. 99-109, San Diego, CA, Academic Press.

[32] **Haralick, R.M., Shanmugam, K. and Dinstein, I.**, 1973. Textural features for image classification, *IEEE Trans. Systems Man. and Cybernetics,* **SMC-3,** 610-621.

[33] **Baraldi, A. and Parmiggiani, F.**, 1995. An investigation of the textural characteristics associated with gray level co-occurance matrix statistical parameters, *IEEE Trans. Geosci. Remote Sensing,* **33,** 293-304.

[34] **Sim, J. and Wright, C.W.,** 2005. The kappa statistics in reliability studies: use, interpretation, and sample size requirements, *Physical Therapy*, **85,** 257-268.

[35] **Landgrebe, D.A.**, 2003. Signal theory and methods in multispectral remote sensing, John Wiley & Sons, New Jersey, USA.

[36] **Landgrebe, D.A.**, 2002. Hyperspectral image data analysis, *IEEE Signal Processing Magazine,* **19,** 17-28.

[37] **Blanzieri, E. and Melgani, F.**, 2006. An adaptive nearest neighbor classifier for remotely sensed imagery, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS'06),* Denver, CO, USA, 31 July-04 August.

[38] **Shahshahani, B.M. and Landgrebe, D.A.**, 1994. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon, *IEEE Trans. Geosci. Remote Sensing,* **32,** 1087-1095.

[39] **Redner, R.A. and Walker, H.F.**, 1984. Mixture densities, maximum likelihood and the EM algorithm, *SIAM Rev.,* **26,** 195-239.

[40] **Cover, T.M. and Hart, P.E.**, 1967. Nearest neighbor pattern classification, *IEEE Trans. Information Theory,* **13,** 21-27

[41] **Fukunaga, K. and Narendra, P.M.**, 1975. A branch and bound algorithm for computing k-nearest neighbors, *IEEE Trans. Computers,* **24,** 750-753.

[42] **Djouadi, A. and Bouktache, E.**, 1997. A fast algorithm for the nearest neighbor classifier, *IEEE Trans. Pattern Analysis and Machine Intelligence,* **19,** 277-282

[43] **Ritter, G.L., Woodruff, H.B., Lowry, S.R. and Isenhour, T.L.**, 1975. An algorithm for a selective nearest neighbor decision rule, *IEEE Trans. Information Theory,* 21, 665-669.

[44] **Chang, C.L.**, 1974. Finding prototypes for nearest neighbor classifiers, *IEEE Trans. Computers,* **23,** 1179-1184.

[45] **Xia, C., Lu, H., Ooi, B.C. and Hu, J.**, 2004. Gorder: An efficient method for KNN joint processing, *in Proceedings of the 30th VLDB Conference,* Toronto, Canada.

[46] **Xia, C., Hsu, W., Lee, M.N. and Ooi, B.C.**, 2006. Border: Efficient computation of boundary points, *IEEE Trans. Knowledge and Data Eng.,* **18,** 289-303.

[47] **Alpaydin, E.**, 1990. Neural models of incremental supervised and unsupervised learning, PhD Thesis, Ecole Polytechnique Federale de Lausanne, Switzerland.

[48] **Alpaydin, E.**, 1990. Grow and learn: An incremental method for category learning, *Int. Neural Network Conf.,* Paris, France.

[49] **Haykin, S.**, 1999. Neural networks: a comprehensive foundation, Prentice Hall, New Jersey, USA.

[50] **Vapnik, V.N.**, 1998. Statistical learning theory, Wiley, New York.

[51] **Foody, G.M. and Mathur, A.**, 2004. A relative evaluation of multiclass image classification by support vector machines, *IEEE Trans. Geosci. Remote Sensing,* **42,** 1335-1343.

[52] **Atkinson, P.M. and Tatnall, A.R.L.**, 1997. Neural networks in remote sensing, Int. Journal of Remote Sensing, **18,** 699-709.

[53] **Foody, G.M. and Arora, M.K.**, 1997. An evaluation of some factors affecting the accuracy of classification by an artificial neural network, Int. Journal of Remote sensing, **18,** 799-810.

[54] **Mather, P.M.**, 2004. Computer processing of remotely sensed images, Wiley, Chichester.

[55] **Van Niel, T.G. and Datt, B.**, 2005. On the relationship between training sample size and data dimensionality of broadband multi-temporal classification, *Remote Sensing of Environment,* **98,** 416-425.

[56] **Foody, G.M.**, 1999. The significance of border training patterns in classification by a feedforward neural network using back propagation learning, *Int. J. Remote Sensing,* **20,** 3549-3562.

[57] **Foody, G.M. and Mathur, A.**, 2006. The use of small training sets containing mixed pixels for accurate hard image classification: Training on mixed spectral responses for classification by a SVM, *Remote Sensing of Environment,* **103,** 179-189.

[58] **Dawson, M.S., Manry, M.T.**, 1993. Surface parameter retrieval using fast learning neural networks, Remote sensing Reviews, **7,** 1-18.

[59] **Valls, G.C., Chova, L.G., Mari, J.M., Frances, J.V. and Marivilla, J.C.**, 2006. Composite kernels for hyperspectral image classification, *IEEE Geosci. and Remote Sensing Letters,* **3,** 93-97.

[60] **Huang, C., Davis, L.S. and Townshend, J.R.G.**, 2002. An assessment of support vector machines for land cover classification, *Int. J. Remote Sensing,* **23,** 725-749.

[61] **Carpenter, G.A. and Grosberg, S.**, 1987. A Massively parallel architecture for a self organizing neural pattern recognition machine, *Computer Vision, Graphics, and Image Processing,* **37,** 54-115.

[62] **Landgrebe, D. and Biehl, L**., 1992. Multispec and AVIRIS NW Indiana's Indian Pines data set [Online]. http://www.ece.purdue.edu /~biehl /MultiSpec/index.html.

[63] **Landgrebe, D.A.**, 2003. Signal theory and methods in multispectral remote sensing, John Wiley & Sons, New Jersey, USA.

[64] **Varshney, P.K. and Arora, M.K.**, 2004. Advanced image processing techniques for remotely sensed hyperspectral data, springer, New York, USA

[65] Satimage data set : Data set available at http://www.liacc.up.pt/ML/old/statlog/ datasets.html.

[66] GLCF web site: Landsat 7 ETM + data available at http://glcf.umiacs. umd.edu/data.

[67] **Arikan, M.**, 2004. Parcel based crop mapping through multi-temporal masking classification of Landsat 7 images in Karacabey, Turkey. *Proceedings of the ISPRS Symposium, Istanbul International Archives of Photogrammetry. Remote Sensing and Spatial Information Science,* **34.**

[68] **Joachims, T.**, 1999. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press Cambridge, MA. .

[69] **Schwaighofer, A.**, MATLAB Interface to SVM light. Institute for Theoretical Computer Science at Graz University of Technology, Software available at http://www.cis.tugraz.at/igi/aschwaig/software.html.

[70] **Chang, C. and Lin, C.**, 2001. LIBSVM : a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/ ~cjlin/libsvm.

[71] **Belousov, A.I., Verzakov, S.A., and Von Frese, J.**, 2002. A flexible classification approach with optimal generalization performance :support vector machines, *Chemometrics and Intelligent Labotory Systems,* **64,** 15-25.

[72] **Hsu, C. and Lin, C-J.**, 2002. A comparison of methods for multiclass support vector machines, *IEEE Trans. Neural Networks,* **13,** 415-425.

[73] **Hsu, C-W, Chang, C-C, Lin, C-J.**, A practical guide to support vector classification. Softcopy of the paper available at http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[74] **Hausler, S**., Neural Network Toolbox: A tutorial for the course computational intelligence, Softcopy and matlab code are available at http://www.igi.tugraz.at/lehre/CI.

[75] ENVI Version 3.6, 2003. The Environment for Visualizing Images Research Systems, Inc. CO, USA. Software information available at http://www.RSInc.com/envi.

[76] **Duda, R.O., Hart, P.E. and Stork, D.G.**, 2001. Pattern Classification, John Wiley & Sons, New York, USA.

CURRICULUM VITAE

N. Gökhan Kasapoğlu received the B.Sc. degree from the Yıldız Technical University (YTU) in 1995 and the M.Sc. degree from the İstanbul Technical University (ITU) in 2000 both in Istanbul, Turkey and in Electronics and Communication Engineering. From 1999-2005, he was a senior system engineer at the Center for Satellite Communication & Remote Sensing (ITU-CSCRS). He is currently a research assistant in the Department of Electronics and Communication Engineering. His main research interests include pattern recognition for remote sensing, neural networks, synthetic aperture radar processing and SAR raw data compression. Mr. Kasapoğlu is a student member of IEEE, Geoscience and Remote Sensing Society.