İSTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

SUPPORT VECTOR SELECTION AND ADAPTATION AND ITS APPLICATION TO EARTHQUAKE DAMAGE ASSESSMENT

Ph.D. Thesis by
Gülşen TAŞKIN KAYA

Department	:	Computational Science and Engineering
Programme	:	Computational Science and Engineering

MAY 2011

İSTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

SUPPORT VECTOR SELECTION AND ADAPTATION AND ITS APPLICATION TO EARTHQUAKE DAMAGE ASSESSMENT

Ph.D. Thesis by
Gülşen TAŞKIN KAYA
(702032034)

Date of submission	:	**01 January 2011**
Date of defence examination	:	**16 May 2011**

Supervisor (Chairman)	:	**Asst. Prof. Mustafa E. KAMAŞAK (I.T.U.)**
Co-advisor	:	**Prof. Okan K. ERSOY (Purdue U.)**
Members of the Examining Committee	:	**Assoc. Prof. Dr. Cem ÜNSALAN (Yeditepe U.)**
		Prof. Muhittin GÖKMEN (I.T.U.)
		Prof. M. Serdar ÇELEBİ (I.T.U.)
		Prof. Nebiye MUSAOĞLU (I.T.U.)
		Asst. Prof. Şinasi KAYA (I.T.U.)

MAY 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

DESTEK VEKTÖR SEÇİMİ VE UYARLAMASI VE DEPREM HASAR TESPİT UYGULAMALARI

DOKTORA TEZİ
Gülşen TAŞKIN KAYA
(702032034)

Tezin Enstitüye Verildiği Tarih	:	**01 Ocak 2011**
Tezin Savunulduğu Tarih	:	**16 Mayıs 2011**
Tez Danışmanı	:	**Yard. Doç. Dr. Mustafa E. Kamaşak (İ.T.Ü.)**
Tez Eş Danışmanı	:	**Prof. Dr. Okan K. ERSOY (Purdue Ü.)**
Diğer Jüri Üyeleri	:	**Doç. Dr. Cem ÜNSALAN (Yeditepe Ü.)** **Prof. Dr. Muhittin GÖKMEN (İ.T.Ü.)** **Prof. Dr. M. Serdar ÇELEBİ (İ.T.Ü.)** **Prof. Dr. Nebiye MUSAOĞLU (İ.T.Ü.)** **Yard. Doç Dr. Şinasi KAYA (İ.T.Ü.)**

MAYIS 2011

FOREWORD

I would like to thank my advisor, Asst. Prof. Mustafa E. KAMAŞAK for his scientific support throughout my thesis. I would also like to extend my deepest appreciation to my co-advisor, Prof. Dr. Okan K. ERSOY for his valuable comments regarding my thesis and all the publications, assistance in developing this thesis and giving responses to my e-mails so quickly.

I would like to thank, Prof. Dr. Muhittin GÖKMEN, Prof. Dr. M. Serdar ÇELEBİ and Prof. Dr. Nebiye MUSAOĞLU for their valuable comments and suggestions regarding my thesis.

I would like to thank my friends Oğuzhan CEYLAN and E. Fatih YETKİN for their encouragement when it was most needed, and for all the unforgettable times together. I would also like to thank Birkan TUNÇ for his help about LaTeX style file.

A special thanks to my love, my beloved husband Huseyin KAYA, who has encouraged me, stayed with me whenever i needed, listened to me carefully although i talked too much and also accompanied with me in this exciting adventure of working on my thesis at Purdue University, USA.

I would also like to thank my family in Turkish, so the following will be in Turkish.

Kendisi ailemden olmasa da ailemden biriymiş gibi hissettiğim çok değerli sanatçı ve bestekar sayın Orhan GENCEBAY'a çok teşekkür ederim. Tezimin yazım aşamasındaki en zor günlerimde, yapmış olduğu birbirinden güzel eserleri, eşsiz yorumu ve sesi ile yanımda oldu.

Tez çalışmalarım boyunca bana tam destek veren ve uzun süren gecelerde uyumayıp beni bekleyen sevgili ablam Neşe ŞAHİN'e ve Amerika'da yalnız başıma olduğum sürede benimle birlikte kalan sevgili kardeşim Zeynep TAŞKIN'a çok teşekkür ederim. Akademik hayatım boyunca, her türlü zorlukta yanımda olan, bana bilimsel çalışmayı sevdiren, üniversiteye başlar başlamaz beni hiçbir zaman yalnız bırakmayan ve destekleyen hem eniştem hem de abim çok değerli Muhammed ŞAHİN'e ayrıca teşekkür ederim. En stresli zamanlarımda, kendi güzel dünyalarında beni de ağırlayan ve bana herşeyi unutturan canım yeğenlerim Defne ve Efe ŞAHİN'e çok teşekkür ederim. Beni doğuran, büyüten, bana emek veren ve hayatım boyunca beni her zaman çok seven hayatımın en değerli kadını sevgili annem Ayşe TAŞKIN'a ve akademik hayata atılmama vesile olan biricik babam Adem TAŞKIN'a çok teşekkür ediyor ve doktora tezimi canımdan çok sevdiğim anneme ve babama adıyorum.

MAY 2011 Gülşen TAŞKIN KAYA

TABLE OF CONTENTS

Page

ABBREVIATIONS

1NN	:	1 Nearest Neigbour
5NN	:	5 Nearest Neigbour
ASM	:	Angular Second Moment
B	:	Blue
C	:	C Programming Language
DEM	:	Digital Elevation Model
ERS	:	European Remote Sensing Satellite
G	:	Green
GIS	:	Geographic Information System
GLCM	:	Gray Level Co-occurrence Matrix
HYDICE	:	Hyperspectral Digital Imagery Collection Experiment
KKT	:	Karush Khun Tucker
KNN	:	K Nearest Neighbour
LANDSAT MSS	:	LANDSAT Multispectral Scanner System
LIBSVM	:	Library Support Vector Machines
LSVM	:	Linear Support Vector Machines
LVQ	:	Learning Vector Quantization
LVQPAK	:	Learning Vector Quantization Package
MSS	:	MultiSpectral Sensor
NIR	:	Near Infrared
NN	:	Neural Network
NSVM	:	Nonlinear Support Vector Machines
NSVM-1	:	Nonlinear SVM with Radial Basis Function
NSVM-2	:	Nonlinear SVM with Polynomial Function
NWFE	:	Nonparametric Weighted Feature Extraction
OA	:	Overall Accuracy
OAO	:	One Against One
PA	:	Producer's Accuracy
QP	:	Quadratic Programming
R	:	Red
RBF	:	Radial Basis Function
RBF	:	Radial Basis Function
RGB	:	Red Green Blue
RVM	:	Relevance Vector Machine
SAR	:	Synthetic Aperture Radar
SPOT HRVIR	:	SPOT High Visible Visualization Resolution
SRM	:	Structural Risk Minimization
STD	:	Standard Deviation
SV	:	Support Vectors
SVM	:	Support Vector Machines

SVSA	:	Support Vector Selection and Adaptation
UA	:	User’s Accuracy
UTM	:	Universal Transverse Mercator
VC	:	Vapnik-Chervonenkis

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF SYMBOLS

θ	:	Angle between pixels
d	:	Distance between pixels
η_0	:	Initial value of learning rate parameter
Φ	:	Kernel function operator
γ	:	Kernel parameter
α	:	Lagrangian coefficient
η	:	Learning rate parameter
C	:	Positive regularization function
ξ	:	Positive slack variable

SUPPORT VECTOR SELECTION AND ADAPTATION AND ITS APPLICATION TO EARTHQUAKE DAMAGE ASSESSMENT

SUMMARY

Remote sensing technology is a powerful tool to identify damaged regions after an earthquake. There are two methodological approaches in detection of earthquake damage, mono- and multi-temporal approaches. Especially for providing effective emergency management, mono-temporal approach is generally preferred in extraction of earthquake damage as it does not depend on availability of pre-earthquake image.

The support vector machine (SVM) is a nonparametric classifier and represents a very attractive approach in classification of linearly and nonlinearly separable data. It has often been found to provide higher classification accuracy than other widely used pattern recognition systems, such as parametric statistical classifiers and neural networks. However, classification of nonlinearly separable data with support vector machines conducted by using a nonlinear kernel function is often a difficult task especially due to the necessity of choosing a convenient kernel type. Moreover, in order to get the optimum classification performance with the SVM, a kernel and its parameters should be determined in advance. This process takes more computational time than SVM with linear kernel function.

In order to overcome or alleviate these difficulties of choosing kernel function and its parameters, a novel nonparametric method called Support Vector Selection and Adaptation (SVSA) has been introduced. It is applicable to both linearly and nonlinearly separable data and aims at achieving classification performance competitive with nonlinear support vector machines without using any kernel function with less computation time.

The proposed method consists of two steps: selection and adaptation. In the selection step, first the support vectors are obtained by a linear SVM. Then, these support vectors are classified by using the K-Nearest Neighbor method, and some of them are rejected if they are misclassified. In the adaptation step, in order to generate the reference vectors to be used for the classification, the remaining support vectors are iteratively adapted with respect to the training data excluding the support vectors. Afterwards, classification of the test data is carried out by 1 nearest neighbour method with the reference vectors using an adaptive distance metric.

The performance of the SVSA was tested with synthetic and remote sensing data and compared to SVMs and KNN methods. The results showed that the SVSA has better classification performance than linear SVM and a competitive classification performance with NSVM. Additionally, in terms of computational performance, it was observed that the SVSA is faster than the NSVM during the training process.

DESTEK VEKTÖR SEÇİMİ VE UYARLAMASI VE DEPREM HASAR TESPİT UYGULAMALARI

ÖZET

Uzaktan algılama teknolojisi, deprem sonrası oluşan hasarlı bölgelerin tespit edilmesinde oldukça güçlü bir araçtır. Deprem hasarlarının belirlenmesinde, tekli ve çoklu zamansal yaklaşım olmak üzere iki türlü yaklaşım vardır. Özellikle acil durum yönetimi açısından, deprem öncesi veriye gereksinim duyulmaması nedeniyle deprem hasarının belirlenmesinde genellikle tekli-zamansal yaklaşımlar tercih edilmektedir.

Destek Vektör Makineleri (DVM), parametrik olmayan bir sınıflandıcı olup, doğrusal ve doğrusal olarak ayrılamayan verilerin sınıflandırılmasında kullanılan oldukça etkili bir yöntemdir. Parametrik istatistiksel sınıflandırma yöntemleri ve yapay sinir ağları gibi oldukça sık kullanılan diğer örüntü tanıma yöntemlerine nazaran daha yüksek bir sınıflandırma başarımı verdiği gözlemlenmiştir. Ancak, doğrusal olmayan verilerin DVM ile sınıflandırılmasında, özellikle uygun bir çekirdek fonksiyonu seçimi gereksiniminden ötürü belirlenmesi oldukça zor olan doğrusal olmayan bir çekirdek fonksiyonu seçmek gerekmektedir. Ayrıca, destek vektör makineleri ile optimum bir sınıflandırma başarımı elde edebilmek için, çekirdek fonksiyonu ve parametrelerinin önceden belirlenmesi gerekmektedir. Bu işlem, doğrusal çekirdek fonksiyonlu destek vektör makinelerine nazaran daha fazla hesaplama zamanı gerektirmektedir.

Çekirdek fonksiyonu ve parametrelerinin şeçimi zorluklarının üstesinden gelmek veya bu zorlukları azaltmak için, Destek Vektör Seçimi ve Uyarlaması (DVSU) isimli yeni bir yöntem geliştirilmiştir. Yöntem hem doğrusal hem de doğrusal olmayan verilerin sınıflandırılmasında kullanılabilmekte ve herhangi bir çekirdek fonksiyonu kullanmaksızın, daha az bir hesaplamama zamanı ile doğrusal olmayan destek vektör makinelerinin performansına ulaşabilmektedir.

Önerilen yöntem, seçim ve uyarlama olmak üzere iki adımdan oluşmaktadır. Seçim aşamasında, ilk olarak doğrusal destek vektör makinesi ile destek vektörleri elde edilir. Ardından, bu destek vektörleri K en yakın komşuluk yöntemi ile sınıflandırılır ve yanlış sınıflandırılan destek vektörleri elenir. Uyarlama adımında, sınıflandırmada kullanılacak referans vektörlerini üretmek için, destek vektörlerinin çıkarılması ile güncellenen eğitim verilerine göre kalan destek vektörleri iteratif olarak uyarlanırlar. Ardından, uyarlanabilir uzaklık ölçütü kullanılarak, referans vektörlerine göre 1 en yakın komşuluk yöntemi ile test verisi sınıflandırılır.

DVSU yöntemi, sentetik ve uzaktan algılama verileri üzerinde kullanılarak, yöntemin performansı DVM ve KNN yöntemleri ile karşılaştırılmıştır. Sonuçlar, DVSU yönteminin doğrusal DVM yönteminden daha iyi ve doğrusal olmayan DVM yöntemi ile de yarışabilir nitelikte olduğunu göstermiştir. Ayrıca, eğitim sırasındaki hesaplama performansı açısından, DVSU yönteminin doğrusal olmayan DVM yönteminden daha hızlı olduğu gözlemlenmiştir.

1. INTRODUCTION

Remote sensing data have been used in a wide range of subjects such as land use monitoring, and management, drought, flood and earthquake damage assessment. It has been a popular ongoing research field especially for providing fast solutions to earthquake damage assessment. Change detection techniques, statistical classification methods and machine learning approaches have been applied to remote sensing data in order to identify the earthquake damage. Supervised classification methods such as statistical classification methods, and the machine learning approaches have been widely preferred to be used in damage assessment, as they do not require a pre-earthquake image.

Classification approaches using Bayesian methods are one of the basic approaches in statistical pattern recognition [1]. The most common supervised classification technique used in remote sensing area is the maximum likelihood classifier. As Bayesian concepts are density based and assume specific density functions for each information class, final decision rule (also referred to as the maximum as-posterior rule) is based on the a-posteriori probabilities. However, the conditional probability density function is generally not known and must be estimated from the training data. Hence, the use of a Bayesian classifier might be critical if an adequate multivariate statistical model is not available and the number of training samples is limited with respect to the dimensionality of the dataset [2]. The performance of the classifier can be affected, resulting in a lower classification accuracy. In such cases, operating a non-parametric method is generally preferred if the form of the density function is unknown. Approaches such as K-Nearest Neighbour (KNN)[1], Neural Networks (NN) and kernel-based methods are the alternative approaches to statistical classification approaches.

[1] See Appendix 6 for detailed information.

The K-Nearest-Neighbours is a non-parametric classification method, which is simple but effective in many cases [3]. The original KNN does not need any training process to make a decision. It uses all training data in classification, and that is why it is called "lazy learning". However, heavy computational load that is proportional to the number of samples, and the number of dimensions of the feature space during testing are important disadvantages of the algorithm. Moreover, it is necessary to choose an appropriate value for neighbourhood K, and the performance of classification is very much dependent on this value. Choosing $K = 1$ is also the most sensitive to noisy samples, and it generally reduces the classification performance.

Neural networks have an advantage over traditional statistical methods as they are distribution free [4]. In other words, they do not depend on any underlying statistical distribution of the data. In the context of classification, a neural network can be considered as a black box model that receives a set of input vectors and produces responses from its output neurons, where the number of neurons depends on the number of information classes (i.e. land cover classes). However, the NN approach can be computationally complex and requires a large number of training samples.

Support Vector Machines (SVM) have been introduced as a non-parametric classifier for pattern recognition and machine learning [5, 6, 7, 8]. SVM is a relatively recent development in the context of remote sensing and still needs further improvement, modification and development of new kernel concepts. SVM is based on an optimal linear separating hyperplane that is fitted to training samples of two classes within a multi-dimensional feature space. The optimization problem that has to be solved relies on Structural Risk Minimization (SRM) and is aimed at maximization of the margins between the hyperplane and the closest training samples. For linearly nonseparable samples, the input data is mapped into a high dimensional space, and the computationally extensive mapping process is handled by using a positive definite kernel function.

Using SVMs, several applications have been conducted in the field of remote sensing image analysis [9,10,11,12,13]. In many studies, they performed more accurately than other classifiers or performed at least equally well. On the other hand, the majority of the studies uncovered common limitations to SVM methodologies, for example, selection of the kernel functions. Choosing type of kernel function considerably affects the classification performance, and it is not generally known before classification. Moreover, even if the kernel function is determined before the classification, choice of the parameter values for the kernel function, which controls the tradeoff between maximizing the margin and minimizing the training error is another important consideration in SVM applications.

In order to overcome such difficulties with nonlinear SVM (NSVM), Support Vector Selection and Adaptation (SVSA) is introduced as a nonparametric supervised classification method. The SVSA uses the support vectors obtained by the linear SVM (LSVM) and selects some which are useful in terms of classification performance. The selected vectors are next called reference vectors and adapted with respect to the training data by using Learning Vector Quantization (LVQ)[2] [14]. During classification, the selected and adapted vectors, called reference vectors, are used with 1 Nearest Neighbour method (1NN) using adaptive learning distance metric. The SVSA handles both linearly and nonlinearly separable data with a competitive classification performance to NSVM. It also outperforms the LSVM in terms of classification performance. The main advantage of the method is to achieve competitive classification performance without choosing any kernel function and its parameters at a lower computational load.

In order to analyse the performance of the proposed method, synthetic data, UCI-repository and Statlog-repository data [15], and different types of remote sensing data including multisource, multispectral and hyperspectral data were used in the experiments. As an earthquake implementation, a small urban region selected from a high resolution satellite image from city of Bam in Iran was used in order to identify the damage patterns by the SVSA using a pixel-based classification. Morever, in order to improve the classification performance of the SVSA, texture properties of the area of interest were additionally included to the pixel features, and damage assessment was carried out with both pixel and textural features. All the experimental results showed that the SVSA is competitive with nonlinear SVM and better than linear SVM.

[2] See Appendix B for detailed information.

1.1 Methodological Literature Review

Recently, particular attention has been dedicated to SVM for the classification of multispectral and hyperspectral remote sensing images [16, 17]. SVMs have often been found to provide higher classification accuracy than other widely used pattern recognition techniques, such as maximum likelihood or multilayer perceptron classifiers. SVMs are also competitive with other kernel-based methods such as radial basis function, neural networks and kernel Fisher discriminant analysis at a lower computational cost [18]. There have been many research studies that use the SVM model in the classification problem. Waske and Benediktsson (2010) addressed the classification of multisensor datasets, consisting of multitemporal Synthetic Aperture Radar (SAR) data and optical imagery [19]. The original outputs of each SVM discriminant function were used in the subsequent fusion process, and it was shown that the SVM-based fusion approach significantly improves the results of a single SVM. Mari et al. (2010) presented two semisupervised one-class support vector machine classifiers for remote sensing applications [20]. The first proposed algorithm was based on modifying the one-class SVM kernel by modelling the data marginal distribution. The second one was based on a simple modification of the standard SVM cost function.

A Linear Support Vector Machine (LSVM) is based on determining an optimum hyperplane that separates the data into two classes with the maximum margin [21]. The LSVM typically has high classification accuracy for linearly separable data. However, for nonlinearly separable data, LSVM has poor performance. For this type of data, a Nonlinear Support Vector Machine (NSVM) is preferred. The NSVM transforms the input data using a nonlinear kernel followed by the LSVM. Although the NSVM can achieve higher classification performance, it requires high computation time to map the input to a higher dimensional space by a nonlinear kernel function which is usually a fully dense matrix [22]. The computational complexity of the NSVM grows with the cube of the total number of training samples. It is a critical drawback especially when dealing with hyperspectral image classification, where the dimensionality of the original data is high, and kernel mapping is more vulnerable to dimensionality problems.

It is well known that the major task of the NSVM approach lies in the selection of its kernel. Choosing different kernel functions produces different SVMs and may result in different performances [23, 24, 25]. Therefore, exploration of new techniques and systematic methodology to construct an efficient kernel function for designing SVMs in a specific application are important research directions [26]. It is also desirable to have a classifier model with both the efficiency of linear SVM and the power of the nonlinear SVM. For this purpose, Chi and Ersoy (2002) presented SVM-based binary decision trees [27]. Kasapoglu and Ersoy (2007) developed another classification algorithm for remote sensing images to detect border vectors determined by using class centers and misclassified training samples, followed by their adaptation with a technique similar to the Learning Vector Quantization (LVQ) [28]. Fu and Robles-Kelly (2008) proposed another approach which is a mixture model combining linear SVMs for the classification of nonlinear data based on divide-and-conquer strategy that partitions the input space into hyper-spherical regions [29].

It is also worth to underline that the kernel-based implementation of SVMs involves the selection of multiple parameters, including the kernel parameters such as the parameters for the Gaussian and polynomial kernels and the regularization parameters. For optimal performance, the kernel parameters have to be adjusted before classification. Since these parameters have a regularization effect on the cost function that is minimized during the training process, they may decrease the overall classification performance if they are not well chosen. Selection of the kernel parameters is empirically done by trying a finite number of values and keeping those that provide the highest classification accuracy. This procedure requires a grid search process over the space of the parameter values. It may not be easy to locate the grid interval without *prior* knowledge of the problem. Since the optimal parameter values vary from kernel to kernel as well, kernel parameter selection might be a challenging task.

Since choosing kernel parameters is critical to the performance of the NSVM, other methodologies have also been introduced in the literature. Two automatic techniques have been developed which are based on the idea of estimating the parameter values so that 1) they maximize the margin; and 2) they minimize the estimate of the expected generalization error. In one of them, Chapelle and Vapnik (2002) discussed an analytical criterion that is a proxy of Vapnik-Chervonenkis (VC) dimension known as the radius-margin bound [30]. This criterion is a quadratic function of the support vectors multipliers. An automatic minimization algorithm is proposed for determining the best kernel parameters. In the second one, Chung et al. (2003) discussed use of differentiable bounds of the leave-one-out error [31]. Optimization of the parameters is then carried out using a gradient descent search over the space of the parameters. Ayata et al. (2005) discussed a SVM model selection criterion based on the minimization of an empirical error estimate so that generalization error can drastically be reduced. The SVM hyperparameters can be adequately chosen based on this empirical error criterion [32]. Recently, Foody (2008) introduced a new method called a Relevance Vector Machine (RVM) approach as a way to address the need to define the regularization parameter [33]. RVMs are considered as a Bayesian treatment alternative to SVMs and have several advantages including probabilistic predictions, automatic estimation of parameters, and arbitrary kernel functions. The author argues that the new method leads to reduced sensitivity to the hyperparameter settings.

In the area of hyperspectral remote sensing image classification, Bazi and Melgani (2006) proposed an optimal SVM classification system to detect the best discriminative features and to estimate the best SVM parameters with radius-margin bound minimization by means of a genetic optimization framework [34].

In this thesis, in order to overcome the mentioned drawbacks of the NSVM, a new method called Support Vector Selection and Adaptation (SVSA) is introduced [35]. The proposed method has some advantages over the NSVM. In the training stage, the SVSA requires less computation time compared to NSVM, and no kernels are needed. With nonlinearly separable data, classification performance of the SVSA is competitive with NSVM. During the preliminary tests with the SVSA, it was observed that the SVSA also outperforms LSVM [36, 37, 38, 39].

1.2 Literature Review of Earthquake Damage Assessment

Remote sensing technology is the most effective technology for disaster relief agencies and civil protection units to assess damage for planning purposes [40]. Especially in the aftermath of extreme earthquakes, in terms of providing critical information for emergency responders, rapidly quantifying the extent and severity of building damage is a high priority.

For earthquake damage assessment, mostly used methodological approaches are either mono- or -multi-temporal [41]. In multi-temporal approaches, earthquake-induced damage is assessed by change detection methods that involves the analysis of two geometrically registered multispectral remote sensing images acquired in the same geographical area at two different times.

Considerable amount of studied can be found in the literature utilizing change detection methods [42,43]. Turker and San (2003) used pre-event and post-earthquake images in order to detect earthquake-induced changes on 17 August 1999 in Izmit, Turkey [44]. In order to preserve the spatial and spectral information, the multispectral and panchromatic images were merged. The change areas were detected by subtracting the image brightness values for the near-infrared (NIR) channel of merged pre-earthquake image from the post-earthquake image. Adams (2004) analysed multitemporal change detection methodologies by both using high resolution Quickbird and moderate resolution SPOT and ERS satellite imagery [45]. Kosugi et al. (2004) introduced a change detection system capable of detecting geographical changes from a pair of aerial or satellite images automatically and detected urban change detection related to earthquakes [46]. Turker and Cetinkaya (2005) discussed detection of the collapsed buildings caused by earthquakes using Digital Elevation Models (DEMs) created from pre- and post-earthquake stereo aerial photographs [47]. Rathje et al. (2005) employed a semi-automated thematic classification algorithm to identify damage patterns based on a comparison of pre-and-post event high resolution imagery [48]. Gamba et al. (2006) addressed a problem of change detection from Synthetic Aperture Radar (SAR) images [49]. The authors proposed an approach based on the extraction and comparison of linear features from multiple SAR images to confirm pixel-based changes. The procedure was validated through identifying earthquake damage. Gamba et al. (2007) addressed a problem regarding post-earthquake building damage assessment performed by combining pre- and post-earthquake images by using SAR images and ancillary data [50].

For post-earthquake damage assessment, multi-temporal approach using change detection approaches has some disadvantages in terms of limitations like short time gap between pre- and post-earthquake imagery or changes in brightness values caused by external factors. Therefore, using only a post-earthquake image is commonly preferred to identify earthquake damage. In the literature, there are several studies performed using only post-earthquake satellite and aerial images, called mono-temporal approach, to detect earthquake-induced damages and changes [51].

Gamba et al. (1998) developed a system which provides a rapid and reliable damage detection without using pre-earthquake image for damage assessment [52]. The pre-earthquake image was only used to extract data about buildings and infrastructures, followed by analysis exploiting Geographical Information System (GIS) capabilities. Immediately after the occurrence of the earthquake, post-earthquake image of the affected area was next obtained to be compared with the pre-earthquake dataset in order to identify earthquake damage. Turker and San (2004) used post-earthquake aerial photographs in order to detect the collapsed buildings caused by the Izmit, Turkey earthquake on 17 August 1999 by using corresponding shadows of the buildings and GIS integration [53]. Saito and Spence (2004) introduced the qualitative and quantitative methods for damage assessment with some results using both types of methodologies, comparison of the results with ground truth data and some suggestions for the application of these damage maps [54]. Balz and Liao (2010) presented theoretical assumptions about the appearance of collapsed buildings in high-resolution SAR images and verified with visual feature interpretations of real SAR images from the area for building-damage assessment using only post-earthquake SAR images [55].

Another way of identifying earthquake damage in mono-temporal approach is to use a supervised classification algorithm. Gong et al. (2010) focused on the detection of geological hazards in the Wenchuan earthquake by using support vector machines and visual interpretation which were applied under the hierarchical stripping classification framework to extract objects separately [56].

1.3 Thesis Outline

This thesis consists of six chapters. A brief summary of each chapter is given below.

Chapter 1 introduces the overall thesis topic and discusses the motivation for the thesis work. In addition, it gives the literature review of the thesis in terms of both methodological approaches and earthquake damage assessment and outlines the thesis manuscript.

Chapter 2 reviews general overview of SVM for both linearly and nonlinearly separable data. Classification of multiclass data by the SVM is also summarized.

Chapter 3 discusses the theoretical formulation of the proposed method, SVSA, and its advantages over LSVM and NSVM. It also discusses the classification procedure by the SVSA and describes the algorithm of the SVSA.

Chapter 4 presents the basic criteria used for comparisons between the algorithms, and the computational complexities of the proposed method, SVMs and KNN. Additionally, it presents the results of classification conducted by the proposed algorithm in comparison to the classification performances of the other methods. For this purpose, three different types of synthetic datasets were used to demonstrate the classification mechanism of the proposed SVSA and to show differences with some other classification methods. The synthetic data used were generated in a two-dimensional feature space, which made it possible to visually display the decision boundaries and to help understand the classification behaviour of the classifiers. In order to show the power of the proposed algorithm with data having more than two features, some datasets from UCI-repository and Statlog-repository were also analysed, and classification performances were compared in terms of classification performance. Some remote sensing classifications with different type of source, multisource, multispectral and hyperspectral data were also analysed with the proposed algorithm.

Chapter 5 consists of a detailed earthquake damage assessment analysis with the proposed algorithm. Both pixel- and hybrid pixel- and texture-based classification was carried out with the proposed algorithm. In hybrid pixel- and texture-based classification, Nonparametric Weighted Feature Extraction (NWFE) method was also used to improve the classification performance further. For implementation, a sub-region was chosen from city of Bam in Iran, and earthquake damage pattern was estimated with the proposed method.

Chapter 6 presents the conclusions and the topics to be addressed in feature research.

2. SUPPORT VECTOR MACHINES

Support vector machines (SVMs) constitute a supervised non-parametric statistical learning technique, and there is no assumption on the data distribution. In its original formulation, the method is presented by Vapnik with a set of labelled data samples and the SVM training algorithm aims to find a hyperplane that separates the dataset into a discrete number of classes consistent with the training samples [57]. The term optimal separation hyperplane is used to refer to the decision boundary that minimizes misclassification, obtained in the training step. Learning refers to the iterative process of finding a classifier with optimal decision boundary to separate the training samples (in potentially high dimensional space) and then to separate test samples under the same configurations.

The underlying principle of SVMs involves Structural Risk Minimization (SRM). Under this scheme, SVMs minimize classification error of unseen samples without prior assumptions made on the probability distribution of the samples.

2.1 Support Vector Machines

Support vector machine chooses a maximum-margin hyperplane and splits the sample classes, while maximizing the distance to the nearest split example. The parameters of the solution hyperplane are derived from a Quadratic Programming (QP) optimization problem. In its simplest form, SVM is a binary classification algorithm, but it can also be generalized to multiclass problems.

2.1.1 Linearly separable case

The implementation of a linear SVM assumes that features of data are linearly separable in the input space.

Let us consider a supervised binary classification problem. Let's assume that the training set consisting of n samples, denoted by Equation **2.1**, and separable by a linear hyperplane.

$$X = \Big\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n),\ \mathbf{x}_i \in \mathbb{R}^d,\ y \in \{+1, -1\}\Big\} \tag{2.1}$$

where d and $y \in \{+1, -1\}$ correspond to number of features and the class labels of training data, respectively.

The decision function is based on the function $\text{sgn}(f(\mathbf{x}))$, where $f(\mathbf{x})$ is the discriminant function associated with the hyperplane and defined as

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + \mathbf{b}, \quad \mathbf{w} \text{ and } \mathbf{b} \in \mathbb{R}^d \tag{2.2}$$

The linear hyperplane should satisfy the following constraints that define the separation of the data samples:

$$y_i = 1 \qquad \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + \mathbf{b} \geq 1 \tag{2.3}$$

$$y_i = -1 \qquad \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + \mathbf{b} \leq -1, \quad i = 1, \ldots, n \tag{2.4}$$

Or in terms of one compact equation:

$$y_i\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + \mathbf{b}\right) \geq 1, \quad i = 1, \ldots, n \tag{2.5}$$

It has been shown that classification is affected by another quantity, the margin (Figure 2.1). The margin is defined as the minimal distance from the separating hyperplane to the closest data sample. The margin depends on the length of the weight vector **w**.

Among all the hyperplanes separating two classes, there exists only a unique one yielding the maximum margin of separation between the classes. The optimal hyperplane is geometrically equivalent to maximizing the margin, $2/||\mathbf{w}||$, the perpendicular distance between two parallel hyperplanes. To maximizing the margin is equivalent to minimizing the norm of the weights.

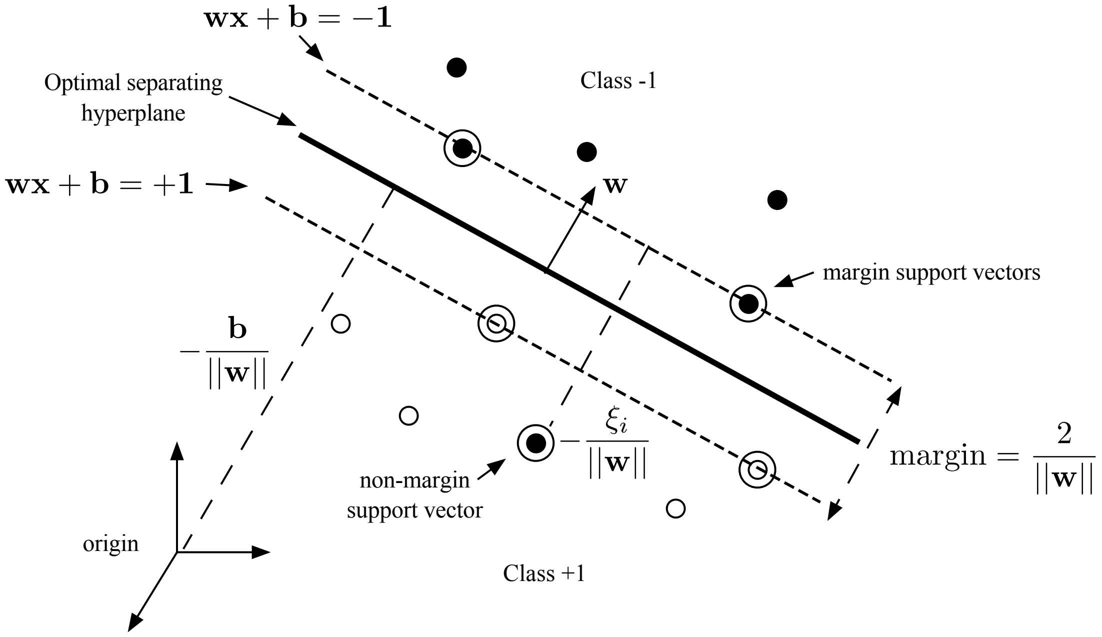

$\mathbf{wx}+\mathbf{b}=-\mathbf{1}$
Optimal separating hyperplane
Class -1
$\mathbf{wx}+\mathbf{b}=+\mathbf{1}$
$\mathbf{w}$
margin support vectors
$-\frac{\mathbf{b}}{||\mathbf{w}||}$
$-\frac{\xi_i}{||\mathbf{w}||}$
margin $=\frac{2}{||\mathbf{w}||}$
non-margin support vector
origin
Class +1

Figure 2.1: Optimal separating hyperplane in SVMs for a linearly nonseparable case. White and black circles refer to the classes $+1$ and -1, respectively. Support vectors are indicated by extra circles.

In order to construct this optimal hyperplane for separable case, the following quadratic optimization problem has to be solved with the following linear constraints:

$$\text{Object function} \;:\; \min_{\{\mathbf{w},\mathbf{b}\}} \frac{1}{2}||\mathbf{w}||^2 \tag{2.6}$$

$$\text{Subject to} \;:\; y_i\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i+\mathbf{b}\right) \geq 1, \quad i=1,\cdots,n \tag{2.7}$$

This constrained optimization problem can be translated to the unconstrained optimization problem by introducing Lagrange multipliers:

$$L(\mathbf{w},\mathbf{b},\alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i \left\{y_i\left[\mathbf{w}^{\mathrm{T}}\mathbf{x}_i+\mathbf{b}\right]-1\right\} \tag{2.8}$$

where α_i are Lagrange multipliers. The Lagrangian $L(\mathbf{w},\mathbf{b},\alpha)$ has to be minimized with respect to **w** and **b** and maximized with respect to $\alpha_i \geq 0$.

Using Karush-Kuhn-Tucker (KKT) conditions, the parameters **w** and **b** in Equation **2.8** can be expressed in terms of only the parameters α_i. Then, Equation **2.8** becomes a dual problem that requires only maximization with respect to the Lagrangian multipliers α_i.

According to the Kuhn-Tucker theorem, the solutions $\mathbf{w}^*$, $\mathbf{b}^*$ and α^* parameters of Equation **2.8** should satisfy the following conditions:

$$\frac{\partial L(\mathbf{w}^*, \mathbf{b}^*, \alpha^*)}{\partial \mathbf{b}} = 0; \quad \frac{\partial L(\mathbf{w}^*, \mathbf{b}^*, \alpha^*)}{\partial \mathbf{w}} = 0 \qquad \textbf{(2.9)}$$

Solving partial derivatives gives the following properties of optimal hyperplanes:

1. The coefficients α_i^* should satisfy the constraints:

$$\sum_{i=1}^{n} \alpha_i^* y_i = 0 \qquad \alpha_i \geq 0 \qquad \textbf{(2.10)}$$

2. The vector $\mathbf{w}^*$ is a linear combination of the vectors in the training set

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \qquad \alpha_i \geq 0 \qquad \textbf{(2.11)}$$

By the KKT complimentary conditions of optimization, the α_i must be nonzero for all the constraints, which are met with equality, in Equation **2.5**, thus

$$\alpha_i^* \left[y_i \left(\mathbf{w}^{*\mathrm{T}} \mathbf{x}_i + \mathbf{b}^* \right) - 1 \right] = 0, \qquad i = 1, \ldots, n. \qquad \textbf{(2.12)}$$

The training samples corresponding to the nonzero Lagrangian multipliers are called support vectors that lie at a distance exactly equal to $1/||\mathbf{w}||$ from the optimal separating hyperplane (Figure 2.1). Since the support vectors are the data samples closest to the decision function, conceptually they are the samples that are the most difficult to classify. Therefore, the decision function of the optimal hyperplane is written in terms of the support vectors while all the remaining training samples are irrelevant to the solution.

In order to construct the dual problem, the $\mathbf{w}$ and $\mathbf{b}$ parameters in Equation **2.8** are replaced with Equation **2.10** and Equation **2.11**. After substituting the expressions into the Lagrangian, the dual form of QP problem becomes

$$\begin{aligned} L(\alpha) &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j \right) \\ \text{subject to constraints} \quad & \sum_{i=1}^{n} \alpha_i^* y_i = 0 \qquad \alpha_i \geq 0. \qquad \textbf{(2.13)} \end{aligned}$$

By solving the QP problem, the Lagrangian parameters α_i^*'s can be determined. Afterwards, with the representation of the hyperplane decision function in terms of $\alpha_i^*, \cdots \alpha_n^*$ and $\mathbf{b}^*$, the function $f(\mathbf{x})$ in Equation **2.2** is achieved as follows:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i \in S} \alpha_i^* y_i \left(\mathbf{x}_i^{\mathrm{T}} \mathbf{x}\right) + \mathbf{b}^*\right) \tag{2.14}$$

where S is the subset of training samples corresponding to vectors with nonzero Lagrange multipliers α_i^*'s. It is worth noting that the Lagrange multipliers effectively weigh each training sample according to its importance in determining the decision function. The parameter $\mathbf{b}^*$ is computed as following from Equation **2.12** and from the set of support vectors $\mathbf{x}_i,\ i \in I \equiv \{i : \alpha_i \neq 0\}$.

$$\mathbf{b}^* = \frac{1}{|I|} \sum_{i \in I} \left(y_i - \sum_{j=1}^{n} \alpha_j y_j \left(\mathbf{x}_i . \mathbf{x}_j\right) \right) \tag{2.15}$$

2.1.2 Linearly nonseparable case

In practice, data samples of different class labels overlap one another. This makes linear separability difficult as the basic linear decision boundaries are often not sufficient to classify patterns with high accuracy. In this case, soft margin method is used [58].

In the hyperplane formulation, a data sample is nonseparable only if it does not satisfy Equation **2.5**. This corresponds to a data sample that falls within the margin or wrong side of the decision boundary (see Figure 2.1). In order to handle nonseparable training data, the concept of optimal separating hyperplane is generalized by introducing a positive slack variable ξ for each training vector to separate with a minimum number of errors.

$$y_i \left(\mathbf{w}^{\mathrm{T}} \mathbf{x}_i + \mathbf{b}\right) \geq 1 - \xi_i, \quad i = 1, \cdots, n \quad \xi_i > 0 \tag{2.16}$$

The new cost function under Equation **2.16** constraints is defined as the solution that minimizes the cost function that expresses a combination of two criteria: margin maximization (as in the case of linearly separable data) and error minimization (to penalize the wrongly classified samples).

$$\psi(\mathbf{w}, \xi) = \frac{1}{2} \|\mathbf{w}\| + C \sum_{i=1}^{m} \xi_i \tag{2.17}$$

where C is a positive regularization constant which controls the degree of penalization of the slack variables ξ_i, so that, when C increases, fewer training errors are permitted, although the generalization performance may degrade. The resulting classifier is usually called soft margin classifier. If $C = \infty$, no value for ξ_i except 0 is allowed; it is the so-called hard margin SVM case.

This optimization problem can also be translated to its dual form as follows:

$$\begin{aligned} L(\alpha) &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j \left(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j\right) && \textbf{(2.18)} \\ \text{subject to :} \quad & \sum_{i=1}^{n} y_i\alpha_i = 0, \qquad 0 \le \alpha_i \le \frac{C}{n}. && \textbf{(2.19)} \end{aligned}$$

Solution of this optimization problem by KKT condition gives the optimum hyperplane as follows:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i\in S} y_i\alpha_i^* \left(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j\right) - \mathbf{b}^*\right) \qquad \textbf{(2.20)}$$

where S is the set of support vectors, and $\mathbf{b}^*$ is found by averaging over all the training samples, which are calculated by using the following KKT conditions:

$$\begin{aligned} \alpha_i \left(y_i(\mathbf{w}\,\mathbf{x}_i - \mathbf{b}^*) - 1 + \xi_i\right) &= 0 \\ (C - \alpha_i)\,\xi_i &= 0 \qquad \textbf{(2.21)} \end{aligned}$$

Equation **2.21** also indicates that $\xi = 0$ if $\alpha_i < C$. Therefore, $\mathbf{b}^*$ can be averaged over only those samples which $0 \le \alpha_i \le C$.

In the nonseparable case, two kinds of support vectors exist: 1) margin support vectors and 2) non-margin support vectors. The margin support vectors lie on the hyperplane margin (If $\alpha_i < C$, then $\xi = 0$). The non-margin support vectors fall on the wrong side of this margin. In other words, when $\alpha_i = C$, the support vectors are missclassified if $\xi > 0$. When $0 < \xi \le 1$, the support vectors are correctly classified but are closer than the $1/\|\ \mathbf{w}\ \|$ from the hyperplane (see Figure 2.1).

2.2 Nonlinear Support Vector Machines

In case of nonlinearly separable data, using a nonlinear function Φ, the training data are mapped as $\Phi : \mathbb{R}^n \to \mathbb{R}^H$ into a higher dimensional feature space in order to make the data linearly separable. In this way, nonlinear SVM makes the maximum margin hyperplane be fit in a feature space (Figure 2.2).

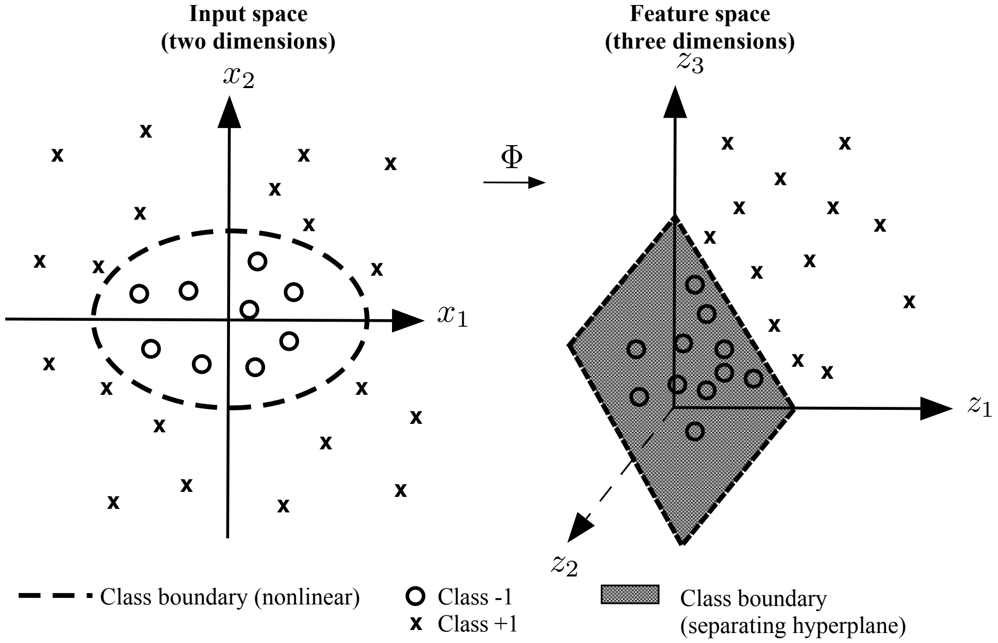


Figure 2.2: The idea of nonlinear SVM is to map the training data into a higher dimensional feature space via Φ, and construct a separating hyperplane with maximum margin there. This yields a nonlinear decision boundary in input space.

A linear classification problem is next formulated in that feature space [21]. Accordingly, the Lagrangian of the dual optimization problem (Equation **2.13**) has to be changed to the following equation:

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K\left(\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j\right) \qquad \textbf{(2.22)}$$

Notice the input vectors are involved in the expression through a kernel function

$$K(\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j) = \Phi(\mathbf{x}_i)^{\mathrm{T}} \Phi(\mathbf{x}_i) \qquad \textbf{(2.23)}$$

Kernel functions have to satisfy the condition stated in Mercer's theorem so as to correspond to some inner product in the transformed (higher) dimensional feature space [59]. This kind of kernel function allows to simplify the solution of the dual problem considerably, since it avoids the computation of the inner products in the transformed space.

The final result is a discriminant function conveniently expressed as a function of the data in the original (lower) dimensional feature space.

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i\in S} y_i\alpha_i K\left(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j\right) - b\right) \tag{2.24}$$

Some commonly used kernels in the literature are tabulated in Table 2.1:

Table 2.1: Commonly used types of kernel functions in the literature.

Kernel	$K(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j)$
Linear	$\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$
Radial Basis Function	$\exp\left(-\gamma\lVert\mathbf{x}_i-\mathbf{x}_j\rVert^2\right),\ \gamma>0$
Polynomial	$\left(\gamma\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j+r\right)^d,\ \gamma>0$
Sigmoid	$\tanh\left(\gamma\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j+r\right)$
Multi quadratic	$\sqrt{\lVert\mathbf{x}_i-\mathbf{x}_j\rVert+\eta}^{1/2}$
Inverse multi quadratic	$\sqrt{\lVert\mathbf{x}_i-\mathbf{x}_j\rVert+\eta}^{-1/2}$

Where γ, r, η and d are kernel parameters. The most commonly used kernel function is Radial Basis Function (RBF) kernel which nonlinearly maps data into a high dimensional space. Unlike the linear kernel, it can handle a case when the relation between class labels and features is nonlinear. Furthermore, the linear kernel is a special case of RBF as the linear kernel with a penalty parameter C has the same performance as the RBF kernel with some parameters (C,γ) [60]. In addition, the sigmoid kernel behaves like RBF for certain parameters [61].

2.3 SVM: Multiclass Strategies

The original SVM method was intrinsically designed for two class problems. However, the classification of multispectral and hyperspectral remote sensing data usually involves more than two classes. In the literature, there have been many strategies of combination of SVMs considered to evaluate the impact of the multiclass problem in the context of remote sensing data classification [62]. In this thesis, One-Against-One (OAO) strategy was used to generalize the SVM to classify multiclass data [63].

The OAO strategy, also known as pair-wise coupling, consists of constructing one SVM for each pair of classes. The final decision in OAO strategy is based on the winner-takes-all rule. In other words, the learning model generated by the SVMs with training of each pair-wise class gives one vote to the winning class, and the sample is labelled with the class having the most votes.

Let's assume $W = \{w_1, w_2, \cdots, w_T\}$ be the set of T possible labels corresponding to information classes. The OAO strategy involves $T(T-1)/2$ SVMs trained for all possible pair-wise classifications in order to distinguish the samples of one class from the samples of another class. In this case, each SVM carries out a binary classification in which two information classes w_i and $w_j = (w_i \in W,\ w_j \in W,\ i \neq j)$ are analysed against each other by means of a discriminant function $f_{ij}(\mathbf{x})$. Consequently, the grouping becomes

$$\begin{cases} W_A = w_i \\ W_B = w_j \end{cases} \tag{2.25}$$

Before the decision process, it is necessary to compute for each class $w_i \in W$ a score function $S_i(x)$, which sums the favourable and unfavourable votes expressed for the considered class

$$S_i(x) = \sum_{\substack{j=1 \\ j \neq i}}^{T} \operatorname{sgn}\{f_{ij}(x)\} \tag{2.26}$$

The final decision in the OAO strategy corresponds to the following maximization

$$w^* = \arg \max_{i=1,\cdots,T} \{S_i(x)\} \tag{2.27}$$

Sometimes, conflicting may occur between two different classes characterized by the same score. Such ambiguities can be solved by selecting the class with the highest prior probability.

3. SUPPORT VECTOR SELECTION AND ADAPTATION

SVMs are particularly appealing in the remote sensing field due to their ability to successfully handle small training datasets, often producing higher classification accuracy than the traditional methods [64].

Alongside the benefits derived from the SVM formulation, there are also several challenges in application of SVM. The major setback concerning the applicability of SVMs is the choice of kernels. Although many types of kernel functions are available in the literature, some of the kernel functions may not provide optimal SVM configuration for remote sensing applications. Empirical evidence indicates that kernels such as radial basis function and polynomial kernels applied on SVM-based classification of satellite image data produce different results. In addition to kernel function, choice of kernel parameters also affects performance of SVM. Therefore, difficulty in selection of proper kernel and its parameters generally limits effective cross-disciplinary applications of SVMs [65].

In this thesis, our motivation is to overcome some of these general classification problems of nonlinear SVM by developing a classification algorithm which is directly based on using the support vectors and their adaptation with respect to the training data to optimize partitioning in the feature space. The main contribution is to introduce a new non-parametric supervised nonlinear classifier, called Support Vector Selection and Adaptation (SVSA) method, without choosing any kernel function and parameters [66].

The SVSA is a novel supervised classification method for classification of both linearly and nonlinearly separable data [35]. In terms of classification performance, the SVSA usually outperforms the LSVM. It is also competitive with the NSVM in the classification of nonlinearly separable data. Therefore, a nonlinear as well as linear classification performance can be achieved by the SVSA without the need for a kernel.

Only the support vectors of the LSVM, which can be considered as the most important vectors (closest to the decision boundary) are used in the SVSA. The method consists of two stages: selection of support vectors obtained by LSVM and adaptation of the selected support vectors. Some of the support vectors are selected based on their contribution to overall classification accuracy, and they are then called reference vectors. Afterwards, they are adaptively modified by using LVQ with respect to the training data [67]. At the end of the adaptation process, the reference vectors are finalized and used in classification during testing with the 1 Nearest Neighbour (1NN) method with adaptive learning metric [68]. The learning schema of the SVSA method is shown in Figure 3.1.

The proposed algorithm replaces the use of kernel functions by the following steps: a selection step to choose the most significant linear support vectors for classification, subsequent adaptation of the chosen linear support vectors for optimal classification performance by using the LVQ adaptation scheme, and finally the 1NN rule for final classification. It is known that the LVQ adaptation also maximizes the hypothesis margin and also the sample margin, since the sample margin is larger than the hypothesis margin [69]. Choosing the most significant linear support vectors reduces the number of reference vectors. Such reduction is also known to result in better generalization performance [70]. The SVSA also keeps the step of determining the linear SVM the same. Therefore, the support vectors used in the SVSA are based on Structural Risk Minimization (SRM) and VC dimension for a LSVM.

3.1 Support Vector Selection

Let n, d, and k denote the number of training samples, the number of features, and the number of support vectors, respectively. Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ represent the training data with $\mathbf{x}_i \in \mathbb{R}^d$, $y \in \mathbb{R}^n$ represent the class labels with $y_i \in \{-1, +1\}$, and $S \in \{\mathbf{s}_1, \dots, \mathbf{s}_k\}$ represent the support vectors with $\mathbf{s}_i \in \mathbb{R}^d$.

The linear SVM is employed to obtain the support vectors (S) from the training data (X) as follows:

$$S = \left\{ (\mathbf{s}_i, y_i) \;\middle|\; (\mathbf{s}_i, y_i) \in X, \quad 1 \leq i \leq k \right\} \tag{3.1}$$

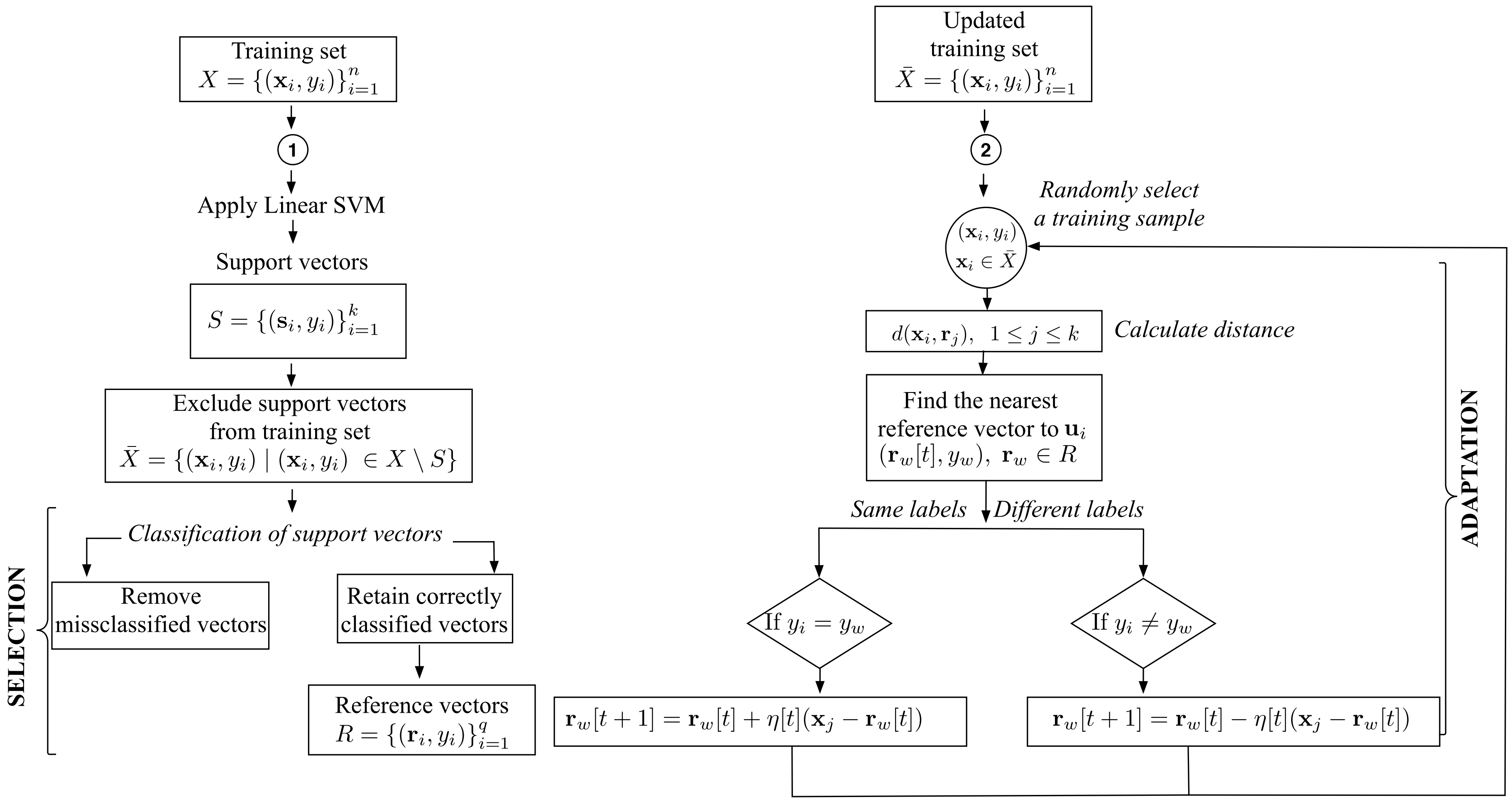

Training set
$X = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$
1
Apply Linear SVM
Support vectors
$S = \{(\mathbf{s}_i, y_i)\}_{i=1}^{k}$
Exclude support vectors
from training set
$\bar{X} = \{(\mathbf{x}_i, y_i) \mid (\mathbf{x}_i, y_i) \in X \setminus S\}$
Classification of support vectors
Remove
missclassified vectors
Retain correctly
classified vectors
Reference vectors
$R = \{(\mathbf{r}_i, y_i)\}_{i=1}^{q}$
SELECTION
Updated
training set
$\bar{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$
2
Randomly select
a training sample
$(\mathbf{x}_i, y_i)$
$\mathbf{x}_i \in \bar{X}$
$d(\mathbf{x}_i, \mathbf{r}_j), \ 1 \le j \le k$
Calculate distance
Find the nearest
reference vector to $\mathbf{u}_i$
$(\mathbf{r}_w[t], y_w), \ \mathbf{r}_w \in R$
Same labels
Different labels
If $y_i = y_w$
If $y_i \neq y_w$
$\mathbf{r}_w[t+1] = \mathbf{r}_w[t] + \eta[t](\mathbf{x}_j - \mathbf{r}_w[t])$
$\mathbf{r}_w[t+1] = \mathbf{r}_w[t] - \eta[t](\mathbf{x}_j - \mathbf{r}_w[t])$
ADAPTATION

Figure 3.1: The flowchart of the SVSA algorithm during learning. 1 and 2 show selection and adaptation process of the algorithm, respectively.

where $y_i \in \{-1,+1\}$ is the class label of the i^{th} support vector. The number of support vectors is data dependent. If the data is linearly separable, the number of support vectors is typically 20% of the training data. If not, the number of support vectors is about 40% of the training data.

The training dataset (X) is next updated to exclude the support vectors as they are used in the selection stage in order to choose some support vectors having the most contribution to classification accuracy.

$$\bar{X} = \left\{(\mathbf{x}_i, y_i) \mid (\mathbf{x}_i, y_i) \in X \backslash S, \quad 1 \leq i \leq n-k\right\} \qquad \textbf{(3.2)}$$

The exclusion of support vectors from the training set is based on the observation that the classification accuracy is increased by excluding them in the experiments. Moreover, since the size of the training data is decreased, the computation time in the selection stage is also decreased.

In the selection stage, the support vectors in the set S are first classified by using the updated training dataset, $\bar{X}$, with the KNN algorithm. The leave-one-out algorithm is used to determine the size of the neighbourhood, K, for KNN classification. The result is given by

$$y_j^p = \left\{y_l \mid l = \arg\,\min_j \left\{\|\mathbf{s}_j - \mathbf{x}_i\|\right\}, \mathbf{s}_j \in S, \mathbf{x}_i \in \bar{X}\right\} \qquad \textbf{(3.3)}$$

where y_j^p is the predicted class label of the j^{th} support vector. If the original and the predicted label of a support vector are different, then this support vector is excluded from the set of support vectors.

The remaining support vectors are called reference vectors and constitute the set R:

$$R = \left\{(\mathbf{r}_j, y_j) \mid (\mathbf{r}_j, y_j) \in S \quad \text{and} \quad y_j^p = y_j\right\}, \quad 1 \leq j \leq k \qquad \textbf{(3.4)}$$

3.2 Support Vector Adaptation

The reference vectors are iteratively adapted based on the training data in a way to make them more representative for classification of data by the nearest neighbour rule. The main logic of adaptation is that a reference vector causing a wrong decision should be further away from the current training vector. Similarly, the nearest reference vector with the correct decision should be closer to the current training vector.

Adaptation is achieved by the Learning Vector Quantization (LVQ) algorithm. The main idea of LVQ is to adapt the reference vectors so that they more optimally represent the classes, thereby, diminishing misclassification errors. These reference vectors result from an update procedure based on the training dataset. The learning procedure consists of iteration over the training data and updating the reference vectors in order to describe optimal class boundaries.

It is assumed that $\mathbf{r}_w$ is the nearest reference vector with y_w class label to randomly chosen vector, $\mathbf{x}_i \in \bar{X}$ with label y_i. The adaptation is applied as follows:

$$\mathbf{r}_w[t+1] = \begin{cases} \mathbf{r}_w[t] - \eta[t](\mathbf{x}_i - \mathbf{r}_w[t]) & \text{if } y_w \neq y_i \\ \mathbf{r}_w[t] + \eta[t](\mathbf{x}_i - \mathbf{r}_w[t]) & \text{if } y_w = y_i \end{cases} \tag{3.5}$$

It means that if the class label of the winner reference vector $\mathbf{r}_w$ matches the class label of randomly selected the training sample $\mathbf{x}_i$, then the reference vector is moved towards $\mathbf{x}_i$. Otherwise, it is moved away from the $\mathbf{x}_i$, where $0 \leq \eta[t] \leq 1$ is the corresponding learning rate parameter given by

$$\eta[t] = \eta_0 \left(1 - \frac{t}{T}\right) \tag{3.6}$$

where η_0 is the initial value of η, t is the current number of iteration, and T is the maximum number of iterations during learning. In this step, LVQ1 algorithm, which is one of the improved algorithms of LVQ, was used [67]. In LVQ1, single set of best matching prototypes is selected and moved closer or further away from each data vector per iteration according to whether the classification decision is correct or wrong, respectively.

3.3 Classification with Reference Vectors

The adaptation is an iterative process resulting in the reference vectors to be used for classification of the test data by the nearest neighbour rule using adaptive nearest neighbour rule [68]. In the classification of test data, unseen samples are classified by using 1NN with the finalized reference vectors.

To define the locally adaptive distance between an unseen samples and a reference vector, a largest sphere is first constructed centered on the reference vector that excludes all the other reference vectors from other classes. The locally adaptive distance between unseen data, $\bar{\mathbf{x}}_i$, and the reference vector, $\bar{\mathbf{x}}_j$ is defined as

$$d_{\text{new}}(\bar{\mathbf{x}}_i, \mathbf{x}_j) = \frac{d(\bar{\mathbf{x}}_i, \mathbf{x}_j)}{r_j} \tag{3.7}$$

where $d(\bar{\mathbf{x}}_i, \mathbf{x}_j)$ is the Euclidean distance between $\bar{\mathbf{x}}_i$ and $\mathbf{x}_j$, and r_j is the radius of the largest sphere for the $j^{th.}$ reference vector, respectively.

Since the SVSA represents the feature space by using a small number of reference vectors, the nearest reference vectors typically have different class labels, causing $K > 1$ to yield worse results [28]. In all the experiments conducted, the highest classification accuracy by the SVSA was obtained with $K = 1$.

4. EXPERIMENTAL RESULTS

In order to test and to compare the proposed algorithm with other classifiers, experiments were conducted on both synthetic and different types of remote sensing data.

In synthetic data experiments, synthetic data with two or more classes and with two features were created by using a Matlab Toolbox for Pattern Recognition [71, 72]. For binary classification, six different degrees of synthetic data having different types of nonlinearity were generated to analyse the proposed algorithm. For multiclass problem, a dataset with 8 classes was generated. The data with more features obtained from UCI machine learning repository and Statlog repository were also used in classification.

In remote sensing experiments, three different types of satellite data were used in classification. In the first experiment, it was of interest to see how well the SVSA method works in classification of multisource data. For this purpose, the Colorado dataset consisting of ten classes and seven features was used for classification. In the second experiment, a multispectral image with 10 m. resolution taken after the earthquake in Adapazarı in Turkey was used to classify buildings as damaged and undamaged. In the third experiment, a hyperspectral image was classified by using the proposed algorithm in order to check the effectiveness of the proposed method in a high dimensional space.

The SVSA algorithm was implemented in Matlab interface with the Library SVM (LIBSVM) for finding the support vectors and LVQPAK software created with C programming language for the adaptation part of the algorithm [73, 67].

4.1 Assessing and Comparing Classification Algorithms

A comparison with the conventional SVMs was provided in terms of classification accuracy. LSVM, NSVM with radial basis function (RBF) kernel, and NSVM with polynomial function kernel were used in comparisons. The SVMs were computed in Matlab software using the LIBSVM.

The KNN classifier was also considered in our experiments since it represents a reference classification method in pattern recognition. In the experiments, KNN was performed with both $K = 1$ and $K = 5$. As it is known, 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. $K = 1$ is most sensitive for noisy samples.

The classification performance was assessed based on the following criteria:

1. An overall accuracy (OA)[1] which is the number of correctly classified samples divided by the total number of samples used in the classification.

2. An average accuracy which represents the average of class classification accuracy.

3. A kappa coefficient[2] of agreement which is the percentage of agreement corrected by the amount of agreement that could be expected due to chance alone [74].

These criteria used to compare classification results of the SVSA to other methods were computed by using the confusion matrix[3] [75].

In most of the experiments, especially when the test data was not given, the accuracy assessment was conducted by 10-fold cross validation[4] i.e. training on 90% of the data and testing on the remaining 10% for 10 times. For classification accuracy comparison, average accuracy percentage from 10-fold cross validation is used. In other words, with 10-fold cross validation, 10 number of datasets were randomly generated to do

[1] See Appendix C for detailed information.

[2] See Appendix D for detailed information.

[3] See Appendix E for detailed information.

[4] See Appendix F for detailed information.

averaging of the results to minimize the variance of the results and increase certainty of conclusions.

In terms of classification performance and computational cost, scaling of data is also an important preprocessing step for SVM. The main advantage of scaling is to avoid features in larger numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during computations. Because kernel values usually depend on the inner products of feature vectors, e.g. the linear kernel and the polynomial kernel, large feature values might cause numerical problems. Therefore, each feature of a data vector was scaled to the range between -1 and +1 before using all the algorithms. The same method was used to scale both training and testing data [76].

To be able to compare the computation time between the algorithms, all the codes used in the experiments have to be implemented with the same programming language for a fair comparison of computational time. Otherwise, the computation time between different algorithms could lead to a wrong decision. Moreover, the code optimization is also required, which means removing the redundancy from the code, for the time comparison.

Hence, in order to make a fair comparison between the algorithms, either it is necessary to rewrite all the codes in the same language or give the computational complexity of the algorithms. We present below the computational complexities for the SVSA, the SVMs and KNN methods. In addition to computational complexity, we also present the space complexity in order to compare the memory storage requirements used by the methods.

4.1.1 Computational complexity

During training, SVM needs to solve a quadratic programming (QP) problem in order to find a separation hyperplane, which has high computational complexity. In order to speed-up the SVM, some decomposition methods faster than QP are introduced in the literature [77, 78]. In this way, the large QP problem is broken into a series of smaller QP problems, and a fast solution of QP is obtained.

The computational complexity of the SVM can change depending on the linear or nonlinear kernel used during training. Therefore, the complexity degree of NSVM is higher than the linear SVM due to the use of kernel function. The computational complexity of the linear SVM is $O(n^2)$, where n is the number training data. In NSVM, the computational complexity is $O(n^3)$ because of computing the kernel function values [79].

The computational complexity of the SVSA during training is analyzed for selection and adaptation parts of the algorithm step by step as follows:

1. $O(n^2)$ for obtaining support vectors by LSVM.

2. $O(n^2 logn)$ for selection of support vectors by KNN [80].

3. $O(nlogn)$ for adaptation part in order to find the nearest reference vector to the training data which is randomly selected.

Step number 3 is repeated T times, which is the maximum number of iterations, so the worst-case computational complexity for this process is $O(Tnlogn)$. Including all the processes, the computational complexity of the SVSA is $O(n^2 logn)$, which is much smaller than the complexity of a NSVM which equals to $O(n^3)$.

During testing, the computational complexities for both linear and nonlinear SVM are $O(n)$, where n corresponds to number of test data. Since the SVSA requires sorting the distances from the reference vectors to an unclassified vector in order to find the nearest reference vector, the computational complexity of the SVSA during testing is $O(nlogn)$. The computational and space complexity for each method is also summarized in Table 4.1.

Therefore, it can be stated that the SVSA takes a longer time than the LSVM in terms of speed performance during training stage because of the selection and the adaptation of support vectors in addition to obtaining them. On the other hand, it requires less time than NSVM since the method does not contain time consuming kernel processes. The advantage of the SVSA method is that the classification performance of the NSVM

Table 4.1: The computational complexity and space complexity for each method during training and testing process.

	Training			
	SVM	**NSVM**	**KNN**	**SVSA**
Comp. Complexity	$O(n^2)$	$O(n^3)$	-	$O(n^2 logn)$
Space Complexity	$O(n)$	$O(n^2)$	-	$O(n)$
	Testing			
Comp. Complexity	$O(n)$	$O(n)$	$O(n^2 logn)$	$O(nlogn)$
Space Complexity	$O(n)$	$O(n)$	$O(n)$	$O(n)$

can be reached with faster calculations during training. During testing, the SVSA takes a bit longer time compared to LSVM but faster than NSVM.

4.2 Choice of Parameters

The parameters of the linear SVM related to the penalty parameter, C, and slack variable, ξ were chosen as the default parameters of the LIBSVM tool since these parameters were not so sensitive to the results obtained with all the experiments. The values of C and ξ were selected as 1 and 1/(number of features), respectively.

For NSVM with RBF kernel function, two parameters have to be estimated while using RBF kernels: kernel parameter γ and penalty parameter C. In order to provide the best parameters for the kernel, 10-fold cross validation was utilized in the model selection with a grid search strategy. The potential combinations of C and γ were tested in a user defined range, where $C = 2^{-5}$ to 2^{15} and $\gamma = 2^{-15}$ to 2^3, and the best combination for C and γ was selected based on the performance of cross validation.

For the parameters of polynomial kernel function, the default parameters of LIBSVM were used in all the experiments; $d = 3$ and $\gamma = 1/$(number of features).

The initial value of learning rate parameter, η_0 for the adaptation part of SVSA was determined by using 10-fold cross-validation as well. The training data was randomly divided into ten groups of equal size. Then, each of the ten groups served as a validation set in turn, and the remaining data of the original training as a new training set. Validations were carried out on these ten fold data to examine the classification accuracies associated with a number of initial values of learning rate parameter, $\eta_0 = 0.1$ to 1.0, in steps of 0.1 [81]. During the experiments conducted, it was observed that the value of learning rate parameter slightly affects the performance of the SVSA. In other words, the learning rate parameter does not have much influence on the decision function obtained by the SVSA.

4.3 Synthetic Data Implementations

In these experiments, synthetic data with 2 and more classes, and 2 features were created. In binary classification, six different types of data with different nonlinearities were created. In the classification of multiclass problem, the data with 2 features and 8 classes were generated. The data from UCI and Statlog databases were also used in the classification to provide the performance of the SVSA compared to other methods.

The performance of the SVSA method was compared to the LSVM, nonlinear SVM with radial basis kernel function (NSVM-1), nonlinear SVM with polynomial kernel function (NSVM-2) and KNN with 1 nearest neighbours (1NN) and 5 nearest neighbours (5NN) in terms of class by class and overall classification accuracy (OA). The standard deviations for each method were also calculated and averaged by 10-fold cross validation.

4.3.1 Binary classification problem

For binary classification, six different types of synthetic data with different types of nonlinearity were generated to analyse the proposed algorithm, and the data types used in the experiments are visualized in Figure 4.1.

The number of samples for each dataset is 4000, and 40% and 60% of each dataset were used as training and test data, respectively. All the algorithms were trained on the training dataset, and tested on the test dataset generated by 10-fold cross validation. The distribution of the reference vectors in the feature space for the banana-shaped data is shown in Figure 4.2.

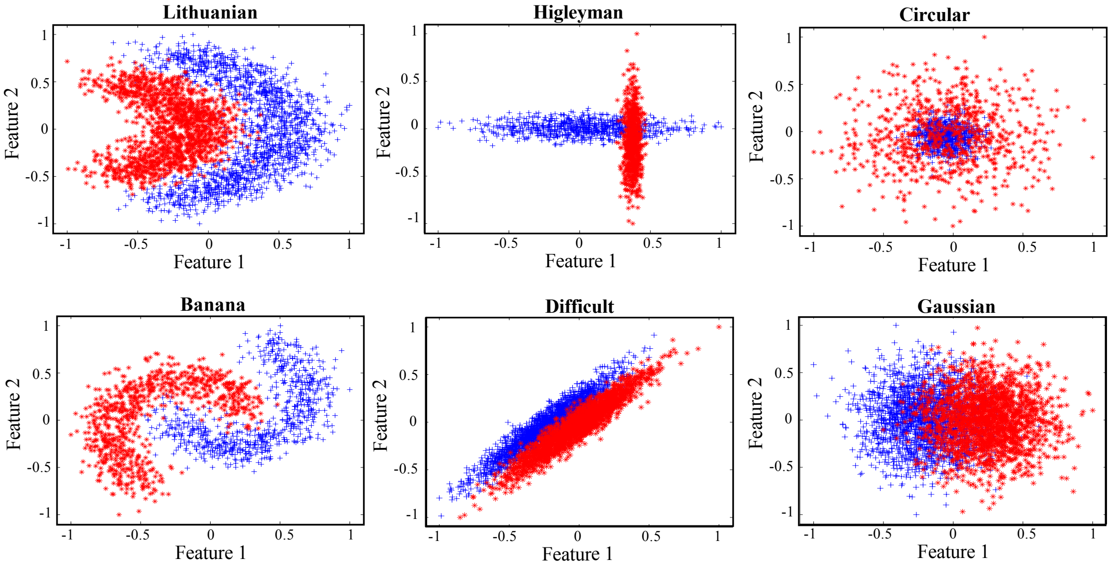


Figure 4.1: Synthetic data types used in binary classification with 2 features.

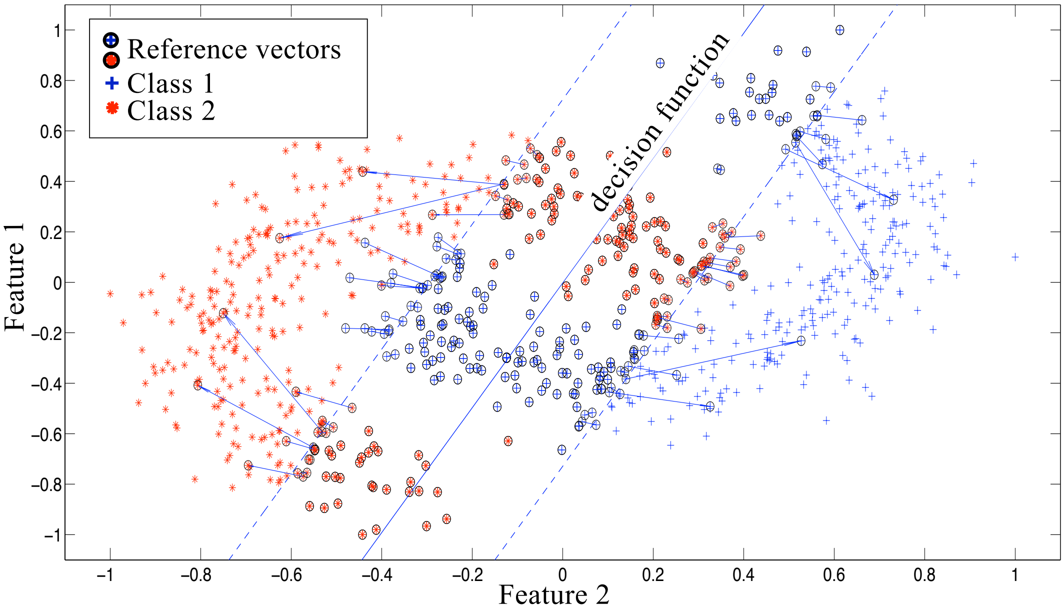


Figure 4.2: The distribution of reference vectors in the feature space for the banana-shaped data.

According to Figure 4.2, the support vectors which are comparatively far away from the decision function obtained by the LSVM are the most adapted vectors by the SVSA. The remaining support vectors keeps their position the same as before the adaptation since there are no training data available around for the support vectors to adapt. All the reference vectors, adapted or not, are used in the classification.

Figure 4.3 shows how selection and adaptation part of the SVSA method affects the classification performance on the difficult, banana and circular type of synthetic data. The results are obtained based on the validation set by 10-fold cross validation.

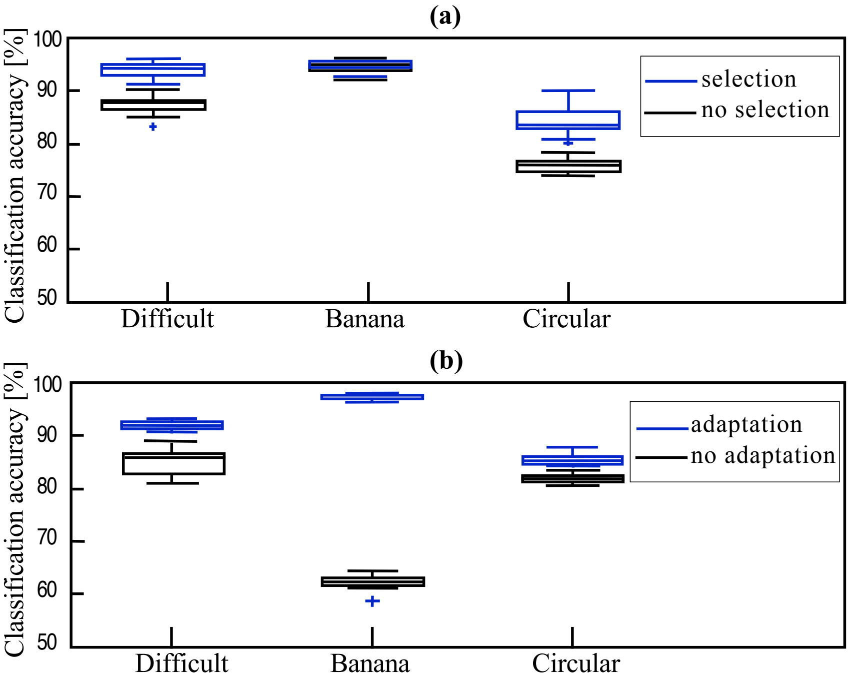


Figure 4.3: (a)The effect of selection stage to the classification performance. (b)The effect of adaptation stage to the classification performance.

In Figure 4.3(a), the selection part of the SVSA was discarded from the algorithm and classification was carried out with the reference vectors obtained without selection stage, depicted with black line. Especially on the difficult and circular type of data, the selection part of SVSA considerably improves the classification accuracy since the class distribution is much more correlated to each other. In other words, when the data is very difficult to classify, the selection stage of SVSA can be observed to be more important in terms of classification performance.

In Figure 4.3(b), the classification accuracies conducted by using only support vectors and adapted support vectors were compared to each other. Especially on the banana-shaped dataset, the adaptation part seems more effective as the banana-shaped data has a highly nonlinear distribution compared to other datasets. In the circular data, the effect of adaptation is minor. Since the classes of circular data are fully overlapped, almost all the training data are selected as the support vectors by the SVSA. So, as there are less number of training data around the support vectors, the adaptation stage of SVSA is not effectively conducted.

The classification accuracies and standard deviations averaged over 10-fold cross validation for all the methods are tabulated in Table 4.2.

Table 4.2: Classification of test accuracies for synthetic data with 2 features and 2 classes. OA and STD refer to overall accuracy and standard deviation, respectively.

DATASETS	**CLASS**	**METHODS**					
		LSVM	**NSVM-1**	**NSVM-2**	**SVSA**	**1NN**	**5NN**
Lithuanian	c_1	79.1	97.0	70.9	96.4	95.8	96.4
	c_2	88.4	97.1	99.7	97.1	96.3	97.0
	OA	83.7	**97.1**	85.3	96.8	96.0	96.7
	STD	1.0	**0.5**	0.6	**0.5**	0.7	0.6
Higleyman	c_1	72.6	87.6	96.0	91.8	93.0	92.2
	c_2	100	99.7	35.0	97.9	93.4	96.5
	OA	86.5	93.7	65.1	**94.9**	93.2	94.4
	STD	1.4	1.2	1.7	**0.7**	1.1	0.8
Circular	c_1	100	94.1	100	93.8	77.1	87.4
	c_2	1.1	74.6	1.1	73.9	76.2	75.3
	OA	51.0	**84.5**	51.1	83.9	76.7	81.4
	STD	1.3	2.2	**0.3**	2.1	2.4	1.6
Banana	c_1	84.7	98.4	77.3	98.4	97.4	98.0
	c_2	85.2	98.0	95.6	98.0	97.6	97.8
	OA	85.0	**98.2**	86.5	**98.2**	97.5	97.9
	STD	1.3	**0.5**	1.4	0.6	0.7	0.6
Difficult	c_1	93.3	93.1	18.0	92.4	90.1	92.4
	c_2	93.8	94.1	99.1	92.8	90.0	92.4
	OA	93.5	**93.6**	59.3	92.6	90.1	92.4
	STD	1.4	**1.4**	1.5	1.2	**1.3**	1.4
Gaussian	c_1	84.7	84.8	96.2	85.1	77.7	82.0
	c_2	83.0	83.0	50.5	82.1	77.6	81.5
	OA	**83.9**	**83.9**	73.5	83.6	77.7	81.7
	STD	1.7	1.7	2.3	**1.6**	2.0	1.9

According to the experimental results provided in Table 4.2, it is observed that the SVSA usually has better classification accuracy than the LSVM. Only for the higleyman, circular, difficult and gaussian datasets, LSVM performs better than SVSA in terms of overall classification accuracy.

For the higleyman dataset, especially for the c_2 class, the performance of LSVM is better than the SVSA. However, there is an unbalanced distribution in the classification accuracies of the higleyman dataset in binary classification. The classification accuracy of c_2 class is quite high while the classification accuracy obtained by the LSVM for c_1 class is low. This is because the optimal hyperplane found by the LSVM is such that all the members of c_2 are correctly classified. For the circular dataset, LSVM and NSVM-2 give the highest classification performance only for the c_1 class compared to all the other methods. Since the classes of circular dataset is highly overlapped, the optimal hyperplane found by the LSVM is somewhere that all the members of the c_1 class is correctly classified while the most members of the c_2 is missclassified. For both higleyman and circular datasets, the SVSA has steady classification performance over the classes compared to LSVM.

For the difficult dataset, the LSVM performs better than the SVSA. This is because the class distribution of this dataset is quite linearly separable compared to other datasets.

The SVSA is also competitive with the NSVM with radial basis function kernel and even better than the NSVM with polynomial function kernel. Besides, it is noted that the NSVM with different types of kernels gives different classification performances, and hence it is required to know which type of kernel is supposed to be chosen before the learning. The standard deviation of SVSA for each dataset shows that the SVSA is a robust method in the classification of these datasets.

In terms of computational time, since the size of training and test data were not so large, the computational time spent during classification of test data for all the methods was approximately less than one second. Besides, the computational time during learning for the SVSA is less than NSVM due to the estimation of optimized kernel function parameters.

In all these experiments, the maximum number of iterations for the SVSA was fixed as 50000, and the initial learning rate parameter for the SVSA was determined as 0.5. The optimized kernel parameters of NSVM-1 for each dataset is tabulated in Table 4.3. The kernel parameters were obtained by using the grid search strategy with 10-fold cross validation in a user defined range, where $C = 2^{-15}$ to 2^{15} and $\gamma = 2^{-15}$ to 2^{3}.

Table 4.3: The optimized kernel parameters of the NSVM-1 for RBF kernel used in the experiments.

Dataset	**C**	γ
Banana	64	8
Lithuanian	0.5	0.5
Gaussian	128	0.5
Difficult	0.125	1
Circular	0.03125	8
Higleyman	8192	0.5

4.3.2 Classification of multiclass dataset

For multiclass classification, a synthetic dataset with 8 classes and 2 features was created from the same data used in binary classification, and is visualized in Figure 4.4. The number of samples used in the classification is 4000. The way of generating training and test data in multiclass classification is the same as in binary classification.

The average classification accuracies of all the methods on the multiclass dataset are listed in Table 4.4.

The standard deviations based on the 10-fold cross validation for all the methods during the classification of the test data can also be visually seen with the box plot illustration in Figure 4.5.

In terms of overall classification performance, the SVSA yields better classification accuracy than LSVM and NSVM with RBF and polynomial kernel functions. The SVSA has also competitive standard deviation compared to the NSVM-1. While the SVSA gives the higher performance based on the average accuracy compared to 1NN method, it has almost the same classification accuracy as 5NN. This result shows that the SVSA is not affected by noisy data although it uses 1NN classification approach with the reference vectors during the classification.

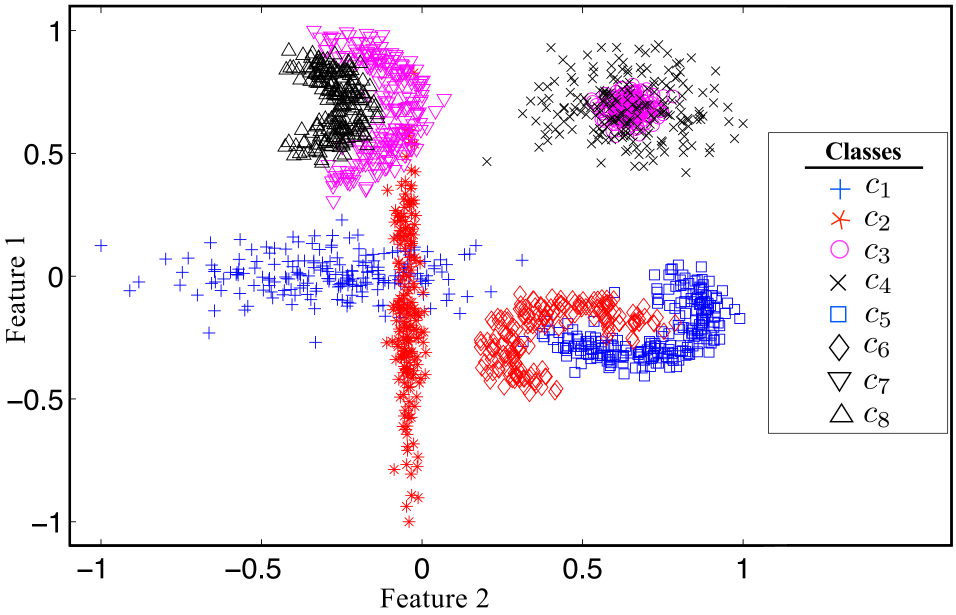


Figure 4.4: Multiclass data with different types of nonlinearities.

Table 4.4: Classification of test accuracies for the multiclass dataset with 2 features and 8 classes. OA and STD refer to overall accuracy and standard deviation, respectively.

CLASS	**METHODS**					
	LSVM	**NSVM-1**	**NSVM-2**	**SVSA**	**1NN**	**5NN**
c_1	69.1	78.4	17.4	88.4	**90**	89.6
c_2	95.3	97	**99.2**	93.6	89.8	93.8
c_3	0	**95.5**	0	87.9	74.6	84.4
c_4	**100**	67.7	97.8	70.9	73.7	73.5
c_5	84.4	95.1	66.8	97.5	97.3	**98**
c_6	83.3	93.3	0	**98.4**	96.9	97.8
c_7	77.3	**98**	63.1	97.3	93.9	96.9
c_8	90.2	97.3	95.6	96.9	96.1	**97.5**
OA	75.7	90.3	55.7	91.5	89.2	**91.6**

We also note that linear SVM classifications is done with respect to the hyperplane whereas SVSA classifications is done with reference vectors using 1NN. Hence, the two methods do not need to give the same accuracy even with a linear classification application.

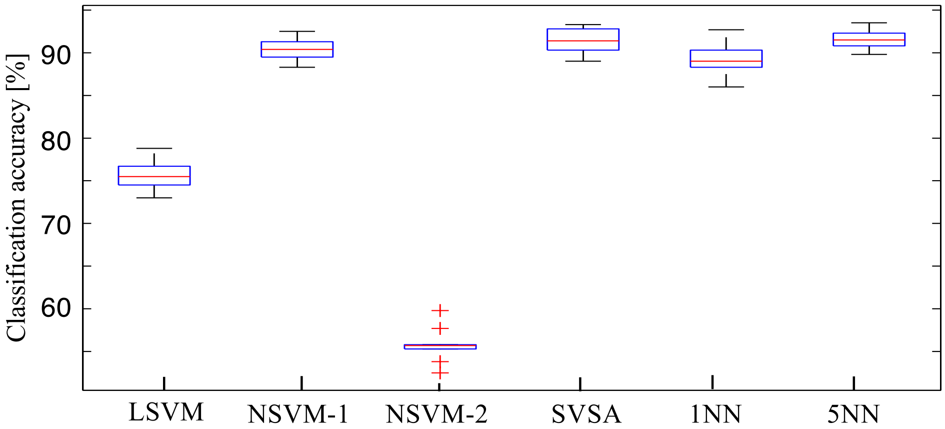


Figure 4.5: The box plot of overall classification accuracies for the multiclass dataset obtained with 10-fold cross validation for each method.

The initial learning rate parameter and the maximum number of iterations for adaptation part of the SVSA were determined as 0.5 and 50.000, respectively. The kernel parameters of NSVM-1, C and γ, were estimated as 64 and 8, respectively.

4.3.3 UCI- and Statlog-repository datasets

The UCI- and Statog-repository databases used in the experiments are of different sizes and features [82]. The dataset used are as follows: sonar, ionosphere, abalone, diabet, heart, satellite, vowel and satellite-2. Table 4.5 lists the main characteristics of the datasets used in the experiments.

Table 4.5: The characteristics of the dataset obtained from UCI- and Statlog-repository.

Datasets	**# Data**	**# Feature**	**# Class**
Sonar	208	60	2
Ionosphere	351	34	2
Abalone	4177	8	3
Pima indians diabet (Medical)	768	8	2
Statlog heart (medical)	270	13	2
Statlog Landsat satellite	6435	36	6
Vowel (connectionist bench)	990	10	11
Statlog Landsat satellite-2	883	36	5

According to Table 4.5, the sonar and the ionosphere datasets have less number of data compared to their number of features. Abalone, which is highly overlapped data, is one of the bigger datasets used in the experiment. The second one is satellite dataset which is also not linearly separable.

40% and 60% of each dataset were used as training and test data, respectively. All results presented were obtained based on overall classification accuracy averaged with 10-fold cross validation. The averaged percentages of correctly classified samples in test data are displayed with the box plot illustration in Table 4.6.

Table 4.6: The averaged percentages of correctly classified objects in test samples with each UCI dataset.

UCI Dataset	**Methods**					
	SVM	**NSVM-1**	**NSVM-2**	**SVSA**	**1NN**	**5NN**
Sonar	75.4	88.0	62.1	**88.5**	87.1	82.7
Ionosphere	87.5	94.3	64.1	91.2	86.4	84.9
Abalone	54.9	54.9	53.8	**55.1**	50.1	53.4
Diabet	76.4	**77.5**	68.4	71.2	71.2	73.0
Heart	**84.8**	81.9	83.0	80.4	75.6	79.6
Satellite	86.9	**92.0**	69.3	91.3	90.4	90.9
Vowel	73.8	97.0	28.2	98.8	**98.9**	94.0
Satellite-2	95.1	**96.0**	71.0	**96.0**	95.1	95.1

According to Table 4.6, the SVSA provides considerably high classification accuracy especially for sonar and ionosphere dataset although these datasets have less number of training data compared to their number of features. For the abalone dataset, the SVSA has the highest classification accuracy compared to other methods. Except for diabetic and heart datasets, which are medical datasets, the SVSA has better classification accuracy than LSVM for the remaining datasets. For the heart dataset, the NSVM-1 also gives less classification accuracy compared to LSVM. Considering the classification result of the satellite dataset, it can also be stated that the SVSA is a successful method when the data is nonlinearly separable. As a conclusion, it is also worth to mention that the classification accuracy of the SVSA is generally very close to NSVM-1's classification accuracy, and better than LSVM.

According to the results obtained by the nearest neighbour method, the classification performance generally varies depending on the number of neighbourhood samples used in KNN. Therefore, in order to obtain better performance by KNN, the number of neighbourhood should be estimated before the classification conducted.

Figure 4.6 also shows the box plot illustration of 10-fold cross validation results for each UCI and Statlog datasets.

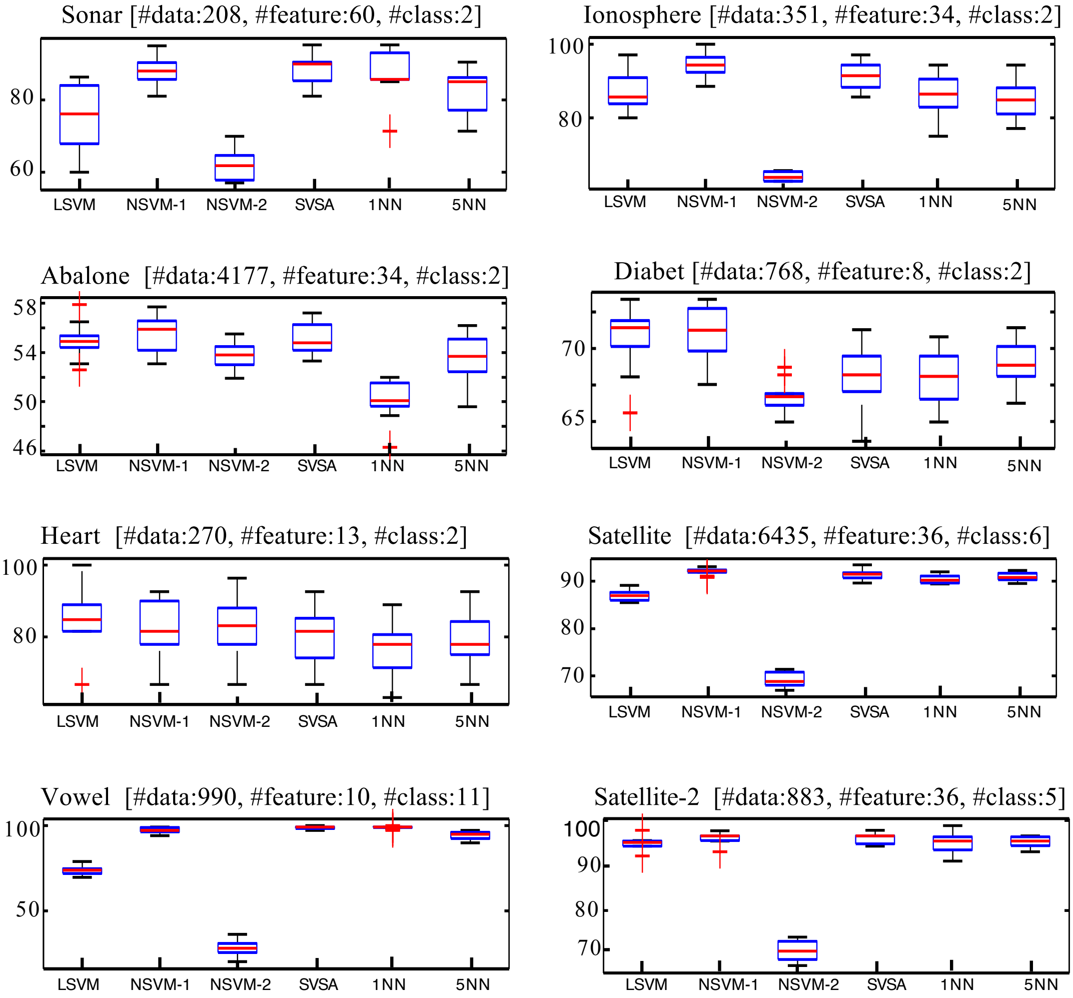


Figure 4.6: The box plot illustration of 10-fold cross validation results for each UCI and Statlog datasets.

The parameters of RBF kernel used by NSVM-1 for each dataset are tabulated in Table 4.7.

Table 4.7: The optimized kernel parameters of RBF kernel function for each dataset in UCI and Statlog datasets.

Dataset	**C**	γ
Sonar	4	0.25
Ionosphere	4	0.25
Abalone	64	2
Diabet	4	0.25
Heart	4	0.25
Satellite	2	2
Vowel	16	1
Satellite-2	8	0.5

4.4 Remote Sensing Implementations

The proposed algorithm was also tested with different types of remote sensing data. Three different remote sensing datasets were used in the experiments; multisource data, multispectral data and hyperspectral data.

4.4.1 Multisource data : Colorado dataset

Classification was performed with the Colorado dataset consisting of the following four data sources [83]:

- Landsat MSS data (four spectral data channels);
- Elevation data (in 10-m contour intervals, one data channel);
- Slope data(0 to 90 in 1 increments, one data channel);
- Aspect data (1 to 180 in 1 increments, one data channel);

Each channel comprises an image of 135 rows and 131 columns, and all channels are spatially co-registered. The area used for classification is a mountainous area in Colorado. Ground reference data are available with ten ground cover classes. One class is water; the others are forest types, as specified in Table 4.8. Therefore, it is very difficult to distinguish among the forest types using the Landsat MSS data alone, since the forest classes show very similar spectral response. For these classes, it clearly helps to add the topographic data sources to the Landsat data in order to make the data more distinguishable. There are ten ground-cover classes listed in Table 4.8.

The Colorado data was classified by the SVSA, LSVM and NSVM with radial basis and polynomial kernel function. The user's and producer's accuracies[5] for each class and overall classification accuracies based on producer's accuracies for all the methods are listed in Table 4.9.

According to the results in Table 4.9, the overall classification performance is generally quite low for all the methods since the Colorado dataset represents a very difficult

[5] See Appendix G for detail information.

Table 4.8: Training and test samples of the Colorado dataset for 10 land use classes with 7 features.

Class	Type of Class	Number of samples Training	 Test
c_1	Water	408	195
c_2	Colorado blue spruce	88	24
c_3	Mountane/ subalpine meadow	45	42
c_4	Aspen	75	65
c_5	Ponderosa pine	105	139
c_6	Ponderosa pine/douglas fir	126	188
c_7	Engelmann spruce	224	70
c_8	Douglas fir/white fir	32	44
c_9	Douglas fir/ponderosa pine/Aspen	25	25
c_{10}	Douglas fir/white fir/aspen	60	39
	Total	**1188**	**831**

classification problem. The overall classification accuracy of the SVSA is better than all the other methods. In addition, it gives higher classification accuracy for many classes individually in comparison to NSVMs. The computational time spent during testing is less than one second for all the methods due to a small number of test data used in the classification.

The initial learning rate parameter and the number of maximum iterations for adaptation part of the SVSA were determined as 0.5 and 40000, respectively. The optimized parameters of nonlinear SVM, C and γ, with radial basis function were determined as 2 and 8, respectively.

4.4.2 Multispectral data : Adapazarı earthquake image

SPOT HRVIR panchromatic images for Adapazarı earthquake were captured with a 10 m. spatial resolution before and after the event on 25 July 1999 and 4 October 1999, respectively (see Figure 4.7). They were geometrically corrected using 26 ground control points from 1:25 000 topographic map of the area [84]. Images were transformed to Universal Transverse Mercator (UTM) coordinates using a first order polynomial transformation and nearest neighbour resampling.

Table 4.9: Test accuracies for classification of Colorado data. PA and UA refer to producer's and user's accuracy, respectively.

Class	**Accuracy**	**METHODS**			
		LSVM	**NSVM-1**	**NSVM-2**	**SVSA**
c_1	*PA*	**100**	94.4	**100**	99.5
	UA	**99.0**	94.2	98.5	**99.0**
c_2	*PA*	37.5	**91.7**	62.5	**91.7**
	UA	11.4	10.7	5.50	**18.0**
c_3	*PA*	4.8	2.40	2.40	**38.1**
	UA	2.80	0.00	0.00	**23.9**
c_4	*PA*	33.9	36.9	20.0	**41.5**
	UA	**45.8**	25.9	23.1	34.4
c_5	*PA*	3.60	1.44	0.00	**20.9**
	UA	21.7	0.00	0.00	**32.3**
c_6	*PA*	59.0	47.3	**61.2**	34.6
	UA	38.3	46.6	42.1	**60.7**
c_7	*PA*	92.9	**100.0**	85.7	94.3
	UA	**89.0**	53.3	52.0	83.1
c_8	*PA*	0.00	0.00	0.00	0.00
	UA	0.00	0.00	0.00	0.00
c_9	*PA*	0.00	0.00	0.00	0.00
	UA	0.00	0.00	0.00	0.00
c_{10}	*PA*	20.5	**69.2**	0.00	64.1
	UA	21.1	0.00	0.00	**39.7**
Overall Accuracy		50.2	50.4	48.0	**53.4**

The training samples for the damage class to be assessed are quite difficult to be visually picked from the post-earthquake image due to its low spatial resolution. Therefore, both pre- and post-earthquake images were used together to be able to conduct the SVSA in damage assessment.

In order to choose the damage samples for the training process, a difference image representing changes occurred during the time between pre- and post-earthquake, was generated by subtracting the matrices of pre- and post-earthquake images. However, some vegetation regions might also be changed during the this time interval, so these

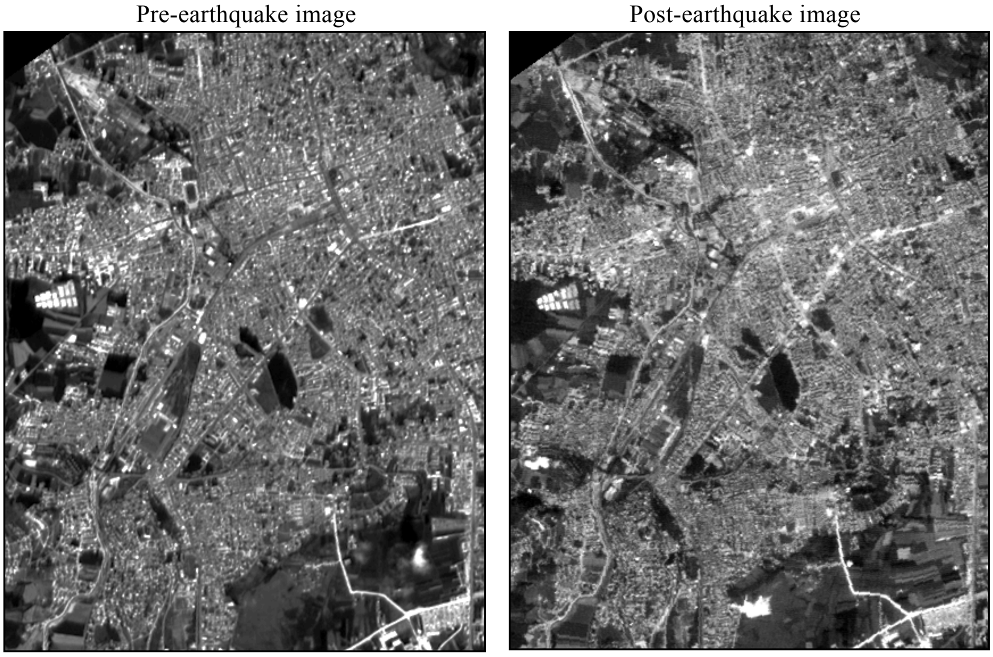


Figure 4.7: Pre- and post-earthquake panchromatic images captured on 25 July 1999 and 4 October 1999, respectively, for a small region in Adapazarı in Turkey. The heavy damage area can visually be identified from the post-earthquake image (brighter intensity values of the urban area compared to pre-earthquake image).

areas can be wrongly interpreted as damage despite not being damage. In order to prevent this confusion, a thematic map of urban area was also created by the SVSA method using the intensity values of pre-earthquake image (Figure 4.8).

The difference image having changes occurred due to both earthquake and vegetation was overlapped with the thematic map of urban area, and the changes occurred due to vegetation were removed from the map of urban area. In other words, only the urban area with earthquake damage was extracted from the difference image. Hence, the training samples for the damage class were selected from this image as it is easier to visually pick the damage samples. The training samples for the urban and vegetation area were chosen from the pre-earthquake image.

After having training data, the post-earthquake image was classified by the SVSA and the thematic map of damage for the area of interest was obtained as in Figure 4.9.

In order to test the classification performance of the SVSA in comparison to other methods in the area of Adapazarı, ten different dataset combinations of the test data for the urban, vegetation and damage class were randomly created, and all the methods

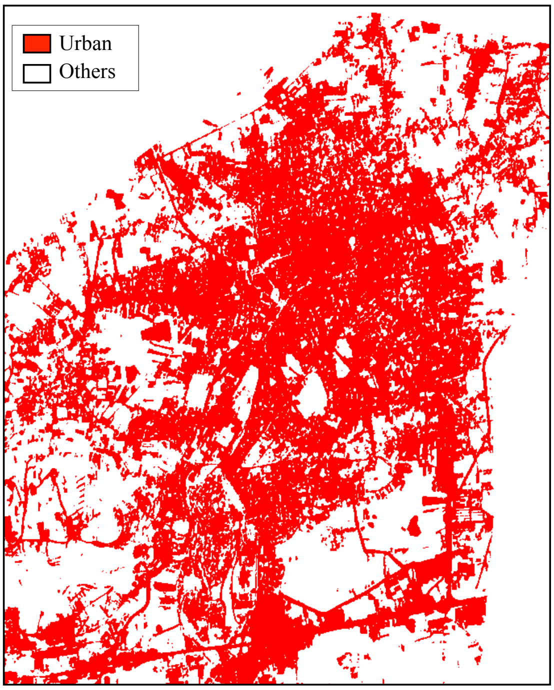


Figure 4.8: The thematic map of the urban area identified by the SVSA from the pre-earthquake image.

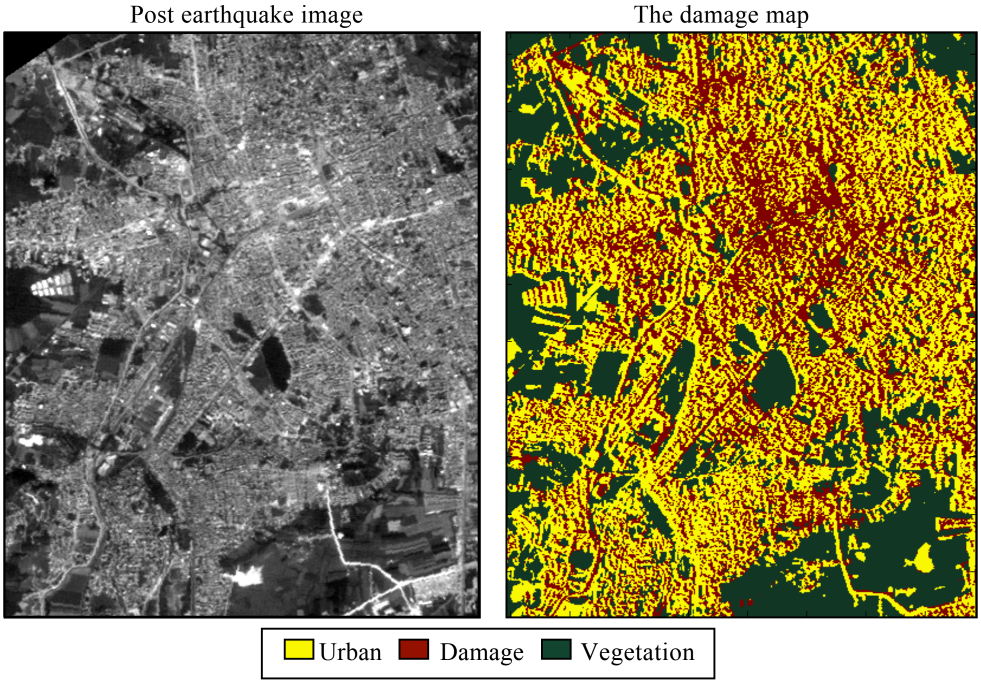


Figure 4.9: The collapsed buildings indicated by the SVSA from the difference map.

were used to classify each dataset individually. The box plot of the classification error for each method with these datasets is illustrated in Figure 4.10.

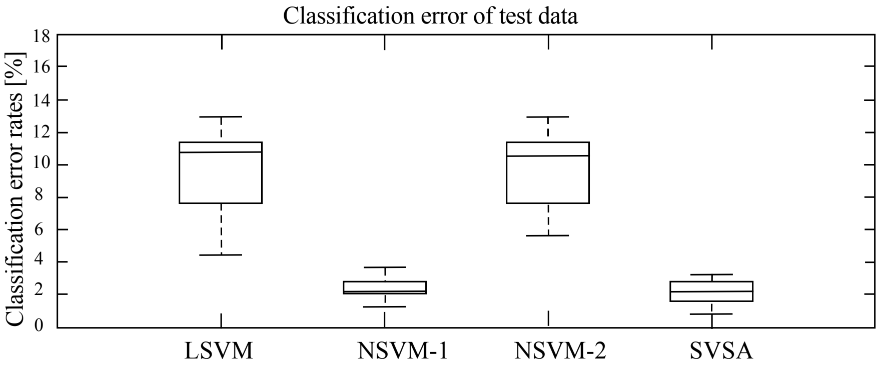


Figure 4.10: The illustration of box plots for the classification error based on ten different datasets for each method.

The SVSA has very low classification error with very small deviations in comparison to linear SVM and nonlinear SVM with polynomial kernel. Additionally, the SVSA method gives competitive classification performance compared to nonlinear SVM with radial basis kernel function. This indication shows once again the importance of kernel type on the classification performance.

4.4.3 Hyperspectral data : Washington DC Mall

In this section, main purpose is to provide how the SVSA performs in the high dimensional space without being affected from curse of dimensionality, also called Hughes phenomenon, which refers to the sample size required for training of a specific classifier grows exponentially with the number of spectral bands [85]. Usually, effective ways to overcome this problem, are to increase sample numbers for training or to reduce the dimensionality of hyperspectral remote sensing data. Since increasing the number of samples for training costs human and material source as well as computational time during training, dimensionally reduction of the training data has been generally preferred in the literature. Therefore, feature selection or feature extraction algorithms have been used to overcome Hughes phenomenon in the literature [86, 87, 88].

A higher dimensional dataset captured by an airborne sensor (HYDICE) over Washington DC Mall was used for exploring the performance of the SVSA in a high dimensional feature space [89]. The original dataset consists of 220 spectral bands across $0.4 - 2.5$ μm., where low signal-to-noise ratio bands were discarded, resulting in a dataset of 191 spectral bands.

The DC dataset is a challenging one to analyse since the classes are complex. There is a large diversity in the materials used in constructing rooftops, and consequently, no single spectral response is representative of the class roofs [90].

A small segment of Washington, DC Mall dataset with 279 × 305 pixels was selected for evaluating the classification performance of the proposed method as in Figure 4.11.

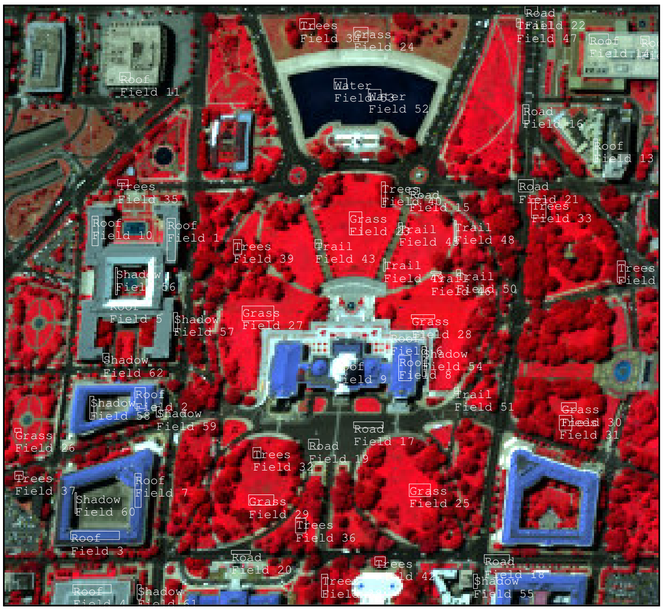


Figure 4.11: Training and test samples for 7 Classes selected in hyperspectral image of Washington DC Mall.

The training and test samples shown in Table 4.10 were collected from a reference dataset which was supplied with the original data. The target land use classes were seven classes used in previous studies with the DC Mall dataset: roofs, road, trail, grass, tree, water and shadow [91].

The proposed method, SVSA, and the SVMs were used in classification of the hyperspectral data. The user's and producer's classification accuracy of the test data for each method are tabulated in Table 4.11.

According to Table 4.11, the SVSA has the highest overall classification accuracy compared to the SVMs. Especially for the class trail, which is quite hard to distinguish

Table 4.10: The number of training and test samples with 191 features and 7 classes.

CLASS	NUMBER OF SAMPLES		
	Training	**Test**	**Total**
Roof	419	273	692
Road	132	109	241
Grass	181	215	396
Trees	226	145	371
Trail	37	28	65
Water	30	30	60
Shadow	73	68	141
Total	**1098**	**868**	**1966**

Table 4.11: Classification accuracies of test data [%]. UA and PA refer to user's accuracy and producer's accuracy, respectively.

Class	Accuracy	Methods			
		LSVM	**NSVM-1**	**NSVM-2**	**SVSA**
Roof	*UA*	**99.6**	91.2	**99.6**	98.5
	PA	94.8	97.3	94.8	**97.5**
Road	*UA*	**100**	99.08	**100**	99.1
	PA	**100**	**100**	99.1	**100**
Grass	*UA*	**100**	99.5	**100**	**100**
	PA	98.6	99.1	**99.5**	**99.5**
Trees	*UA*	**100**	**100**	**100**	**100**
	PA	98.0	98.6	98.6	**99.3**
Trails	*UA*	35.7	67.9	42.9	**71.4**
	PA	90.9	43.2	**100**	76.9
Water	*UA*	**100**	**100**	**100**	**100**
	PA	**100**	**100**	**100**	**100**
Shadow	*UA*	95.6	**98.5**	97.1	**98.5**
	PA	**100**	**100**	**100**	**100**
	Overall accuracy	97.5	95.9	97.8	**98.4**
	Time (s)	15	21	40	**6**

due to the small number of training data, the classification accuracy of the SVSA has the highest performance in terms of user's accuracy as well.

In the SVSA, the classification of the hyperspectral image was carried out with the 45 reference vectors tabulated in Table 4.12. These results suggest that the proposed algorithm is also highly competitive in the classification of hyperspectral data, and does not seem to be affected by the curse of dimensionality [92].

Table 4.12: The number of reference vectors used in the classification.

Type of classes							Total
Roof	**Road**	**Grass**	**Tree**	**Trail**	**Water**	**Shadow**	
4	5	9	6	8	4	9	**45**

Since the number of reference vectors obtained by the SVSA is 45, the computational time spent by the SVSA during testing is less than all the SVMs.

The thematic map generated by the SVSA is shown in Figure 4.12(b).

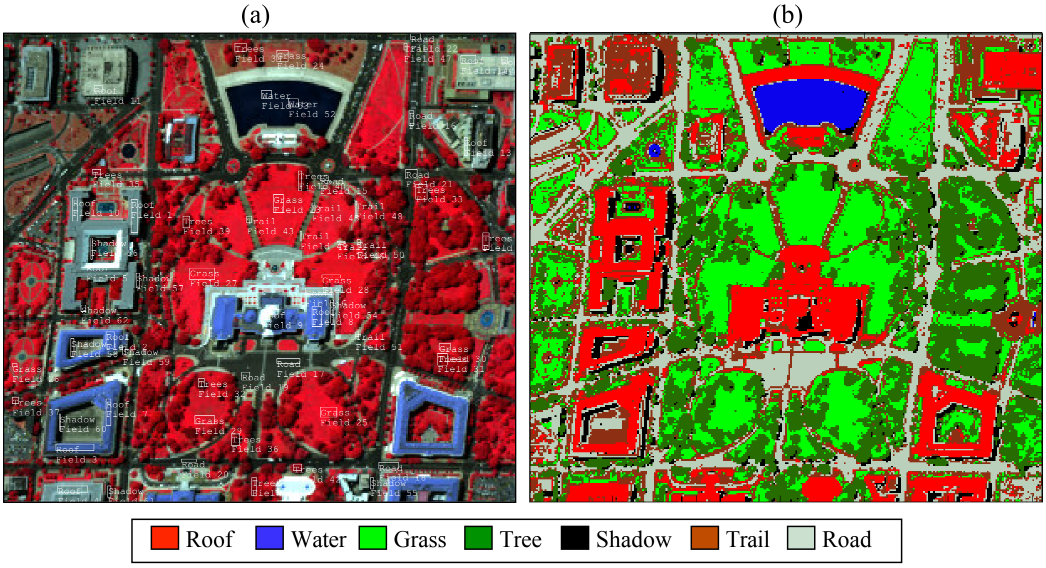


Figure 4.12: (a)Hyperspectral image of Washington DC Mall. (b)Thematic map of DC Mall image obtained by SVSA.

According to Figure 4.12(b), the land use classes were considerably identified by the SVSA with quite low salt-and-pepper classification noise.

The initial learning rate parameter and the number of maximum iterations for the adaptation part of the SVSA algorithm were determined as 0.1 and 40000, respectively. The parameters of nonlinear SVM, C and γ, were determined as 32 and 0.25, respectively.

5. EARTHQUAKE DAMAGE ASSESSMENT

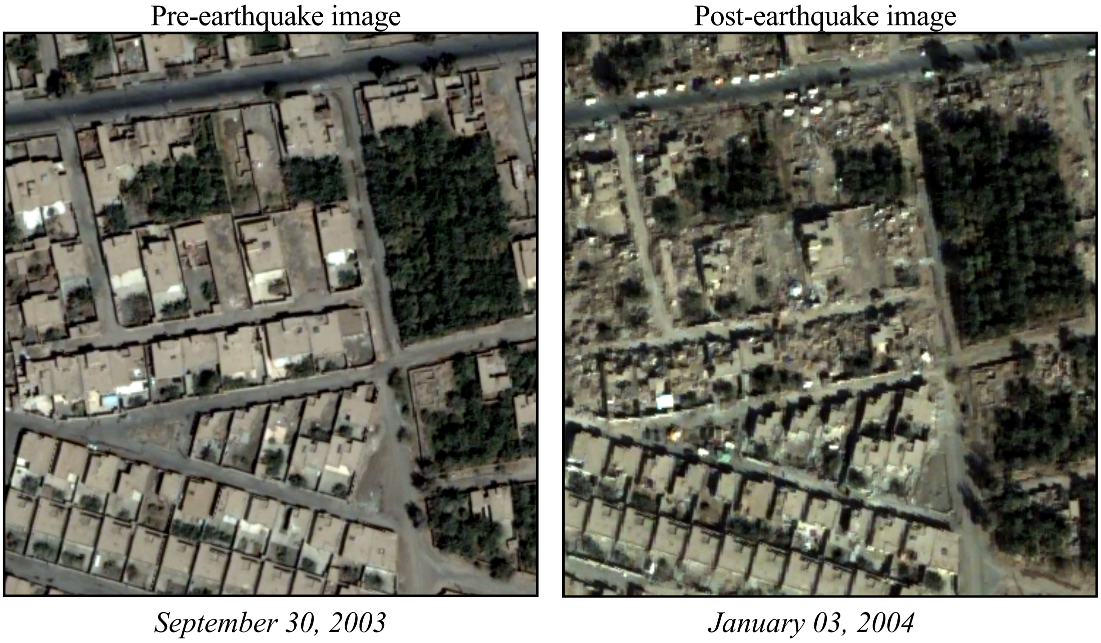
In this chapter, a pansharpened post-earthquake image obtained from city of Bam was used to assess earthquake damage by the proposed algorithm, SVSA. Two different type of experiments were conducted on this image which are pixel-based and hybrid pixel- and texture-based classification.

In pixel-based classification, only the pansharpened image was used with 4 spectral bands, and the effectiveness of the proposed methodology was compared to SVMs and KNN method. In terms of damage assessment, since using of texture information obtained from high resolution image is important, texture information, also called spatial information, were extracted by Gray Level Co-occurrence matrix (GLCM) method from panchromatic image for the area of interest. Both spectral and spatial information were used in classification. Nonparametric Weighted Feature Extraction method (NWFE) was also used during classification in order to weight each feature with hybrid pixel- and texture-based classification. The results showed that the classification performance obtained by the SVSA was improved compared to its pixel-based classification.

5.1 SVSA with Pixel-based Classification

Quickbird satellite images of the city of Bam, acquired on 30 September 2003 (pre-earthquake) and 03 January 2004 (post-earthquake) were obtained. Only a post-earthquake pansharpened image having four spectral bands, which are RGB and near-infrared (NIR), was used to distinguish damage patterns.

All the algorithms were evaluated using a small area of the city, approximately 5 hectares in size. Since there is no ground truth data available regarding to land use classes, pansharpened images with 0.6 m. spatial resolution for pre- and post-earthquake images were obtained and considered as the ground truth data (Figure 5.1).



Pre-earthquake image

Post-earthquake image

September 30, 2003

January 03, 2004

Figure 5.1: Pre- and post-earthquake pansharpened images of area of interest.

In order to obtain the pansharpened images, principle component resolution merge method using a bilinear interpolation resampling technique was used. Since the pansharpened image has high resolution compared to the multispectral image, the pansharpened pre- and post-earthquake data were also used as ground truth data.

For the thematic classification, five classes were identified: Vegetation, shadow, buildings, open ground and damage. Training and test data were visually selected from the pansharpened post-earthquake image after being verified with pansharpened pre- and post-earthquake images. The number of training and test samples used in the classification for each class is tabulated in Table 5.1.

Table 5.1: The number of training and test samples for each class.

Class	**Class number**	**Number of samples** **Training**	 **Test**	 **Total**
Vegetation	1	358	616	974
Shadow	2	276	449	725
Building	3	356	866	1222
Open ground	4	537	480	1017
Damage	5	397	1238	1634
	Total	**1924**	**3549**	**5572**

Additionally, Bhattacharya distance[1] between each pair of classes for each subset of four spectral bands was also calculated in order to measure the separability between the classes. The results are shown in Table 5.2.

Table 5.2: Bhattacharya distances between pairs of classes with respect to different combination of spectral channels. R, G, B and NIR represent red, green and near-infrared channels while 1, 2, 3, 4 and 5 refer to vegetation, shadow, building, open ground and damage classes, respectively. Avg. represents the average value of each row.

Channels		**Class pair**									
	Avg.	**1-2**	**1-3**	**1-4**	**1-5**	**2-3**	**2-4**	**2-5**	**3-4**	**3-5**	**4-5**
R-G-B-NIR	6.3	3.6	15.7	7.3	6.7	18.7	4.6	3.2	1.0	1.1	0.7
G-B-NIR	6.1	3.6	15.3	7.2	6.3	18.7	4.3	2.9	1.0	0.7	0.5
R-B-NIR	6.1	3.4	15.5	7.1	6.6	18.5	4.1	3.1	1.0	1.0	0.6
R-G-NIR	5.6	3.3	14.5	7.0	4.5	18.6	3.6	2.2	0.9	1.0	0.6
R-NIR	4.9	3.3	11.3	6.2	3.5	17.7	3.3	1.7	0.8	0.6	0.5
B-NIR	5.4	3.2	14.2	6.9	4.1	18.6	3.0	1.9	0.9	0.6	0.5
G-NIR	5.5	3.1	15.3	6.8	5.3	18.0	2.6	2.2	0.9	0.6	0.3
NIR	1.5	3.0	0.3	0.2	0.1	8.5	0.8	1.2	0.7	0.5	0.0
R-G-B	4.5	1.9	12.4	2.4	3.3	16.0	3.8	2.7	0.9	1.0	0.6
G-B	4.2	1.8	12.3	1.2	3.2	16.0	3.5	2.5	0.8	0.7	0.4
R-G	3.8	0.8	10.7	2.3	1.0	15.9	3.0	1.9	0.8	1.0	0.3
G	3.5	0.7	10.7	1.2	0.9	15.9	2.2	1.6	0.8	0.6	0.0
R-B	4.0	0.5	12.2	2.1	2.3	14.9	3.5	2.6	0.8	0.9	0.4
R	3.2	0.4	9.8	1.6	0.9	13.7	2.7	1.4	0.7	0.5	0.1
B	3.4	0.3	12.1	1.2	1.2	14.6	1.8	1.7	0.8	0.6	0.0

In Table 5.2, the interclass pair distances for all classes appear in rows placed in descending order based on the average distance in the row. The results suggest that in terms of classification performance, using all the spectral bands is a good choice for classifying the five classes.

In terms of correlation between the class pairs, it can be stated that the most confused classes are the damage versus open ground, damage versus building and building versus open ground class. This is mainly because the rubble of collapsed buildings was scattered along the open grounds, and the most buildings were made of clay. Due to these reasons, the spectral response of the samples belonging to damage, open ground and building classes reflect very similar spectral responses.

[1] See Appendix H for detail information.

In the pixel-based classification, all the methods were first trained on the training samples only by using four spectral bands to identify the damage patterns from the post-earthquake image, and the test data were classified with respect to the learning model generated by the methods. The overall accuracy and producer's accuracy of the test data for each method obtained are tabulated in Table 5.3.

Table 5.3: The overall accuracy and the producer's accuracy of test data for each method. OA refers to overall classification accuracy.

	Classification accuracies [%]					
Methods	**Vegetation**	**Shadow**	**Building**	**Open ground**	**Damage**	**OA**
LSVM	99.5	94.4	**91.8**	62.7	33.5	69.8
NSVM-1	99.5	96.7	88.1	59.6	71.6	81.7
NSVM 2	98.4	94.9	0.0	**100.0**	1.1	41.8
KNN	**99.7**	**97.3**	76.7	61.0	77.6	81.4
SVSA	**99.7**	**97.3**	79.7	58.8	**79.3**	**82.4**

According to Table 5.3, in terms of the overall classification accuracy, the SVSA gives the highest classification accuracy compared to both linear and nonlinear SVM with different kernels and KNN classifier. Since the classes of building, damage and open ground are the most confused classes due to their similarity on spectral response, the classification accuracies of these classes are generally low for all the methods while the vegetation and the shadow classes can be easily separated from each other with higher producer's accuracy. In terms of damage identification, the SVSA gives the highest producer's accuracy as well.

In order to make a detailed analysis of the classification performance for each method, except NSVM-2 as it has the lowest classification accuracy, a confusion matrix for each method on the test data is also provided with the four statistical measures of accuracy derived from the confusion matrix as tabulated in Table 5.4, listing user's accuracy, producer's accuracy, kappa value and the overall accuracy.

Table 5.4: Confusion matrix of test data for each method.

LSVM	**Number of samples**						**User's**
Classification	**Vegetation**	**Shadow**	**Building**	**Open ground**	**Damage**	**# Samples**	**accuracy**
Vegetation	**613**	14	0	0	2	629	**97.5**
Shadow	2	**424**	0	0	16	442	**95.9**
Building	0	0	**795**	60	152	1007	**78.9**
Open ground	0	11	19	**301**	653	984	**30.6**
Damage	1	0	52	119	**415**	587	**70.7**
	616	449	866	480	1238	**# Samples**	
	99.5	**94.4**	**91.8**	**62.7**	**33.5**	**Producer's accuracy**	
					62.3	**Kappa value**	
					69.8	**Overall accuracy**	

NSVM-1	**Number of samples**						**User's**
Classification	**Vegetation**	**Shadow**	**Building**	**Open ground**	**Damage**	**# Samples**	**accuracy**
Vegetation	**613**	10	0	0	6	629	**97.5**
Shadow	2	**434**	0	0	22	458	**94.8**
Building	0	0	**763**	46	88	897	**85.1**
Open ground	0	0	14	**286**	236	536	**53.4**
Damage	1	5	89	148	**886**	1129	**78.5**
	616	449	866	480	1238	**# Samples**	
	99.5	**96.7**	**88.1**	**59.6**	**71.6**	**Producer's accuracy**	
					76.3	**Kappa value**	
					81.7	**Overall accuracy**	

KNN	**Number of samples**						**User's**
Classification	**Vegetation**	**Shadow**	**Building**	**Open ground**	**Damage**	**# Samples**	**accuracy**
Vegetation	**614**	10	0	0	6	630	**97.5**
Shadow	2	**437**	0	0	38	477	**91.6**
Building	0	0	**664**	32	90	786	**84.5**
Open ground	0	0	31	**293**	143	467	**62.7**
Damage	0	2	171	155	**961**	1289	**74.6**
	616	449	866	480	1238	**# Samples**	
	99.7	**97.3**	**76.7**	**61.0**	**77.6**	**Producer's accuracy**	
					75.7	**Kappa value**	
					81.4	**Overall accuracy**	

SVSA	**Number of samples**						**User's**
Classification	**Vegetation**	**Shadow**	**Building**	**Open ground**	**Damage**	**# Samples**	**accuracy**
Vegetation	**614**	10	0	0	11	635	**96.7**
Shadow	2	**437**	0	1	38	478	**91.4**
Building	0	0	**690**	37	97	824	**83.7**
Open ground	0	0	25	**282**	110	417	**67.6**
Damage	0	2	151	160	**982**	1295	**75.8**
	616	449	866	480	1238	**# Samples**	
	99.7	**97.3**	**79.7**	**58.8**	**79.3**	**Producer's accuracy**	
					76.9	**Kappa value**	
					82.4	**Overall accuracy**	

According to Table 5.4, the SVSA has the highest Kappa value, 76.9%, which represents a moderate agreement, while LSVM has the lowest one, 62.3%. Since LSVM is a linear classifier, it is not effective to separate the classes which are much correlated because of their similar spectral response. In the NSVM-1 classification, due to the nonlinear decision function used in the classification, the accuracies are improved compared to LSVM, so the Kappa value is also increased from 62.3% to 76.3%.

Not only the producer's accuracy, the user's accuracy should also be taken into account in assessment of classification performance. Figure 5.2 shows the user's and producer's accuracies for each class with respect to the methods compared to the SVSA classifier.

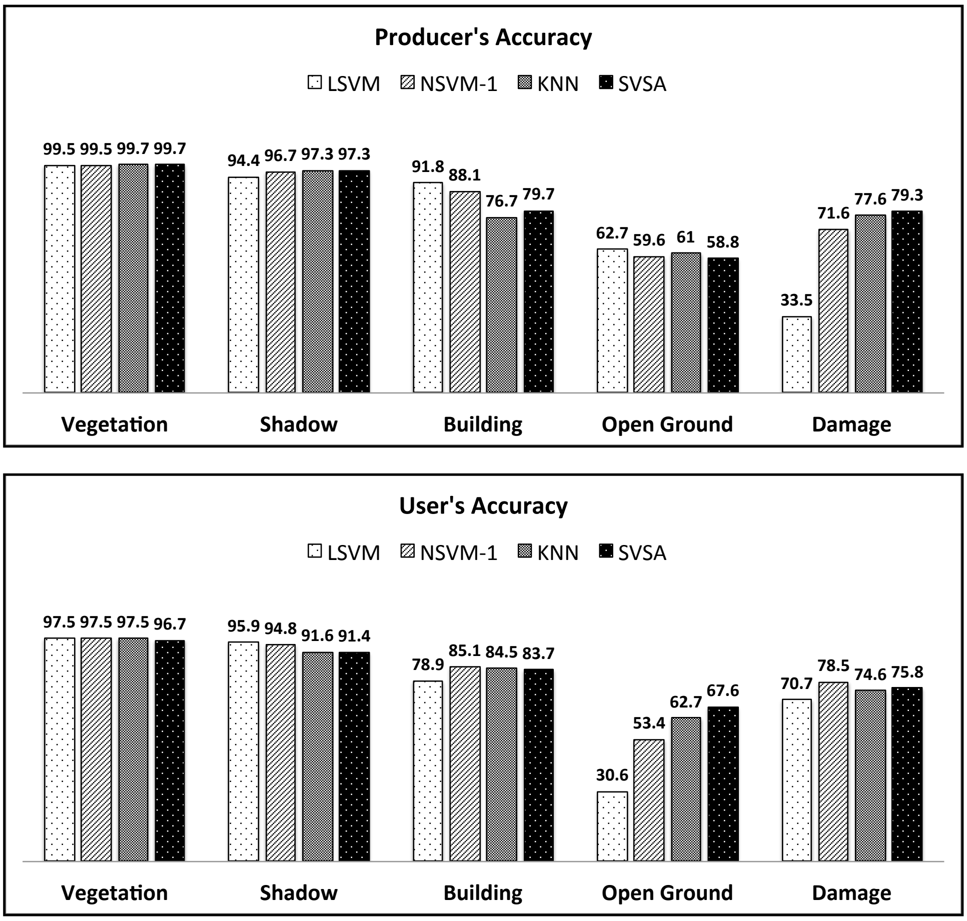


Figure 5.2: User's and producer's accuracies of each class for all the methods.

Especially for the damage class, the SVSA has the highest producer's accuracy, and thus the SVSA can estimate 79.3% of damage pixels as correctly classified. Although the damage class identified by the SVSA shows the highest producer's accuracy, only 75.8% of the area labelled damage is actually covered by the damage. In other words, 24.2% of pixels classified by the SVSA as damage actually belong to other classes.

The labels of damage class predicted by the SVSA and NSVM-1 were also compared to each other to show the performance of each method for the classification of damage class (Figure 5.3).

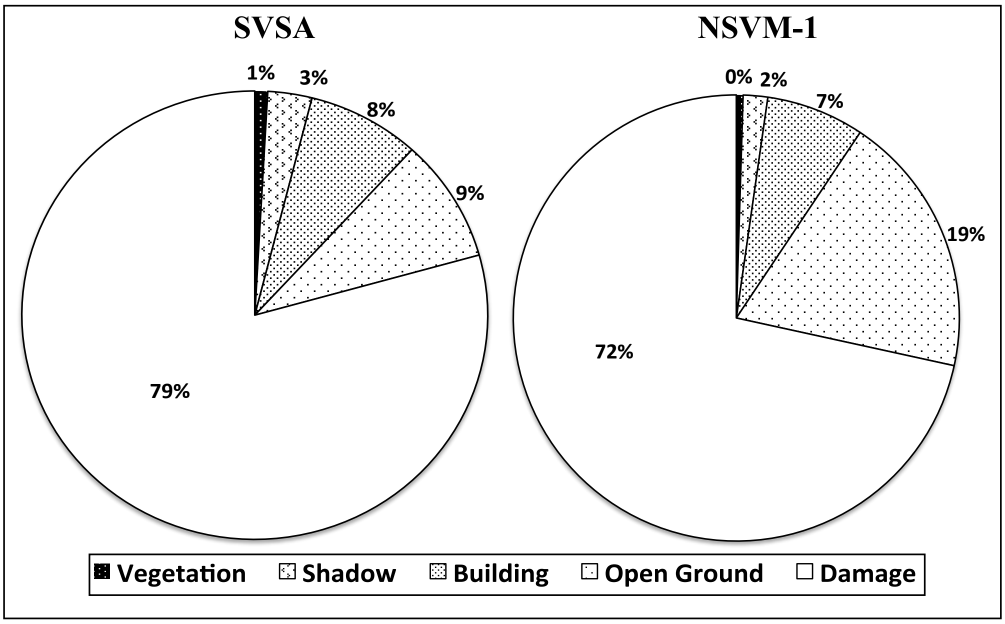


Figure 5.3: The approximate rates of the damage class labels predicted by SVSA and NSVM-1.

Figure 5.3 shows the rates of other classes confused with the damage class for both SVSA and NSVM-1. According to Figure 5.3, the most confused classes with the damage class predicted by the SVSA are the open ground and building class. 9% and 8% of the damage samples were classified as open ground and building by the SVSA. In NSVM-1, 19% of the damage samples were classified as open ground which is worse for NSVM-1 compared to SVSA.

Figures 5.4(a) and (b) show the comparison of user's and producer's accuracies for the NSVM-1 and the SVSA. In Figure 5.4(c), the differences between the producer's and user's accuracy, achieved by the NSVM-1 and the SVSA, are shown.

According to Figure 5.4(c), especially for the open ground and the damage class, the SVSA is significantly better than NSVM-1, while the NSVM-1 has better accuracy

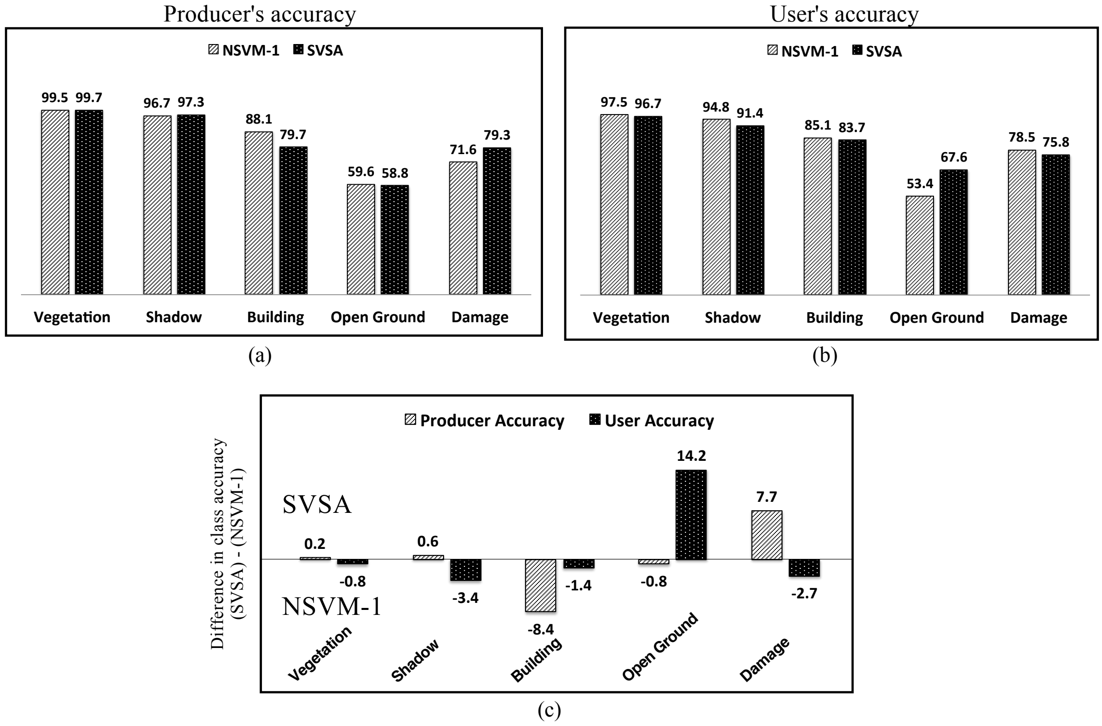


Figure 5.4: (a)Comparison of producer’s accuracy. (b)Comparison of user’s accuracy. (c)Difference between user’s and producer’s accuracy achieved by SVSA and NSVM-1. The positive differences show that the SVSA has better accuracy than NSVM-1 while negative differences show that the NSVM-1 has better accuracy than the SVSA.

for the building class. For the shadow class, NSVM-1 performs well compared to the SVSA in terms of the user’s accuracy. Looking at the Figure 5.4(c), it can also be stated that the SVSA achieves slightly better class accuracies than the NSVM-1.

The post-earthquake image was classified by the SVSA, and the thematic map was obtained as in Figure 5.5.

According to Figure 5.5, the land use classes are generally identified well by the SVSA, but there are some salt-and-pepper type of noise in the thematic map.

As a result of the pixel-based classification, it can be stated that the confusion between classes can not be removed only by the spectral responses. Therefore, the textural information of the classes should also be taken into account in order to reduce the confusion between classes and obtain better classification performance.

For Iran, Bam earthquake damage assessment, the confusion matrix reveals that the most confused classes are damage and open ground. The main reason of this confusion

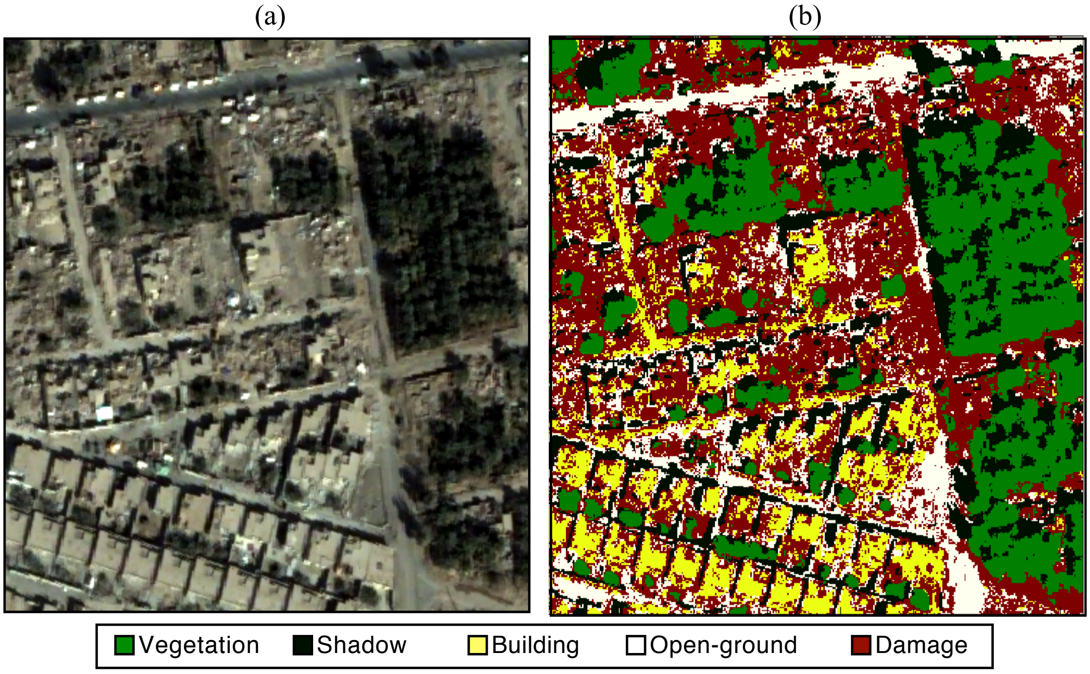


Figure 5.5: (a)Post-earthquake image. (b)Thematic map obtained by the SVSA with pixel-based classification.

is that the classification carried out by the SVSA is a pixel-based classification. It means that any object can be misclassified as another object in the image if their spectral responses are similar enough. Therefore, in terms of higher classification accuracy, more features are needed to be used in addition to the spectral bands in order to increase the separability between the classes.

5.2 SVSA with Hybrid Pixel- and Texture-based Classification

In classification of remote sensing data of urban areas, a satellite image with high spatial resolution is necessary to identify relatively small objects, i.e., houses and narrow streets. The most commonly available remote sensing data of high spatial resolution is the single band panchromatic data. However, for accurate classification, using only a panchromatic image is not sufficient for discriminating detailed class types. To overcome this problem, the spatial properties, texture features, obtained from the panchromatic image should be used in classification. If the spatial content of the image is not used, the resulting thematic map sometimes looks noisy, with salt-and-pepper classification noise.

There are several studies on extraction of spatial information and for use in classification in order to improve the classification accuracy [93, 94, 95]. Fauvel et al.'s paper (2008) was based on the fusion of the morphological information and the original hyperspectral data, i.e., the two vectors of attributes were concatenated into one feature vector [96]. After a reduction of dimensionality, the final classification was achieved by using a support vector machine classifier. The algorithm was tested on some remote sensing data from urban areas, and signification improvements were achieved in terms of classification accuracy.

Thus, in order to improve the classification performance, spatial information between the values of neighbouring pixels in the remote sensing data should additionally be considered besides the spectral information during the classification process. Spectral-and-spatial classification aims at assigning each image pixel to one class using a feature vector based on its own spectral value (the spectral information) and information extracted from its neighbourhood (the spatial information). Several studies have shown that using both spectral and spatial information improves classification accuracy and visual quality of thematic maps [97].

For extraction of the spatial information from the panchromatic image, the second order Haralick features obtained by Gray Level Co-occurrence matrix were used. Eight texture features calculated using the GLCM method were evaluated; homogeneity, contrast, dissimilarity, mean, standard deviation, entropy, angular second moment and correlation.

With the hybrid pixel- and texture-based classification, the number of features increases during the classification. Therefore, different feature subsets should be taken into account with respect to their contribution to the classification accuracy. In order to assign different weight to each feature, nonparametric feature extraction method was also used before the classification process. Figure 5.6 shows the steps followed in the proposed hybrid pixel- and texture-based classification with the SVSA.

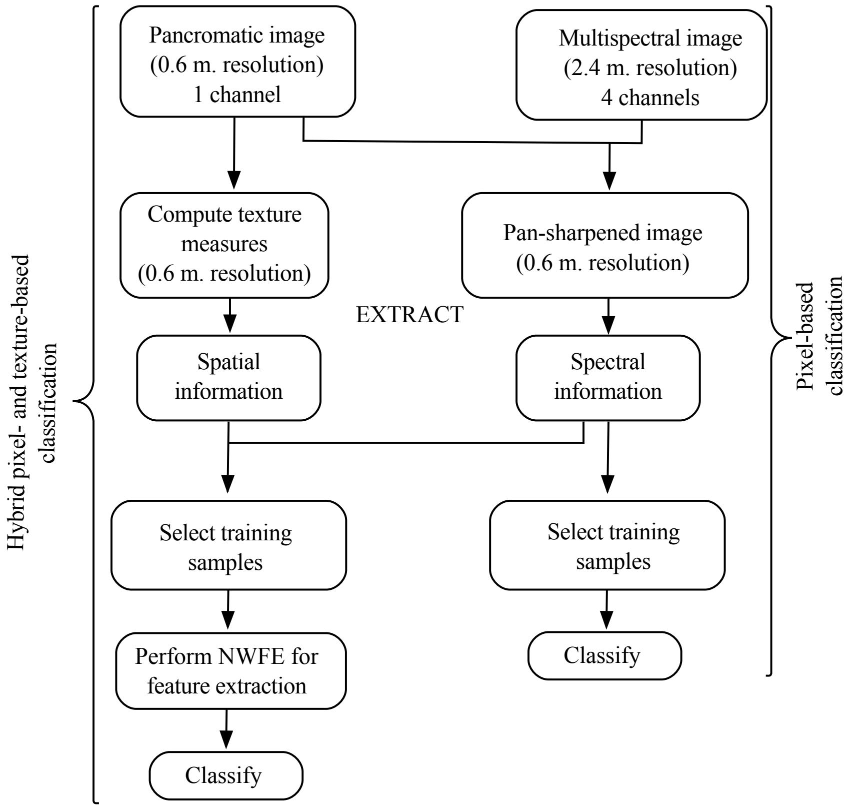


Figure 5.6: The flow chart for both pixel-based and hybrid pixel- and texture-based classification.

5.2.1 Gray Level Co-occurrence Matrix

In the scientific literature, one of the well-known and widely used technique is the Gray Level Co-occurrence Matrix (GLCM), which is a way of extracting second order statistical textural features [98, 99]. GLCM is a matrix derived from the gray-level image, which shows the joint probability of distribution of a pair of gray levels, separated at a certain distance and a certain orientation. Among many methods, the GLCM method was found to be more effective [100].

Consider an image with N_x row and N_y columns. Let N_g corresponds to the number of gray levels in the image. Therefore, the image in two dimensional spatial domain, can be indexed with $L_x = \{1, 2, \cdots, N_x\}$ and $L_y = \{1, 2, \cdots, N_y\}$ corresponding to horizontal and vertical spatial domains, respectively. The texture context information is specified by the matrix of relative frequencies $\mathbf{P}_{d,\theta}(i,j)$ as follows:

$$\begin{aligned}\mathbf{P}_{d,\theta}(i,j) = \#\{(x_1,y_1)(x_2,y_2) \mid \; & f(x_1,y_1) = i, f(x_2,y_2) = j, \\ & \|(x_1,y_1) - (x_2,y_2)\|^2 = d, \\ & \angle((x_1,y_1),(x_2,y_2)) = \theta\} \end{aligned} \quad \textbf{(5.1)}$$

where the intensity level of a pixel pair changes from i to j, the location of the first pixel is (x_1,y_1) and that of the second pixel is (x_2,y_2), d is the Euclidean distance between the pixel pair, θ is the angle between the two pixels, and # denotes the number of elements in the set. $p_{d,\theta}(i,j)$ is the $(i,j)^{th}$ element of the normalized co-occurence matrix.

The gray level co-occurence matrix (GLCM) is calculated as a symmetric matrix, and normalized as follows:

$$\mathbf{P}_{d,\theta}(i,j)^{\mathrm{S}} = \frac{\mathbf{P}_{d,\theta}(i,j) + \mathbf{P}_{d,\theta}(i,j)^{\mathrm{T}}}{2} \quad \textbf{(5.2)}$$

$$\mathbf{P}_{d,\theta}(i,j)^{\mathrm{N}} = \frac{\mathbf{P}_{d,\theta}(i,j) + \mathbf{P}_{d,\theta}(i,j)^{\mathrm{T}}}{\sum_{i=1}^{N_g}\sum_{i=1}^{N_g}\mathbf{P}_{d,\theta}(i,j)} \quad \textbf{(5.3)}$$

where $\mathbf{P}_{d,\theta}(i,j)^{\mathrm{S}}$ and $\mathbf{P}_{d,\theta}(i,j)^{\mathrm{N}}$ are the symmetric and normalized GLCM matrices, respectively.

5.2.2 Texture features from GLCM

In order to calculate texture features for each pixel in an image, a moving window is usually used to define the neighbourhood of a pixel, and the texture measurement calculated using the window is assigned to the center pixel (Figure 5.7).

A number of texture features, called Haralick's features, can be extracted using GLCM [101]. The Haralick's features used in the implementation are as follows:

1. **Contrast** measures local spatial frequency. If the GLCM has large off-diagonal elements, the local window has high contrast.

$$\text{Contrast} = \sum_{i}^{N_g}\sum_{j}^{N_g}(i-j)^2 p_{d,\theta}(i,j) \quad \textbf{(5.4)}$$

2. **Correlation** is a measure of gray level linear dependence between the pixels at the specified positions relative to each other. It shows how the reference pixel is related

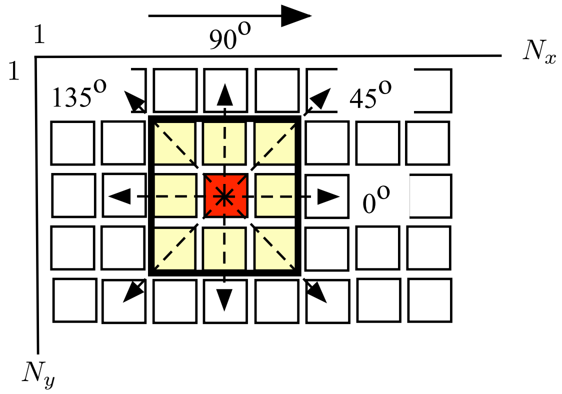


Figure 5.7: A representation of sliding window in an image with 3 × 3 window size and at different angles. The center pixel coloured with red corresponds to texture measurement calculated within the window.

to its neighbour.

$$\text{Correlation} = \sum_{i}^{N_g}\sum_{j}^{N_g} \frac{(i-\mu_x)(j-\mu_y)p_{d,\theta}(i,j)}{\sigma_x\sigma_y} \qquad \textbf{(5.5)}$$

where μ_x and σ_x are the mean and the standard deviation of the row sums of the co-ocurrence matrix; μ_y and σ_y are the mean and the standard deviation of the column sums of the co-ocurrence matrix.

3. **Dissimilarity** is similar to contrast. The high contrast of the local window indicates high dissimilarity value.

$$\text{Dissimilarity} = \sum_{i}^{N_g}\sum_{j}^{N_g} \mid i-j \mid p_{d,\theta}(i,j) \qquad \textbf{(5.6)}$$

4. **Entropy** measures disorder in the image. The higher entropy corresponds to a greater contrast from one pixel to the its neighbours.

$$\text{Entropy} = \sum_{i}^{N_g}\sum_{j}^{N_g} \left(\frac{p_{d,\theta}(i,j)}{\log p_{d,\theta}(i,j)}\right) \qquad \textbf{(5.7)}$$

5. **Homogeneity** or inverse difference moment measures the similarity of the pixels, and results in a larger value if the elements of the GLCM are concentrated on the main diagonal.

$$\text{Homogeneity} = \sum_{i}^{N_g}\sum_{j}^{N_g} \left(\frac{p_{d,\theta}(i,j)}{1+\mid i-j \mid}\right) \qquad \textbf{(5.8)}$$

6. **Mean** defines the average level of intensity of the image.

$$\text{Mean} \quad = \quad \sum_{i}^{N_g}\sum_{j}^{N_g} i p_{d,\theta}(i,j) \tag{5.9}$$

7. **Angular Second Moment (ASM)** also known as uniformity or energy. It measures local uniformity. ASM is higher when pixels are very similar.

$$\text{Angular second moment} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p_{d,\theta}(i,j)^2 \tag{5.10}$$

8. **Variance** puts relatively higher weights on the elements that differ from the average value of $P_{i,\theta}(i,j)$.

$$\text{Variance} \quad = \quad \sum_{i}^{N_g}\sum_{j}^{N_g} p_{d,\theta}(i,j)(i-\mu_x)^2 \tag{5.11}$$

5.2.3 Nonparametric Weighted Feature Extraction

Nonparametric Weighted Feature Extraction (NWFE) is based on discriminant analysis by focusing on samples near the eventual decision boundary [102]. The main idea of the NWFE is as follows: 1) assigning different weights on every sample to compute the local means and 2) defining nonparametric between-class and within-class scatter matrices. Many experiments showed the effectiveness of the approach for the classification of remote sensing data [103, 104, 105].

The NWFE defines new non-parametric between-class and within-class scatter matrices to get more features by assigning different weights on every sample of training set. The nonparametric between-class scatter matrix and within-scatter matrix are defined as

$$\mathbf{S_b}^{\mathrm{NW}} \quad = \quad \sum_{i=1}^{L} P_i \sum_{\substack{j=1 \\ j\neq i}}^{L} \sum_{k=1}^{n_i} \frac{\lambda_k^{(i,j)}}{n_i} \left(x_k^{(i)} - M_j(x_k^{(i)})\right)\left(x_k^{(i)} - M_j(x_k^{(i)})\right)^{\mathrm{T}} \tag{5.12}$$

$$\mathbf{S_w}^{\mathrm{NW}} \quad = \quad \sum_{i=1}^{L} P_i \sum_{k=1}^{n_i} \frac{\lambda_k^{(i,j)}}{n_i} \left(x_k^{(i)} - M_j(x_k^{(i)})\right)\left(x_k^{(i)} - M_j(x_k^{(i)})\right)^{\mathrm{T}} \tag{5.13}$$

where $x_k^{(i)}$ refers to kth sample from class i, n_i is training sample size of class i, and P_i denotes the prior probability of class i. The scatter matrix weight $\lambda_k^{(i,j)}$ is defined as:

$$\lambda_k^{(i,j)} = \frac{\operatorname{dist}\left(x_k^{(i)}, M_j(x_k^{(i)})\right)^{-1}}{\sum_{i=1}^{n_i} \operatorname{dist}\left(x_l^{(i)}, M_j(x_l^{(i)}\right)^{-1}} \tag{5.14}$$

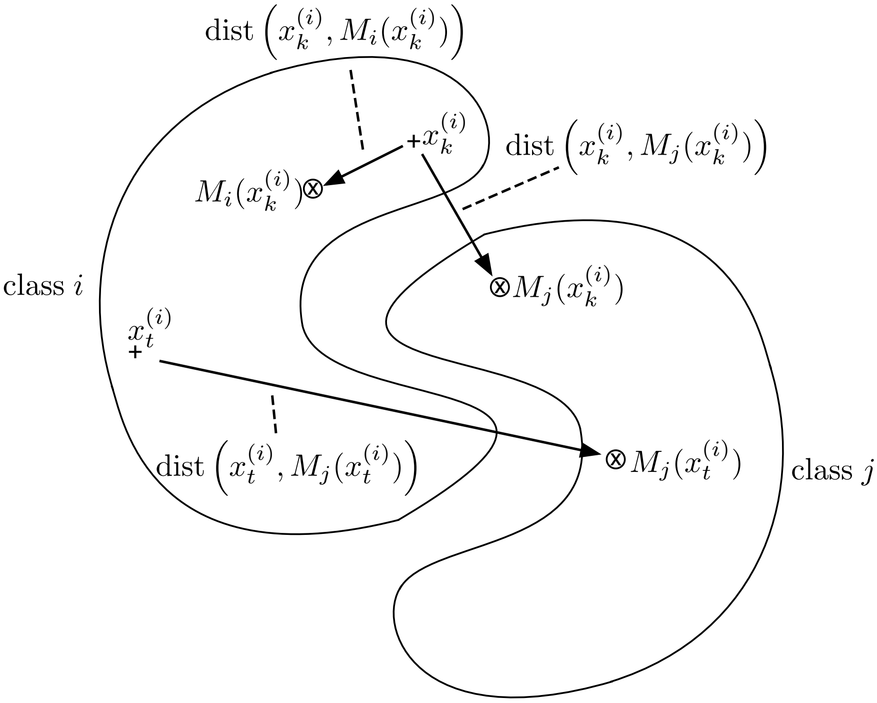
where dist(a,b) refers to Euclidean distance from a and b, and $M_j(x_k^{(i)})$ is the local mean of $x_k^{(i)}$ in the class j. It is defined as:

$$M_j(x_k^{(i)}) = \sum_{l=1}^{n_j} w_{kl}^{(i,j)} x_l^{(j)} \tag{5.15}$$

where $w_{kl}^{(i,j)}$ is as follows:

$$w_{kl}^{(i,j)} = \frac{\text{dist}\left(x_k^{(i)}, x_l^{(j)}\right)^{-1}}{\sum_{l=1}^{n_i} \text{dist}\left(x_k^{(i)}, x_l^{(j)}\right)^{-1}} \tag{5.16}$$

Figure 5.8 is the visualization of the parameters used in NWFE to compute between-class and within-class scatter matrices. It shows the importance of using boundary points and local means.



$\text{dist}\left(x_k^{(i)}, M_i(x_k^{(i)})\right)$
$+x_k^{(i)}$
$\text{dist}\left(x_k^{(i)}, M_j(x_k^{(i)})\right)$
$M_i(x_k^{(i)})\otimes$
class i
$\otimes M_j(x_k^{(i)})$
$x_t^{(i)}$ +
$\text{dist}\left(x_t^{(i)}, M_j(x_t^{(i)})\right)$
$\otimes M_j(x_t^{(i)})$
class j

Figure 5.8: The relationship between sample points and their local means in the feature space. The $+$ symbols are neighbours of $x_k^{(i)}$, and $\otimes$ represents local means.

The goal of NWFE is to find a linear transformation $\mathbf{A} \in \mathrm{R}^{d \times p}$, $p \leq d$, which maximizes the between-class scatter and minimizes the within-class scatter. The columns of $\mathbf{A}$ are the optimal features obtained by optimizing the following criterion:

$$\mathbf{A} = \arg\max_{\mathbf{A}} \text{tr}\left((\mathbf{A}^{\mathrm{T}}\mathbf{S_w}^{\mathrm{NW}}\mathbf{A})^{-1}\mathbf{A}\mathbf{S_b}^{\mathrm{NW}}\mathbf{A}\right) \tag{5.17}$$

This maximization is equivalent to finding eigen-pairs $(\lambda_i, v_i)_{i=1}^n$, $\lambda_1 \geq \lambda_2, \cdots, \geq \lambda_d$ for the generalized eigenvalue problem:

$$\mathbf{S_b}^{\mathrm{NW}}\mathbf{v} = \lambda \mathbf{S_w}^{\mathrm{NW}}\mathbf{v} \tag{5.18}$$

To reduce the effect of cross-products of within-class distances and prevent the singularity, some regularization techniques can be applied to within-class scatter matrix. In NWFE, within class scatter matrix is regularized by

$$\mathbf{S_w}^{\mathrm{NW}} = 0.5\mathbf{S_w}^{\mathrm{NW}} + 0.5\mathrm{diag}(\mathbf{S_w}^{\mathrm{NW}}) \tag{5.19}$$

where $\mathrm{diag}(\mathbf{S_w}^{\mathrm{NW}})$ means diagonal members of $\mathbf{S_w}^{\mathrm{NW}}$.

Solving Equation **5.18**, the eigenvalues and eigenvectors are obtained as follows:

$$\mathbf{Q} = [\mathbf{v_1}, \cdots, \mathbf{v_n}], \qquad \Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \tag{5.20}$$

Where $\mathbf{Q}$ and Λ represent the orthogonal matrix and the eigenvalue matrix, respectively.

The data in the transformed feature space is obtained via the following equation:

$$\bar{\mathbf{x}} = \Lambda^{1/2}\mathbf{Q}\mathbf{x} \tag{5.21}$$

where $\bar{\mathbf{x}}$ and $\mathbf{x}$ correspond to the data in the transformed feature space and the data in the original feature space, respectively. The equation **5.21** is also equivalent to the following equation:

$$d_{\mathbf{xy}} = (\mathbf{x} - \mathbf{y})^{\mathrm{T}}\Sigma(\mathbf{x} - \mathbf{y}) \tag{5.22}$$

where $d_{\mathbf{xy}}$ is the distance by weighted Σ between $\mathbf{x}$ and $\mathbf{y}$ data, and $\Sigma = \mathbf{Q}^{\mathrm{T}}\Lambda\mathbf{Q}$.

Equation **5.22** can also be written by substituting Σ as follows:

$$\begin{aligned}
(\mathbf{x} - \mathbf{y})^{\mathrm{T}}\Sigma(\mathbf{x} - \mathbf{y}) &= (\mathbf{x} - \mathbf{y})^{\mathrm{T}}(\mathbf{Q}^{\mathrm{T}}\Lambda\mathbf{Q})(\mathbf{x} - \mathbf{y}) \\
&= (\mathbf{x} - \mathbf{y})^{\mathrm{T}}(\Lambda^{1/2}\mathbf{Q})^{\mathrm{T}}(\Lambda^{1/2}\mathbf{Q})(\mathbf{x} - \mathbf{y}) \\
&= \Big((\Lambda^{1/2}\mathbf{Q})(\mathbf{x} - \mathbf{y})\Big)^{\mathrm{T}}\Big((\Lambda^{1/2}\mathbf{Q})(\mathbf{x} - \mathbf{y})\Big) \\
&= \Big(\Lambda^{1/2}\mathbf{Q}\mathbf{x} - \Lambda^{1/2}\mathbf{Q}\mathbf{y}\Big)^{\mathrm{T}}\Big(\Lambda^{1/2}\mathbf{Q}\mathbf{x} - \Lambda^{1/2}\mathbf{Q}\mathbf{y}\Big)
\end{aligned} \tag{5.23}$$

Thus, by using Equation **5.23**, $d_{\mathbf{xy}}$ can be rewritten as below:

$$d_{\mathbf{xy}} = (\bar{\mathbf{x}} - \bar{\mathbf{y}})^{\mathrm{T}}(\bar{\mathbf{x}} - \bar{\mathbf{y}}) \tag{5.24}$$

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ refer to the data in the transformed feature space as in Equation **5.25**.

$$\bar{\mathbf{x}} = \Lambda^{1/2}\mathbf{Q}\mathbf{x}, \quad \bar{\mathbf{y}} = \Lambda^{1/2}\mathbf{Q}\mathbf{y} \tag{5.25}$$

5.2.4 Earthquake damage assessment with hybrid pixel- and texture-based

In hybrid pixel- and texture-based classification, 8 texture features including homogeneity, contrast, dissimilarity, mean, standard deviation, entropy, angular second moment and correlation were obtained with the same pixel distance (1 pixel) and the horizontal directional invariance by using the GLCM method from the panchromatic image (Figure 5.9).

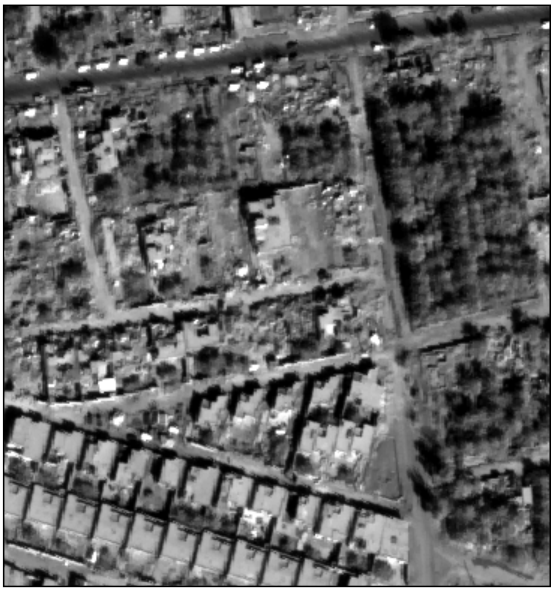


Figure 5.9: The panchromatic image, used for texture extraction, with 0.6 m. spatial resolution and one spectral channel.

The textural features were added to spectral features during the classification. Figure 5.10) shows the textural features with their ID constituted with respect to their row of using during the classification.

In order to calculate the texture features for each pixel in the panchromatic image, a sliding window is usually used to define the neighbourhood of a pixel, and the texture measurement calculated from the window is assigned to the center pixel. Since choice

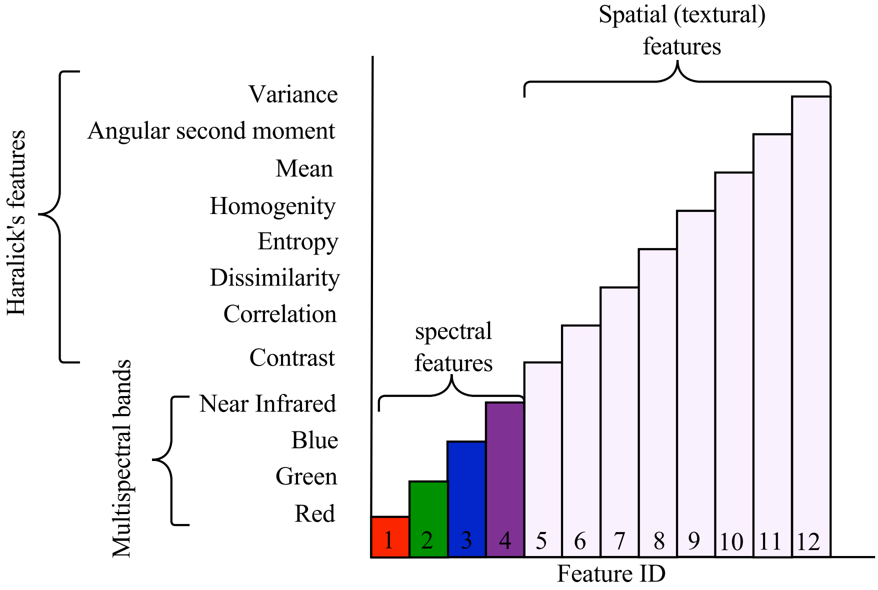


Figure 5.10: The numeric representation of spatial and spectral features used in the hybrid pixel- and texture-based classification. The feature id shows the cumulative features to be used in the classification, i.e., the feature ID 4 and 12 correspond to first four spectral bands and all the features (totally 12 features), respectively.

of the window size affects the classification accuracy, windows with different sizes were used to find the best window size in terms of classification accuracy. As the window size can not be bigger than the size of the object to be identified, only the windows with 3×3, 7×7 and 9×9 sizes were used in the classification of the test data. For each window size, 8 spatial feature images were individually obtained based on GLCM.

In order to use the spatial features in the classification of the test data, the training data were first updated by adding the spatial feature to the current training data used in the pixel-based classification. The training data for the spatial features were extracted from the associated spatial feature image by using the image coordinates of training data used for the pixel-based classification. The test data was also updated with the same process. The training and test data were next transformed by using the transformation matrix obtained by the NWFE method, and transformed training and test data were used in the classification.

Figure 5.11 shows the behaviour of overall classification accuracy of the test data with respect to spatial features added during the classification for each window size.

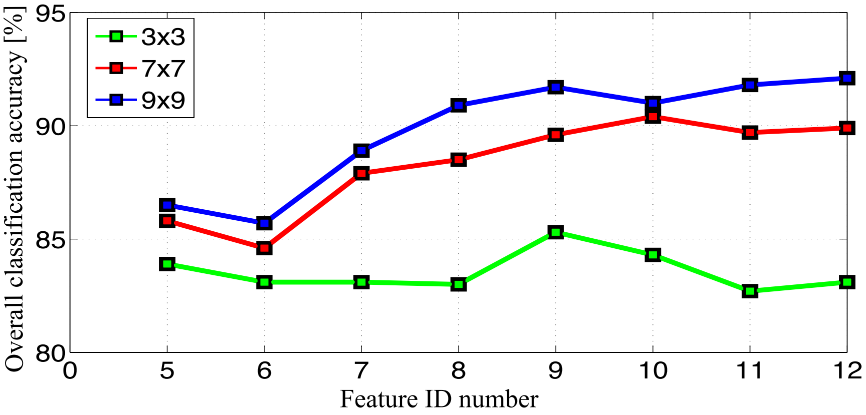


Figure 5.11: Classification accuracies used for number of features used with respect to window size used in GLCM.

According to Figure 5.11, as the number of features used in the classification grows, the overall accuracy for each feature ID obtained by 9 × 9 window size uniformly better compared to other window sizes. Thus, 9 × 9 window size was found to be optimum window size in terms of classification accuracy. The classification accuracy for each class obtained with the 9 × 9 window size is tabulated in Table 5.5.

Table 5.5: The classification accuracies obtained with 9 × 9 window size.

Feature ID Number	**Classification Accuracy [%]**					
	Vegetation	**Shadow**	**Building**	**Open Ground**	**Damage**	**OA**
5	**99.4**	98.9	78.5	**82.9**	82.6	86.5
6	**99.4**	98.0	81.8	78.3	80.1	85.7
7	98.7	98.4	85.3	77.7	87.4	88.9
8	97.6	96.2	91.7	76.0	**90.8**	90.9
9	98.4	98.7	93.6	76.9	90.1	91.7
10	96.9	99.3	93.1	81.9	87.1	91.0
11	97.7	99.1	94.7	81.5	88.2	91.8
12	95.8	**99.8**	**94.8**	79.6	90.4	**92.1**

According to classification performances in Table 5.5, it is worth to state that after the feature with ID 9, there is almost no contribution to the classification performance in terms of overall classification accuracy. In other words, the textures which are the mean, the angular moment and the variance do not have much influence on the classification performance.

The spatial feature images for the area of interest based on Haralick's features obtained with 9 × 9 window size are shown in Figure 5.12.

For classification, features of training and test data were transformed using NWFE. In order to show the performance of the transformed data in terms of classification accuracy, the original features and the features transformed by the NWFE were used in the classification with the SVSA, and their overall classification accuracies are compared in Figure 5.13.

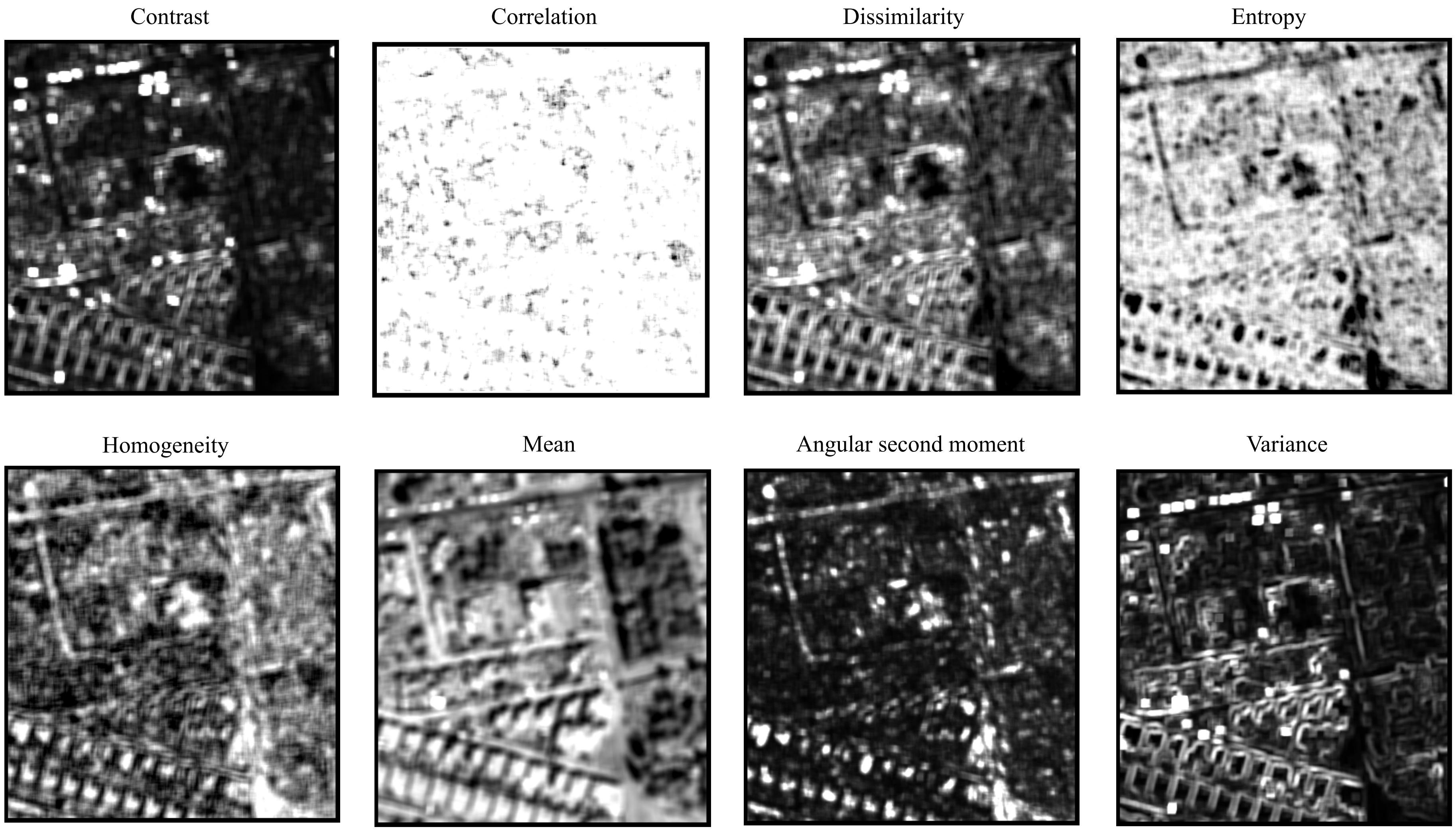
Figure 5.13 shows that the features transformed by the NWFE has better classification accuracy than the classification of original features by the SVSA. Therefore, all the data used in the classification were transformed by the NWFE.

For the hybrid pixel- and texture-based classification, four spectral features and eight spatial features were combined to be used during classification. The confusion matrix of the SVSA method for both pixel-based and hybrid pixel- and texture-based classification, including user's accuracy, producer's accuracy and kappa value are tabulated in Table 5.6.

According to Table 5.6, in comparison to pixel-based classification, the confusion matrix of hybrid pixel- and texture-based classification is diagonally more dominant for all the classes, which means that the classification accuracy for all the classes was increased with the hybrid pixel- and texture-based classification. The overall classification accuracy was increased from 82.4% to 92.4%, and the kappa value was also increased from 76.9% to 89.7%. The user's and producer's accuracy were also increased with the hybrid pixel- and texture-based classification as shown in Figure 5.14.

According to Figure 5.14, especially for the most confused classes, which are the building, open ground and damage, user's accuracy and producer's accuracy dramatically increased with the hybrid pixel- and texture-based classification. In other words, by using the textural features, the class separability of the most confused classes were increased.

Regarding the high dimensionality resulting in using both the spectral and the spatial information together during classification, the computational load of the SVSA



Contrast

Correlation

Dissimilarity

Entropy

Homogeneity

Mean

Angular second moment

Variance

Figure 5.12: The spatial images based on Haralick’s features obtained from the panchromatic image with 9×9 window size.

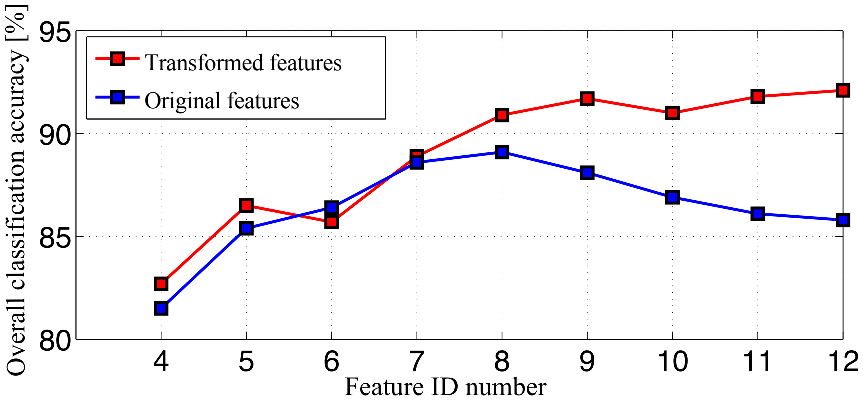


Figure 5.13: The classification accuracy of test data obtained by the SVSA with transformed and not transformed features.

Table 5.6: Confusion matrices for both pixel-based and hybrid pixel- and texture-based classification obtained by the SVSA.

SVSA Classification	Pixel-based classification						
	Vegetation	**Shadow**	**Building**	**Open ground**	**Damage**	**# Samples**	**User's accuracy**
Vegetation	**614**	10	0	0	11	635	**96.7**
Shadow	2	**437**	0	1	38	478	**91.4**
Building	0	0	**690**	37	97	824	**83.7**
Open Ground	0	0	25	**282**	110	417	**67.6**
Damage	0	2	151	160	**982**	1295	**75.8**
	616	449	866	480	1238	**# Samples**	
	99.7	**97.3**	**79.7**	**58.8**	**79.3**	**Producer's accuracy**	
					76.9	**Kappa value**	
					82.4	**Overall accuracy**	

SVSA classification	Hybrid pixel- and texture-based classification						
	Vegetation	**Shadow**	**Building**	**Open ground**	**Damage**	**# Samples**	**User's accuracy**
Vegetation	**590**	1	0	0	17	608	**95.8**
Shadow	26	**448**	1	3	0	478	**99.8**
Building	0	0	**821**	72	68	961	**94.8**
Open Ground	0	0	17	**382**	34	433	**79.6**
Damage	0	0	27	23	**1119**	1169	**90.4**
	616	449	866	480	1238	**# Samples**	
	97.0	**99.8**	**85.4**	**88.2**	**95.7**	**Producer's accuracy**	
					89.7	**Kappa value**	
					92.1	**Overall accuracy**	

algorithm is also of interest. Since separability between the classes increases due to using more features in the classification, the reference vectors representing the feature space accordingly decrease. Therefore, in addition to improvement on the classification accuracy with the hybrid pixel- and texture-based classification, the

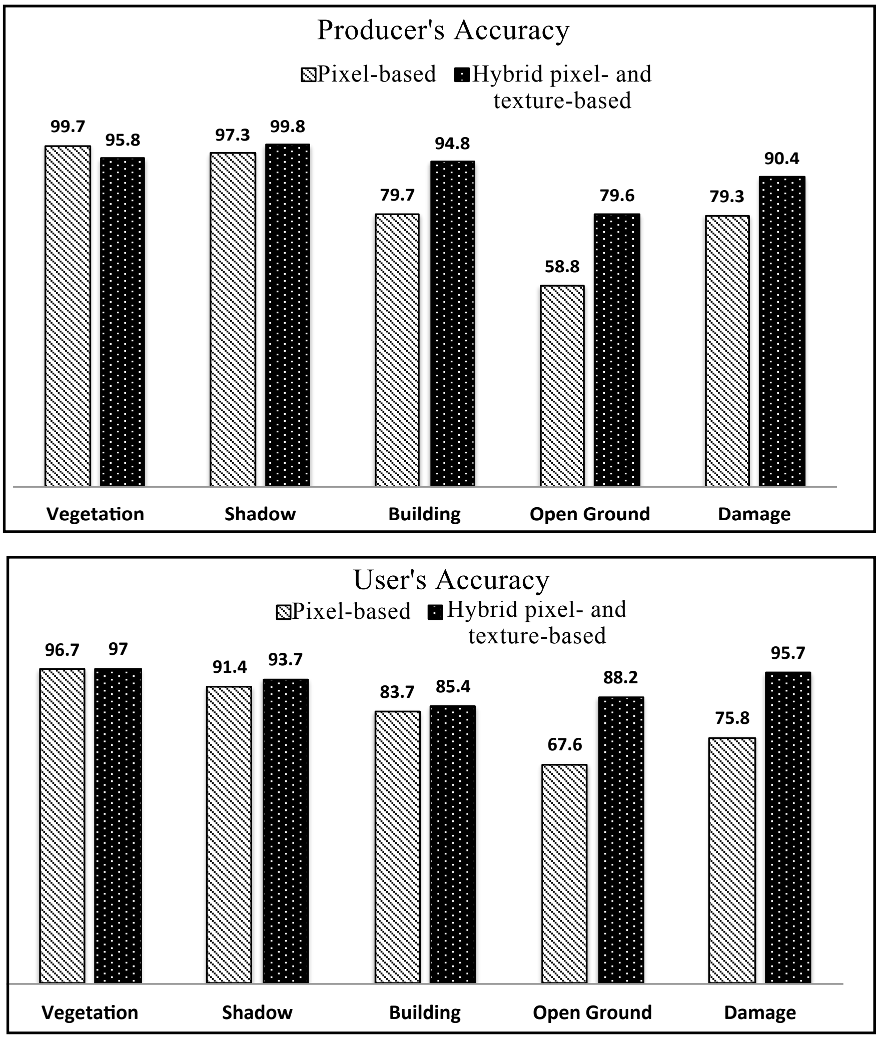


Figure 5.14: User's and producer's accuracy for each class in comparison to to pixel- and hybrid pixel- and texture-based classification.

computational time required for classification also decreases with use of less reference vectors.

Figure 5.15 shows the decrease of reference vectors as the number of features used in the classification increase, and thus also proves increase on separability of the classes. After the feature with ID 9, the number of the reference vectors slightly decrease. It means that after the feature ID 9, no more textural features may require to be used in the classification in terms of increasing separability. This result is also consistent with the overall classification accuracies obtained after the feature ID 9 in Table 5.5.

The thematic map with hybrid pixel- and texture-based classification produced by the SVSA method is shown in Figure 5.16.

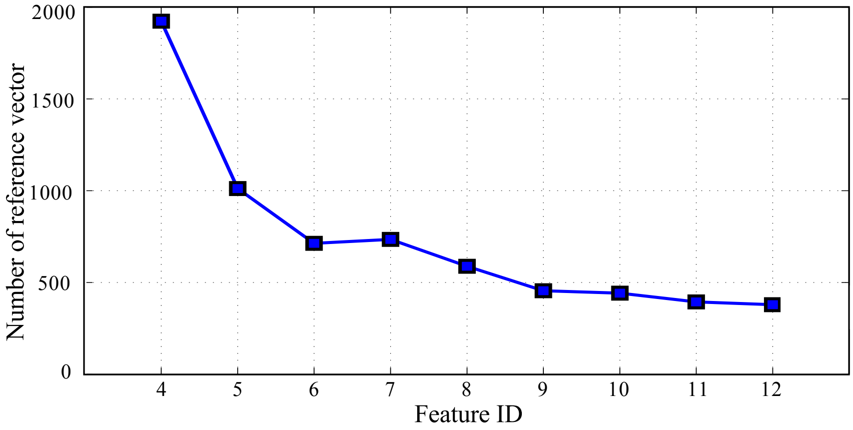


Figure 5.15: The number of reference vectors obtained by the SVSA with respect to number of features used during the learning.

The thematic map obtained by the SVSA with the pixel-based classification looks more noisy, salt-and-pepper classification noise, compared to the thematic map obtained with the hybrid pixel- and texture-based classification. The result of the thematic map obtained by the SVSA with the hybrid pixel- and texture-based classification was also visually compared to the thematic map obtained by European Space Agency (Figure 5.17).

According to Figure 5.17(c), it seems visually from the post-earthquake image that there were almost no damage on the bottom left corner of the image where the buildings were located. However, according to Figure 5.17(d), the bottom left corner of the image was moderately destroyed compared to rest of the places. Therefore, this result is compatible with the thematic map obtained by the SVSA.

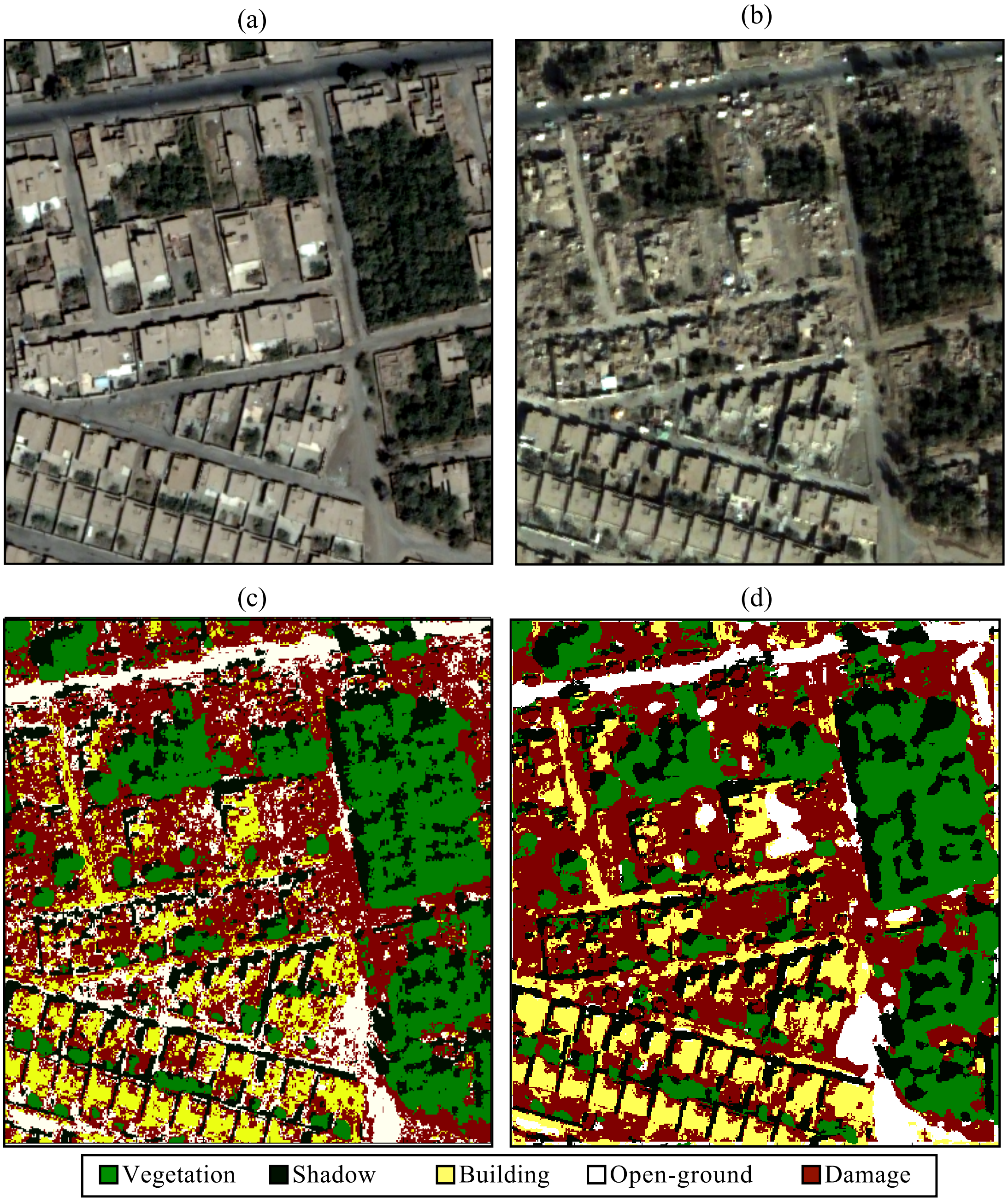


Figure 5.16: (a)Pre-earthquake pansharpened image. (b)Post-earthquake pansharpened image. (c)The thematic map obtained by the SVSA with the pixel-based classification. (d)The thematic map obtained by the SVSA with the hybrid pixel- and texture-based classification.

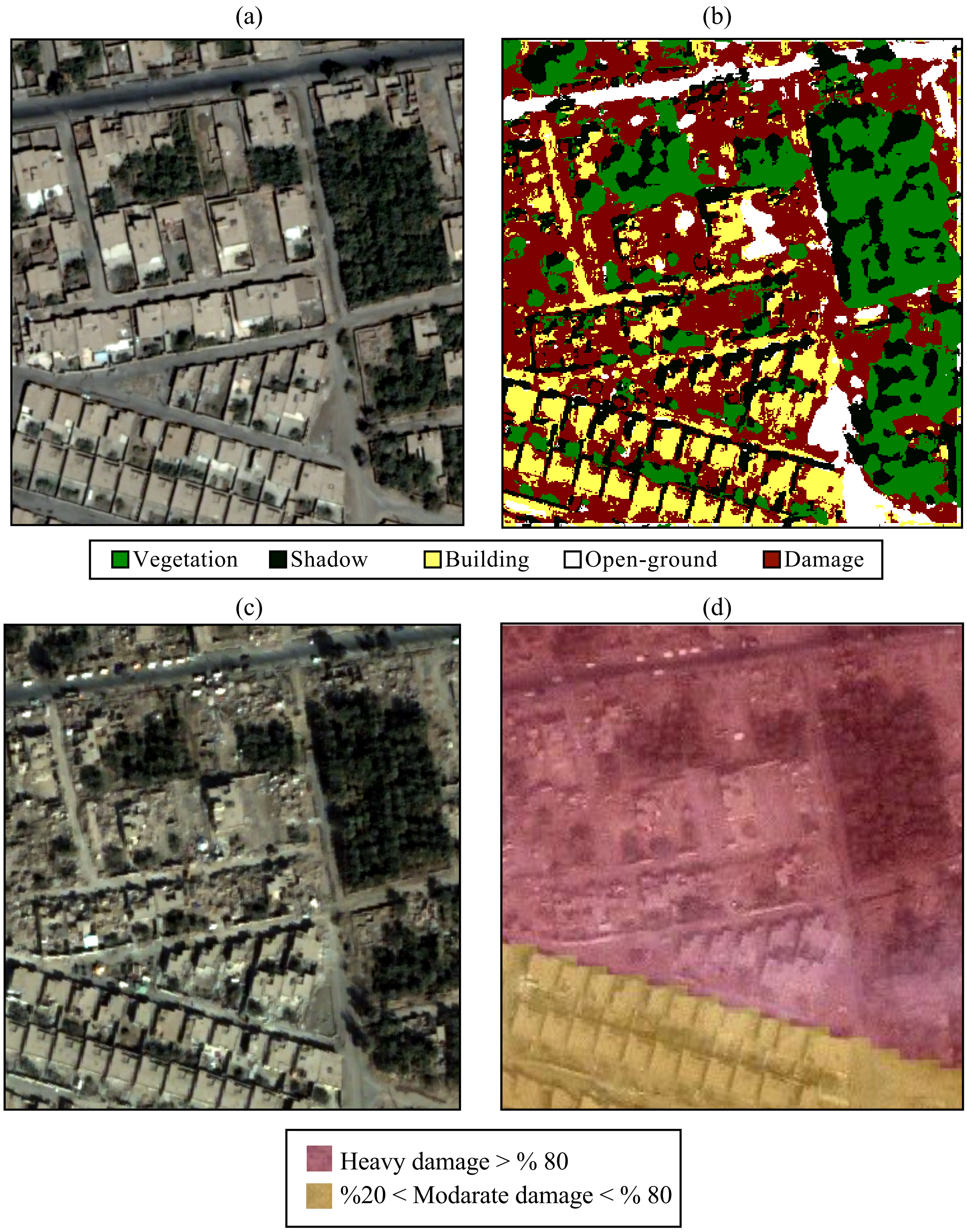


Figure 5.17: (a)Pre-earthquake image. (b)The thematic map obtained by the SVSA with hybrid pixel- and texture-based classification. (c)Post-earthquake image. (d)The damage map obtained by European Space Agency.

6. CONCLUSIONS AND DISCUSSIONS

In this thesis, a novel method called Support Vector Selection and Adaptation (SVSA) was proposed in order to overcome the drawbacks of NSVM on choosing a proper kernel type and associated parameters. The proposed method is reliable for classification of both linearly separable and nonlinearly separable data without need for a kernel. The SVSA method consists of selection of support vectors which most contribute to classification accuracy and adaptation of them with respect to training data. Their adaptation generates the reference vectors which are next used for classification of test data by the 1NN method with adaptive distance metric.

The proposed algorithm was tested with synthetic data and remote sensing images which are multisource, multispectral and hyperspectral, in comparison to linear, nonlinear SVM and KNN algorithms. The results showed that the SVSA method gave better classification results as compared to LSVM with nonlinearly separable data. The SVSA performance was also competitive with and often better than NSVM with RBF kernel function for both synthetic data and remote sensing images without requiring any kernel function.

For earthquake damage assessment, a pixel-based classification was conducted by the SVSA, and the proposed method gave the highest overall classification accuracy in comparison to other methods. It is well known that the performance of a classifier considerably depends on the textural features of an images. For this purpose, more textural features were obtained from the panchromatic image using the GLCM to be used in the classification together with the spectral information. The results obtained by the SVSA showed that the classification performance was improved by including textural features extracted from the image since class separability was increased.

For classification of earthquake damage, it is worth to imply that the quality and accurateness of the training data is very important in terms of classification performance since the SVSA algorithm is a supervised classification algorithm. As the damage patterns in the city of Iran, Bam are very complex, choice of training data was difficult, because any ground truth data was not available. It is well known that, having ground truth data makes choice of training data much easier and accurate.

In terms of computational complexities, the SVSA is faster than NSVM during training but slightly slower than NSVM during testing if kernel function is fixed. In the future, the SVSA method during testing can be speeded up by using parallel programming. In comparison to LSVM, the SVSA is slightly slower than LSVM during training because of the selection and the adaptation steps of the SVSA. In comparison to KNN, the SVSA method needs considerably less computation time and less memory requirements especially when having less number of reference vectors.

The SVSA method can be used in classification of any kind of data. All the experiments carried out throughout the thesis infer that the SVSA is a competitive classification method compared to NSVM without need for a kernel function.

REFERENCES

[1] **Shmilovici, A.**, 2005. *The Data Mining and Knowledge Discovery Handbook*, Springer.

[2] **Fukunaga, K.**, 1990. *Introduction to Statistical Pattern Recognition*, Academic Press, 2 edition.

[3] **Alpaydin, E.**, 2004. *Introduction to Machine Learning*, The MIT Press.

[4] **Atkinson, P.M. and Tatnall, A.R.L.**, 1997. Neural Networks in Remote Sensing, *International Journal of Remote Sensing*, **18**(4), 699–709.

[5] **Duda, R.O.**, **Hart, P.E. and Stork, D.G.**, 2000. *Pattern Classification*, Wiley-Interscience, 2nd edition.

[6] **Tao, Q.**, **Wu, G.W.**, **Wang, F.Y. and Wang, J.**, 2005. Posterior probability Support Vector Machines for unbalanced data, *IEEE Transactions on Neural Networks*, **16**(6), 1561–1573.

[7] **Shilton, A.**, **Palaniswami, M.**, **Ralph, D. and Tsoi, A.C.**, 2005. Incremental training of Support Vector Machines, *IEEE Transactions on Neural Networks*, **16**(1), 114–131.

[8] **Tao, Q.**, **Chu, D. and Wang, J.**, 2008. Recursive Support Vector Machines for dimensionality reduction, *IEEE Transactions on Neural Networks*, **19**(1), 189–193.

[9] **Huang, C.**, **Davis, L.S. and Townshend, J.R.G.**, 2002. An Assessment of Support Vector Machines for Land Cover Classification, *International Journal of Remote Sensing*, **23**(4), 725–749.

[10] **Hermes, L.**, **Frieauff, D.**, **Puzicha, J. and Buhmann, J.M.**, 1999. Support Vector Machines for land usage classification in LANDSAT TM imagery, *IGARSS'99 Proceedings*, volume 1, pp.348–350.

[11] **Brown, M.**, **Lewis, H. and Gunn, S.**, 2000. Linear spectral mixture models and Support Vector Machines for remote sensing, *IEEE Transactions on Geoscience and Remote Sensing*, **38**(5), 2346–2360.

[12] **Waske, B.**, **Van Der Linden, S.**, **Benediktsson, J.**, **Rabe, A. and Hostert, P.**, 2010. Sensitivity of Support Vector Machines to Random Feature Selection in Classification of Hyperspectral Data, *IEEE Transactions on Geoscience and Remote Sensing*, **48**(7), 2880–2889.

[13] **Lardeux, C.**, **Frison, P.L.**, **Tison, C.**, **Souyris, J.C.**, **Stoll, B.**, **Fruneau, B. and Rudant, J.P.**, 2009. Support Vector Machine for Multifrequency SAR Polarimetric Data Classification, *IEEE Transactions on Geoscience and Remote Sensing*, **47**(12), 4143–4152.

[14] **Kohonen, T.**, 1986. Learning Vector Quantization for Pattern Recognition, **Technical Report, TKK-F-A601**, Helsinki University of Technology.

[15] **Frank, A. and Asuncion, A.**, 2010. UCI Machine Learning Repository, **Technical Report**, University of California, Irvine, School of Information and Computer Sciences.

[16] **Pal, M. and Mather, M.**, 2005. Support Vector Machines for Classification in Remote Sensing, *International Journal of Remote Sensing*, **26**(5), 1007–1011.

[17] **Heikkinen, V., Tokola, T., Parkkinen, J., Korpela, I., and Jaaskelainen, T.**, 2010. Simulated Multispectral Imagery for Tree Species Classification Using Support Vector Machines, *IEEE Transactions on Geoscience and Remote Sensing*, **48**(3), 1355–1364.

[18] **Camps-Valls, G., and Bruzzone, L.**, 2005. Kernel-Based Methods for Hyperspectral Image Classification, *IEEE Transactions on Geoscience and Remote Sensing*, **43**(6), 1351–1362.

[19] **Waske, B., and Benediktsson, J. A.**, 2007. Fusion of Support Vector Machines for Classification of Multisensor Data, *IEEE Transactions On Geoscience and Remote Sensing*, **45**(12), 3858–3866.

[20] **Mari, J. M., Bovolo, F., Chova, L. G., Bruzzone, B., and Camps-Valls, G.**, 2010. Semisupervised One-Class Support Vector Machines for Classification of Remote Sensing Data, *IEEE Transactions on Geoscience and Remote Sensing*, **48**(8), 3188–3197.

[21] **Cherkassky, V., and Mulier, F.**, 1998. *Learning From Data: Concepts, Theory, and Methods*, Wiley-interscience, New York.

[22] **Yue, S., Li, P., and Hao P.**, 2003. SVM Classification:Its Contents and Challenges, *Applied Mathematic - A Journal of Chinese Universities*, **18**(3), 332–342.

[23] **Scholkopf, B.**, 1997. Support Vector Learning, *Ph.D. thesis*, Technische Universitat Berlin.

[24] **Shawe-Taylor, J., Bartlett, P.L., Willianmson, R.C, and Anthony, M.**, 1998. Structural Risk Minimization over Data-Dependent Hierarchies, *IEEE Transaction Information Theory*, **44**(5), 1926–1940.

[25] **Kavzoglu, T. and Colkesen, I.**, 2009. A Kernel Functions Analysis for Support Vector Machines for Land Cover Classification, *International Journal of Applied Earth Observation and Geoinformation*, **11**(5), 352–359.

[26] **Tan, Y. and Wang, J.**, 2004. A Support Vector Machine with a Hybrid Kernel and Minimal Vapsnik-Chervonenkis Dimension, *IEEE Transactions on Knowledge and Data Engineering*, **16**(4), 385–395.

[27] **Chi, H.C. and Ersoy, O.K.**, 2002. Support Vector Machine Decision Trees With Rare Event Detection, *International Journal for Smart Engineering System Design*, **4**, 225–242.

[28] **Kasapoglu, N.G. and Ersoy, O.K.**, 2007. Border Vector Detection and Adaptation for Classification of Multispectral and Hyperspectral Remote Sensing Images, *IEEE Transactions on Geoscience and Remote Sensing*, **45**(12), 3880–3892.

[29] **Fu, Z. and Robles-Kelly, A.**, 2008. On Mixtures of Linear SVMs for Nonlinear Classification, *Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pp.489–499.

[30] **Chapelle, O. and Vapnik, V.N.**, 2002. Choosing Multiple Parameters for Support Vector Machines, *Machine Learning*, **46**(1-3), 131–159.

[31] **Chung, K.M., Kao, W.C., Sun, T., Wang, L.L. and Lin C.J.**, 2003. Radius Margin Bounds for Support Vector Machines with the RBF Kernels, *Neural Computation*, **15**(11), 2643–2681.

[32] **Ayata, N.E., Cherieta, M. and Suenb C.Y.**, 2005. Automatic Model Selection for the Optimization of SVM Kernels, *Pattern Recognition*, **38**(10), 1733–1745.

[33] **Foody, G.M.**, 2008. RVM-Based Multi-Class Classification of Remotely Sensed Data, *International Journal of Remote Sensing*, **29**(6), 1817–1823.

[34] **Bazi, Y. and Melgani, F.**, 2006. Toward an Optimal SVM Classification System for Hyperspectral Remote Sensing Images, *IEEE Transactions on Geoscience and Remote Sensing*, **44**(11), 3374–3385.

[35] **Taskin Kaya, G., and Ersoy, O.K. and Kamasak, M.E.**, 2011. Support Vector Selection and Adaptation for Remote Sensing Classification, *IEEE Transactions on Geoscience and Remote Sensing*, **49**(6), 2071–2079.

[36] **Kaya, G., Ersoy, O.K. and Kamasak, M.E.**, 2009. Support Vector Selection and Adaptation and its Application in Remote Sensing, *4th International on Recent Advances in Space Technologies*, IEEE, pp.408–412.

[37] **Taskin Kaya, G., Ersoy, O.K. and Kamasak, M.E.**, June 29-July 1, 2009. Support Vector Selection and Adaptation, *International Symposium on Innovations in Intelligent Systems and Applications*.

[38] **Taskin Kaya, G., and Ersoy, O.K.**, 2008. Support Vector Selection and Adaptation for Classification of Remote Sensing Images, **Technical Report, TR-ECE-09-2**, Electrical and Computer Engineering, Purdue University.

[39] **Kaya, G., Ersoy, O.K. and Kamasak, M.E.**, 2009. Support Vector Selection and Adaptation for Classification of Earthquake Images, *IEEE International Geoscience and Remote Sensing Symposium*, volume 2, pp.II–851–854.

[40] **Cheema, U.**, 2007. Expert Systems for Earthquake Damage Assessment, *IEEE Aerospace and Electronic Systems Magazine*, **22**(9), 6–10.

[41] **Eguchi, R.T. and Mansouri, B.**, 2005. Use of Remote Sensing Technologies for Building Damage Assessment after the 2003 Bam, Iran, Earthquake-Preface to Remote Sensing Papers, *Earthquake Spectra*, **21**(S1), 2017–212.

[42] **Matsuoka, M. and Yamazaki, F.**, 2001. Image Processing for Building Damage Detection due to Disasters Using SAR Intensity Images, *Proceedings of the Japanese Conference on Remote Sensing*, volume 31, pp.269–270.

[43] **Matsuoka, M. and Yamazaki, F.**, 2002. Application of the Damage Detection Method using SAR Intensity Images to Recent Earthquakes, *IEEE International Geoscience and Remote Sensing Symposium, IGARSS '02.*, volume 4, pp.2042–2044.

[44] **Turker, M. and San, B.T.**, 2003. SPOT HRV data analysis for detecting earthquake-induced changes in Izmit, Turkey, *International Journal of Remote Sensing*, **24**(12), 0143–1161.

[45] **Adams, B.J.**, 2004. Improved Disaster Management through Post-Earthquake Building Damage Assessment using Multitemporal Satellite Imagery, *XXth ISPRS Congress Proceedings*, volumeXXXV, Istanbul, Turkey.

[46] **Kosugi, Y., Sakamoto, M., Fukunishi, M., Lu, W., Doihara, T. and Kakumoto, S.**, 2004. Urban Change Detection Related to Earthquakes Using an Adaptive Nonlinear Mapping of High-Resolution Images, *IEEE Geoscience and Remote Sensing Letters*, **1**(3), 152–156.

[47] **Turker, M. and Cetinkaya, B.**, 2005. Automatic Detection Of Earthquake-Damaged Buildings Using DEMs Created From Pre- And Post-Earthquake Stereo Aerial Photographs, *International Journal of Remote Sensing*, **26**(4), 823–832.

[48] **Rathje, E.M., Crawford, M., Woo, K. and Neuenschwander, A.**, 2005. Damage Patterns from Satellite Images of the 2003 Bam, Iran, Earthquake, *Earthquake Spectra*, **21**(S1), 295–307.

[49] **Gamba, P., Acqua, F.D. and Lisini, G.**, 2006. Change Detection of Multitemporal SAR Data in Urban Areas Combining Feature-Based and Pixel-Based Techniques, *IEEE Transactions on Geoscience and Remote Sensing*, **44**(10), 2820–2827.

[50] **Gamba, P., Dell'Acqua, F. and Trianni, G.**, 2007. Rapid Damage Detection in the Bam Area Using Multitemporal SAR and Exploiting Ancillary Datasets, *IEEE Transactions on Geoscience and Remote Sensing*, **45**(6), 1582–1589.

[51] **Kaya, G.T.**, **Musaoglu, N. and Ersoy, O.K.**, 2011. Damage Assessment of 2010 Haiti Earthquake with Post-Earthquake Satellite Image by Support Vector Selection and Adaptation, *Special issue on Haiti Earthquake for American Society for Photogrammetric Engineering*, accepted.

[52] **Gamba, P. and Casciati, F.**, 1998. GIS and Image Understanding for Near-Real-Time Earthquake Damage Assessment, *Photogrammetric Enginnering & Remote Sensing*, **64**(10), 987–994.

[53] **Turker, M. and San, B.T.**, 2004. Detection of Collapsed Buildings Caused by the 1999 Izmit, Turkey Earthquake through Digital Analysis of Post-Event Aerial Photographs, *International Journal of Remote Sensing*, **25**(21), 4701–4714.

[54] **Saito, K. and Spence, R.**, 2004. Rapid damage mapping using post-earthquake satellite images, *Geoscience and Remote Sensing Symposium. IGARSS '04. Proceedings.*, volume 4, pp.2272–2275.

[55] **Balz, T. and Liao, M.**, 2010. Building-damage detection using post-seismic high-resolution SAR satellite data, *International Journal of Remote Sensing*, **31**(13), 3369–3391.

[56] **Gong, J.**, **Wang, D.**, **Li, Y.**, **Zhang, L.**, **Yue, Y.**, **Zhou, J. and Song, Y.**, 2010. Earthquake-Induced Geological Hazards Detection under Hierarchical Stripping Classification Framework in the Beichuan Areas, *Landslides*, **7**(2), 181–189.

[57] **Vapnik, V.**, 1979. Estimation of Dependences Based on Empirical Data, *Nauka Moscow (English translation: Springer Verlag, New York, 1982)*, **27**, 5165–5184.

[58] **Cortes, C. and Vapnik, V.**, 1995. Support-vector networks, *Machine Learning*, **20**(3), 273–297.

[59] **Scholkopf, B. and Smola, A.**, 2001. *Learning with Kernels*, The MIT Press.

[60] **Keerthi, S. and Lin, C.J.**, 2003. Asymptotic behaviors of support vector machines with Gaussian kernel, *Neural Computation*, **15**(7), 1667–1689.

[61] **Lin H. and Lin C.**, 2003. A study on sigmoid kernels for SVM and the training of non- PSD kernels by SMO-type methods, **Technical Report**, Department of Computer Science, National Taiwan University.

[62] **Debnath, R.**, **Takahide, N. and Takahashi, H.**, 2004. A decision based one-against-one method for multi-class support vector machine, *Pattern Analysis & Applications*, **7**, 164–175.

[63] **Melgani, F. and Bruzzone L.**, 2004. Classification of Hyperspectral Remote Sensing Images with Support Vector Machines, *IEEE Transactions on Geoscience and Remote Sensing*, **42**(8), 1778–1790.

[64] **Mantero, P.**, **Moser, G. and Serpico, S.**, 2005. Partially Supervised Classification of Remote Sensing Images through SVM-Based Probability Density Estimation, *IEEE Transactions on Geoscience and Remote Sensing*, **43**(3), 559–570.

[65] **Mountrakis, G.**, **Im, J. and Ogole, C.**, 2010. Support Vector Machines in Remote Sensing: A Review, *ISPRS Journal of Photogrammetry and Remote Sensing*, in Press.

[66] **Kaya, G.T. and Ersoy, O.K. and Kamasak, M.E.**, 2010. Hybrid SVM and SVSA method for classification of remote sensing images, *IEEE Geoscience and Remote Sensing Symposium*, pp.2828–2831.

[67] **Kohonen, T.**, **Kangas, J.**, **Laaksonen, J. and Torkkola, K.**, 1992. LVQPAK: A Software Package for the Correct Application of Learning Vector Quantization Algorithms, *International Joint Conference on Neural Network (IJCNN)*, volume 1, pp.725–730.

[68] **Wang, J.**, **Neskovic, P. and Cooper, L.N.**, 2007. Improving Nearest Neighbor Rule with a Simple Adaptive Distance Measure, *Pattern Recognition Letters*, **28**(2), 207–213.

[69] **Hammer, B.**, **Strickert, M. and Villmann, T.**, 2005. On the Generalization Ability of GRLVQ Network, *Neural Processing Letters*, **21**(2), 109–120.

[70] **Crammer, K.**, **Gilad-Bachrach, R.**, **Navot, A. and Tishby, N.**, 2002. Margin Analysis of the LVQ Algorithms, *Advances in Neural Information Processing Systems*.

[71] **Duin, R.P.W.**, **Juszczak, P.**, **Paclik, P. and et. al.**, 2007. A Matlab Toolbox for Pattern Recognition, *Delft University of Technology, PRTools4.1*.

[72] **Heijden, F.V.D.**, **Duin, R.**, **Ridder, D.D. and Tax, D.M.J.**, 2004. *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*, Wiley.

[73] **Chang, C.C. and Lin, C.**, 2010, LIBSVM: A Library for Support Vector Machines, http://www.csie.ntu.edu.tw/˜cjlin/libsvm, accessed on May, 2008.

[74] **Congalton, R.G. and Green, K.**, 1999. *Assessing the Accuracy of Remotely Sensed Data*, Lewis.

[75] **Liu, C.**, **Frazier, P. and Kumar, L.**, 2007. Comparative Assessment of the Measures of Thematic Classification Accuracy, *Remote Sensing of Environment*, **107**(4), 606–616.

[76] **Hsu, C.W.**, **Chang, C.C. and Lin, C.J.** A Practical Guide to Support Vector Classification, A Practical Guide to Support Vector Classification, http://www.csie.ntu.edu.tw/˜cjlin/papers/guide/guide.pdf, accessed on May, 2008.

[77] **Dong, J.X.**, 2005. Fast SVM Training Algorithm with Decomposition on Very Large Datasets, *IEEE Transaction Pattern Analysis and Machine Intelligence*, **27**(4), 603–618.

[78] **Platt, J.C.**, 1998, Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.

[79] **Liu, X.**, **Hall, L.O. and Bowyer, K.W.**, 2004. A Parallel Mixture of SVMs for Very Large Scale Problems, *Neural Computation*, **16**(7), 1345–1351.

[80] **Mullin, M. and Sukthankar, R.**, 2000. Complete Cross-Validation for Nearest Neighbor Classifiers, *Proceedings of 17th International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp.639–646.

[81] **Zhu, H. and Basir, O.**, 2005. An Adaptive Fuzzy Evidential Nearest Neighbor Formulation for Classifying Remote Sensing Images, *IEEE Transactions on Geoscience and Remote Sensing*, **43**(8), 1874–1889.

[82] **Blake, C.L. and Merz, C.J.**, 1998, UCI Repository of Machine Learning Databases, University of California, Irvine, Dept. of Information and Computer Sciences., http://archive.ics.uci.edu/ml/, accessed on July, 2009.

[83] **Benediktsson, J.**, **Swain, P. and Ersoy, O.K.**, 1990. Neural Network Approaches Versus Statistical Methods in Classification of Multisource Remote Sensing Datasets, *IEEE Transactions on Geoscience and Remote Sensing*, **28**(4), 540–552.

[84] **Kaya, S.**, **Curran, P.J. and Llewellyn, G.**, 2005. Post-Earthquake Building Collapse: A Comparison of Government Statistics and Estimates Derived From SPOT HRVIR Data, *International Journal of Remote Sensing*, **46**(13), 2731–2740.

[85] **Shahshahani, B. and Landgrebe, D.A.**, 1994. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon, *IEEE Transactions on Geoscience And Remote Sensing*, **32**(5), 1087–1095.

[86] **Hsu, P.H.**, 2007. Feature extraction of hyperspectral images using wavelet and matching pursuit, *ISPRS Journal of Photogrammetry and Remote Sensing*, **62**(2), 78–92.

[87] **Pal, M. and Foody, G.M.**, 2010. Feature selection for classification of hyperspectral data by SVM, *IEEE Transactions on Geoscience and Remote Sensing*, **48**(5), 2297–2307.

[88] **Gidudu, A. and Ruther, H.**, 2007. Comparison of feature selection techniques for SVM classification, *10th International Symposium Physical Measurement Spectral Signatures Remote Sensing*, volumeXXXVI, pp.258–263.

[89] **Landgrebe, A.D.**, 2003. *Signal Theory Methods in Multispectral Remote Sensing*, John Wiley and Sons.

[90] **Huang, X. and Zhang, L.**, 2008. An Adaptive Mean-Shift Analysis Approach for Object Extraction and Classification From Urban Hyperspectral Imagery, *IEEE Transactions on Geoscience and Remote Sensing*, **46**(12), 4173–4185.

[91] **Kuo, B.C. and Chang, K.Y.**, 2007. Feature Extractions for Small Sample Size Classification Problem, *IEEE Transactions on Geoscience and Remote Sensing*, **45**(3), 756–764.

[92] **Gualtieri, J.A. and Chettri, S.**, 2000. Support vector machines for classification of hyperspectral data, *IEEE Geoscience and Remote Sensing Symposium*, volume 2, pp.813–815.

[93] **Zhang Q, W.J.**, 2001. Texture analysis for urban spatial pattern study using SPOT imagery, *IEEE Transactions on Geoscience and Remote Sensing*, **28**(4), 513–519.

[94] **Karakahya, H.**, **Yazgan, B. and Ersoy, O.K.**, 2003. A Spectral-Spatial Classification Algorithm for Multispectral Remote Sensing Data, *Lecture Notes in Computer Science*, **2714/2003**(173), 1011–1017.

[95] **Kaya, G.T.**, **Ersoy, O.K. and Kamasak, M.E.**, 2010. Spectral and spatial classification of earthquake images by support vector selection and adaptation, *International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp.194–197.

[96] **Fauvel, M.**, **Benediktsson, J.A.**, **Chanussot, J. and Sveinsson, J.R.**, November 2008. Spectral and Spatial Classification of Hyperspectral Data Using SVMs and Morphological Profiles, *IEEE Transaction on Geoscience and Remote Sensing*, **46**(11), 3804–3814.

[97] **Ryherd, S. and Woodcock, C.**, 1996. Combining Spectral and Texture Data in the Segmentation of Remotely Sensed Images, *Photogrammetric Engineering and Remote Sensing,*, **62(2)**, 181–194.

[98] **J.S. Weszka, C.D. and Rosenfeld, A.**, 1976. A comparative Study of Texture Measures for Terrain Classification, *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-6**, 269–285.

[99] **Baraldi, A. and Parmiggiani, F.**, 1995. An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters, *IEEE Transactions on Geoscience and Remote Sensing*, **33**(2), 293–304.

[100] **Conners, R. and Harlow, C.**, 1980. A Theoretical Comparison of Texture Algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**(3), 204–221.

[101] **Haralick, R. M., Shanmugam, K., and Dinstein I.**, 1973. Textural Features For Image Classification, *IEEE Transactions on Systems, Man, and Cybernetics*, **3**(6), 610–621.

[102] **Kuo, B.C. and Landgrebe, D.A.**, 2004. Nonparametric Weighted Feature Extraction for Classification, *IEEE Transactions on Geoscience and Remote Sensing*, **42**(5), 1096–1105.

[103] **Kuo, B. and Landgrebre, D.**, 2002. Hyperspectral Data Classification Using Nonparametric Weighted Feature Extraction, *IEEE Geoscience and Remote Sensing Symposium*, volume 3, pp.1428–1430.

[104] **Kuo, B.C.**, **Hung, C.C.**, **Chang, C.W. and Wang, H.P.**, 2006. A Modified Nonparametric Weight Feature Extraction Using Spatial and Spectral Information, *IEEE International Geoscience and Remote Sensing Symposium*, volume1-8, IEEE, pp.172–175.

[105] **Kuo, B.C.**, **Li, C.H. and Yang, J.M.**, 2009. Kernel Nonparametric Weighted Feature Extraction for Hyperspectral Image Classification, *IEEE Transactions on Geoscience and Remote Sensing*, **47**(4), 1139–1155.

[106] **Blanzieri, E. and Melgani, F.**, 2008. Nearest Neighbor Classification of Remote Sensing Images with the Maximal Margin Principle, *IEEE Transactions on Geoscience and Remote Sensing*, **46**(6), 1804–1811.

[107] **Cohen, J.**, 1960. A coefficient of agreement for nominal scales, *Educational and Psychological Measurement*, **20**(1), 37–46.

[108] **Richards, J.A. and Jia, X.**, 1998. *Remote Sensing Digital Image Analysis: An Introduction*, Springer.

[109] **Lewis, H.G. and Brown, M.**, 2001. A generalized confusion matrix for assessing area estimates from remotely sensed data, *International Journal of Remote Sensing*, **22**(16), 3223–235.

APPENDICES

A: K-Nearest Neighbour

The K-Nearest Neighbour classification of a sample is made on the basis of the classifications of the selected number of K neighbours [106].

Given a point x' of the d-dimensional input feature space, an ordering function f'_x : $\mathbb{R}^d \rightarrow \mathbb{R}$ is defined. The typical ordering function is based on the Euclidean metrics: $fx'(x) = \| x - x \|$. By means of an ordering function, it is possible to order the entire set of training samples X with respect to x'. This corresponds to define a function $r_{x'} : \{1,\dots,N\} \rightarrow \{1,\dots,N\}$ that reorders the indexes of the N training points. We define this function recursively, as follows:

$$\begin{cases} r_{x'}(1) = \arg\min_i f_{x'}(x_i) & \text{with } i \in \{1,\dots,N\}, \\ r_{x'}(j) = \arg\min_i f_{x'}(x_i) & \text{with } i \in \{1,\dots,N\} \text{ and} \\ & i \neq r_{x'}(1),\dots,i \neq r_{x'}(j-1) \text{ for } j = 2,\dots,N. \end{cases}$$

In this way, $x_{r_{x'}}(j)$ is the point of the set X in the jth position in terms of distance from x', namely the jth nearest neighbour, and $f_{x'}(x_{r_{x'}}(j)) = \|x_{r_{x'}}(j) - x'\|$ is its distance from x'. In other words, the following holds: $j < k \Rightarrow f_{x'}(x_{r_{x'}}(j)) \leq f_{x'}(x_{r_{x'}}(k))$. Given the above definition, the decision rule of the KNN classifier for binary classification problems is defined by the following majority rule:

$$\text{KNN}(x) = sign\left(\sum_{i=1}^{k} y_{r_x}(i)\right) \tag{A.1}$$

where $y_{r_x}(i) \in \{-1,+1\}$ is the class label of the i^{th} nearest training sample.

B: Learning Vector Quantization

The aim of Learning Vector Quantization (LVQ) is to define optimal class regions in the input data space. In general, from a given training set X of n prototypes, several codebook vectors m_i are initially selected. Then these codebook vectors are adaptively modified in such a way that the nearest neighbourhood rule minimizes the average expected misclassification probability.

The learning process consists of iteratively moving some of the codebook vectors m_i in the neighbourhood of a prototype x towards it and some away from it according to the result of 1NN rule. Differences among the various LVQ algorithms refer to the modifications to the codebook vectors.

Let $x \in X$ be an input sample, let m_c be the nearest codebook vector to x, and let $m_c(t)$ represent the codebook vector m_c at step t. The learning process in the basic LVQ1 algorithm consists of updating the position of m_c. If the class label of the codebook vector m_c matches the class label of the training prototype x, then the codebook vector is moved towards x. Otherwise, it is moved away from the given input prototype. The modifications to the codebook vector m_c are performed according to the following general rule:

$$m_c[t+1] = \begin{cases} m_c[t] + \eta[t]\,(x - m_c[t]) & \text{if class}(x) = \text{class}(m_c), \\ m_c[t] - \eta[t]\,(x - m_c[t]) & \text{if class}(x) \neq \text{class}(m_c). \end{cases} \qquad \textbf{(B.1)}$$

where $0 < \eta[t] < 1$ denotes the corresponding learning rate, and it may be constant or decrease monotonically with time.

C: Overall accuracy

The confusion matrix's[1] cell values on the diagonal represent a pixel count of correctly classified pixels. The sum of the diagonal cells in the matrix represents the total number of correctly classified pixels. The proportion of the total number of correctly classified pixels to the total number of pixels in the matrix gives the *overall accuracy* for the classification.

$$\text{Overall accuracy} = \frac{(TP+TN)}{(P+N)} \qquad \textbf{(C.1)}$$

There are also several measures of classification accuracy which can be derived from the confusion matrix; namely user's and producer's accuracy.

[1] See Appendix E for detailed information.

D: Kappa statistic

The accuracy measurements, the overall accuracy, producer's accuracy, and user's accuracy, are based on either the principal diagonal, columns, or rows of the confusion matrix only, which does not use the information from the whole confusion matrix[1]. Kappa coefficient found by Cohen (1960) uses all of the information in the confusion matrix in order for the chance allocation of labels to be taken into consideration [107].

Let's suppose **A** is a confusion matrix with $c \times c$ dimension, where c is the number of classes. The Kappa coefficient is defined by

$$K = \frac{P_{OA} - P_{CA}}{1 - P_{CA}} \tag{D.1}$$

$$P_{OA} = \frac{\sum_{i=1}^{c} a_{ii}}{N} \tag{D.2}$$

$$P_{CA} = \frac{\sum_{i=1}^{c}(a_{i+}a_{+i})}{N^2} \tag{D.3}$$

where P_{OA} and P_{CA} correspond to overall accuracy and chance agreement, respectively. a_{ii} is the entry of the confusion matrix, a_{i+} and a_{+i} are the marginal totals of row i and column j, respectively, and N is the total number of samples.

The Kappa coefficient takes not only the principal diagonal entries but also the off-diagonal entries into consideration. The higher the value of Kappa, the better the classification performance. If all classes are correctly identified, Kappa takes the value 1. As the values of the off-diagonal entries increase, the value of Kappa decreases.

Generally, Kappa coefficient greater than 0.80 (80%) represent strong agreement between the remotely sensed classification and the reference data while the values between 0.4 and 0.8 represent moderate agreement. The coefficient below 0.4 is indicative of poor agreement.

[1] See Appendix E for detailed information.

E: Confusion matrix

Accuracy is determined empirically, by selecting a sample (desirably an independent random sample) of pixels from the thematic map and checking their labels against classes determined from reference data [108]. Often reference data is referred to as ground truth, and the pixels selected for accuracy checking are called *test* pixels. From these checks the percentage of pixels from each class in the image labelled correctly or labelled not correctly by the classifier can be estimated. These results are then expressed in tabular form, often referred to as a confusion matrix, also known as a error matrix or contingency table, and is illustrated in Figure E.1.

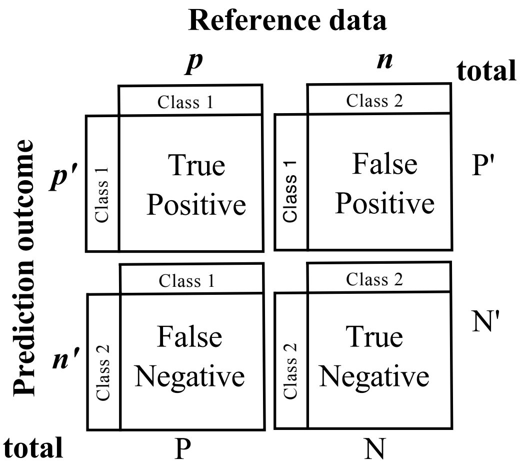


Figure E.1: A confusion matrix for two classes. **p**, **n** are the type of classes class 1 and class 2, respectively. $\mathbf{p}'$ and $\mathbf{n}'$ are the predicted classes by the classifier.

A confusion matrix contains the classifier performance about reference and predicted classifications. The confusion matrix is a square matrix by $c \times c$, where c is the number of classes. The values listed in the matrix represent the number of ground truth pixels, in each case, correctly and incorrectly labelled by the classifier. Information in the horizontal rows generally corresponds to the thematic classes resulting from image classification. The vertical columns show the thematic information of the reference data.

The basic attributes of a confusion matrix are true positive, true negative, false positive, and false negative, respectively. If the outcome from a prediction is **p** and the reference label is also **p**, then it is called a true positive (TP); however if the reference label is **n** then it is to be a false positive (FP). Conversely, a true negative (TN) occurs when both the prediction outcome and the reference label are **n**, and false negative (FN) is when the prediction outcome is **n** while the reference label is **p**.

F: K-Fold Cross Validation

In K-fold cross-validation, the dataset X is divided randomly into K equal sized parts, X_i, $i = 1, \ldots, K$. In order to generate each pair, one of the K parts is kept out as the validation set, and the remaining $K-1$ parts are combined to form the training set. Doing this K times, each time leaving out another one of the K parts out, K pairs are obtained as follows:

$$\begin{aligned}
V_1 = X_1 \quad T_1 &= X_2 \cup X_3 \cup \ldots \cup X_K \\
V_2 = X_2 \quad T_2 &= X_1 \cup X_3 \cup \ldots \cup X_K \\
&\vdots \\
V_K = X_K \quad T_K &= X_1 \cup X_2 \cup \ldots \cup X_{K-1}
\end{aligned}$$

The cross-validation process is repeated K times, the folds, with each of the K sub-samples used exactly once as the validation data. The K results from the folds then can be averaged (or otherwise combined) to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once.

As K increases, the percentage of training samples increases and a more robust estimator can be obtained, but the validation sets become smaller. 10-fold cross-validation is commonly used.

G: User's and Producer's accuracy

The user's accuracy is the proportion of correctly classified pixels (TP and TN) to the total number of pixels in that row (P' and N') for each row in the confusion matrix[1] [109].

$$\begin{aligned}\text{User's accuracy} &= \frac{TP}{P'} \quad \text{for the first row} \qquad \textbf{(G.1)}\\ &= \frac{TN}{N'} \quad \text{for the second row} \qquad \textbf{(G.2)}\end{aligned}$$

The user's accuracy provides a measure for the thematic map user of the probability that the pixels on the classified map is being correctly assigned during the classification process. In other words, the user accuracy denotes the probability that a classified pixel actually represents that information class on the ground. It can also be called a measure of a commission error which is a percentage of the pixels incorrectly assigned to a particular class that actually belong to other classes.

The producer's accuracy is calculated for each column by comparing the proportion of correctly classified pixels in that column with the total number of pixels in the column.

$$\begin{aligned}\text{Producer's accuracy} &= \frac{TP}{P} \quad \text{for the first column} \qquad \textbf{(G.3)}\\ &= \frac{TN}{N} \quad \text{for the second column} \qquad \textbf{(G.4)}\end{aligned}$$

It shows what percentage of a particular ground class is correctly classified. The producer's accuracy provides a measure of how well the classification is achieved when producing the thematic map. In other words, it tells us the proportion of pixels in the test dataset that are correctly recognized by the classifier. Producer's accuracy is also a measure of omission error which is the percentage of pixels incorrectly excluded from a particular class that actually belong in.

[1] See Appendix E for detailed information.

H: Bhattacharya distance

Bhattacharya distance is widely used as a measure of statistical class separability because of its analytical form and its relation to the Bayes error. It provides a bound of classification accuracy taking into account first-and second-order statistics. Bhattacharyya distance is the sum of two components, one based primarily on mean differences and the other based on covariance differences.

For two normally distributed classes, the Bhattacharya distance is computed by the equation

$$B = \frac{1}{8}(\mu_2 - \mu_1)^{\mathrm{T}} \left[\frac{\Sigma_1 + \Sigma_2}{2}\right]^{-1} (\mu_2 - \mu_1) + \frac{1}{2} \ln \frac{\mid (\Sigma_1 + \Sigma_2)/2 \mid}{\mid \Sigma_1 \mid^{1/2} \mid \Sigma_2 \mid^{1/2}} \qquad \textbf{(H.1)}$$

where μ_i and Σ_i are the mean vector and covariance matrix of class i, respectively.

The bigger Bhattacharya Distance of the selected bands combination indicates the better separability of the two classes.

For multiclasses, a commonly used method is to calculate the average Bhattacharya distance between them. That is, we calculate the Bhattacharya distance of all the class pairs and get the average. It is defined as:

$$\bar{B} = \sum_{i=1}^{N} \sum_{j=1}^{N} p(w_i) p(w_j) B_{ij} \qquad \textbf{(H.2)}$$

where $p(w_i)$ is the weight for class i, and B_{ij} is the Bhattacharya distance for class i and class j.

CURRICULUM VITAE

Name/Surname: Gülşen TAŞKIN KAYA

Birth of Place/Year: Pazar, 1978

Address: ITU, Informatics Institute, Campus of Ayazaga, 34469, Maslak, İstanbul

B.A. ITU Geomatic Engineering, 2001
MSc. ITU Computational Science and Engineering, 2003

PUBLICATION LIST

Journal Papers

- **Taşkın Kaya, G.**, Ersoy, O.K. and Kamaşak, M.E., 2011. Support Vector Selection and Adaptation for Remote Sensing Classification, *IEEE Transactions on Geoscience and Remote Sensing*, **49**(6), 2071-2079.

- **Taşkın Kaya, G.**, Musaoglu, N. and Ersoy, O.K., 2011. Damage Assessment of 2010 Haiti Earthquake with Post-Earthquake Satellite Image by Support Vector Selection and Adaptation, *Special issue on Haiti Earthquake for American Society for Photogrammetric Engineering*, accepted for publication on October issue.

- Demiralp, M., **Taskin, G.**, 2004. A Matrix Algebraic Analytic Continuation ... A Matrix Algebraic Analytic Continuation Method for Hypergeometric Functions of $p+1$ F p Type, *Applied Numerical Analysis & Computational Mathematics*, **1**, 3-13.

Conference Papers

- **Taskin Kaya, G.,** Kaya, H., and Ersoy O.K., 2011. Change Detection in Very High Resolution Imagery Based on Dynamic Time Warping: An Implementation for Haiti Earthquake Damage Assessment, 6th. International Workshop on the Analysis of Multi-Temporal Remote Sensing Images (MultiTemp), doi:10.1109/Multi-Temp.2011.6005035, pp. 13-16.

- **Taşkın Kaya, G.**, Ersoy, O.K. and Kamaşak, M.E., 2010. Hybrid SVM and SVSA method for classification of remote sensing images, IEEE International Geoscience and Remote Sensing Symposium, pp. 2828-2831 .

- **Taşkın Kaya, G.**, Ersoy, O.K. and Kamaşak, M.E., 2010: Spectral and Spatial Classification of Earthquake Images by Support Vector Selection and Adaptation, International Conference on Soft Computing and Pattern Recognition.

- **Taşkın Kaya, G.**, Ersoy, O.K. and Kamaşak, M.E., 2009. Support Vector Selection and Adaptation and Its Application In Remote Sensing, 4th. International Conference on Recent Advances in Space Technologies (RAST 2009), pp. 408-412.

- **Taşkın Kaya, G.**, Ersoy, O.K. and Kamaşak, M.E., 2009. Support Vector Selection and Adaptation, 2009 International Symposium on Innovations in Intelligent Systems and Applications.

- **Taşkın Kaya, G.**, Ersoy, O.K. and Kamaşak, M.E., 2009. Support Vector Selection and Adaptation For Classification of Earthquake Images, IEEE International Geoscience and Remote Sensing Symposium, pp. II-851-II-854.

- **Taskin, G.,** Demiralp, M., 2005. Random Data Partitioning via Generalized High Dimensional Model Representation for Digital Elevation Models by Using Cubic Spline Interpolation, International Conference of Numerical Analysis and Applied Mathematics 2005 (ICNAAM 2005).

- Demiralp, M., **Taskin, G.,** 2004. Relations Between the Matrix Algebraic Factorized Type Solutions at Different Singular Points for Generalized Hypergeometric Functions of Type $p+1\ F\ p$., International Conference of Numerical Analysis and Applied Mathematics 2004 (ICNAAM 2004).

- **Taskin, G.,** Uskuplu, S., Saygin, H., Ergintav, S., 2003. Optimization of GPS Observation Strategy for Improvement of Tectonic Measurements, The IASTED International Conference on Applied Simulation and Modelling.

- **Taskin, G.,** Saygin, H., Demiralp M., Yanalak, M., 2002. Least Squares Curve Fitting via High Dimensional Model Representation for Digitial Elevation Model, International Symposium on Geographic Information Systems.

- **Taskin, G.,** Saygin, H., Demiralp, M., 2002. Approximation Scheme for Multivariate Hypersurface Fitting Problems, International Symposium on Geographic Information Systems.

Technical Report

- **Taskin Kaya, G.** and Ersoy, O.K., 2009. Support vector selection and adaptation for classification of remote sensing images, Electrical and Computer Engineering, Purdue University, West Lafayette, IN, *TR-ECE-09-2*.