REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CLASSIFICATION OF HYPERSPECTRAL IMAGES WITH ENSEMBLE LEARNING METHODS

Uğur ERGÜL

DOCTOR OF PHILOSOPHY THESIS
Department of Computer Engineering
Program of Computer Engineering

Advisor
Assoc. Prof. Dr. Gökhan BİLGİN

May, 2019

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CLASSIFICATION OF HYPERSPECTRAL IMAGES WITH ENSEMBLE LEARNING METHODS

A thesis submitted by Uğur ERGÜL in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY** is approved by the committee on 21.05.2019 in Department of Computer Engineering, Program of Computer Engineering.

Assoc. Prof. Dr. Gökhan BİLGİN
Yildiz Technical University
Advisor

Approved By the Examining Committee

Assoc. Prof. Dr. Gökhan BİLGİN, Advisor
Yildiz Technical University _____

Assoc. Prof. Dr. Songül VARLI, Member
Yildiz Technical University _____

Assoc. Prof. Dr. Behçet Uğur TÖREYİN, Member
Istanbul Technical University _____

Assist. Prof. Dr. Oğuz ALTUN, Member
Yildiz Technical University _____

Prof. Dr. Mustafa Ersel KAMAŞAK, Member
Istanbul Technical University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Classification of Hyperspectral Images With Ensemble Learning Methods supervised by my supervisor, Assoc. Prof. Dr. Gökhan BİLGİN. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Uğur ERGÜL

Signature

This study was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) Grant No: 2211-A, and Yildiz Technical University, Scientific Research Projects Coordination Department (Project Number: 2016-04-01-DOP03).

Dedicated to my beloved wife
and my family

ACKNOWLEDGEMENTS

Above all, I would like to express my infinite gratitude to my supervisor, Dr. Gökhan BİLGİN, for his valuable guidance and advice. He has been more than an advisor to me. He helped me to stand up for what I think, led me, encouraged me, and increased my hope. Without his support, I would have no doubt lost my sense of direction.

I would additionally like to thank my thesis monitoring committee members, Dr. Songül VARLI and Dr. Ayşe Betül OKTAY, for their valuable suggestions and comments.

I also want to thank the Scientific and Technological Research Council of Turkey, Directorate of Science Fellowships and Grant Programmes (BIDEB) for supporting my doctoral study in the 2211 program.

To my dearest mother, father, and my lovely sister. I want to show my appreciation for their endless support. Their belief in me has always been an outstanding source of encouragement.

Last, to my beloved wife. I do not know how to thank you enough. It has been long and tiring journey, and I could not have succeeded in such a challenging process without your patience, understanding, and continuous love.

Uğur ERGÜL

TABLE OF CONTENTS

LIST OF SYMBOLS

n	Constant Value That Represents Sample Number
d	Constant Value That Represents Dimension Number
i	Index Variable
x_i	i^{th} Input Data
y_i	Label Value of i^{th} Input Data
$\mathbb{R}$	Real Numbers
B	MI Bag, Symmetric Positive Semi-Definite Matrix.
N	Training Sample Pair Number
$\tilde{N}$	Extreme Learning Machine Hidden Layer Node number
$g(.)$	Activation Function of the Hidden Layer of Extreme Learning Machine
$\boldsymbol{w}_i$	Weight Vector Between Input Layer and i^{th} Hidden Layer in an Extreme Learning Machine
$\boldsymbol{\beta}_i$	Weight Vector Between i^{th} Hidden Layer and Output Layer in an Extreme Learning Machine
$\boldsymbol{Y}$	Expected Output Values
$\boldsymbol{H}$	Hidden Layer Output Values in an Extreme Learning Machine
b_i	Bias Value of i^{th} Hidden Layer Node
$\boldsymbol{H}^{\dagger}$	Generalized Inverse of $\boldsymbol{H}$
$\boldsymbol{H}^{T}$	Transpose of $\boldsymbol{H}$
ζ	Constant Value
$\boldsymbol{I}$	Unite Matrix
$K(.)$	Kernel Function
P	Number of Predefined Kernels
η	Kernel Weights

f	Hypothesis Function
C	Regularization Parameter
$\ell(.)$	Error Measurement Function
$\boldsymbol{\alpha}$	Vector of Lagrangian Dual Variables
$\boldsymbol{v}$	Unite Vector, Data Vector
v	Specific Feature Value in MIL
$\boldsymbol{u}$	Data Vector
$\circ$	Hadamard Product
T	Number of Boosting Trials
D_t	Sub-sampling Probability Distribution of t^{th} Boosting Trial
m	Kernel index value
ϵ_t^m	Error Rate for t^{th} Trial and m^{th} Kernel
$sign(.)$	Sign Function
$\Phi(.)$	Feature Mapping Function
λ	Positive Real Value
K_s	Spatial Kernel Matrix
K_ω	Spectral Kernel Matrix
κ	Kappa Statistics Coefficient
θ_1	Agreement Measurement of Kappa
θ_2	Disagreement Measurement of Kappa
F	Contingency Matrix of Kappa
$\boldsymbol{F}_i$	i^{th} Feature in Multiple Decision Tree
μ	Mean Value, Coefficient in HCKBoost Computation
$s_{\bar{d}}$	Standard Deviation of d vector
Z	McNemar's Test Result Value
Z_t	Normalization Factor for t^{th} trial in HCKBoost
Q	McNemar's Test Contingency Matrix
k	Window Width
E	Entropy

$\boldsymbol{S}$	Instances in Training Data Set in MIL
$p(\boldsymbol{S})$	Instances Belong to Positive Bags in MIL
$n(\boldsymbol{S})$	Instances Belong to Negative Bags in MIL
ξ	Slack Variable in Support Vector Machine for MIL
$\boldsymbol{w}$	Normal Vector in Support Vector Machine for MIL
δ	Bias in Support Vector Machine for MIL
$\mathscr{L}$	Log Likelihood for MIL-Boosting
$\boldsymbol{X}^s$	Spatial Data
$\boldsymbol{X}^w$	Spectral Data
P^s	Number of Spatial Kernels
P^w	Number of Spectral Kernels
γ	Very Small Positive Real Value in Calculation of HCKBoost
$\mathcal{O}$	Complexity Function
$C(n)$	Time Complexity
$S(n)$	Space Complexity

LIST OF ABBREVIATIONS

1D	1 Dimensional
2D	2 Dimensional
3D	3 Dimensional
AA	Average Accuracy
AdaBoost	Adaptive Boosting
AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
BHC	Binary Hierarchical Classifier
CART	Classification and Regression Tree
CK	Composite Kernel
CK-ELM	Composite Kernel Extreme Learning Machine
CK-SVM	Composite Kernel Support Vector Machine
ELM	Extreme Learning Machine
EnLe	Ensemble Learning
FN	False Negative
FP	False Positive
HCK	Hybridized Composite Kernels
HK	Hybrid Kernel
HSI	Hyperspectral Image
ICA	Independent Component Analysis
IG	Information Gain
L1-MK-ELM	L1 Norm Multiple Kernel Extreme Learning Machine
L1-MKL	L1 Norm Multiple Kernel Learning
L2-MK-ELM	L2 Norm Multiple Kernel Extreme Learning Machine

L2-MKL	L2 Norm Multiple Kernel Learning
LFDA	Local Fisher Discriminant Analysis
LTF	Logarithmic Transfer Function
KELM	Kernel Extreme Learning Machine
KNN	K-Nearest Neighborhood
KOMP	Kernel Orthogonal Matching Pursuit
KSOMP	Kernel Simultaneous Orthogonal Matching Pursuit
KSVM	Kernel Support Vector Machine
MASR	Multiscale Adaptive Sparse Representation
MCS	Multiple Classifier System
MI	Multiple Instance
MIL	Multiple Instance Learning
milFr	MIL Forest
MKBoost	Multiple Kernel Boosting
MKBoost-D1	First Deterministic Multiple Kernel Boosting
MKBoost-D2	Second Deterministic Multiple Kernel Boosting
MKL	Multiple Kernel Learning
ML	Machine Learning
MWIR	Middle-Wave Infrared
NIR	Near Infrared
OA	Overall Accuracy
PCA	Principal Component Analysis
PTF	Polynomial Transfer Function
RF	Random Forest
ROSIS-03	Reflective Optics System Imaging Spectrometer
RTF	Radial Transfer Function
SimpleMKL	Simple Multiple Kernel Learning
SM1MKL	Soft Margin Multiple Kernel Learning
SWIR	Short-Wave Infrared

SVM	Support Vector Machine
TN	True Negative
TP	True Positive
USA	United States of America

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Classification of Hyperspectral Images With Ensemble Learning Methods

Uğur ERGÜL

Department of Computer Engineering
Doctor of Philosophy Thesis

Advisor: Assoc. Prof. Dr. Gökhan BİLGİN

Hyperspectral imaging is a remote sensing technology that enables the acquisition of hundreds of consecutive bands in high frequencies. Hyperspectral sensors allow to capture images between 10-20nm wavelength intervals by operating in an area called the optic region of the electromagnetic spectrum. The application of this technology is increasing day by day in a number of disciplines including the defense industry, chemistry, forestry, agriculture, urban planning, and medicine.

Developing hyperspectral imaging increases the need for advanced analysis of these images. For this reason, hyperspectral image processing subjects are being processed frequently in the fields of pattern recognition and machine learning. In this thesis, multiple instance learning, multiple classifier systems and kernel methods are emphasized in order to increase classification performance and to perform advanced image analysis. The use of spatial information in the proposed methods is also emphasized. In the proposed multiple instance ensemble learning approach, the use of unlabeled areas on hyperspectral images was provided. Methods such as bagging and random feature subspace selection have been used to increase the classification performance. In this approach, base classifiers such as decision trees, support vector machines, and k-nearest neighbors are used.

Combining more than one kernel methods provides an efficient way to manage data with a compound distribution, such as hyperspectral images. However, the proposed multiple kernel learning methods often require complex optimization procedures. In order to address this issue, a boosting-based ensemble learning method is presented.

Hybrid kernels are taken into consideration together with composite kernels which allow the use of spatial information, and it is aimed to perform advanced hyperspectral image analysis. Although this proposed method has shown high performance, the ratio of hybrid and composite kernels should be determined manually. For this reason, another method called multiple composite kernel extreme learning machine is proposed. In this method, hybrid and composite kernels are presented as an aggregated input, and the weight value of each kernel is determined automatically with an extreme learning machine based optimization algorithm. Since the extreme learning machine allows for multiple classification, the overloading calculation time is avoided and the result is achieved in a less complicated way.

The proposed methods have been tested on hyperspectral images with ground-truth information. Obtained results are compared with state-of-the-art methods in the literature. Both numerical and statistical methods are used in these comparisons. In addition to that, the obtained classification maps are presented together with the experimental results for comparison purposes.

Keywords: Hyperspectral imaging, ensemble learning, multiple instance learning, hybrid kernels, composite kernels, extreme learning machine

ÖZET

Hiperspektral Görüntülerin Topluluk Öğrenme Yöntemleri ile Sınıflandırılması

Uğur ERGÜL

Bilgisayar Mühendisliği Anabilim Dalı
Doktora Tezi

Danışman: Doç. Dr. Gökhan BİLGİN

Hiperspektral görüntüleme yüksek frekanslarda yüzlerce sıralı bant elde edilmesine imkan tanıyan bir uzaktan algılama teknolojisidir. Hipersepktral görüntüleyiciler, elektromanyetik spektrumda optik bölge olarak adlandırılan kısımda çalışarak 10-20nm dalga boyu aralıklarında görüntü elde edilmesini sağlarlar. Bu teknolojinin her geçen gün savunma sanayi, kimya, ormancılık, tarım, şehir planlama, tıp gibi bir çok disiplinde kullanımı artmaktadır.

Gelişmekte olan hiperspektral görüntüleme, beraberinde bu görüntülerin ileri düzeyde analiz edilmesine olan ihtiyacı da artırmaktadır. Bu sebeple örüntü tanıma ve makina öğrenmesi alanlarında hiperspektral görüntü işleme konuları sıklıkla işlenmeye başlanmıştır. Sınıflama başarımlarının artırılması ve gelişmiş görüntü analizinin yapılabilmesi için bu tez çalışmasında çoklu örnek öğrenme, çoklu sınıflayıcı sistemler ve çekirdek yöntemler üzerinde durulmuştur. Önerilen yöntemlerde uzamsal bilginin de kullanımına önem gösterilmiştir. Önerilen çoklu sınıflayıcı topluluk örnek öğrenme yaklaşımında hiperspektral görüntüler üzerinde bulunan etiketsiz alanların da kullanımına olanak sağlanmıştır. Torbalama ve rastsal özellik alt uzayı seçimi gibi yöntemler kullanılmış ve sınıflama başarımının artırılması hedeflenmiştir. Bu yaklaşımda karar ağaçları, destek vektör makineleri, k-enyakın komşu gibi temel sınıflayıcı yöntemler kullanılmıştır.

Birden fazla çekirdek yönteminin bir araya getirilerek kullanılması hiperspektral görüntüler gibi bileşik dağılıma sahip veriler için etkin bir yöntem sunar. Fakat önerilmiş olan çoklu çekirdek öğrenme yöntemleri genellikle karmaşık optimizasyon

prosedürlerine ihtiyaç duyar. Bunun önüne geçmek için artırım (boosting) tabanlı bir topluluk öğrenme yöntemi sunulmuştur. Melez çekirdeklerle beraber uzamsal bilginin kullanımına olanak tanıyan kompozit çekirdekler de işin içine katılmış ve gelişmiş hiperspektral görüntü analizinin yapılması amaçlanmıştır. Melezleştirilmiş kompozit çekirdek artırım adı verilen bu yöntemde temel sınıflayıcı olarak aşırı öğrenme makinası kullanılmıştır. Önerilen bu yöntem her ne kadar yüksek başarımlar vermiş olsa da melez çekirdekler ile kompozit çekirdeklerin oranının manuel belirlenmesi gerekmektedir. Bu sebeple çoklu kompozit çekirdek aşırı öğrenme makinesi ismi verilen bir diğer yöntem önerilmiştir. Bu yöntemde, melez ve kompozit çekirdekler topluca girdi olarak sunulmuş ve aşırı öğrenme makinesi tabanlı optimizasyon algoritması ile her bir çekirdeğin ağırlık değeri otomatik olarak belirlenmiştir. Aşırı öğrenme makinesi çoklu sınıflamaya olanak tanıdığı için hesaplama zamanından tasarruf edilmiş ve daha az karmaşık bir yolla sonuca ulaşılmıştır.

Önerilen yöntemler yer doğrusu bilgisine sahip hiperspektral görüntüler üzerinde test edilmiştir. Elde edilen sonuçlar literatürde bulunan önde gelen yöntemlerle kıyaslanmıştır. Bu kıyaslamalarda hem nümerik hem de istatistiki yöntemler kullanılmıştır. Ayrıca elde edilen sınıflandırma haritaları da karşılaştırma amacıyla deneysel sonuçlarla beraber sunulmuştur.

Anahtar Kelimeler: Hiperspektral görüntüleme, topluluk öğrenme, çoklu örnek öğrenme, melez çekirdekler, kompozit çekirdekler, aşırı öğrenme makinası

1 Introduction

1.1 Literature Review

Hyperspectral remote sensing is part of a new generation of remote sensing technology that operates within the visible wavelength ($0.4-0.7\mu m$), near-infrared wavelength (NIR) ($0.7-1.5\mu m$), short-wave infrared wavelength (SWIR) ($1.5-3\mu m$), middle-wave infrared wavelength (MWIR) ($3-5\mu m$), and long-wave infrared wavelength ($5-14\mu m$) in the electromagnetic spectrum. It also provides hundreds of bands with high resolution [1]. Whereas standard imaging systems work within the visible wavelength and obtain few bands, hyperspectral images (HSIs) contain numerous bands that belong to a wide range of wavelength intervals within $10nm-20nm$. That feature of HSIs enables the sensing of different kind of features that cannot be detected by other imaging systems.

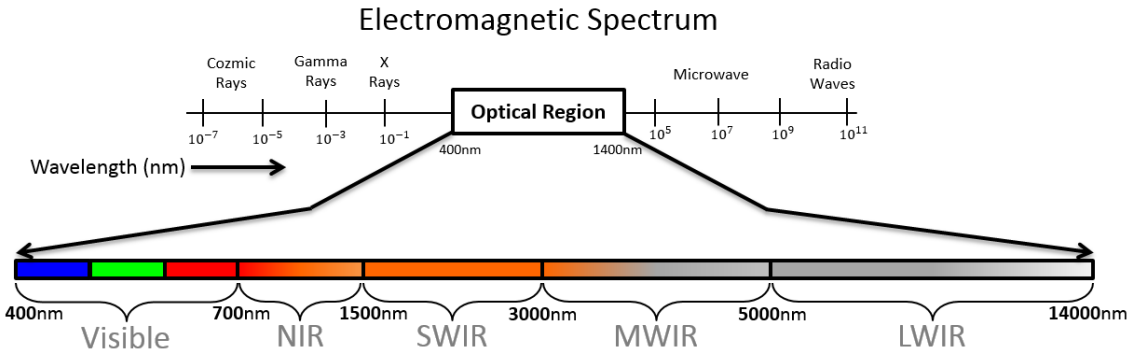

Figure 1.1 Electromagnetic spectrum

Remotely sensed images can be organized into three groups according to their spectral resolutions: panchromatic, multispectral, and hyperspectral. In particular, panchromatic sensors create single band images with respect to the energy level of reflected rays from objects [2]. Such sensors ordinarily operate between the visible and NIR wavelengths in the optic region of the electromagnetic spectrum. By contrast, multispectral sensors usually operate between the visible and SWIR wavelengths, in which they produce images with from four to seven bands. In multispectral images,

each band is obtained at wavelength intervals within $0.3 - 0.4\mu m$ [3], [4].

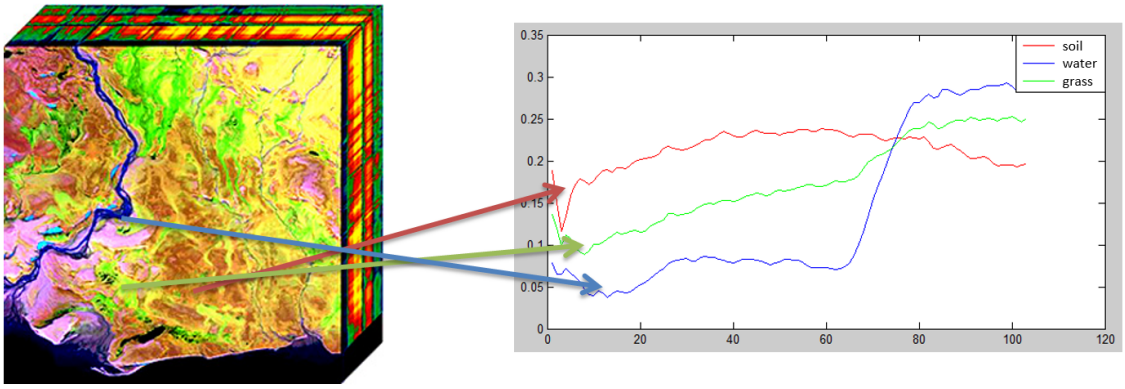

Figure 1.2 Hyperspectral signatures of different objects on the hypercube

By further contrast, hyperspectral imaging can be regarded as an advanced version of multispectral imaging. They contain hundreds of bands obtained at wavelengths between $10nm$ – $20nm$ [5]. The three dimensional (3D) structure of HSIs that presents all dimensions together is known as the *hyperspectral cube* or *hypercube*. Although a hypercube contains spatial information on x (i.e., horizontal) and y (i.e., vertical) dimensions, it also contains spectral information on the complete band slices. Each pixel residing in an HSI has spectral information that varies for the different objects and thus differentiates the objects' reflective features at different wavelengths. The spectral information of HSI pixels is also called the *spectral signature*, because different objects have unique spectral information that can identify them. By extension, hyperspectral signatures have more information than other signatures of images obtained with other kinds of sensors, which affords HSIs high performance in methods of applied machine learning (ML) and pattern recognition.

Hyperspectral remote sensing has wide range of applications. Its capacity to obtain significant information after processing remotely sensed images with rich content for diverse domains has promoted its use in various fields of practice and research. In particular, the ability of hyperspectral optic sensors to operate within infrared wavelengths has transformed means of military defense. To date, hyperspectral sensor systems have been used for target detection, the detection of mined regions, and the identification of different types of military vehicles, [6]. By extension, microscopic hyperspectral imaging has begun to arouse interest in the fields of medicine and biomedicine, in which samples gathered from tissues and organs can now be analyzed to diagnose diseases [7], [8]. At the same time, given the importance of cultivating fresh, high quality products in agricultural and livestock industries, hyperspectral imaging affords an efficient way to detect the deterioration and abnormal conditions of such products automatically [9], [10]. In addition, hyperspectral imaging has been used to determine the quality of soil that supports agricultural activities, to distinguish

arid regions, and to evaluate damage to agricultural products as a result of external factors such as precipitation, disease, and drought.

However, according to the so-called "no free lunch" theorem in ML, no single system can solve all problems while offering superior performance [11]. For example, a classifier may have high accuracy in a particular dataset but perform weakly in another. To partly overcome that problem, ensemble learning (EnLe) methods can be used as statistical paradigms for selecting and fusing the decisions of a group of classifiers in multiple classifier systems. In selection, the most accurate (i.e. strongest) classifier is selected among a group of classifiers; however such a strong classifier might not be preferable, since a weak classifier may be able to produce more accurate results than a strong classifier for a specific dataset. In fusion, a dataset is presented to all classifiers, and the decisions of all classifiers are singularized by various procedures. In terms of EnLe each of those classifiers are known as *weak classifiers*.

EnLe methods not only increase the accuracy of but also provide more robust classifier systems [12]. The error of a classifier system can be calculated via binomial distribution as shown in equation (1.1) [13].

$$P(r) = \binom{T}{r} \epsilon^r (1-\epsilon)^{(T-r)} \tag{1.1}$$

Here T is total number of classifiers, r is number of classifiers to be selected, and ϵ is error rate of the each classifier.In a classifier system consisting of 21 weak classifiers, if each weak classifier's error is 0.3, then the error of the system can also be calculated via binomial distribution as shown in equation (1.2). Since such a system needs at least 11 accurate classifiers to make desired decisions, r should be greater than or equal to 11 [14].

$$P(i \geq 11) = \sum_{i=11}^{21} \binom{21}{i} \epsilon^i (1-\epsilon)^{(21-i)} = 0.026 \tag{1.2}$$

Figure 1.3 illustrates general topologies of ensemble systems, among which parallel architecture has dominated in studies in the literature [15]. In parallel architecture, each weak classifier is trained individually, and the final decision is made according to a specific combination rule. The total number of weak classifiers is usually determined before the classification process begins. By Contrast, in serial architecture, each weak classifier is trained according to the output of the previous classifier and the total number of weak classifiers is usually determined during the classification process [16],

[17].

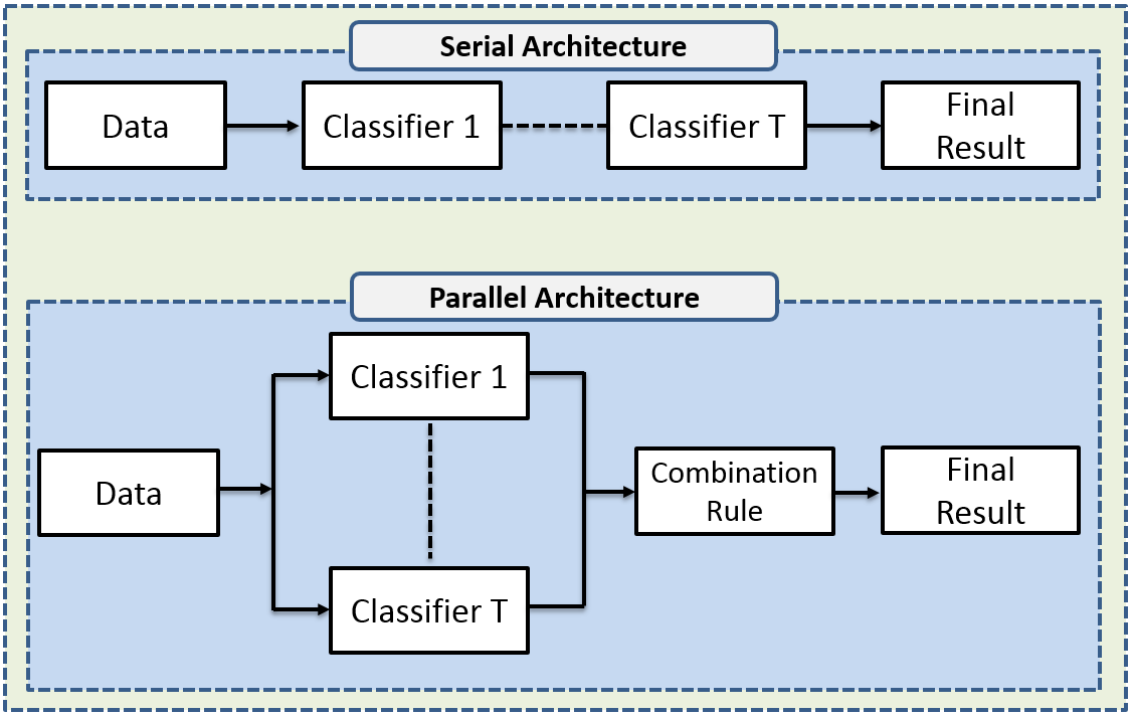

Figure 1.3 Architectures of ensemble systems

In a multiple classifier system (MCS) (i.e. ensemble system), it is desirable to have weak classifiers that are both accurate and disagree about particular parts of the input data. Therefore, the diversity of a system plays an important role in the success of any ensemble classifier. There are two kinds of approaches that ensure such diversity: homogeneous and heterogeneous [15]. In homogeneous approaches, the same kinds of classifiers are trained by feeding them input data that have been manipulated, usually by changing the input samples or features of the samples. In heterogeneous approaches, by contrast, different kinds of classifiers are trained by using the same input data [13], [18].

The manipulation of input data is a method of increasing the diversity of an MCS. Therein, original input data are generally divided into sub-parts, or else some changes are made to the original input data for each weak classifier. Bootstrap aggregation, or bagging [19] is one such method that offers particularly simple implementation as well as good generalization. In bagging, a system's diversity is ensured by sampling a subset of training data by choosing samples randomly from original training data. That new subset is constructed by means of replacement, in which the original data do not change, and every subset is constructed by using those original data. In relatively small training samples, bagging provides an appealing method in which training subsets are drawn at a high percentage from the original training data. Given the likelihood that each constructed subset will contain mostly overlapping data, an ensemble system

might not have satisfactory diversity. In response, unstable classifiers such as decision trees and neural networks may be used such that, variant decision boundaries can be obtained for minor changes [20].

Manipulating input features is another method of increasing the diversity of a system. In that method, the original sample size of training data is preserved although features of the input space are randomly selected with a replacement for each weak classifier. The random subspace method [21] is another well-known approach of feature manipulation for ensuring diversity that is not only simple but also affords surprisingly high performance. Selected features in the training phase of random subspace should be retained for classification phase. Random subspace is a useful method for preventing overfitting during classifier design. Although decision tree-based [22] and support vector machine (SVM)-based classifiers [23] have been employed with random subspace to obtain remarkable results, the random subspace method is not applicable for datasets with few dimensional feature spaces. Manipulation of both input data samples and input spaces allow the creation of random forests (RFs) [22], so called because decision trees are utilized as weak classifiers that take randomly selected samples and feature subspaces into account. RFs exhibit characteristics of bagging as well as random subspace methods but outperform both of those approaches.

An important development in multiple classifier systems has been the use of boosting—that is, a set of operations that convert individual weak classifiers into strong decision makers [24]. As in bagging, instance resampling is performed in boosting to ensure the diversity of a system. Unlike bagging, however, resampling in boosting is performed in a way that can identify more informative instances. The original boosting method proposes the creation of three weak classifiers. For the first classifier (C_1), the training process is performed with randomly selected samples without replacement. After obtaining the model C_1, another two-part training subset is created from the original input data by having the desired size of input data passed through C_1. Each part is of equal size; the first contains data samples that are correctly classified by C_1, whereas the second contains data samples that are misclassified by C_1. Thereafter, the second weak classifier (C_2) is trained by using that dataset. Last, the final weak classifier (C_3) is constructed by using data samples about which C_1 and C_2 disagree. Ultimately, the final ensemble decision is made by way of majority voting.

Proposed in 1997, Adaptive Boosting (AdaBoost) is a kind of generalized version of boosting [25]. Since AdaBoost can accommodate both multiclass classification and regression problems, it has garnered considerable interest. AdaBoost iteration commences with subsampling training data from original data in which all samples

have initially equal probability distribution. During boosting trials, as the probabilities of misclassified samples increase, hard-to-classify samples are drawn for each weak classifier trial. AdaBoost uses the individual hypothesis of each weak classifier to obtain results; more specifically, it combines each hypothesis via weighted majority voting in order to reach the final hypothesis. Weighted majority voting is an intuitive method in which the right to decide is not equal for each weak classifier. If a weak classifier shows high performance in the training phase, then the effect of that classifier's decision on the final result will be high. Conversely, if a classifier shows weak performance in the training phase, then the weight of the decision of that classifier will decrease. Freund and Schapire have demonstrated that the training error of the AdaBoost ensemble is limited in the following equation:

$$E < 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t (1-\epsilon_t)} \tag{1.3}$$

Where E is the training error of ensemble, ϵ_t is training error of t^{th} weak classifier, and T indicates the total number of weak classifiers [25]. In AdaBoost, the training error of weak classifiers usually yields values that are too small after a few iterations since $\epsilon_t < 1/2$. In other methods, that situation causes overfitting, in which a classifier learns not only training data but also the noise of training data in excess. Consequently, the classifier memorizes and performs well with the training data but performs dismally with testing data. However, AdaBoost does not suffer from overfitting, as Schapire et al. have explained as part of margin theory [26], according to which an ensemble error linked to the margin created from the classifier system decreases as the margin increases. In AdaBoost, the margin of an instance is defined as the difference between the total votes of correct classifiers and the maximum votes of any incorrect classifier. Owing to its increasing margin capacity, the error of AdaBoost is typically slight.

Since any multiple classifier system should ideally be more diverse and have more accurate weak classifiers, another method known as rotation forest has been proposed [27] in which each weak classifier is constructed by using rotated features. Training space rotation can be accomplished by way of various feature extraction techniques, including principal component analysis (PCA), independent component analysis (ICA), and local fisher discriminant analysis (LFDA) . Rotation forest can complete multiclass classification tasks because it uses decision trees as weak classifiers. In the beginning of the training process, the feature space of a training dataset is divided into K subsets created to be either disjoint or intersecting, the former of which has been recommended by authors when maximizing the diversity of a system is the goal [27]. After that, for each feature subset, an instance subset is created using

bootstrap aggregation with a 75% subsampling percentage. In that step, it is important to ensure that the instance subset contains instances from all classes. Afterward, for each bootstrapped subset, feature extraction is performed, and a sparse rotation matrix constructed via eigenvectors is used for both training and testing samples in the classification process. A decision tree classifier can be built with that rotation matrix, via which a test sample is rotated and passed through the classifier. It is reported that the rotation forest method produces more diverse and accurate results than bagging [27].

Applications of EnLe methods on HSIs have shown better results than standard ML methods [15]. Among them, SVM has been widely used as a weak learner for EnLe methods on HSIs, and a particular SVM-based EnLe method has been proposed that combines spectral, structural, and semantic features [28]. That SVM-based EnLe approach outperforms multi-feature SVM methods such as vector stacking, feature selection, and composite kernels. Another EnLe approach [29] that initially splits original hyperspectral data into a few data sources according to the similarity of spectral features also uses SVM to obtain results. An ensemble algorithm that combines a mixture of Gaussian functions and support cluster machine models for classification has additionally been proposed to deal with insufficient numbers of training samples and the misrepresentation of the real distribution of the entire data space [30]. As demonstrated by Waske et. al. [31], SVM ensembles based on random feature selection have shown significant improvement compared to single SVM and RFs. In other work, the application of the random subspace method using SVM ensembles as weak classifiers revealed that noisy features or outliers have a reduced effect on multiple classifier systems [32]. Random subspace optimization based on a genetic algorithm has additionally been proposed that has shown better performance than SVM ensemble methods based on standard random subspace [33]. Moreover, an adaptive boosting-based multiple classifier system has been proposed to eliminate problems caused by insufficient training samples and spectral band redundancies [34]. In that method, base SVM learners are trained after removing redundant and uninformative bands. In another method that tries to suppress limitations caused by insufficient training samples, a multiple kernel learning (MKL) framework is used that applies a boosting strategy [35]. Stump functions have also been used as weak classifiers in an EnLe method for HSI classification [36], and bootstrap aggregation without involving replacement has been employed to enhance the stability and accuracy of the AdaBoost process. A kernel-based random feature subspace ensemble technique has also been proposed for HSIs [37], in which it was aimed to combine subdecisions by optimizing both the base classifier's (i.e., SVM's) hyperplane and corresponding weights of the subdecisions. Applying an SVM-based transfer learning

boosting strategy to HSIs [38] can be performed by labeling the pixels in the target image manually and re-weighting instances from the source image in the training set.

Since HSIs typically have high feature space, multiple classifier systems based on random feature subspace selection (e.g., RFs) are regarded as being more attractive and more robust against overfitting. An RF approach with embedded feature selection and a Markov random field has been reported as being highly compatible with HSIs [39]. In another RF-based EnLe method [40], a binary hierarchical multiclassifier system is implemented to improve the performance of generalization. Another method of adaptive random feature selection with a binary hierarchical classifier (BHC) returned results thought to be more accurate than those achieved by classical classification and regression trees. Such work has been extended to transfer learning by using binary hierarchical classifiers [41] in which extracted information from existing labeled data is leveraged to test data. It is useful to apply that approach when no labeled data are available instead of directly applying the original classifier. RF-based HSI classification can be performed to detect harmful plants in ecosystems [42]. In that work, two methods capable of unsupervised classification—BHC and classification and regression tree (CART)—based approaches representing two different RF-based approaches were compared in HSI analysis [43]. An object-based hyperspectral classification is proposed that involves steps of multiresolution segmentation (MRS) and RF classifier (RFC) [44]. In the method, when classes starkly differ from each other, some segmentation errors may occur; accordingly, the scale of segmentation needed to be calibrated at an appropriate level. Recently, decision making methods based on deep learning have gained popularity; however, such learning models require a great deal of training samples in order to tune abundant parameters, which requires an exceptionally high capacity for calculation in order to extrapolate. To overcome such adversities, a densely connected deep RF has been proposed [45]. Since rotation forests [27] operate as a special kind of RF ensemble learners, their application in HSIs has been reported outperform other sorts of EnLe methods such as AdaBoost, RF, and bagging [46]. Authors who have proposed the application of rotation forest on HSIs have also proposed using Markov random fields [47] and extended morphological map features [48] for exploiting advanced spatial features together with multiple classifier systems.

1.2 Objective of the Thesis

Since a variety of materials reside within HSIs obtained by remote sensors at long distances, inferences need to be made for different applications of HSIs. However, for experts working in fields involving the use of HSIs, it remains difficult to manually

obtain meaningful results, because each material constitutes different pixel patterns. Although that circumstance facilitates the use of ML methods, distorted signals, format errors, and spectral clutters make HSIs incredibly challenging to work with. Moreover, mislabeling and insufficiently labeled samples are frequently encountered. To eliminate those adversities, advanced techniques remain necessary.

Using spatial information along with spectral information is highly significant in HSI analysis. In any HSI, a pixel of remotely sensed image may cover from $1m^2$ to $10m^2$ depending on the distance and resolution of the sensor. Thus, a pixel may contain more than one object. Moreover, it is often possible that neighboring pixels consist of similar materials and thus naturally have similar spectral signatures. Consequently, contextual information means far more than other kinds of information in 1D or 2D datasets used in ML. For that reason, the spatial feature utilization of HSIs play an important role in classification.

Remotely sensed HSI data are usually composed of compound distribution, which in some circumstances reduces the linear separability of the data. When using spectral classifiers, the success of the classifier heavily depends on the distribution of data. To increase linear separability, kernel methods should be employed; however, the selection of the proper kernel method continues to be a problem that needs to be addressed. In most cases, a classifier exhibiting good performance with training data might not perform well with unseen testing data. At the same time, different classifiers might show divergent performance when generalizing the same training and testing data, and such a high volume of data might be too complex to solve for a single classifier. Therefore, multiple classifier systems also play important role in classification.

Because the aim of the research reported in this thesis was to incorporate spatial and spectral information jointly, spatial feature extraction procedures were used. Another aim was to obtain classes within a reasonable level of separation. Since HSI data usually exhibit compound distribution, different kinds of kernel methods were taken into account during analysis such that linearly separable class distribution could be achieved. Moreover, to obtain a more generalized classification performance, different kinds of multiple classifier systems are proposed, which should provide less complex, more stable classifier systems. In that way, complicated HSI data analysis can be simplified for many kind of applications in diverse fields of research and practice.

1.3 Hypothesis

Remote sensors located on the bodies of airplanes, satellites, and other kinds of aircraft capture images from long distances, often with very large areas. A single pixel in an HSI may also represent a fairly large area. Consequently, supervised as well as unsupervised ML methods may not succeed in assigning a pixel to a class or cluster. Instead of using those kinds of approaches based exclusively on spectral signatures, the use of neighboring information may improve the success of those methods. Compared to unlabeled areas, most labeled ones within HSIs occupy little space. Therefore, a great deal of valuable data remain unused during supervised classification applications. Even labeled samples can cause uncertain class labeling problems because of ground-truth labeling by humans and format errors. Since all of those situations negatively affect standard ML algorithms, consolidating uncertain ground-truth information and using unlabeled areas during supervised learning might yield better results. To that end, multiple-instance learning (MIL) seems to be the most appropriate method, for it allows working with uncertain target information. As a result, applying MIL in HSI analysis should afford similar improvements.

The success of any ML algorithm depends on the separation ability of the algorithm. Since HSI data usually contain intertwined class distribution, it remains difficult to generate models with a high capacity for linear separation. At that stage, kernel methods facilitate data transformation from original input space to higher or even infinite-dimensional Hilbert space, which might increase a classifier's ability to perform linear separation. Although kernel methods increase linear separability, transforming whole data to kernel space is an expensive process. Therefore, some special kernel based methods that take advantage of the so-called "kernel trick" have been proposed, including SVMs and extreme learning machines (ELMs). Kernel versions of those methods, kernel ELM (KELM) and kernel SVM (KSVM), provide elegant ways to facilitate kernel space transformation at a reasonable computational cost. Although applying KELM and KSVM with HSI should increase the success of classification, some kernel transfer functions show superior generalization, whereas others show a strong learning ability. Since having both with a single kernel may be impossible, combining multiple kernels could represent an approach to increase both the generalization and learning ability of the kernel classifier. In the same sense, hybridizing different kernels should provide improved kernel based classification for HSIs.

A single classifier may perform in a desired way with training data, although that classifier's performance may not always be as high as with unseen data. Likewise, different classifiers may achieve different kinds of success with the same data. For those reasons, ensemble classifiers might provide more robust classification models.

At the same time, some proposed approaches such as the feature subspace method could provide a more efficient means for processing high volumes of data, for which divide-and-conquer methods might be used with less complexity. Accordingly, using multiple classifier systems may significantly improve HSI classification, since using spatial information, kernel methods, hybrid kernels (HKs), and ensemble methods can enhance HSI analysis, each in their own way. Combining those abilities in a single classifier system should yield superior improvements.

2
Theoretical Background

In this chapter, some theoretically related concepts of the proposed works are briefly described. First of all, multiple instance learning method is introduced. Afterwords, extreme learning machine, multiple kernel learning, and multiple kernel boosting (MKBoost) methods are explained respectively. Finally, hybrid kernels (HKs) and composite kernels (CKs) are described.

2.1 Multiple Instance Learning

In conventional supervised machine learning methods, all training samples in a dataset contain a particular label and sample/label pairs are used to construct classifier models. However, in many real world applications, whole training samples may not have a corresponding label information. In addition to having a large amount of data, only small portion of the data found as labeled just as in the hyperspectral images. In multiple instance learning, sample/label pair ambiguity is tried to be handled. More specifically, instead of assigning a specific label to each training sample, labels are assigned to group of training samples which are called "bag" in MIL terminology.

MIL is a learning paradigm that is first introduced by Dietterich et al. [49] for drug discovery. This problem is inspired from the shape of molecules which are found in different shapes for same molecules. Thus, ambiguity arises for the molecules that actually belong to same class but have different properties. Therefore, it is considered that a bag is positive if at least one positive labeled sample resides in that bag. A MIL bag is considered as negative on the opposite case. Obviously, only binary classification is supported in MIL algorithms.

In standard machine learning methods it is straightforward to represent instance/label pairs. Such that, supposing that we have n training samples $(x_1, x_2, ..., x_n)$ where $x_i \in \mathbb{R}^d$ and n labels $(y_1, y_2, ..., y_n)$ corresponding to each sample where $y_i \in \{0, 1\}$. It is aimed to train a supervised model and classify each unknown test samples according

to trained model. On the other hand, classifier model training phase is accomplished by MI bags. An MI bag (B) gets positive label ($Y_B = 1$) if it has at least one positive sample and it gets negative label ($Y_B = -1$) if it does not have any positive labeled sample.

2.2 Extreme Learning Machine

In a single hidden layer feed forward neural network architecture that contains N training sample pairs ($\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$), $\boldsymbol{x}_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in \mathbb{R}^n$ represents instances that exist in n dimensional space and $\boldsymbol{y}_i = [y_{i1}, y_{i2}, ..., y_{im}]^T \in \mathbb{R}^m$ shows m possible output nodes of the instances. A neural network structure that contains $\tilde{N}$ hidden layer node can be expressed as in equation (2.1).

$$\sum_{i=1}^{\tilde{N}} \boldsymbol{\beta_i} g(\boldsymbol{w}_i.\boldsymbol{x}_j + b_i) = \boldsymbol{o}_j, j = 1, ..., N \tag{2.1}$$

In this expression, $g(.)$ refers to activation function of the hidden layer. $\boldsymbol{w}_i = [w_{i1}, w_{i2}, ..., w_{in}]^T \in \mathbb{R}^n$ is weight vector between input layer and i^{th} hidden layer and $\boldsymbol{\beta_i} = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T \in \mathbb{R}^m$ vector denotes the weights between i^{th} hidden layer and output layer nodes. b_i is bias value of i^{th} hidden layer node. Stated equation can be rewritten in matrix format as in equation (2.2) in which $\boldsymbol{Y}$ indicates the expected output values.

$$\boldsymbol{H\beta} = \boldsymbol{Y} \tag{2.2}$$

$\boldsymbol{H}$ denotes hidden layer output values and can be rephrased in matrix form with respect to $\boldsymbol{\beta_i}$, $\boldsymbol{w}$, b_j parameters as in equation (2.3).

$$\boldsymbol{H} = \begin{bmatrix} g(\boldsymbol{w}_1.\boldsymbol{x}_1 + b_1) & ... & g(\boldsymbol{w}_{\tilde{N}}.\boldsymbol{x}_1 + b_{\tilde{N}}) \\ ... & ... & ... \\ g(\boldsymbol{w}_1.\boldsymbol{x}_N + b_1) & ... & g(\boldsymbol{w}_{\tilde{N}}.\boldsymbol{x}_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \tag{2.3}$$

$\boldsymbol{\beta}$ and $\boldsymbol{Y}$ are shown in equation (2.4). According to ELM theorem [50], [51] training error is minimized together with the output norm ($min : \|\boldsymbol{H\beta} - \boldsymbol{Y}\|^2 \quad and \quad \|\boldsymbol{\beta}\|^2$)

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ ... \\ \boldsymbol{\beta}_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad and \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{Y}_1^T \\ ... \\ \boldsymbol{Y}_N^T \end{bmatrix}_{N \times m} \tag{2.4}$$

In Moore-Penrose generalized matrix inverse theorem [52] for an equation system such as $\boldsymbol{H\beta} = \boldsymbol{Y}$, inverse of a non-square matrix providing minimum norm and least

square solutions can be expressed as in equation (2.5).

$$\boldsymbol{\beta}^* = \boldsymbol{H}^{\dagger}\boldsymbol{Y} \tag{2.5}$$

Different methodologies are used for Moore-Penrose generalized inverse operation. One of them is orthogonal projection method and it has two kind of usage: 1) $\boldsymbol{H}^{\dagger} = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T$ if $\boldsymbol{H}^T\boldsymbol{H}$ is not singular and 2) $\boldsymbol{H}^{\dagger} = \boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T)^{-1}$ if $\boldsymbol{H}\boldsymbol{H}^T$ is not singular. According to Ridge regression theorem [53], adding a positive value to the diagonal of the $\boldsymbol{H}^T\boldsymbol{H}$ or $\boldsymbol{H}\boldsymbol{H}^T$ matrices allows to have more stable results and higher generalized performance [51]. Equation in the (2.5) can be expanded as in (2.6), where ζ refers a constant variable and $\boldsymbol{I}$ is a unit matrix.

$$\boldsymbol{\beta}^* = \boldsymbol{H}^T(\frac{\boldsymbol{I}}{\zeta} + \boldsymbol{H}\boldsymbol{H}^T)^{-1}\boldsymbol{Y} \tag{2.6}$$

After obtaining weights between hidden layer and output layer ($\boldsymbol{\beta}^*$), hidden layer output values are calculated for test data $\boldsymbol{x}_t$ as in equation (2.7).

$$h(\boldsymbol{x}_t) = g(\boldsymbol{W}^T\boldsymbol{x}_t + \boldsymbol{b}) \tag{2.7}$$

In order to finalize the classification task, $\boldsymbol{\beta}^*$ is utilized as shown in equation (2.8).

$$f(\boldsymbol{x}_t) = h(\boldsymbol{x}_t)\boldsymbol{\beta}^* = h(\boldsymbol{x}_t)\boldsymbol{H}^T(\frac{\boldsymbol{I}}{\zeta} + \boldsymbol{H}\boldsymbol{H}^T)^{-1}\boldsymbol{Y} \tag{2.8}$$

$\boldsymbol{\beta}$, weights between hidden layer and output layer, is calculated analytically during the training phase. However, if it is not needed to be known, $h(\boldsymbol{x}_t)\boldsymbol{H}^T$ and $\boldsymbol{H}\boldsymbol{H}^T$ dot products can be transferred to a kernel function as shown in (2.9).

$$K(\boldsymbol{x}_t, \boldsymbol{x}^T) = \begin{bmatrix} K(\boldsymbol{x}_t, \boldsymbol{x}_1^T) \\ \dots \\ K(\boldsymbol{x}_t, \boldsymbol{x}_N^T) \end{bmatrix} \quad and \quad \boldsymbol{\Omega}_{ELM} = K(\boldsymbol{x}_i, \boldsymbol{x}_j^T) \tag{2.9}$$

Thus, final form of kernel based ELM (KELM) can be expressed as in equation (2.10).

$$f(\boldsymbol{x}_t) = K(\boldsymbol{x}_t, \boldsymbol{x}^T)(\frac{\boldsymbol{I}}{\zeta} + \boldsymbol{\Omega}_{ELM})^{-1}\boldsymbol{Y} \tag{2.10}$$

2.3 Multiple Kernel Learning

MKL algorithms aim to find the optimal combination of P predefined kernels $\{K_m : \boldsymbol{R}^{d_m} \times \boldsymbol{R}^{d_m}\}_{m=1}^{P}$. Linear or non-linear functions may be used to combine kernels in an efficient way. It is reasonable to take linear combination methods into consideration

when using complex kernel functions like radial base functions for the purpose of reducing complexity [54]. Weighted sum for multiple kernel constitution is formulated as a linear function in (2.11):

$$K_\eta = f_\eta(\{K_m\}_{m=1}^P|\eta) = \sum_{m=1}^{P} \eta_m K_m \tag{2.11}$$

where η denotes the kernel weights and usually used as convex combination ($\eta > 0$ and $\sum_{m=1}^{P} \eta_m = 1$) to maintain positive-definiteness.

Hypothesis function ($f \in H_K$) learned from a kernel classifier is desired to produce minimum norm and error and this can be achieved by a maximum margin optimization task as shown in equation (2.12):

$$\min_\eta \min_f = \frac{1}{2}\|f\|_{H_K}^2 + C\sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i)) \tag{2.12}$$

where C is regularization parameter and ℓ is error measurement function for a given specific input value $\boldsymbol{x}_i$ via f hypothesis. This optimization can be expressed in dual formulation as in equation (2.13).

$$\min_\eta \max_{\boldsymbol{\alpha}} = \left\{ \boldsymbol{\alpha}^T \boldsymbol{\nu} - \frac{1}{2}(\boldsymbol{\alpha} \circ \boldsymbol{y})^T \left(\sum_{m=1}^{P} \eta_m K_m \right) (\boldsymbol{\alpha} \circ \boldsymbol{y}) \right\} \tag{2.13}$$

In this Lagrangian equation, $\boldsymbol{\alpha}$ expresses a vector that corresponds to dual variables for each separation constraint, $\boldsymbol{\nu}$ stands for a vector with all elements being one, and $\circ$ is Hadamard product.

2.4 Multiple Kernel Boosting

Adaptive boosting (AdaBoost) is a popular ensemble learning algorithm that combines decisions of base learners through weighted majority voting [25], [55]. In AdaBoost, input data distribution is updated adaptively with respect to previous misclassified samples. Thus, samples more difficult to classify (i.e. probably more informative) will have more influence on the system.

MKBoost idea is heavily based on boosting technique to train a classifier with multiple kernels. MKBoost runs for T boosting trials to train f_t kernel classifiers ($t = 1, ..., T$). At each boosting round, firstly n instances are sub-sampled from training data set according to D_t distribution. After sub-sampling operation, two deterministic approaches are introduced to train a kernel-based classifier f_t and these are named

as MKBoost-D1 and MKBoost-D2 respectively. In MKBoost-D1, one classifier f_t^m is trained with each kernel K_m from the P kernel collection using SVM method. Each f_t^m classifiers' performance with K_m is measured for whole training data over D_t distribution as in equation (2.14).

$$\epsilon_t^m = \epsilon(f_t^m) = \sum_{i=1}^{N} D_t(i)\left(f_t^m(\boldsymbol{x_i}) \neq y_i\right) \tag{2.14}$$

Final classifier for t^{th} boosting round is built by choosing best classifier according to performance measure as in (2.15).

$$f_t = \arg\min_{f_t^m}\left(\epsilon(f_t^m)\right) \tag{2.15}$$

It is clear that the MKBoost-D1 is only interested with the best one among P kernel classifiers and rest of them is discarded. In MKBoost-D2, contribution of all P kernel classifiers are calculated for t^{th} round as in equation (2.16):

$$f_t(\boldsymbol{x}) = sign\sum_{m=1}^{P}\left(\alpha_t^m f_t^m(\boldsymbol{x})\right) \tag{2.16}$$

where α is referred to as coefficient of f and calculated with the inverse ratio of the misclassification measure. Thus, contribution of more successful classifiers are kept high accordingly.

Rest of the steps are similar with the AdaBoost computation for both deterministic approaches. The misclassification rate ϵ_t for the combined classifier f_t is computed over D_t distribution on whole training data. In the last step of each boosting round, distribution weights are updated for the next trial as in the regular AdaBoost.

It is reported that the MKBoost approach gives better results compared to some state of the art MKL methods such as Subgradient Descent MKL, Semi-Infinite Linear Programming, Lp-Norm MKL [56]. Two stochastic MKBoost methods (MKBoost-S1 and MKBoost-S2) are introduced along with deterministic ones. Stochastic methods are claimed to reduce computation time. However, deterministic methods produce more accurate results than stochastic approaches. Therefore, MKBoost-D1 and MKBoost-D2 are utilized for comparison in the experimental design and results section.

2.5 Hybrid Kernels

In compliance with the general concept of pattern recognition, data taking place in higher feature space have more linear separability. Thus, it is desired to have linearly more separable data by transferring it into high dimensional space from low dimensional space via K transfer function ($K : \boldsymbol{R}^d \rightarrow \boldsymbol{F}; \boldsymbol{x}_n \rightarrow K(\boldsymbol{x}_n)$). Transferring data individually into a high-dimensional domain is a high-cost job. In order to reduce computational load and storage area, a method called "kernel trick" is used. This method takes dot products results' into account instead of data itself. Kernel functions utilized in this thesis are radial transfer function (RTF), polynomial transfer function (PTF), and logarithmic transfer function (LTF). These functions can be seen in equations (2.17), (2.18), and (2.19) respectively.

$$\boldsymbol{RTF} : K(\boldsymbol{u},\boldsymbol{v}) = exp\Big(-\frac{\|\boldsymbol{u}-\boldsymbol{v}\|^2}{2\sigma^2}\Big) \tag{2.17}$$

$$\boldsymbol{PTF} : K(\boldsymbol{u},\boldsymbol{v}) = (\gamma(\boldsymbol{u}-\boldsymbol{v})+r)^d, \gamma > 0 \tag{2.18}$$

$$\boldsymbol{LTF} : K(\boldsymbol{u},\boldsymbol{v}) = -\log\big(\|\boldsymbol{u}-\boldsymbol{v}\|^d + 1\big) \tag{2.19}$$

Defining a kernel function for transferring data to high dimensional space is made with respect to Mercer's Theorem [57]. According to this theorem; a gram matrix or any finite subset of this matrix should be positive semi-definite as a result of transferring training data to $\boldsymbol{R}$ domain via K kernel function. This theorem plays a key role during the constitution of kernel based classifiers in order to acquire global solution. Following functions are valid according to Mercer's Theorem:

1-)$K(\boldsymbol{u},\boldsymbol{v}) = K_1(\boldsymbol{u},\boldsymbol{v}) + K_2(\boldsymbol{u},\boldsymbol{v})$

2-)$K(\boldsymbol{u},\boldsymbol{v}) = \lambda K_1(\boldsymbol{u},\boldsymbol{v})$

3-)$K(\boldsymbol{u},\boldsymbol{v}) = K_1(\boldsymbol{u},\boldsymbol{v})K_2(\boldsymbol{u},\boldsymbol{v})$

4-)$K(\boldsymbol{u},\boldsymbol{v}) = f(\boldsymbol{u})f(\boldsymbol{u})$

5-)$K(\boldsymbol{u},\boldsymbol{v}) = K_3(\Phi(\boldsymbol{u}),\Phi(\boldsymbol{v}))$

6-)$K(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{u}^T\boldsymbol{B}\boldsymbol{v}$

K_1 and K_2 are positive semi-definite kernels $K : \boldsymbol{X} \times \boldsymbol{X} \rightarrow \boldsymbol{R}^d$, $f(.)$ is a real-valued function over X, $\Phi(.)$ is a feature mapping function on n dimensional space, K_3 is a kernel over $\boldsymbol{X}^n \times \boldsymbol{X}^n$, λ is positive real value, and $\boldsymbol{B}$ is a symmetric positive semi-definite matrix.

Having both superior generalization performance and strong learning ability with only

single kernel function may not be practically possible. While some kernels like RTF are locally effective and have strong learning ability, some kernels like PTF are globally effective and have high generalization performance. Mixing/Hybridizing different kind of kernel functions is an approach applied in the literature for the purpose of increasing both generalization performance and learning ability of the kernel classifier [58], [59], [60].

The most common hybrid kernel constitution method is convex combination method and can be shown in (2.20).

$$K_H(\boldsymbol{u},\boldsymbol{v}) = \lambda K_a(\boldsymbol{u},\boldsymbol{v}) + (1-\lambda)K_b(\boldsymbol{u},\boldsymbol{v}) \tag{2.20}$$

K_a and K_b are different kernel functions. Using direct summation Mercer's condition (first Mercer's condition above) and multiplication by a constant coefficient (second Mercer's condition above), K_H ensures Mercer's theorem subject to $0 < \lambda < 1$ constraint.

2.6 Composite Kernels

In a HSI, a pixel is usually correlated with neighbor pixels. Thus, it is desired to produce joint spatial-spectral combination during HSI classification. For this purpose, local spatial feature extraction based composite kernels (CKs) are used to exploit the information that have high-separating ability. Given a pixel entity $\boldsymbol{x}_i \in \mathbb{R}^N$ is denoted as spectral feature $\boldsymbol{x}_i^{\omega} \in \mathbb{R}^{N_\omega}$. Spectral feature is exactly same as a pixel which reflects continuous spectral band characteristic. Spatial features $\boldsymbol{x}_i^{s} \in \mathbb{R}^{N_s}$ are extracted from a defined small area by using spectral signatures. Simple contextual feature extraction methods such as mean and standard deviation are the most common and high-yielding techniques.

Stacked features approach, direct summation kernel, weighted summation kernel, and cross-information kernels are proposed kernel combination methods in [61]. Weighted summation kernel combination method is presented as a superior technique compared to others and can be seen as in equation (2.21):

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \lambda K_s(\boldsymbol{x}_i^{s},\boldsymbol{x}_j^{s}) + (1-\lambda)K_\omega(\boldsymbol{x}_i^{\omega},\boldsymbol{x}_j^{\omega}) \tag{2.21}$$

where K_s and K_ω symbolize spatial and spectral kernel matrices respectively. The λ is a positive constant value ($0 < \lambda < 1$) which determines contribution of spatial and spectral information to the composed kernel.

3
Hyperspectral Datasets and Validation Methods

In this chapter, utilized hyperspectral datsets and validation methods are described. In section 3.1, AVIRIS Indian Pines, ROSIS-03 Pavia University, and AVIRIS Salinas datasets are introduced. Afterwords in section 3.2, overall accuracy (OA), Kappa statistics, t-test, and McNemar's test are explained respectively.

3.1 Hyperspectral Datasets

3.1.1 AVIRIS Indian Pines

AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) Indian Pines hyperspectral scene was captured over Northwest Indiana, USA, in June 1992. The scene consists of 145 rows/scene, 145 pixels/row. Indian Pines HSI contains 224 spectral bands with a wavelength range of 400 - 2500 nm. Band number is reduced to 200 after the removal of water absorption caused noisy bands. Ground-truth is available and contains sixteen different classes. Some of the ground-truth labels occupy very small spaces and they are not convenient for ensemble learning process. Therefore, nine classes are selected to operate ensemble formation process. The selected classes are corn-notill, corn-mintill, grass-pasture, grass-trees, hay-windrowed, soybean-notill, soybean-mintill, soybean-clean, and woods. These nine classes have 9345 labeled samples in total. Indian Pines HSI's 25th band in gray scale and ground-truth map with nine classes are shown in Figure 3.1. The class names and class sequence numbers with the corresponding number of labeled samples are given in Table 3.1.

3.1.2 ROSIS-03 Pavia University

The ROSIS-03 (Reflective Optics System Imaging Spectrometer) Pavia University hyperspectral scene was obtained over the Pavia University area in Italy by the Deutsches Zentrum für Luft- und Raumfahrt (DLR, German Aerospace Center). Pavia University HSI consists of 115 spectral bands with variable wavelengths between 430 - 860 nm and has 610 rows/scene, 340 pixels/row. The original band number is reduced

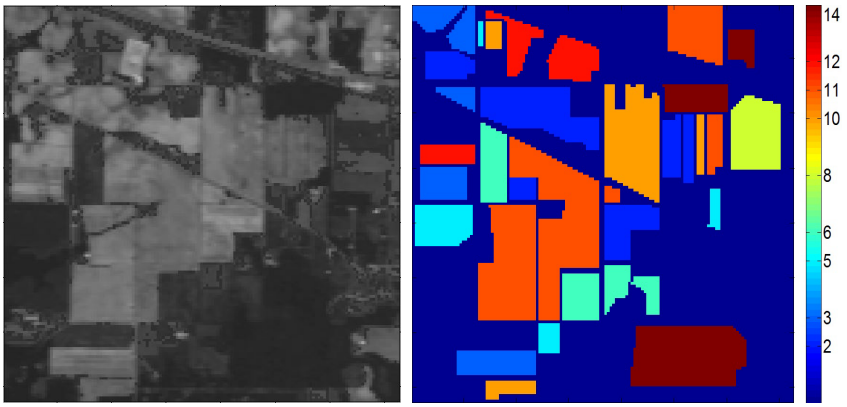

Figure 3.1 (a) Image of 25^{th} band sample, (b) 9 class ground-truth information of AVIRIS Indian Pines hyperspectral scene.

Table 3.1 Class names and number of labeled samples for AVIRIS Indian Pines hyperspectral scene.

Indian Pines		
Class No	Class Names	# of Labeled Samples
2	Corn no till	1428
3	Corn min till	830
5	Grass pasture	483
6	Grass trees	730
8	Hay widrowed	478
10	Soybean no till	972
11	Soybean min till	2455
12	Soybean clean till	593
14	Woods	1265
Total # of labeled samples		**9234**

to 103 after the removal of water absorption caused noisy bands. Nine different classes are defined for this HSI; asphalt, meadows, gravel, trees, painted metal sheets, bare soil, bitumen, self-blocking brick, and shadows. Pavia University HSI's 50th band in gray scale and ground-truth map with nine different classes are shown in Figure 3.2. Class names and the number of labeled samples with corresponding class sequence numbers are given in Table 3.2.

3.1.3 AVIRIS Salinas

AVIRIS Salinas data set was captured over Salinas Valley, California, USA, in October 1998. The scene consists of 512 row/scene, 217 pixel/row. Salinas HSI originally contains 224 spectral bands. After the removal of water absorption caused noisy bands, band number is reduced to 204. Spatial resolution of the scene is 3.7 meter

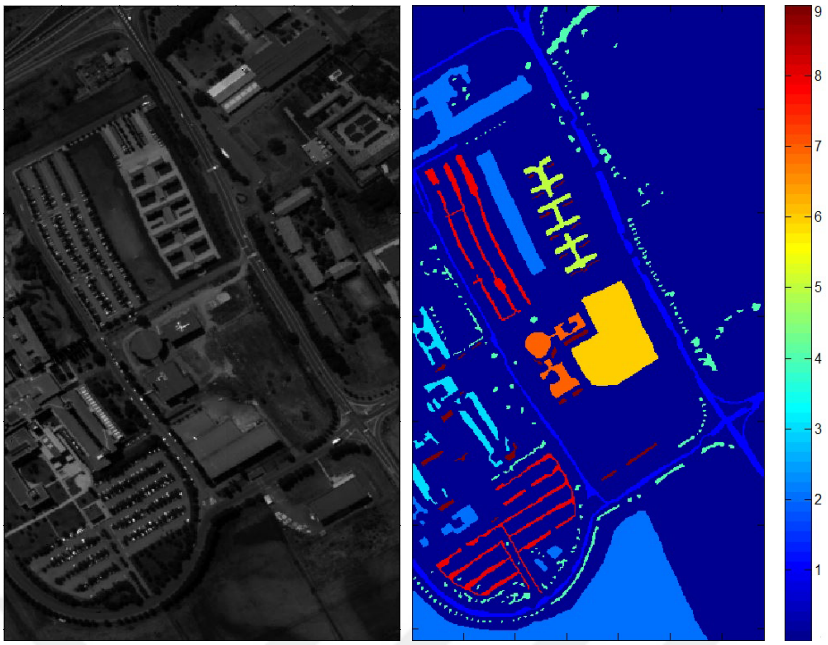

Figure 3.2 (a) Image of 50^{th} band sample, (b) 9 class test ground-truth information of ROSIS-03 Pavia University hyperspectral scene.

Table 3.2 Class names and number of labeled samples for ROSIS-03 Pavia University hyperspectral scene.

Pavia University		
Class No	Class Names	# of Labeled Samples
1	Tree	3064
2	Asphalt	6631
3	Bitumen	1330
4	Gravel	2099
5	Metal Sheet	1345
6	Shadow	947
7	Bricks	3682
8	Meadow	18649
9	Soil	5029
Total # of labeled samples		**42776**

per pixel. Ground-truth is available with 54129 labeled samples and sixteen different classes. Some of them are herbs like broccoli, celery, grape, corn lettuce, vineyard and non-plant areas like fallow, stubble, soil-winyard. Salinas HSI's 75th band in gray scale and ground-truth map with sixteen different classes are shown in Figure 3.3. Class names and the number of labeled samples with corresponding class sequence numbers are given in Table 3.3.

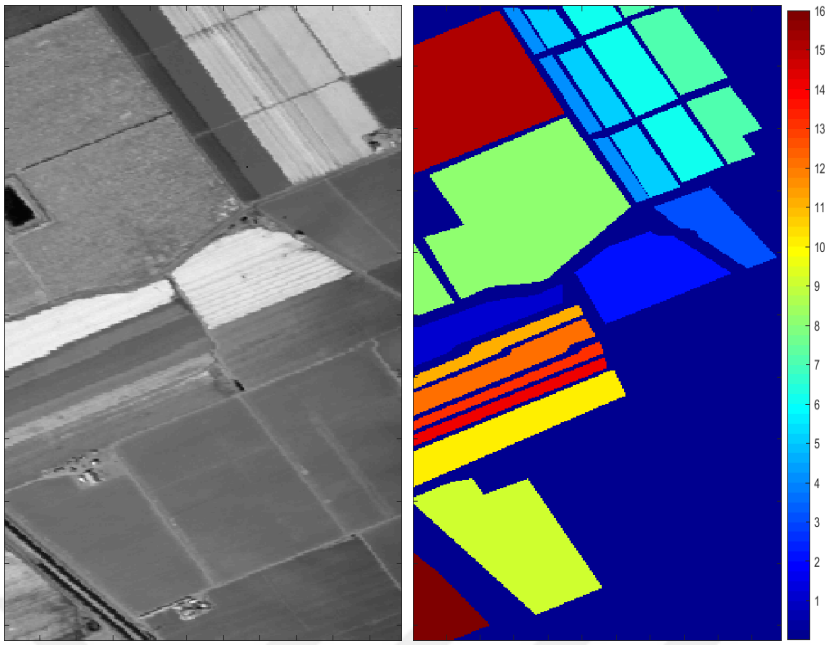

Figure 3.3 (a) Image of 75^{th} band sample, (b) 16 class test ground-truth information of Salinas hyperspectral scene.

Table 3.3 Class names and number of labeled samples for Salinas hyperspectral scene.

Salinas		
Class No	Class Names	# of Labeled Samples
1	Brocoli_green_weeds_1	2009
2	Brocoli_green_weeds_2	3726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807
Total # of labeled samples		**54129**

3.2 Validation Methods

3.2.1 Overall Accuracy

Overall accuracy (OA) metric is used to evaluate the obtained classification results. OA is determined by confusion matrix. A confusion matrix is constructed by comparing

ground-truth information with the result of a classifier. Sample confusion matrix of a binary classifier is shown in Figure 3.4.

Confusion Matrix		Actual	
		Positive	Negative
Predicted	Positive	**TP**	**FP**
	Negative	**FN**	**TN**

Figure 3.4 A binary confusion matrix illustration

This matrix consists of TP (true positive), FP (false positive), FN (false negative), and TN (true negative) values. An OA value for a classifier is calculated with dividing total number of correctly classified instances by total number of samples as in equation (3.1).

$$OA = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.1}$$

For a multi-class classifier, confusion matrix becomes $u \times u$, here u indicates total number of classes in a data set. OA calculation for a multi-class classifier is accomplished with the same logic: sum up the values residing on the diagonal of the confusion matrix (total number of correctly classified samples) and then divide it by the total number of samples.

3.2.2 Kappa Statistics

Another evaluation metric, kappa statistic [62], is used for pairwise diversity measurement and proposed for revealing agreement of classifiers' decisions [63]. Kappa is calculated over the measurement of agreement (θ_1) and disagreement (θ_2). Calculation of θ_1 is shown in equation (3.2).

$$\theta_1 = \frac{\sum_i F_{ii}}{N} \tag{3.2}$$

Measurement of disagreement is calculated as in equation (3.3). Here N is total number of test samples and F is a contingency matrix. F_{ij} indicates the number of pairs for which decision of first classifier equals to i and second one equals to j. Since, we are trying to find out how similar the classification results are as compared to the ground-truth, we have utilized ground-truth information instead of second classifier's decision.

$$\theta_2 = \sum_i \left(\sum_j \frac{F_{ij}}{N} * \sum_j \frac{F_{ji}}{N} \right) \tag{3.3}$$

Kappa statistic produces 1 in case of full agreement and -1 on the contrary. Kappa coefficient (κ) is calculated as in equation (3.4).

$$\kappa = \frac{\theta_1 - \theta_2}{1 - \theta_2} \tag{3.4}$$

3.2.3 T-test

Pairwise t-test each vs. each is utilized [64] for the purpose of comparing the obtained classification results with the other methods. T-test is a common way to determine whether the results of classifiers are statistically different or not. T-test compares overall accuracy vector of each method pairs by matching each overall accuracy value with its corresponding element of the other method's results. This statistical test is based on the pairwise differences for the values of matched observations of two samples ($\boldsymbol{d}_i = y_{1i} - y_{2i}$). The paired t-test formulation can be expressed as in equation (3.5):

$$t_{\overline{d}} = \frac{\overline{d} - \mu_{12}}{s_{\overline{d}}/\sqrt{n}} \sim t_{n-1}(\alpha) \tag{3.5}$$

where n corresponds to the number of data in $\boldsymbol{d}$ vector. Expression of $\overline{d}$ represents the mean value of $\boldsymbol{d}$ and $s_{\overline{d}}$ represents the standard deviation of $\boldsymbol{d}$. Zero value is assigned to μ_{12} which also shows our null hypothesis value.

$$\overline{d} = \frac{\sum d_i}{n} \quad and \quad s_{\overline{d}} = \sqrt{\frac{\sum (d_i - \overline{d})}{n-1}} \tag{3.6}$$

The null hypothesis is defined as $H_0 : \mu_1 - \mu_2 = 0$ and that means there is no difference between the mean values of two samples. In this case, the alternative hypothesis could be presented as $H_1 : \mu_1 - \mu_2 \neq 0$. The t-test result, calculated by equation (3.5), should be compared to the t-distribution value according to the degree of freedom and an alpha (α) number. The degree of freedom parameter of t-test is found by the $n-1$ value. Alpha is a significance level. The most commonly used significance levels are 0.01, 0.05, and 0.1.

3.2.4 McNemar's Test

McNemar's test is an objective and statistical criterion, utilized for comparing the obtained classification results with the state-of-the-art methods. This test is a common way to understand whether classifiers' results are statistically different or not. Nonparametric McNemar's test is also suitable for thematic map comparison [65] and can be calculated as in equation (3.7):

$$Z = \frac{Q_{12} - Q_{21}}{\sqrt{Q_{12} + Q_{21}}} \tag{3.7}$$

Another kind of 2×2 contingency matrix (Q) is used for McNemar's test. Q_{ij} shows the value in i^{th} row and j^{th} column. Q_{12} is the number of samples that are correctly classified by the first classifier and misclassified by the second one. Evidently, Q_{21} shows the value that is acquired by the opposite case. Z value indicates the difference between two classifiers. For comparison reason, different degree of freedom and significance levels can be chosen. For example if first degree of freedom and 5% significance level are chosen from the chi square distribution table, square root of this value corresponds to 1.96. That means, two classifiers' results are statistically different from each other if the obtained $|Z|$ value is bigger than 1.96. Null hypothesis is defined as $H_0 : No$ and accepted when there is no significant difference between two classifiers (if $|Z| \leq 1.96$) and the alternative hypothesis is defined as $H_1 : Yes$ and accepted when there is a significant difference between two classifiers (if $|Z| > 1.96$).

4
Multiple Instance Ensemble Learning

In this chapter, an ensemble framework for multiple-instance (MI) learning (MIL) is introduced to use in hyperspectral images (HSIs) by inspiring the bagging (bootstrap aggregation) method in ensemble learning. Ensemble-based bagging is performed by a small percentage of training samples, and MI bags are formed by a local windowing process with variable window sizes on selected instances. In addition to bootstrap aggregation, random subspace is another method used to diversify base classifiers. The proposed method is implemented using four MIL classification algorithms. The classifier model learning phase is carried out with MI bags, and the estimation phase is performed over single-test instances. In the experimental part of the study, two different HSIs that have ground-truth information are used, and comparative results are demonstrated with state-of-the-art classification methods. In general, the MI ensemble approach produces more compact results in terms of both diversity and error compared to equipollent non-MIL algorithms.

4.1 Introduction

Hyperspectral remote sensors operate from visible wavelength to the long-wave infrared range of the electromagnetic spectrum. Hyperspectral imaging takes place in the field of remote sensing and has attracted the attention of researchers from different disciplines over the past few decades. Hyperspectral images (HSIs) consist of quite rich content and contain lots of adjacent and narrow bands that are constituted by a bunch of rays reflected from different materials. Due to the fact that HSIs are obtained from long distances, it is commonplace to receive distorted signals, format errors and spectral clutters [66]. Ground-truth information labelled by humans and aforementioned adversities give rise to uncertain class label problems, and this situation affects standard machine learning algorithms negatively. Multiple instance (MI) learning (MIL) is a paradigm developed to provide the ability to work with uncertain target information [67]. Multiple instance learning is an assumption system first introduced by Dietterich et al. [49] for drug activity prediction. MIL has become

the subject of intense interest in recent years in the field of computer vision. This learning approach is being used for classification [68] and material detection in hyperspectral scenes and remotely sensed images [66, 69, 70].

Ensemble Learning (EnLe) is another subject of interest in recent years in the field of machine learning. Basically, EnLe aims to train base learners by increasing the diversity of ensemble classifier systems. Ensemble classifiers are able to reach a quite high success ratios compared to single classifiers and can be used in the classification of hyperspectral images. In the literature, for the purpose of increasing the classification success, it is proposed to use EnLe on hyperspectral images that have inadequate training data, which do not represent the whole feature space distribution [30]. A considerable improvement on the hyperspectral image classification is observed in a different EnLe approach that uses support vector machines (SVMs) as base classifiers [71]. Feature extraction based rotation forest ensemble learning is proposed [46] to attain high classification ratios to HSIs. The spatial circle-neighborhood information is combined with a semi-supervised classifier approach [72], and applied to HSIs using classifier fusion in order to improve the classification ability.

There are quite a few studies in the literature that present ensemble learning along with multiple instance learning. In one of these studies, the MIL is adapted to EnLe by using bootstrap aggregation over preformed MIL bags, and EnLe methods are applied to them [73]. Random forests with multiple instance learning is performed in [68], and an optimization strategy is proposed to preserve the diversities of base classifiers. The concept lattice infrastructure based ensemble learning approach is used for content based image acquisition in [74]. MIL and boosting approaches are combined in [75] and optimization of boosting is made over Noisy-OR, which views boosting as a gradient descent process. Similarly, MIL and boosting are studied together in [76], but in an online manner for real-time object tracking. A multi-instance multi-labelled SVM-based ensemble classification framework is proposed in another work [77] for the purpose of automatic video annotation. The classification capability of decision trees [78] on data sets with more than two classes and binary soft margin classifiers [79] are our preliminary works on multiple instance learning. In the aforementioned studies, multiple instance bags are created over labeled samples and samples with known label information are located in each EnLe bag in order to ensure randomness and optimization during the training phase of ensemble classifiers.

In this work, a new EnLe framework for HSI is presented by the motivation of bagging strategy in the EnLe methods. Ensemble learning-based bagging is made using a small percentage of the training samples on a hyperspectral scene, and local multiple instance bags are defined upon selected training samples. The proposed method

is operated using four MIL classification algorithms, and experimental results are demonstrated with state of the art classification algorithms comparatively.

4.2 Formation of ensemble framework with multiple-instance bagging approach

In the multiple instance learning scenarios, a group of different sized data are presented to the learners. These group of data are named multiple instance bags in MIL literature. An MI bag (**B**) gets a positive label when at least one of the samples in the bag is positive ($Y_{\mathbf{B}} = 1$). If all the target samples in a bag stay negative or unlabelled, then **B** MI bag gets a negative label ($Y_{\mathbf{B}} = -1$). Since hyperspectral images mostly consist of more than two different classes for land cover classification, one against all classification strategy is applied during the MIL bag creation, training and classification phases for consistency [80]. Ensuring to obtain different learning models as a result of diversifying the same kind of base classifiers is the common method followed during an ensemble classifier formation. Bootstrap aggregation structure (so-called bagging) is one of these methods based on random sampling from a training data set for the purpose of increasing the diversity of the ensemble model [19]. The multiple instance bagging strategy proposed in this work relies on randomly selected samples from the training data set. Each of the T base classifier's training stage is performed with m multiple instance bags that are generated by using randomly selected m labelled spectral signatures. In the creation phase of the multiple instance bags, spatial information of HSI is also utilized. Local $k \times k$ neighborhood are defined in a windowed structure for each component/spectral signature $\boldsymbol{x}_l \in R^d$ within m randomly selected sample subset. $\mathbf{B}_l$, l=$(1, ..., m)$ multiple instance bags are constituted with the samples around $\boldsymbol{x}_l$. Note that, $\boldsymbol{x}_l$ is the centralized pixel in the $k \times k$ windowed area, and it will be referred to as $\boldsymbol{x}_{ij}$ for two-dimensional space. The construction of $\mathbf{B}_l$ is formulated as in equation (4.1).

$$\mathbf{B}_l = \begin{bmatrix} \boldsymbol{x}_{i-\frac{k-1}{2}j-\frac{k-1}{2}} & \dots & \boldsymbol{x}_{i-\frac{k-1}{2}j+\frac{k-1}{2}} \\ \dots & \boldsymbol{x}_{ij} & \dots \\ \boldsymbol{x}_{i+\frac{k-1}{2}j-\frac{k-1}{2}} & \dots & \boldsymbol{x}_{i+\frac{k-1}{2}j+\frac{k-1}{2}} \end{bmatrix} \quad and \quad l = 1, ..., m \tag{4.1}$$

Each of the obtained $\mathbf{B}_l$ multiple instance bags contain k^2 (k has an odd integer value such as 3, 5, 7, 9) instances including the centralized spectral signature $\boldsymbol{x}_l$. Exceptionally, an MI-bag may contain less than k^2 instances when the centre pixel of the bag is located close to the border of the HSI. Samples in an MI bag may exist in three different cases: 1) Whole samples in the bag may have the same labels with

$\boldsymbol{x}_l$. 2) Some of the samples may stay unlabelled, while the rest of them have the same label information with the label of $\boldsymbol{x}_l$. 3) Some samples may have positive and negative label information, while others have no labels. All these three cases are illustrated in Figure 4.1. As can be seen from Figure 4.1, it is guaranteed that there will be at least one labelled sample $\boldsymbol{x}_l$ with $Y_{\boldsymbol{x}_l}$ label information in the $\mathbf{B}_l$. An MI-bag gets a positive label in case $\boldsymbol{x}_l$ has a positive label. If $\boldsymbol{x}_l$ has negative label, the MI-bag $\mathbf{B}_l$ is more likely to get a negative label. However, it should be considered that the possibility of the existence of a positive labelled instance in the $\mathbf{B}_l$.

In standard machine learning methods, all training instances unexceptionally belong to a particular class. Whereas in MIL scenarios, it is more likely to have bags with unlabelled samples in the training data. Some special MIL algorithms have been proposed to overcome this weakness of the standard machine learning algorithms in the MIL area. Diverse density, maximum likelihood, boosting, and logistic regression are some MIL-based classification algorithms [81]. Citation-KNN, multiple instance-based SVM (miSVM), multiple decision tree and MIL-boosting algorithms are MIL adapted versions of standard KNN, SVM, decision tree and boosting algorithms. These are examined in detail in the following subsections. In this work, we followed the “MI-bag based training, single instance based classification" way. That means the classifier model learning phase is carried out with MI-bags and the estimation phase is performed over single test instances.

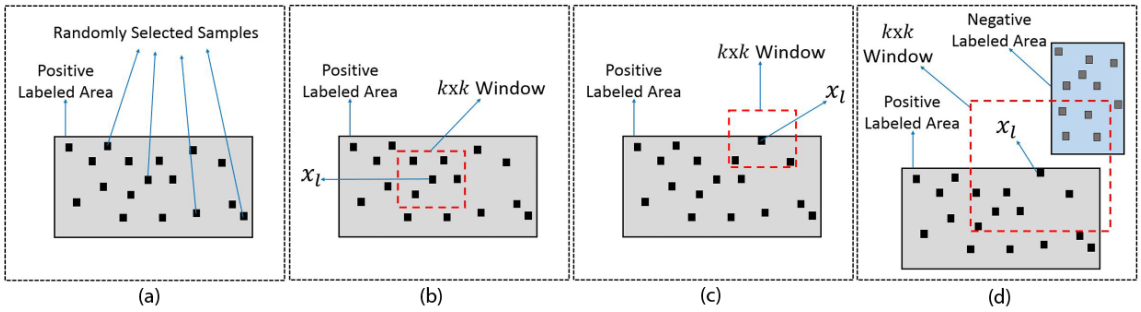

Figure 4.1 (a) Randomly selected samples in a labelled area, $\boldsymbol{x}_l$ centred $k \times k$ window that (b) stays completely inside labelled area, (c) stays partially inside labelled and unlabelled area, (d) stays partially inside positive labelled, negative labelled and unlabelled area.

4.2.1 Citation-KNN

Citation-KNN is a distance-based classification algorithm first introduced by Wang and Zucker [82]. The popular distance-based KNN algorithm is adapted to MIL by defining bag-level distance metric using the minimum Hausdorff distance. This algorithm could be expressed as the shortest distance between members of two MI-bags. “MI-bag based

training, single instance based classification" scenario is expressed as in equation (4.2) for citation KNN with respect to the Hausdorff distance between a training MI-bag and an unlabeled sample :

$$Dist(\mathbf{B}, x) = \min_{\boldsymbol{b}_i \in \mathbf{B}} ||\boldsymbol{b}_i - \boldsymbol{x}|| \quad and \quad \forall i, 1 \leq i \leq k^2 \tag{4.2}$$

An unlabelled instance in an MI-bag may not always belong to the class to which the bag belongs (see Figure 4.1). Because the majority voting is an underlying concept of the KNN's prediction mechanism, it can easily make wrong decisions due to false positive instances. The citation idea is suggested to overcome this drawback. This idea is inspired by reference and citer concepts. Neighbours (references) of MI-bag $\mathbf{B}_l$ are considered along with the other bags (citers) that have $\mathbf{B}_l$ as a neighbour. The final decision is made through the combination of references and citers. It is empirically proven that much more robust results are taken in this way.

4.2.2 Multiple Decision Tree

Multiple decision tree is presented to solve the MIL problems by Zucker and Chevaleyre [83]. The standard machine learning method, C4.5, uses information gain to split nodes of a decision tree. It makes node splits with respect to information gain (IG) of the training samples' features. The multiple decision tree is a customized form of C4.5 developed for multiple instance solutions in particular. The notion of information gain is intrinsically associated with the entropy (E) term. Computations of these two statements are adapted to multiple instance learning as in equation (4.3) and equation (4.4). Also, probability computations of two binary classes are given explicitly in equation (4.5).

$$IG(\mathbf{S}, \boldsymbol{F}_i) = E(\mathbf{S}) - \sum_{v \in \boldsymbol{F}} \frac{p(\mathbf{S}_v) + n(\mathbf{S}_v)}{p(\mathbf{S}) + n(\mathbf{S})} * E(\mathbf{S}_v) \tag{4.3}$$

$$E(\mathbf{S}) = -\sum_{i=1}^{2} g_i \log_2(g_i) \tag{4.4}$$

$$g_1 = \frac{p(\mathbf{S})}{p(\mathbf{S}) + n(\mathbf{S})} \quad and \quad g_2 = \frac{n(\mathbf{S})}{p(\mathbf{S}) + n(\mathbf{S})} \tag{4.5}$$

$\mathbf{S}$ indicates all instances in the training data set, and $p(\mathbf{S})$ and $n(\mathbf{S})$ show the samples that belong to positive and negative bags respectively. Node splitting is performed

in respect to feature $\boldsymbol{F}_i$. The $\mathbf{S}_v$ statement in the information gain equation (4.3) corresponds to the collection of instances that have specific feature value $v \in \boldsymbol{F}_i$. Similar to standard C4.5, in this algorithm, instances from $p(\mathbf{S})$ and $n(\mathbf{S})$ are utilized instead of MI-bags. Tree growing is sustained until pure class instances are obtained in each leaf. For the classification of an unknown bag, each member of the bag is passed through the tree and classified individually. The MI-bag is labelled as positive in case at least one positive instance appears in a bag, otherwise negative label is assigned. In the proposed "MI-bag based training, single instance classification" scenario, we consider each test sample as an MI-bag that has exactly one instance.

4.2.3 Support Vector Machines for MIL

While defining the maximum margin solution for the MIL problem, the soft margin definition algorithm supposes that all instances have negative labels in negative labeled bags and at least one instance has a positive label in positive labeled bags. For negative labeled bags, the maximum margin is defined as in the regular SVM case. However, for positive labeled bags, the maximum margin optimization turns out to be a mixed integer problem. Andrews et al. [84] proposed an SVM approach for MIL problems called multiple instance-based SVM (miSVM) and also introduced a heuristic in order to prevent the mixed integer problem. In miSVM, instances that do not belong to any negative bag are considered as unknown integer variables. Thus, the equation has to be maximized by a soft-margin criterion over the possible label assignments to these unknown integer variables.

$$\min_{\{y_i\}} \min_{\boldsymbol{w},\xi} \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_i \xi_i \tag{4.6}$$

The soft margin is maximized as in equation (4.6). This maximization is made subject to $\forall i$ as shown in equation (4.7). Where $\boldsymbol{w}$ stands for the normal vector, C is the regularization parameter, and ξ is referred to as the slack variable.

$$\forall i : y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + \delta) \geq 1 - \xi_i, \xi_i \geq 0, y_i \in \{-1, 1\} \tag{4.7}$$

For the purpose of obtaining the optimal hyperplane and selecting the optimal pattern from unknown integer variables, a heuristic method is suggested. According to this heuristic method, labels are imputed to the instances in positive bags according to maximum values achieved from $(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + \delta)$. Then, for the positive bags, all outputs are computed with the current classifier. These steps are repeated until convergence

(i.e., until imputed labels stop changing). When the margin computation is finalized, "single instance classification" is performed as in the regular case.

4.2.4 MIL-Boosting

The MILBoost algorithm is proposed by Viola et al. [75] as a gradient descent process [85]. Maximization of log likelihood ($\max \sum_l \log p(y_l|\mathbf{B}_l)$) is desired. Log likelihood is defined over the probability of training bags $p(y_l|\mathbf{B}_l)$ in terms of the probability of training instances. For doing this, the Noisy-OR MIL cost function is adopted and log likelihood is assigned to training MIL bags as in equation (4.8) and equation (4.9) respectively.

$$p(y_l|\mathbf{B}_l) = 1 - \prod_{i,j} \left(1 - p(y_l|\boldsymbol{x}_{ij})\right) \tag{4.8}$$

$$\mathscr{L} = \prod_l p(y_l|\mathbf{B}_l)^{y_l} (1 - p(y_l|\mathbf{B}_l))^{(1-y_l)} \tag{4.9}$$

In this model, the weight of each instance is obtained from the derivative of log likelihood as in equation 4.10.

$$\boldsymbol{w}_{lij} = \frac{\delta \log \mathscr{L}}{\delta y} = \frac{y - p(y_l|\mathbf{B}_l)}{p(y_l|\mathbf{B}_l)} p(y_l|\boldsymbol{x}_{ij}) \tag{4.10}$$

According to this approach, each round is a searching operation that maximizes the likelihood. Intuitively, high probability of an instance increases the probability of its bag. In this case, the likelihood of negative bags is always -1 since all instances are assumed to be negative in negative bags. Boosting is a special case of instance subspace selection and the MIL-Boosting method perfectly fits our proposed ensemble formation framework on hyperspectral images.

4.2.5 Multiple Classifier System Design

So far, bootstrap aggregation has been sufficiently discoursed which is also the starting point of our proposed approach. Besides bootstrap aggregation, random subspace is another method used to diversify base classifiers. Random subspace for support vector machines [23], k-nearest neighbors [86], decision trees [21], and AdaBoost algorithms [55] are non-MIL versions of the MIL methods aforementioned in the previous subsections. Briefly, the random subspace method relies on features that are

randomly selected from the whole feature space for each base classifier. We applied this method to each base MIL-classifier in order to increase diversity of the multiple classifier system. The ensemble formation process steps for Citation-KNN, Multiple Decision Tree and miSVM are shown in Algorithm 1. In contrast to the others, the MILBoost algorithm does not use random subspace method. Instead of that it uses weighted bootstrap aggregation with gradient descent process just as explained in subsection 4.2.4.

Algorithm 1 The steps of the ensemble formation process

$\mathbf{X}_{h(tr)}$: Hyperspectral Training Data set
Y: Class Label Information
T: Total Number of the Base Classifier
k: Window Size
$\boldsymbol{F}$: Feature Space of Hyperspectral Data

1: **for** t=1 to T **do**
2: Select random feature subspace $\boldsymbol{F}_s$ for $\mathbf{X}_{h(tr)} : \left(\mathbf{X}_{h(tr)} \xrightarrow{F_s} \mathbf{X'}_{h(tr)}\right)$
3: Select random m training samples from $\mathbf{X'}_{h(tr)}\big)$
4: Define $k \times k$ windows for each $\boldsymbol{x}_l \in R^d, l = \{1, ..., m\}$ sample where $\boldsymbol{x}_l$ center pixel of window and also referred to as $\boldsymbol{x}_{ij}$
5: Create $\mathbf{B}_l$ MI-bags with the instances inside the window
6: Train C_t base classifier with using $\mathbf{B}_l$ MI-bag and $Y_{\mathbf{B}_l}$ label information
7: **end for**

As a common characteristic of hyperspectral images, classes have unbalanced ratio compared to each other. Besides that, hyperspectral data mostly have more unlabelled data than labelled data. In order to simulate these situations and have a more challenging environment; $\mathbf{X}_{h(tr)}$: Hyperspectral training data set is subsampled from the original training data set with 1%, 5% and 10% subsampling ratios. After that, the proposed bagging-based windowing structure is applied to MIL classifiers as formulated above. The complete training and testing phases are illustrated in the Figure 4.2.

Profits of the proposed framework can be listed as below:

1-) Mostly, neighbor pixel correlation is high in hyperspectral images. So, taking neighbor pixels into account by defining local windowing function during the MIL bag creation step is evaluated as spatial-spectral association and it is thought to have a positive effect on classification task.

2-) Unlabeled pixels on the HSI are valuable as much as labeled ones and usually contain meaningful information. Thus, making use of the unlabeled pixels may provide significant advantage. Proposed framework naturally allows unlabeled pixel

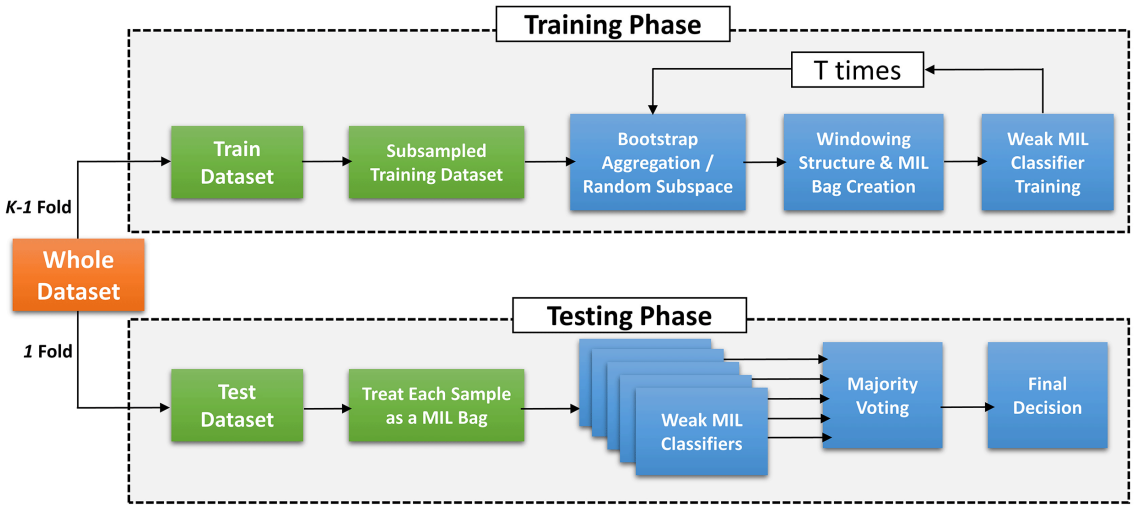

Figure 4.2 Flowchart of the training and testing phases

usage and perfectly fits on MIL classification on HSI.

3-) Data sets with the small training size is arduous to classify. However ensemble methods with bootstrap aggregation and random subspace adaptation on MIL based classification methods increase the performance.

As an innovative way, multiple instance learning is combined with ensemble learning by allowing the use of presumably valuable unlabeled data. Nevertheless, windowing function in the bag creation phase may thought to be spatial-spectral association that is considered to be of major importance on hyperspectral image classification task. At the same time, proposed framework works as a pure MIL method on training phase and works as a standard machine learning method on testing phase. As a consequence of this, proposed approach can be considered as a hybrid method that enables to use advantages of both multiple instance learning and standard machine learning methodologies.

4.3 Experimental Design and Results

In the experiments, ROSIS-03 (Reflective Optics System Imaging Spectrometer) Pavia University hyperspectral scene and AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) Indian Pines hyperspectral scene are used. Obtained results are compared with the state-of-the art non-MIL and non-ensemble methods. Overall accuracy of the algorithms calculated and t-test is utilized as a statistical criterion. Another comparison criterion used for ensemble learners is pairwise diversity. Diversity is defined to measure the difference level of two base classifiers. Kappa statistic is proposed for revealing the agreement of classifier decisions. Kappa yields

1 when the results of two classifiers are identical; otherwise, it yields result between 1 and -1. Since kappa measures the agreement level of the classification results, a lower kappa value means more diverse classifiers. The diversity of each base learner is calculated in a pairwise manner because of the definition of the kappa. In order to calculate the pairwise kappa value, the first part of the pair vector is selected from the base learner's own classification decision over the test samples, and the second one is determined as the majority voted result vector of the ensemble classifier system over the test samples. The final diversity is obtained by averaging diversities of all classes.

For each case, algorithms are run for 10 times according to 10-fold cross validation, and the $\boldsymbol{d}$ result vector is created with dual differences of the algorithms. KNN, SVM, and decision tree classification algorithms are employed for ensemble creation by bootstrap aggregation and random subspace selection. Therefore, the decision tree is referred to as Random Forest (RF) [22, 87], and the obtained results from these non-MIL ensemble methods are utilized for the purpose of comparison. Note that the non-MIL classification processes are operated with the 'instance base training – instance based classification' strategy. The K value of the KNN and citation-KNN algorithms are selected as 1 and 3. Original training data set is sub-sampled by 10%, 5%, and 1% ratios. In the bagging phase, half of the training samples are picked out randomly, and the feature space is halved in random subspace selection phases for both MIL and non-MIL base classifiers. Thus, the MIL-based decision tree algorithm is referred to as MIL Forest (milFr). In the formation of MI-bags, the window size is first set to 3×3 and increased to 5×5, 7×7, and 9×9 respectively. Ensemble classifier sizes are determined as 1, 10, 50, 100, and the final classification result of the ensemble is ascertained by majority voting.

Overall classification accuracies obtained with non-MIL methods are shown in Table 4.1 for both HSIs. The obtained results for MIL methods are shown in Table 4.2 and Table 4.3 for Pavia University and Indian Pines scenes respectively. In addition to that, 1×1 windowing function (i.e practically no windowing function) is applied with the selected MIL algorithms, and the obtained results are placed to Table 4.2 and Table 4.3 under the 1×1 column. The t-test was run for a 0.05 significance level and applied to the resulting pairs of MIL and non-MIL versions of algorithms, like 1NN with 1% versus cit1NN with 1%, and all results for different window sizes under this training sample percentage. In Table 4.4 and Table 4.5 the win\loss values are presented for the Pavia University and Indian Pines hyperspectral scenes. Numerically, the win\loss values appear along with statistically win\loss values in parentheses.

The proposed method's overall classification accuracies, shown in Table 3.2 and Table 4.3, are marked in bold font if they are statistically better than the equipollent

non-MIL version of the algorithm, and they are marked in underlined italic font for the opposite circumstances. The total comparisons of the given results are simply summarized numerically and statistically and can be seen in Table 4.4 and Table 4.5.

All cases, except for Cit1NN and Cit3NN of Indian Pines, yield better classification results than non-MIL versions of the algorithms. For example, one can interpret the win\loss Table 4.4 for the first cell as Cit1NN wins 42 times and loses 6 times against 1NN in numerical manner, and Cit1NN wins 30 times and loses 1 time against 1NN in statistical manner. The most extreme point of the win\loss Tables confronts us in Table 4.5's milFr vs. RF and MILBoost vs. AdaBoost cells. It shows that milFr and MILBoost win both numerically and statistically for 48 different cases, which also indicates the maximum possible case number.

The overall classification accuracies of four different MIL algorithms show the superiority of the proposed framework against non-MIL versions. Random sample selection originated multiple instance bag creation has not only empowered classifiers, but also made it more consistent despite the use of a limited/small number of training samples. In Table 4.2, one can see that almost all cases of Pavia University's results statistically increased in comparison with non-MIL versions of algorithms. In Table 4.3, the obtained classification results of the Indian Pines scene indicate that the maximum margin, decision tree and boosting based MIL methods performed better performance. However, the distance based MIL algorithm did not perform satisfactory in a manner. The experimental results of both hyperspectral scenes clearly show that ensemble classifiers are convenient to have a more improved performance than single classifiers. Also, increasing window size mostly affects the obtained accuracies positively.

Kappa-error diagrams are informative about ensemble characteristics as well. In Figure 4.5 and Figure 4.6, scatter plots of 100 base classifiers' kappa-error diagrams are shown for Pavia University and Indian Pines scenes respectively. These diagrams are drawn for only decision tree and support vector based algorithms using 10% of training samples. Yet, they supply a general view of the proposed method about diversity. Since ensemble classifier systems are desired to have more diverse base classifiers as much as their higher accuracies, it is appropriate to say that the proposed framework works as desired. In general, the multiple instance bagging approach produces more compact results in terms of diversity and error compared to equipollent non-MIL algorithms. Conspicuously, there is an inconsistency about window size and diversity between the two different data sets. In the Pavia University data set, increasing the window size made the ensemble more diverse, whereas increasing the window size in Indian Pines decreased the diversity of the ensemble classifier system. However, the proposed MIL-based method shows more success in classification error

Table 4.1 The Pavia University and Indian Pines hyperspectral scenes' overall accuracy results obtained by non-MIL methods (The value of 1 denotes 100% accuracy).

Non-MIL Methods							
Ensemble Size	Method	PAVIA UNIVERSITY			INDIAN PINES		
		% of train samples			% of train samples		
		1%	5%	10%	1%	5%	10%
1	1NN	0.812	0.818	0.838	0.818	0.821	0.844
	3NN	0.771	0.814	0.836	0.813	0.83	0.859
	SVM	0.763	0.771	0.794	0.851	0.863	0.867
	RF	0.781	0.802	0.819	0.779	0.804	0.835
	AdaBoost	0.787	0.805	0.809	0.789	0.809	0.827
10	1NN	0.827	0.831	0.85	0.842	0.857	0.878
	3NN	0.829	0.841	0.848	0.861	0.872	0.876
	SVM	0.756	0.76	0.778	0.853	0.859	0.866
	RF	0.803	0.837	0.841	0.828	0.837	0.868
	AdaBoost	0.799	0.802	0.822	0.806	0.814	0.821
50	1NN	0.84	0.848	0.856	0.844	0.854	0.882
	3NN	0.843	0.856	0.853	0.863	0.871	0.88
	SVM	0.755	0.774	0.799	0.855	0.862	0.866
	RF	0.811	0.825	0.849	0.859	0.862	0.884
	AdaBoost	0.786	0.822	0.822	0.804	0.823	0.837
100	1NN	0.861	0.86	0.86	0.845	0.859	0.881
	3NN	0.853	0.857	0.859	0.862	0.87	0.881
	SVM	0.754	0.761	0.794	0.855	0.859	0.866
	RF	0.826	0.835	0.85	0.861	0.873	0.885
	AdaBoost	0.794	0.813	0.827	0.819	0.814	0.831

Table 4.2 The Pavia University hyperspectral scene's overall accuracy results obtained by MIL methods (The value of 1 denotes 100% accuracy).

MIL Methods																
Ensemble Size	Method	1% of train					5% of train					10% of train				
		1x1	3x3	5x5	7x7	9x9	1x1	3x3	5x5	7x7	9x9	1x1	3x3	5x5	7x7	9x9
1	Cit1NN	0.819	0.825	0.801	0.824	0.799	0.819	**0.842**	**0.851**	**0.864**	0.834	0.849	**0.877**	**0.886**	**0.882**	0.849
	Cit3NN	0.792	**0.827**	**0.805**	**0.823**	0.774	0.821	**0.845**	**0.847**	**0.856**	**0.83**	0.841	**0.878**	**0.885**	**0.868**	**0.849**
	miSVM	0.781	**0.794**	0.792	**0.809**	**0.801**	0.785	**0.801**	**0.842**	**0.859**	**0.863**	0.808	**0.822**	**0.868**	**0.876**	**0.881**
	MIL-Fr	0.773	0.779	**0.837**	0.77	0.784	0.813	**0.828**	**0.835**	**0.827**	**0.846**	0.82	0.832	**0.841**	**0.854**	**0.869**
	MIL-Boost	0.813	**0.846**	**0.846**	**0.848**	**0.834**	0.838	**0.86**	**0.857**	**0.861**	**0.859**	0.842	**0.852**	**0.859**	**0.862**	**0.861**
10	Cit1NN	0.825	**0.874**	**0.873**	**0.861**	**0.858**	0.843	**0.881**	**0.884**	**0.872**	**0.854**	0.857	**0.893**	**0.886**	**0.88**	0.855
	Cit3NN	0.819	**0.881**	**0.865**	**0.862**	0.858	0.846	**0.883**	**0.877**	**0.87**	0.841	0.846	**0.891**	**0.885**	**0.869**	0.85
	miSVM	0.771	**0.865**	**0.857**	**0.872**	**0.86**	0.773	**0.863**	**0.861**	**0.857**	**0.873**	0.804	**0.869**	**0.874**	**0.88**	**0.884**
	MIL-Fr	0.806	**0.839**	**0.834**	0.822	**0.842**	0.829	**0.856**	**0.859**	**0.862**	**0.871**	0.843	**0.861**	**0.871**	**0.878**	**0.89**
	MIL-Boost	0.83	**0.841**	**0.834**	**0.852**	**0.832**	0.845	**0.859**	**0.867**	**0.862**	**0.865**	0.847	**0.851**	**0.857**	**0.861**	**0.859**
50	Cit1NN	0.832	**0.879**	**0.879**	0.857	0.845	0.852	**0.885**	**0.875**	**0.872**	0.857	0.862	**0.894**	**0.89**	**0.881**	0.863
	Cit3NN	0.841	**0.883**	**0.878**	0.859	0.853	0.856	**0.881**	**0.875**	0.861	0.847	0.859	**0.891**	**0.885**	0.869	0.855
	miSVM	0.746	**0.847**	**0.859**	**0.869**	**0.845**	0.779	**0.861**	**0.864**	**0.871**	**0.878**	0.81	**0.872**	**0.877**	**0.88**	**0.883**
	MIL-Fr	0.812	**0.848**	**0.841**	0.821	0.821	0.829	**0.859**	**0.863**	**0.872**	**0.881**	0.857	**0.87**	**0.872**	**0.878**	**0.892**
	MIL-Boost	0.839	**0.836**	**0.841**	**0.842**	**0.836**	0.834	**0.862**	**0.858**	**0.857**	**0.859**	0.839	**0.859**	**0.863**	**0.86**	**0.862**
100	Cit1NN	0.863	**0.883**	0.876	0.86	*0.842*	0.865	0.875	**0.882**	0.872	0.851	0.86	**0.889**	**0.889**	**0.88**	0.857
	Cit3NN	0.844	**0.884**	**0.873**	0.864	0.86	0.87	**0.888**	**0.873**	0.863	0.858	0.859	**0.891**	**0.885**	0.871	0.856
	miSVM	0.759	**0.843**	**0.876**	**0.87**	**0.866**	0.762	**0.851**	**0.869**	**0.875**	**0.872**	0.794	**0.866**	**0.877**	**0.879**	**0.882**
	MIL-Fr	0.841	**0.845**	**0.844**	0.82	0.826	0.839	**0.869**	**0.867**	**0.858**	0.852	0.859	**0.872**	**0.872**	**0.88**	**0.891**
	MIL-Boost	0.833	**0.859**	**0.842**	**0.831**	**0.856**	0.851	**0.868**	**0.864**	**0.868**	**0.869**	0.845	**0.858**	**0.868**	**0.871**	**0.867**

Bold: Statistically better than non-MIL version of the algorithm at same level of sampling percentage and ensemble size.
Italic: Statistically worse than non-MIL version of the algorithm at same level of sampling percentage and ensemble size.

Table 4.3 Indian Pines hyperspectral scene's overall accuracy results obtained by MIL methods (The value of 1 denotes 100% accuracy).

MIL Methods																
Ensemble Size	Method	1% of train					5% of train					10% of train				
		1x1	3x3	5x5	7x7	9x9	1x1	3x3	5x5	7x7	9x9	1x1	3x3	5x5	7x7	9x9
1	**Cit1NN**	0.821	**0.841**	0.809	0.804	0.821	0.827	**0.851**	**0.842**	0.837	0.819	0.845	0.859	0.856	0.842	0.828
	Cit3NN	0.817	**0.863**	0.823	0.82	**0.85**	0.833	**0.859**	**0.852**	0.839	*0.802*	0.865	0.86	0.852	0.842	*0.808*
	mi-SVM	0.859	0.867	**0.877**	**0.871**	0.87	0.857	0.877	**0.889**	**0.917**	**0.923**	0.862	**0.887**	**0.902**	**0.921**	**0.939**
	MIL-Fr	0.82	**0.855**	**0.844**	**0.865**	**0.835**	0.818	**0.884**	**0.893**	**0.901**	**0.915**	0.839	**0.902**	**0.925**	**0.928**	**0.932**
	MIL-Boost	0.817	**0.852**	**0.85**	**0.849**	**0.851**	0.866	**0.886**	**0.896**	**0.899**	**0.897**	0.875	**0.884**	**0.898**	**0.905**	**0.913**
10	**Cit1NN**	0.849	**0.864**	**0.861**	*0.807*	*0.818*	0.845	0.872	0.863	0.86	*0.828*	0.876	0.882	0.867	0.86	*0.839*
	Cit3NN	0.865	0.878	0.859	0.848	*0.831*	0.867	0.876	0.864	0.854	*0.826*	0.881	0.881	0.86	0.858	*0.833*
	mi-SVM	0.851	**0.873**	**0.874**	**0.871**	**0.875**	0.849	0.863	**0.882**	**0.923**	**0.941**	0.869	0.868	**0.894**	**0.945**	**0.973**
	MIL-Fr	0.827	**0.891**	**0.902**	**0.917**	**0.914**	0.831	**0.915**	**0.937**	**0.948**	**0.952**	0.865	**0.947**	**0.968**	**0.979**	**0.975**
	MIL-Boost	0.825	**0.847**	**0.858**	**0.847**	**0.859**	0.858	**0.886**	**0.892**	**0.895**	**0.897**	0.869	**0.891**	**0.896**	**0.899**	**0.918**
50	**Cit1NN**	0.842	**0.882**	**0.866**	0.843	0.833	0.841	**0.894**	**0.873**	0.85	0.839	0.885	0.89	0.87	*0.856*	*0.845*
	Cit3NN	0.867	0.879	0.866	*0.842*	*0.834*	0.866	0.884	0.867	0.853	*0.846*	0.882	0.891	0.863	*0.853*	*0.843*
	mi-SVM	0.863	**0.874**	**0.874**	**0.878**	**0.873**	0.85	0.865	**0.881**	**0.917**	**0.948**	0.865	0.866	**0.892**	**0.942**	**0.977**
	MIL-Fr	0.862	**0.905**	**0.917**	**0.927**	**0.935**	0.867	**0.937**	**0.947**	**0.964**	**0.97**	0.881	**0.953**	**0.974**	**0.986**	**0.984**
	MIL-Boost	0.826	**0.857**	**0.864**	**0.871**	**0.878**	0.86	**0.884**	**0.894**	**0.898**	**0.895**	0.871	**0.906**	**0.918**	**0.921**	**0.926**
100	**Cit1NN**	0.844	**0.877**	**0.867**	0.848	0.829	0.857	**0.892**	**0.882**	0.847	0.842	0.886	0.891	0.871	*0.857*	*0.847*
	Cit3NN	0.869	0.877	0.863	0.848	*0.83*	0.864	**0.892**	0.861	0.858	*0.848*	0.879	0.897	0.865	*0.854*	*0.843*
	mi-SVM	0.861	**0.874**	**0.874**	**0.882**	**0.878**	0.859	0.871	**0.878**	**0.923**	**0.957**	0.871	0.866	**0.885**	**0.941**	**0.978**
	MIL-Fr	0.871	**0.912**	**0.917**	**0.927**	**0.937**	0.873	**0.952**	**0.963**	**0.969**	**0.971**	0.887	**0.953**	**0.975**	**0.987**	**0.986**
	MIL-Boost	0.858	**0.866**	**0.893**	**0.879**	**0.885**	0.864	**0.887**	**0.895**	**0.902**	**0.909**	0.886	**0.92**	**0.927**	**0.926**	**0.938**

Bold: Statistically better than non-MIL version of the algorithm at same level of sampling percentage and ensemble size.
Italic: Statistically worse than non-MIL version of the algorithm at same level of sampling percentage and ensemble size.

Table 4.4 Win\Loss (statistically Win\Loss) Matrix for Pavia University hyperspectral scene.

Methods	Cit1NN	Cit3NN	miSVM	MIL-Fr	MIL-Boost
1NN	42\6 (30\1)	-	-	-	-
3NN	-	45\2 (31\0)	-	-	-
SVM	-	-	48\0 (47\0)	-	-
RF	-	-	-	44\3 (38\0)	-
AdaBoost	-	-	-	-	48\0 (48\0)

Table 4.5 Win\Loss (Statistically Win\Loss) Matrix for Indian Pines hyperspectral scene.

Methods	Cit1NN	Cit3NN	miSVM	MIL-Fr	MIL-Boost
1NN	23\25 (13\8)	-	-	-	-
3NN	-	19\29 (5\14)	-	-	-
SVM	-	-	46\0 (39\0)	-	-
RF	-	-	-	48\0 (48\0)	-
AdaBoost	-	-	-	-	48\0 (48\0)

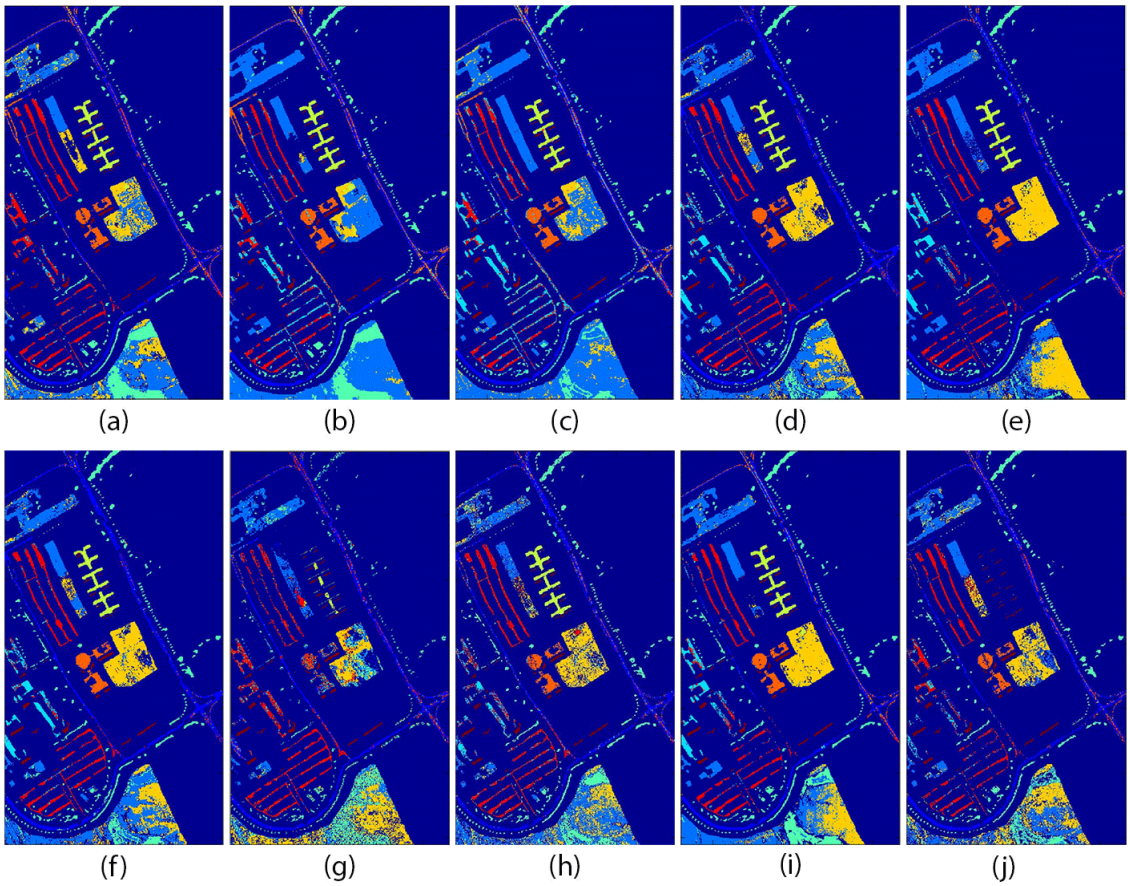

Figure 4.3 Artificial classification maps of Pavia University after binary classification with using (a) 1% of training set, 1NN and 100 base classifiers, (b) 1% of training set, Cit3NN, 3×3 window and 100 base classifiers, (c) 5% of training set, Cit3NN, 3×3 window and 100 base classifiers, (d) 5% of training set, Mil-Fr, 9×9 window and 50 base classifiers, (e) 10% of training set, miSVM, 9×9 window and 100 base classifiers, (f) 10% of training set, Mil-Fr, 9×9 window and 100 base classifiers, (g) 1% of training set, MILBoost, 1×1 window and 1 base classifier, (h) 5% of training set, MILBoost, 3×3 window and 100 base classifiers, (i) 10% of training set, MILBoost, 7×7 window and 100 base classifiers, (j) 10% of training set, AdaBoost and 100 base classifiers.

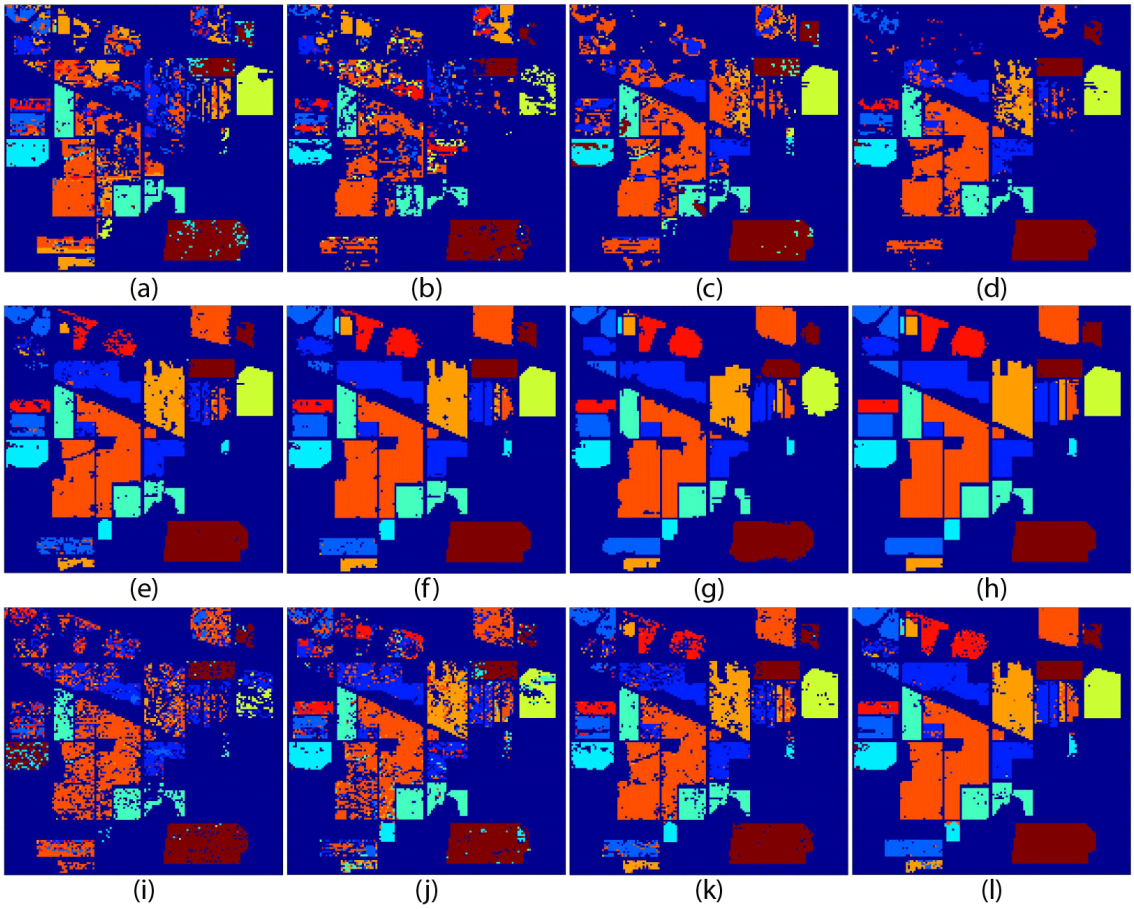

Figure 4.4 Artificial classification maps of Indian Pines after binary classification with using (a) 10% of training set, 1NN and 50 base classifiers, (b) 10% of training set, RF and 100 base classifiers, (c) 1% of training set, cit1NN, 3 × 3 window and 50 base classifiers, (d) 1% of training set, Mil-Fr, 9 × 9 window and 100 base classifiers, (e) 5% of training set, miSVM, 7 × 7 window and 100 base classifiers, (f) 5% of training set, Mil-Fr, 9 × 9 window and 100 base classifiers, (g) 10% of training set, miSVM, 9 × 9 window and 100 base classifiers, (h) 10% of training set, Mil-Fr, 9 × 9 window and 100 base classifiers. (i) 10% of training set, AdaBoost and 100 base classifiers, (j) 5% of training set, MILBooost, 1 × 1 window and 1 base classifer, (k) 10% of training, MILBoost, 3 × 3 window and 50 base classifiers. (l) 10% of training, MILBoost, 9 × 9 window and 100 base classifiers.

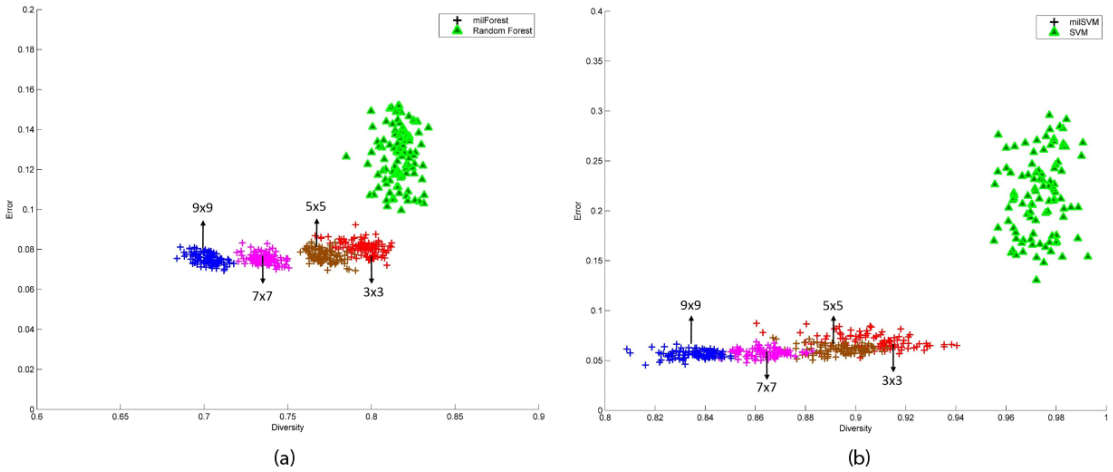

Figure 4.5 ROSIS Pavia University hyperspectral scene's average pairwise kappa – error diagrams for (a) decision tree and (b) SVM based algorithms using k= 3, 5, 7, 9 window sizes and 100 base classifiers with 10% of training samples.

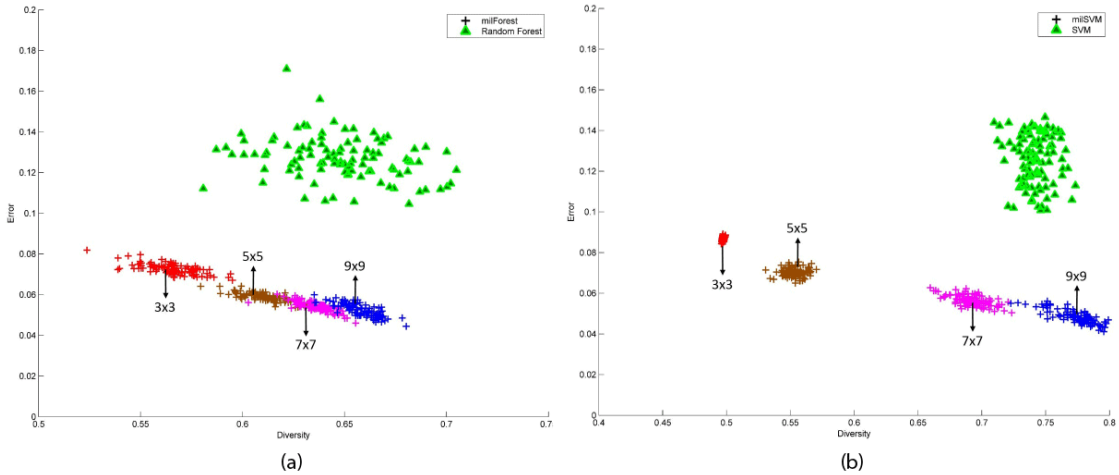

Figure 4.6 AVIRIS Indian Pines hyperspectral scene's average pairwise kappa – error diagrams for (a) decision tree and (b) SVM based algorithms using k=3, 5, 7, 9 window sizes and 100 base classifiers with 10% of training samples.

for both data sets.

4.4 Conclusions

Ensemble methods are becoming so popular and preferable in the area of hyperspectral remote sensing due to their superior performance compared to single classifiers. In this study, a multiple instance-based ensemble approach is proposed for high dimensional spectral images. Having unreliable ground-truth information and insufficiently labelled data make the proposed method more usable for many high dimensional image data sets. The usage of unlabelled samples along with labelled ones makes it possible to expand the limited sample space. The random subspaces method has a positive effect by decreasing classifier error for different types of classification algorithms. Model variances of MIL algorithms are shown to be reduced compared to non-MIL ones. In most cases, increasing the window size decreases classification error. But this situation varies according to different data sets. Effects of window size on the classification success differ from one data set to another because of the labelled class samples' distribution on the hyperspectral scene.

5
Hybridized Composite Kernel Boosting

Utilization of contextual information on the hyperspectral image analysis is an important fact. On the other hand, multiple kernels (MKs) and hybrid kernels (HKs) in connection with kernel methods have significant impact on the classification process. Activation of spatial information via composite kernels (CKs) and exploiting hidden features of the spectral information via MKs and HKs have been shown great successes on hyperspectral images separately. In this chapter, it is aimed to aggregate composite and hybrid kernels to obtain high classification success with a boosting based community learner. Spatial and spectral hybrid kernels are constructed using weighted convex combination approach with respect to individual success of the predefined kernels. Composite kernel formation is realized with certain proportions of the obtained spatial and spectral HKs. Computationally fast and effective extreme learning machine (ELM) classification algorithm is adopted. Since, main objective is to obtain optimal kernel during ensemble formation operation, unlike the standard MKL methods, proposed method disposes off the complex optimization processes and allows multi-class classification. Pavia University, Indian Pines, and Salinas hyperspectral scenes that have ground truth information are used for simulations. Hybridized composite kernels (HCK) are constructed using Gaussian, polynomial, and logarithmic kernel functions with various parameters and then obtained results are presented comparatively along with the state-of-the-art MKL, CK, sparse representation, and single kernel based methods.

5.1 Introduction

Working with hyperspectral images (HSIs) is a challenging task because of the high-dimensional characteristics of the images called Hughes phenomenon [88]. Due to the fact that HSIs are remotely sensed from long distances, it is also possible to receive distorted signals, format errors and spectral clutters [66]. On top of it, small number of labeled samples and mislabeling make it quite arduous to classification process [89]. In order to eliminate aforementioned adversities and have more accurate

results kernel based methods have been proposed on hyperspectral images [89], [90], [91].

Kernel-based methods operate on the basis of mapping original input space to higher dimensional kernel space. These methods provide ability to interpret the learning model of linearly non-separable data. Support vector machines (SVMs), one of the most popular kernel-based algorithm, have been widely used for HSI classification [92], [93], [89]. SVMs are preferable due to their handling capacity of large inputs and surmounting ability of the noisy samples robustly [94]. Extreme learning machine (ELM) is another thriving kernel-based method which has recently gained popularity in the field of HSI classification [95], [96]. ELM, a special kind of single hidden layer feed-forward neural network, provides a key solution for non-linear problems with least norm and least square solutions at very low run-time.

Most of the kernel-based methods utilize a single predefined kernel such as radial base kernel or polynomial kernel. However, real-world learning problems reveal the necessity of multiple kernel usage since most data come from heterogeneous data sources or could be encountered as different representations. Multiple kernel learning (MKL) is a proposed method to combine multiple kernels for the purpose of achieving higher and flexible solutions in the real-world scenarios [97], [54], [98]. Different variations of MKL proposed for the HSI classification have shown the enhanced accuracy results [99], [100], [101]. Although it is widely used, existing regular MKL methods have some limitations such as complicated optimization process, low efficiency, and scalability issues on the real-world large applications. In order to address the limitations of the regular MKL approaches, a new ensemble based algorithm, multiple kernel boosting (MKBoost) is proposed [56]. In MKBoost, it is intended to learn optimal combination of the weak kernel-based learners' results without resolving a complicated optimization task. Performances of the proposed MKBoost variations on the HSI classification job show superiority against regular MKL methods [102], [103], [104].

Taking contextual information into account as much as spatial information has great importance in the analysis of hyperspectral images. In recent years, some HSI classification studies considering spatial information together with spectral information are presented [105], [106], [107]. A full family of kernel-based method that incorporates spatial and spectral information is proposed and referred to as composite kernels (CKs) [61]. In SVM-CK, mean and standard deviation are utilized as spatial information. Thus, it has straightforward implementation. However, it is difficult to find the optimal SVM-CK regularization parameters and processing HSI data is time consuming. In order to reduce the time consumption and increase the

spectral and spatial separability, HSI classification with using CK and kernel-ELM (KELM) is proposed [108].

Since, HSI data mostly contain compound distribution, it is convenient to represent HSI data by a compound kernel function. In general, MKL methods do not fulfill this desire because of the fact that MKL methods fuse classifiers itself instead of kernels. Therefore, hybrid kernel functions are proposed by blending different kind of kernel functions into one compound single kernel in appropriate proportions [58]. We have proposed a hybrid kernel function based classification method with using KELM in order to increase generalization performance [60]. Despite hybrid kernels are easy to implement, proportion parameters should be adjusted properly.

In this work, we have proposed a novel framework by employing spatial information along with spectral information and putting multiple different kernel functions together. We adapted the boosting idea in order to adjust both ratio of the predefined kernel functions on the composed hybrid kernel and sub-sampling likelihood of the instances in the training data set. Together with the use of composite kernels, we have named this framework as "hybridized composite kernel boosting (HCKBoost)". Main contribution of our work can be summarized as follows: 1) We have proposed a new framework that combines both CKs and HKs efficiently. HCKBoost offers adaptive adjustment of the hybrid kernel composition and probability of the instance sub-sampling with respect to the final hybridized composite kernel based classifier. 2) Proposed framework aims to integrate the kernels itself instead of fusing classifiers as in the MKL. Thus, it does not require complicated optimization processes. Alternatively, an intuitive adaptive parameter regulation method is used during the boosting task. It enables to obtain final result from an ensemble classifier that provides robust generalization performance. 3) Computationally fast and effective classification algorithm ELM is utilized. It does not require to tune hidden node parameters and it can be simply adapted to kernel base. ELM also allows multi-class classification. Thus, proposed framework removes the SVM-origin disadvantages.

5.2 Proposed method: hybridized composite kernel boosting

A kernel function provides effectiveness to a learning method. However, in most cases choosing a kernel requires prior knowledge about the data. Moreover, different kind of extracted contextual features or data come from varied sources make it challenging to employ a single kernel. Some particular kernel functions may operate optimally for some circumstances, yet there is no superior kernel function that fits in all applications (no free lunch theorem [109]). As in kernel selection, spectral context may not be

sufficient individually in order to extract meaningful results from high dimensional data. Therefore, we intended to combine different kernels along with the spatial information.

5.2.1 HCKBoost: hybridized composite kernel boosting

Hybridized Composite Kernel Boosting is rely on adaptive boosting trials to combine kernel functions. According to this procedure, some kernel functions are learned repeatedly for both spatial and spectral features during each boosting round ($t = 1, ..., T$). Before the beginning of the training step, whole elements of probability distribution matrix D_0 is set to $1/N$ in order to give equal selection chance to each instance initially. During the boosting iteration, probability of selecting an instance is increased if it is misclassified by the previous classifier and decreased in the opposite case. So, it forces the classifier to focus on the samples that are difficult to classify.

In each boosting round, a subset of training data is sub-sampled with n predefined number of instances. Sub-sampling process is made according to D_t distribution and realized for spatial $\boldsymbol{X}^s$ and spectral $\boldsymbol{X}^w$ data separately. After obtaining subset for both input data, performance of kernel functions on spatial and spectral data are calculated individually in the sequential loops. Performance measure calculation over misclassified samples for $i^{th}(i = 1, ..., P^s)$ and $j^{th}(j = 1, ..., P^w)$ sequential inner loop iterations on t^{th} boosting trial are shown in equation (5.1) and equation (5.2) for both spatial and spectral kernel classifiers respectively:

$$\epsilon_t^s(i) = \sum_{k=1}^{N} D_t(k)\left(f_t^{s_i}(\boldsymbol{x}_k) \neq y_k \right) + \gamma \tag{5.1}$$

$$\epsilon_t^w(j) = \sum_{k=1}^{N} D_t(k)\left(f_t^{w_j}(\boldsymbol{x}_k) \neq y_k \right) + \gamma \tag{5.2}$$

where γ is a very small positive real value and used to prevent division by zero exception. Two different sequential loops allow to use distinct spatial ($\{K_i^s\}_{i=1}^{P_s}$) and spectral ($\{K_j^w\}_{j=1}^{P_w}$) kernel collections and content of them may differ from each other.

After obtaining the performance measurements (μ coefficients) through error ratios (see equation (5.5)), spatial ($\boldsymbol{K}_{H_t}^s$) and spectral ($\boldsymbol{K}_{H_t}^w$) hybrid kernels are constructed for t^{th} trial as in equation (5.3) and equation (5.4) respectively.

$$\boldsymbol{K}_{H_t}^s = \sum_{i=1}^{P^s} \left(\mu_t^s(i) * \boldsymbol{K}_i^s \right) \tag{5.3}$$

$$\boldsymbol{K}_{\boldsymbol{H}_t}^{w} = \sum_{j=1}^{P^w} \left(\mu_t^w(j) * \boldsymbol{K}_j^w \right) \tag{5.4}$$

As shown in the above expression, each kernel is weighted according to its performance. μ coefficients of $\boldsymbol{K}^s$ and $\boldsymbol{K}^w$ are calculated over error rates as in equation (5.5).

$$\mu_t^s(i) = \left(1 - \frac{\epsilon_t^s(i)}{\sum_i \epsilon_t^s(i)}\right) \quad \text{and} \quad \mu_t^w(j) = \left(1 - \frac{\epsilon_t^w(j)}{\sum_j \epsilon_t^w(j)}\right) \tag{5.5}$$

Global hybridized composite kernel ($\boldsymbol{K}_t^g$) for t^{th} boosting trial is constructed as a result of spectral kernel combination together with spatial one as in equation (5.6):

$$\boldsymbol{K}_t^g = \lambda \boldsymbol{K}_{\boldsymbol{H}_t}^s + (1-\lambda)\boldsymbol{K}_{\boldsymbol{H}_t}^w \tag{5.6}$$

where λ is a predefined constant value that balances the contribution of spatial and spectral kernel to global kernel.

After construction of global kernel on t^{th} boosting round, performance of classifier built upon that global kernel is calculated through the error ratio over whole training data set as in equation (5.7).

$$\epsilon_t^g = \sum_{k=1}^{N} D_t(k) \left(f_t^g(\boldsymbol{x}_k) \neq y_k \right) + \gamma \tag{5.7}$$

In the last step of weak classifier training, D_t distribution matrix is updated as in following equation (5.8):

$$D_{t+1}(k) = \frac{D_t(k)}{Z_t} \times \begin{cases} e^{-\alpha_t} & if \quad f_t^g(\boldsymbol{x}_k) = y_k \\ e^{\alpha_t} & if \quad f_t^g(\boldsymbol{x}_k) \neq y_k \end{cases} \tag{5.8}$$

where Z_t is normalization factor and α_t is weighting element for D_{t+1} and calculated with $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t^g}{\epsilon_t^g})$ equation. Entire HCKBoost algorithm can be seen in Algorithm 2.

Note that, since it is intended to find samples hard to classify on each trial, distribution of the spatial and spectral data is kept common (D_t) that is calculated over global kernel classifier which is considered more powerful than the spatial and spectral classifiers individually. Besides that, summation of both μ^s and μ^w are set to 1 and each element of μ is kept greater than or equal to 0. This is also called convex combination which ensures to retain positive definiteness.

Algorithm 2 The HCKBoost Algorithm

Inputs:
$\boldsymbol{X}^s = [(\boldsymbol{x}_1^s, y_1)...(\boldsymbol{x}_N^s, y_N)]$: Spatial training data
$\boldsymbol{X}^w = [(\boldsymbol{x}_1^w, y_1)...(\boldsymbol{x}_N^w, y_N)]$: Spectral training data
$\{K_i^s\}_{i=1}^{P_s}$: Kernel functions for spatial data
$\{K_j^w\}_{j=1}^{P_w}$: Kernel functions for spectral data
$D_1(k) = \frac{1}{N}$: Initial distribution for $k = 1, ..., N$
T: Total Number of boosting trials
λ: constant value where $0 < \lambda < 1$

for t=1 to T **do**
  sub-sample n instances from both spatial and spectral training data ($\boldsymbol{X}^w \xrightarrow{n} \boldsymbol{X}_t'^w$ and $\boldsymbol{X}^s \xrightarrow{n} \boldsymbol{X}_t'^s$) using D_t
  for i =1, ..., P^s **do**
    train weak spatial kernel classifier:
    $f_t^{s_i}$ with $K_i^s(\boldsymbol{X}_t'^s)$
    compute training error of spatial kernel classifier:
    $$\epsilon_t^s(i) = \sum_{k=1}^{N} D_t(k)\left(f_t^{s_i}(\boldsymbol{x}_k) \neq y_k\right) + \gamma$$
  end for
  for j =1, ..., P^w **do**
    train weak spectral kernel classifier:
    $f_t^{w_j}$ with $K_j^w(\boldsymbol{X}_t'^w)$
    compute training error of spectral kernel classifier:
    $$\epsilon_t^w(j) = \sum_{k=1}^{N} D_t(k)\left(f_t^{w_j}(\boldsymbol{x}_k) \neq y_k\right) + \gamma$$
  end for
  compute μ_t^s and μ_t^w coefficients:
  $$\mu_t^s(i) = \left(1 - \frac{\epsilon_t^s(i)}{\sum_i \epsilon_t^s(i)}\right) \quad \text{and} \quad \mu_t^w(j) = \left(1 - \frac{\epsilon_t^w(j)}{\sum_j \epsilon_t^w(j)}\right)$$
  construct hybridized spatial ($\boldsymbol{K}_{\boldsymbol{H}_t}^s$) and spectral ($\boldsymbol{K}_{\boldsymbol{H}_t}^w$) kernels:
  $$\boldsymbol{K}_{\boldsymbol{H}_t}^s = \sum_{i=1}^{P^s}\left(\mu_t^s(i) * \boldsymbol{K}_i^s\right)$$
  $$\boldsymbol{K}_{\boldsymbol{H}_t}^w = \sum_{j=1}^{P^w}\left(\mu_t^w(j) * \boldsymbol{K}_j^w\right)$$
  construct global kernel $\boldsymbol{K}_t^g$:
  $$\boldsymbol{K}_t^g = \lambda \boldsymbol{K}_{\boldsymbol{H}_t}^s + (1-\lambda)\boldsymbol{K}_{\boldsymbol{H}_t}^w$$
  compute training error of global kernel classifier:
  $$\epsilon_t^g = \sum_{k=1}^{N} D_t(k)\left(f_t^g(\boldsymbol{x}_k) \neq y_k\right) + \gamma$$
  choose $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t^g}{\epsilon_t^g})$
  update $D_{t+1}(k)$:
  $$D_{t+1}(k) = \frac{D_t(k)}{Z_t} \times \begin{cases} e^{-\alpha_t} & if \quad f_t^g(\boldsymbol{x}_k) = y_k \\ e^{\alpha_t} & if \quad f_t^g(\boldsymbol{x}_k) \neq y_k \end{cases}$$
end for

MKL methods fuse the classifiers to reach the final decision. In HCKBoost method, each boosting iteration provides the global kernel to converge superiority and final decision is acquired from the created weak kernel classifiers during the boosting round via majority voting process. Thus, HCKBoost preserves being an ensemble learning method.

5.2.2 Complexity analysis of HCKBoost

HCKBoost algorithm shows similarity to KELM algorithm in the sense of complexity. Such that, time complexity calculation of kernel matrix $\mathbf{\Omega}$ (see equation (2.9)) equals to $\mathcal{O}(N^2M)$ and hidden layer output matrix's time complexity can be expressed as $\mathcal{O}(2N^3 + CN^2)$ [110]. Where N shows the sample size, M is feature size of the input data (i.e neuron size of the input layer) and C is class size (i.e neuron size of the output layer). Combining these two time complexity gives the complete KELM's time complexity as $\mathcal{O}(2N^3 + (C + M)N^2)$. During the boosting trials sub-sampling operation allows reduction when choosing the relatively small subset of the training input data ($n \ll N$). So, time complexity of the each of the KELM classifier belonging to HCKBoost can be expressed as $\mathcal{O}(2n^3 + (C + M)n^2)$. We denote $C(n)$ notation to indicate time complexity of the base kernel classifier instead of long form (i.e. $C(n) = \mathcal{O}(2n^3 + (C + M)n^2)$). General worst case time complexity of HCKBoost can be shown as $\mathcal{O}(T \times P \times C(n))$. Where T is the total number of boosting trials and P is total number of spatial and spectral kernels plus one global kernel (i.e. $P = P^s + P^w + 1$).

We denoted the space complexity as $S(n)$ for KELM trained from a small subset of input data ($n \ll N$). In this way, the worst case space complexity of HCKBoost can be expressed as $\mathcal{O}(T \times P \times S(n))$. Since, KELM does not require to store all predefined kernels on the memory as in standard MKL methods, instead micro-kernels are constructed during boosting trials. Thus, significant amount of memory space has been earned. This situation provides effective solution in large-scale applications such as HSI analysis.

5.3 Experimental design and results

In the experiments, proposed HCKBoost algorithm is compared with several state of the art methods and obtained classification results with different parameters are presented. Three different benchmark datasets are utilized to investigate performance of the proposed method against state of the art methods. Pavia University, Indian Pines, and Salinas. Details of the experimental setup is discussed in the following subsections.

5.3.1 Comparison schemes

The proposed HCKBoost algorithm is compared with the various state of the art multiple kernel learning, composite kernel, and sparse representation algorithms. Also, results of the basic machine learning algorithms are inserted into comparison scheme.

- Basic classification algorithms:
  - SVM ([111]): Basic support vector machine.
  - KSVM ([112]): Kernel support vector machine.
  - ELM ([50]): Basic extreme learning machine.
  - KELM ([51]): Kernel extreme learning machine.
- Composite kernel classification algorithms:
  - CK-SVM ([61]): Composite kernel support vector machine.
  - CK-ELM ([108]): Composite kernel extreme learning machine.
- Sparse representation based classification algorithms:
  - KOMP ([113]): Kernel orthogonal matching pursuit sparse representation algorithm.
  - KSOMP ([113]): Kernel simultaneous orthogonal matching pursuit sparse representation algorithm.
  - MASR ([114]): Multiscale adaptive sparse representation algorithm.
- Multiple kernel learning algorithms:
  - SimpleMKL ([115]): Mixed-norm regularization simple multiple kernel learning.
  - SM1MKL ([116]): Soft margin multiple kernel learning.
  - L1-MKL ([117]): L1 norm multiple kernel learning.
  - L2-MKL ([117]): L2 norm multiple kernel learning.
  - MKBoost-D1 ([56]): First deterministic multiple kernel boosting algorithm.
  - MKBoost-D2 ([56]): Second deterministic multiple kernel boosting algorithm.
  - L1-MK-ELM ([118]): L1 norm Extreme learning machine based multiple kernel learning algorithm.
  - L2-MK-ELM ([118]): L2 norm Extreme learning machine based multiple kernel learning algorithm.

5.3.2 Experimental Settings

For the experimental settings, single Gaussian kernel is utilized for the basic classification algorithms (KSVM, KELM), the composite kernel classification algorithms (CK-SVM, CK-ELM), and the kernel based sparse representation algorithms (KOMP, KSOMP). In the state of the art MKL algorithms, 12 different kernels are used as follows: Gaussian kernels with 9 different width parameters $(2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4})$, polynomial kernels with 2 different degrees $(1, 2)$, and 1 logarithmic kernel with the parameter $d = 2$. When using KOMP, KSOMP, and MASR algorithms, dictionary is constructed with using 10% of the training samples. In the MASR method, window widths are set to 3, 5, 7, 9, 11, 13, and 15.

HCKBoost is composed of spatial and spectral kernel parts. Each part is a hybrid kernel in itself. Gaussian kernel has global approximation ability and polynomial kernel is known with its local learning ability. Logarithmic kernel is a robust kernel function against noisy data and its success has been reported as almost equal to Gaussian kernel on the spectral reflectance estimation [119]. That's why, Gaussian, polynomial, and logarithmic kernel functions are used to form spatial and spectral hybrid kernels separately. Each kernel function is utilized with its single parameter. To be more specific, one Gaussian kernel with 2^1 width value, one polynomial kernel with degree of 2, and one logarithmic kernel with $d = 2$ are only kernels used for both spatial and spectral parts. That means, 6 kernels are handled within HCKBoost. Since, logarithmic kernel is conditionally positive definite for $0 < d \leq 2$ [120], in each log kernel, d value is taken as 2 in order to maintain positive definiteness of the composed global kernel.

Spatial domain of the algorithm is managed by a simple feature extraction method named mean statistics [121]. Since, it is considered that the neighbor pixels reflect similar spectral characteristics, mean statistic may expose overall tendency of the central pixels' ($\boldsymbol{x}_i \in \mathbb{R}^N$) surrounding area. A group of pixel belonging to a material more likely does not have identical spectral signatures with each others. So, defining a windowing area around central pixel will produce convergent result to the general characteristic of the desired spectral signature of the material. Therefore, spatial information serves as a complementary content to spectral information. This situation is valid when all or most of the pixels belong to one class (homogeneous case), otherwise (heterogeneous case) spatial information become distorted away from original spectral signature. Thus, determining optimum window size plays significant role. In our experiments, surrounding window widths are set to 3, 5, 7, 9, 11, 13, and 15.

K-fold cross validation is performed over whole data set. At first, data is divided into five parts randomly. One part is reserved for testing phase and remaining four parts

are divided by two in themselves. As a result, obtained 2/5 portion of the data is used for training, 2/5 is used for validation, and 1/5 is utilized for testing phases. This method may also called as 5x2 cross validation and naturally allows us to run each algorithm 10 times. Final result is obtained by averaging 10 results. For the comparison purpose, Overall Accuracy (OA), Average Accuracy (AA), and Kappa (κ) statistics [62] are applied to all methods to disclose the classification successes.

HCKBoost inherits some key features of boosting method such as sub-sampling of the instances hard to classify in an iteration. Sub-sampling ratio is determined as 0.5 (i.e. 50% of the training samples) for trials. In each iteration a weak classifier is trained with sub-sampled instances and obtained results of that classifier is stored to be used in majority voting phase for the purpose of getting final result. During HCKBoost trials, T parameter is set to 10 for all experiments ($T = 10$) presented in this paper. Effects of variable sub-sampling ratios and T parameters are specifically examined in the subsection 5.3.5.

5.3.3 Experimental Results

Overall Accuracy (OA), Average Accuracy (AA), and Kappa (κ) statistic results of single kernel, composite kernel based methods, and sparse representation based methods are demonstrated in Table 5.1, Table 5.2, and Table 5.3. OA, AA, and κ statistic results of MKL algorithms and proposed HCKBoost algorithm are shown in Table 5.4, Table 5.5, and Table 5.6 for Indian Pines, Pavia University, and Salinas HSIs respectively. Classification maps of proposed method and four other methods are inserted into the Figure 5.4, Figure 5.5, and Figure 5.6. Unless otherwise indicated, demonstrated HCKBoost results on the tables and classifications maps are obtained with utilizing 13×13 window size on the spatial feature extraction phase and $\lambda = 0.9$ parameter value on the global kernel composition phase.

Effects of various window sizes used in the spatial feature extraction phase and different λ values on the OA, AA, and κ are investigated. Obtained results for 3×3, 5×5, 7×7, 9×9, 11×11, 13×13, and 15×15 window sizes with $\lambda = 0$, $\lambda = 0.1$, $\lambda = 0.5$, $\lambda = 0.9$, and $\lambda = 1$ are illustrated in the Figure 5.1, Figure 5.2, and Figure 5.3 for three HSI data sets. According to these figures, most accurate results are acquired with the high λ values. That means, contribution of spatial kernel to global kernel has significant impact in the case of high spatial kernel proportion for our proposed method on the HSI data. Also, accuracy increases directly proportional with the window size. But after some certain width of window (mostly after 13 for our cases), accuracy results show declining trend. This circumstance mostly occurs because of distorted and inadequate representation of the center pixel resides in the middle of

the windowing area by the sample collections in that windowing area. Which means, extracting mean statistic based spatial features does not work well with bigger than some specific window sizes. Of course, optimal size of the window depends on the used data sets. Note that, since spatial information is not effective in case of $\lambda = 0$, changes of the window widths are ineffective on the accuracy results. In the $\lambda = 1$ case, no spectral kernel contributes the global kernel. So, obtained results become relatively lower than the cases in which $0 < \lambda < 1$.

Table 5.1 Overall Accuracy (OA), Average Accuracy (AA)(1 denotes 100% accuracy), and Kappa (κ) results obtained from SVM, ELM, KSVM, KELM, CK-SVM, CK-ELM, KOMP, KSOMP, and MASR methods for Indian Pines HSI

Indian Pines									
Class	Standard Methods		Kernel Methods		Sparse Representation Methods			CK Methods	
Label	SVM	ELM	KSVM	KELM	KOMP	KSOMP	MASR	CK-SVM	CK-ELM
2	0.764	0.769	0.755	0.842	0.723	0.831	**0.988**	0.873	0.962
3	0.661	0.667	0.672	0.827	0.675	0.836	**0.982**	0.824	0.951
5	0.942	0.911	0.937	0.931	0.788	0.950	**0.989**	0.957	0.963
6	0.978	0.967	0.971	0.977	0.925	0.981	**0.987**	0.983	0.979
8	0.999	0.988	0.996	0.975	0.971	0.991	**1.000**	0.995	0.994
10	0.729	0.665	0.705	0.805	0.673	0.938	**0.968**	0.761	0.958
11	0.785	0.785	0.809	0.849	0.782	0.862	**0.990**	0.912	0.954
12	0.770	0.763	0.761	0.841	0.610	0.933	**0.977**	0.834	0.957
14	0.993	0.988	0.987	0.994	0.980	**0.999**	0.996	0.992	0.996
AA	0.847	0.834	0.844	0.893	0.792	0.924	**0.986**	0.903	0.968
±std	±0.123	±0.124	±0.122	±0.071	±0.129	±0.062	±0.009	±0.080	±0.016
OA	0.827	0.817	0.828	0.880	0.804	0.907	**0.987**	0.904	0.965
±std	±0.010	±0.010	±0.045	±0.077	±0.112	±0.078	±0.034	±0.069	±0.014
κ	0.807	0.796	0.808	0.865	0.786	0.898	**0.982**	0.878	0.961
±std	±0.011	±0.011	±0.050	±0.084	±0.124	±0.075	±0.038	±0.076	±0.016

5.3.4 Statistical Evaluation

It is required to be defined an objective and statistical criterion for comparing the obtained classification results with the state-of-the-art methods. For this purpose, non-parametric McNemar's test is utilized. We have chosen first degree of freedom and 5% significance level that corresponds to 1.96 value. That means, two classifiers' results are statistically different from each other if the obtained $|Z|$ value is bigger than 1.96. Obtained McNemar's test results are presented in the Table 5.7 for all data sets. As shown in the Table 5.7, all Z values are bigger than 1.96 and produce statistically different results except MASR method result on the Salinas HSI. MASR exposes the similar thematic map to the HCKBoost. However, since the Z value is greater than 0, the proposed method is still not statistically but numerically better than MASR.

Table 5.2 Overall Accuracy (OA), Average Accuracy (AA)(1 denotes 100% accuracy), and Kappa (κ) results obtained from SVM, ELM, KSVM, KELM, CK-SVM, CK-ELM, KOMP, KSOMP, and MASR methods for Pavia University HSI

Pavia University									
Class	Standard Methods		Kernel Methods		Sparse Representation Methods			CK Methods	
Label	SVM	ELM	KSVM	KELM	KOMP	KSOMP	MASR	CK-SVM	CK-ELM
1	0.859	0.854	0.938	0.951	0.832	0.745	0.963	0.963	**0.992**
2	0.941	0.929	0.964	0.982	0.714	0.918	0.985	0.984	**0.998**
3	0.763	0.747	0.839	0.853	0.751	0.721	0.984	0.924	**0.964**
4	0.908	0.917	0.951	0.974	0.704	0.928	0.981	0.975	**0.992**
5	0.998	0.997	0.991	0.997	0.996	**0.999**	**0.999**	0.997	0.993
6	0.816	0.756	0.897	0.938	0.553	0.748	0.965	0.958	**0.988**
7	0.727	0.716	0.877	0.912	0.631	0.781	0.994	0.936	**0.982**
8	0.829	0.819	0.916	0.929	0.783	0.767	0.917	0.958	**0.967**
9	**0.999**	0.988	**0.999**	0.997	0.998	0.995	0.997	0.998	0.998
AA	0.871	0.858	0.930	0.948	0.774	0.845	0.976	0.966	**0.986**
±std	±0.092	±0.100	±0.050	±0.044	±0.142	±0.107	±0.024	±0.024	±0.012
OA	0.868	0.857	0.933	0.949	0.785	0.830	0.956	0.967	**0.981**
±std	±0.016	±0.003	±0.030	±0.003	±0.146	±0.053	±0.045	±0.012	±0.004
κ	0.832	0.816	0.914	0.934	0.773	0.823	0.953	0.957	**0.978**
±std	±0.02	±0.004	±0.039	±0.041	±0.178	±0.072	±0.047	±0.016	±0.005

Table 5.3 Overall Accuracy (OA), Average Accuracy (AA)(1 denotes 100% accuracy), and Kappa (κ) results obtained from SVM, ELM, KSVM, KELM, CK-SVM, CK-ELM, KOMP, KSOMP, and MASR methods for Salinas HSI

Salinas									
Class	Standard Methods		Kernel Methods		Sparse Representation Methods			CK Methods	
Label	SVM	ELM	KSVM	KELM	KOMP	KSOMP	MASR	CK-SVM	CK-ELM
1	**1.000**	0.999	0.988	0.999	0.989	**1.000**	0.999	0.994	0.999
2	0.999	0.998	0.992	0.999	0.993	**1.000**	0.999	0.996	0.999
3	0.998	0.949	0.976	0.997	0.889	0.913	**1.000**	0.991	0.998
4	0.993	0.987	0.992	0.994	0.976	0.965	**0.995**	0.991	0.994
5	0.993	0.955	0.984	0.994	0.971	0.984	**0.998**	0.990	0.995
6	0.999	0.998	0.998	0.999	0.996	0.996	0.999	0.999	**1.000**
7	0.999	0.997	0.998	0.999	0.993	0.996	**1.000**	0.999	**1.000**
8	0.831	0.842	0.838	0.897	0.738	0.856	**0.996**	0.891	0.960
9	0.996	0.992	0.994	0.997	0.990	0.997	**1.000**	0.999	0.999
10	0.984	0.953	0.918	0.985	0.859	0.962	**0.998**	0.983	0.985
11	0.994	0.940	0.961	0.993	0.851	0.890	**1.000**	0.974	0.995
12	0.998	0.972	0.995	0.995	0.976	0.981	**1.000**	0.997	0.998
13	**0.998**	0.958	0.983	0.993	0.931	0.962	**0.998**	0.996	0.997
14	0.986	0.947	0.969	0.985	0.924	**1.000**	0.997	0.991	0.993
15	0.671	0.714	0.717	0.832	0.646	0.777	**0.996**	0.831	0.952
16	0.996	0.991	0.991	0.996	0.926	0.985	0.998	0.994	**0.999**
AA	0.965	0.950	0.956	0.978	0.915	0.954	**0.998**	0.976	0.991
±std	±0.086	±0.072	±0.073	±0.045	±0.098	±0.062	±0.002	±0.045	±0.014
OA	0.919	0.917	0.918	0.953	0.871	0.928	**0.998**	0.951	0.983
±std	±0.002	±0.003	±0.020	±0.013	±0.085	±0.079	±0.002	±0.007	±0.004
κ	0.912	0.910	0.910	0.949	0.856	0.917	**0.998**	0.947	0.986
±std	±0.002	±0.003	±0.022	±0.014	±0.081	±0.064	±0.002	±0.008	±0.005

Table 5.4 Overall Accuracy (OA), Average Accuracy (AA)(1 denotes 100% accuracy), and Kappa (κ) results obtained from SimpleMKL, L1MKL, L2MKL, SM1MKL, MKBoost-D1, MKBoost-D2, L1-MK-ELM, L2-MK-ELM, and proposed HCKBoost methods for Indian Pines HSI

Indian Pines									
Class	MKL Methods								**Proposed**
Label	SimpleMKL	L1MKL	L2MKL	SM1MKL	MKBoost-D1	MKBoost-D2	L1-MK-ELM	L2-MK-ELM	**HCKBoost**
2	0.891	0.891	0.910	0.910	0.904	0.908	0.831	0.869	**0.992**
3	0.887	0.898	0.916	0.917	0.902	0.913	0.853	0.887	**0.995**
4	0.948	0.956	0.933	0.955	0.958	0.955	0.950	0.958	**0.992**
6	0.989	0.989	0.978	0.984	**0.994**	0.993	0.991	0.991	0.991
8	**1.000**	0.999	0.987	0.992	0.996	0.997	0.997	0.996	**1.000**
10	0.898	0.907	0.918	0.919	0.916	0.918	0.890	0.903	**0.992**
11	0.896	0.917	0.886	0.888	0.886	0.889	0.863	0.881	**0.995**
12	0.879	0.918	0.937	0.937	0.929	0.926	0.857	0.900	**0.989**
14	0.985	0.985	0.982	0.993	0.994	0.993	0.984	0.987	**0.998**
AA	0.930	0.940	0.939	0.944	0.942	0.944	0.913	0.930	**0.994**
±std	±0.047	±0.052	±0.068	±0.056	±0.075	±0.071	±0.023	±0.016	±0.001
OA	0.921	0.932	0.927	0.931	0.929	0.931	0.898	0.917	**0.994**
±std	±0.098	±0.068	±0.086	±0.082	±0.085	±0.074	±0.014	±0.009	±0.001
κ	0.873	0.884	0.876	0.882	0.879	0.882	0.859	0.866	**0.993**
±std	±0.154	±0.011	±0.137	±0.132	±0.134	±0.119	±0.006	±0.003	±0.001

Table 5.5 Overall Accuracy (OA), Average Accuracy (AA)(1 denotes 100% accuracy), and Kappa (κ) results obtained from SimpleMKL, L1MKL, L2MKL, SM1MKL, MKBoost-D1, MKBoost-D2, L1-MK-ELM, L2-MK-ELM, and proposed HCKBoost methods for Pavia University HSI

Pavia University									
Class	MKL Methods								**Proposed**
Label	SimpleMKL	L1MKL	L2MKL	SM1MKL	MKBoost-D1	MKBoost-D2	L1-MK-ELM	L2-MK-ELM	**HCKBoost**
1	0.972	0.972	0.969	0.977	0.973	0.975	0.954	0.964	**0.999**
2	0.972	0.973	0.971	0.980	0.979	0.980	0.949	0.951	**0.999**
3	0.965	0.969	0.969	0.965	0.956	0.968	0.919	0.920	**0.999**
4	0.984	0.978	0.977	0.987	0.986	0.987	0.912	0.914	**0.997**
5	0.982	0.992	0.995	0.997	0.995	0.997	0.965	0.966	**1.000**
6	0.961	0.965	0.939	0.968	0.965	0.969	0.923	0.922	**1.000**
7	0.972	0.968	0.956	0.972	0.971	0.974	0.922	0.921	**0.999**
8	0.967	0.974	0.969	0.967	0.965	0.968	0.931	0.933	**0.997**
9	0.998	**1.000**	**1.000**	**1.000**	0.997	**1.000**	0.989	0.990	**1.000**
AA	0.975	0.977	0.972	0.979	0.976	0.980	0.940	0.942	**0.999**
±std	±0.038	±0.032	±0.039	±0.029	±0.046	±0.033	±0.090	±0.010	±0.001
OA	0.973	0.977	0.973	0.976	0.975	0.977	0.940	0.943	**0.998**
±std	±0.034	±0.024	±0.034	±0.026	±0.037	±0.027	±0.001	±0.001	±0.001
κ	0.950	0.957	0.950	0.955	0.953	0.956	0.922	0.926	**0.998**
±std	±0.061	±0.044	±0.061	±0.048	±0.068	±0.050	±0.001	±0.001	±0.001

5.3.5 Boosting parameters evaluation

In boosting trials, sub-sampling ratio and ensemble size (T) parameters may effect the obtained results of the HCKBoost. Sub-sampling ratio determines the proportion of training samples which are selected from whole training data for each boosting round. We have evaluated the different sub-sampling ratios starting from 10% to 90% as shown in the Figure 5.7 for Indian Pines, Pavia University, and Salinas HSIs. In general, we have found that the small sub-sampling ratios reduce the performance

Table 5.6 Overall Accuracy (OA), Average Accuracy (AA)(1 denotes 100% accuracy), and Kappa (κ) results obtained from SimpleMKL, L1MKL, L2MKL, SM1MKL, MKBoost-D1, MKBoost-D2, L1-MK-ELM, L2-MK-ELM, and proposed HCKBoost methods for Salinas HSI

Salinas									
Class	MKL Methods								**Proposed**
Label	SimpleMKL	L1MKL	L2MKL	SM1MKL	MKBoost-D1	MKBoost-D2	L1-MK-ELM	L2-MK-ELM	**HCKBoost**
1	0.998	0.999	0.999	0.999	0.999	0.999	0.998	0.998	**1.000**
2	0.997	0.999	0.999	0.999	0.999	0.999	0.998	0.999	**1.000**
3	0.998	0.998	0.998	0.998	0.997	0.997	0.997	0.997	**1.000**
4	**0.998**	**0.998**	**0.998**	**0.998**	**0.998**	**0.998**	**0.998**	0.996	0.996
5	0.996	0.996	0.996	0.996	0.996	0.996	0.995	0.995	**0.998**
6	0.999	**1.000**	**1.000**	**1.000**	0.999	**1.000**	0.999	0.999	**1.000**
7	0.999	0.999	0.999	0.999	0.999	**1.000**	0.998	0.998	**1.000**
8	0.701	0.937	0.939	0.937	0.940	0.942	0.791	0.790	**0.999**
9	0.998	0.998	0.998	0.998	0.999	0.999	0.998	0.998	**1.000**
10	0.983	0.983	0.985	0.985	0.989	0.990	0.985	0.982	**1.000**
11	0.993	0.993	0.992	0.993	0.997	0.997	0.992	0.993	**1.000**
12	0.999	0.999	0.999	0.999	**1.000**	0.999	0.999	0.999	**1.000**
13	0.996	0.997	0.997	0.997	0.999	0.999	0.996	0.997	**1.000**
14	0.993	0.994	0.993	0.993	0.994	0.992	0.992	0.993	**0.999**
15	0.846	0.918	0.919	0.919	0.921	0.925	0.901	0.902	**0.999**
16	0.996	0.996	0.996	0.996	0.997	0.997	0.995	0.995	**1.000**
AA	0.968	0.988	0.988	0.988	0.989	0.989	0.977	0.977	**0.999**
±std	±0.112	±0.034	±0.042	±0.036	±0.025	±0.027	±0.011	±0.009	±0.001
OA	0.914	0.974	0.973	0.974	0.975	0.976	0.941	0.940	**0.999**
±std	±0.138	±0.019	±0.021	±0.020	±0.019	±0.018	±0.001	±0.002	±0.001
κ	0.876	0.954	0.953	0.954	0.955	0.957	0.935	0.935	**0.999**
±std	±0.184	±0.036	±0.04	±0.037	±0.036	±0.034	±0.002	±0.002	±0.001

of the algorithm. Ratios around the 50% produce best accuracy results. While approximating to the 100% sub-sampling ratio, performance slightly slides down. However, results are still remain being satisfactory.

We have examined effect of the total number of boosting rounds and presented obtained results in the Figure 5.8. At first, T value is set to 1 which actually does not yield an ensemble algorithm but single classifier. Afterwards, T value is set to 5, 10, 25, 50, 100, and 200. It is observed that the increasing ensemble size has raised the performance. However, improvements on classification accuracies become very small when using $T > 10$. This proves that the proposed algorithm works well even if the small ensemble size is used. Certainly, bigger T values produce better accuracies, but tradeoff between time complexity and obtained accuracy should be considered according to time and accuracy sensitivity of the applications.

5.4 Conclusion

In this chapter, we proposed a boosting based new methodology which combines composite and hybrid kernels efficiently. In combination with the extreme learning machine which is considered as a computationally effective classification algorithm,

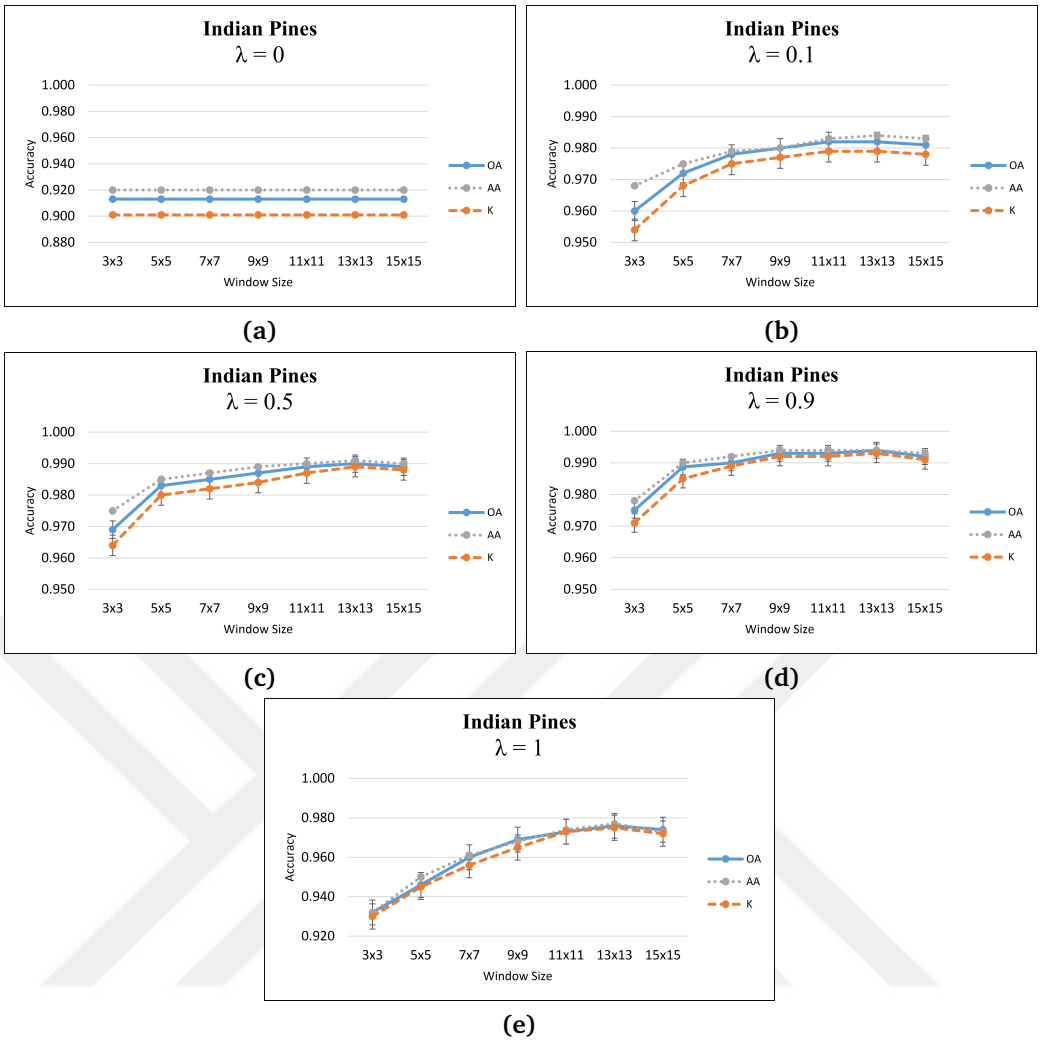

Figure 5.1 Overall Accuracy (OA), Average Accuracy (AA), and Kappa (κ) results of proposed HCKBoost method constructed with contribution of spatial kernels (i.e. relatively spatial features extracted with utilizing various window sizes) using (a) $\lambda = 0$, (b) $\lambda = 0.1$, (c) $\lambda = 0.5$, (d) $\lambda = 0.9$, (e) $\lambda = 1$ values for Indian Pines HSI

proposed hybridized composite kernel boosting method confront multi-class classification tasks rapidly. The proposed method is compared with numerous state-of-the-art CK, MKL, and sparse representation methods along with the basic classification algorithms. The results presented in the previous sections show superiority of HCKBoost compared to others in terms of accuracy and kappa statistics. HCKBoost is designed as an ensemble method, thus it may also be considered as a low-cost method against optimization based MKL methods since it does not require complicated optimization tasks. Spatial and spectral domains are hybridized within themselves by using different type of kernels with various parameters. As a consequence of this, global kernel comes up with HK as much as CK. Although, simple spatial feature extraction method (mean statistic) has been used via windowing structure, it yielded pleasing results. Each HSI data may have different pixel resolution

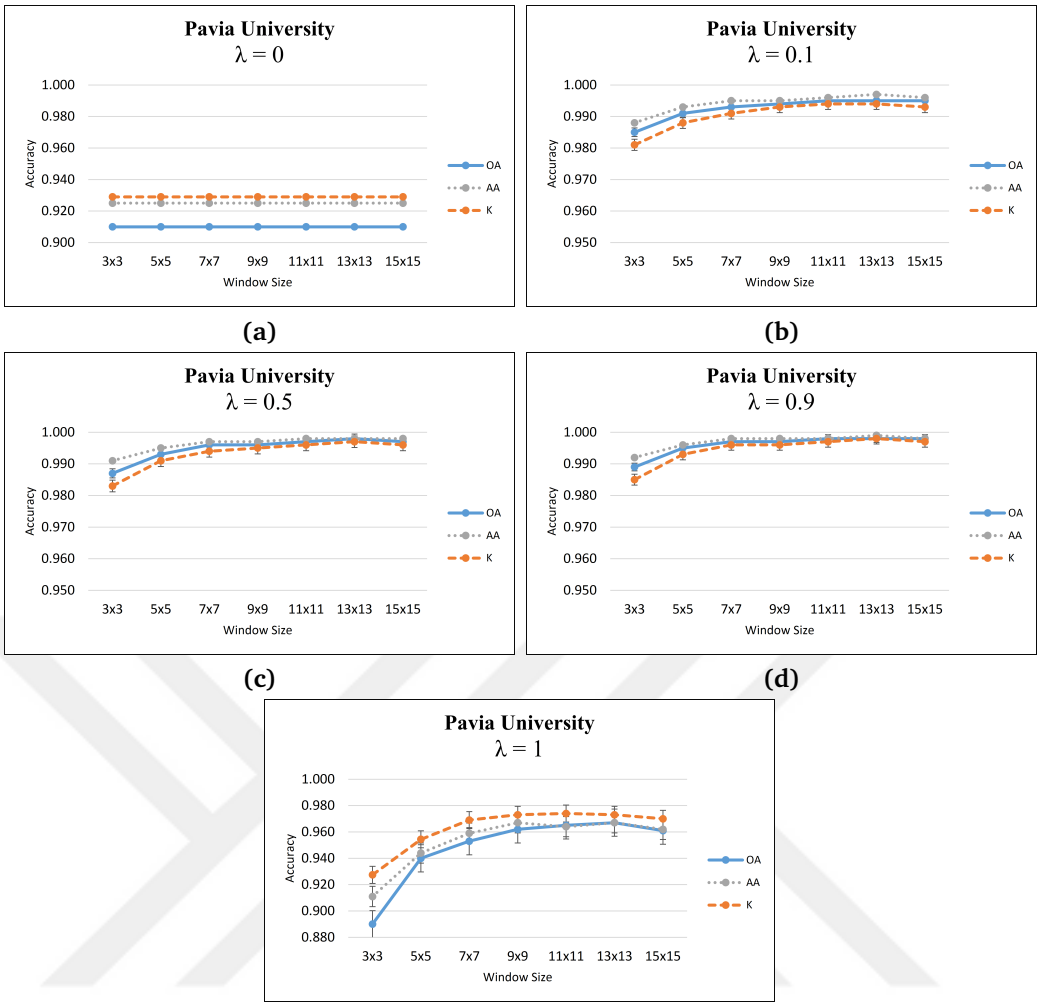

Figure 5.2 Overall Accuracy (OA), Average Accuracy (AA) and Kappa (κ) results of proposed HCKBoost method constructed with contribution of spatial kernels (i.e. relatively spatial features extracted with utilizing various window sizes) using (a) $\lambda = 0$, (b) $\lambda = 0.1$, (c) $\lambda = 0.5$, (d) $\lambda = 0.9$, (e) $\lambda = 1$ values for Pavia University HSI

and settlement diversity of classes on 2 dimensional plane, so optimal window size may differ from one HSI data to another.

In most cases, ensemble performance increases with the increasing number of trials. However, final results show the robustness of our proposed method even in the case of having a small ensemble. In summary, the use of spatial information has been influential on the hyperspectral image analyses if the parameters are set properly. So, window size is a situation that needs to be adjusted and may be considered as a future work. Also, extracting more sophisticated spatial features may provide further improvements.

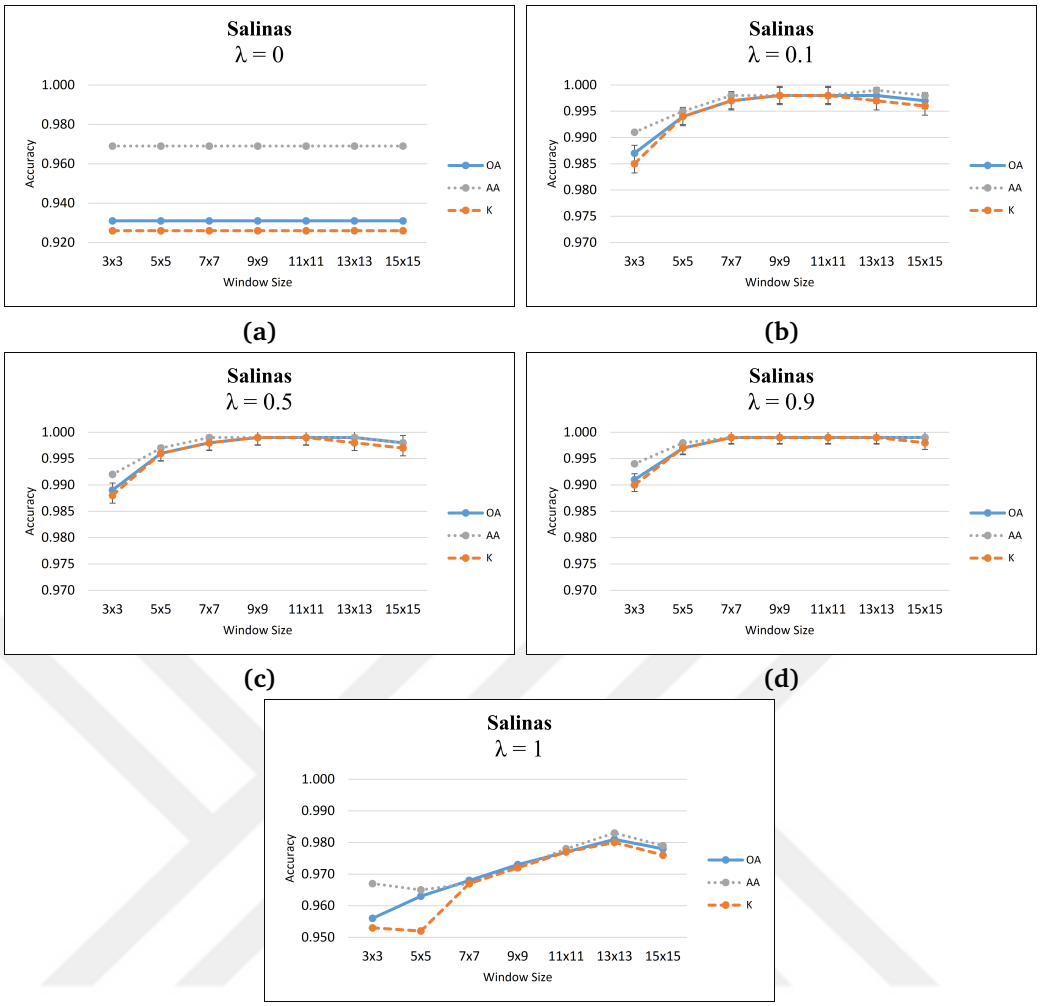

Figure 5.3 Overall Accuracy (OA), Average Accuracy (AA) and Kappa (κ) results of proposed HCKBoost method constructed with contribution of spatial kernels (i.e. relatively spatial features extracted with utilizing various window sizes) using (a) $\lambda = 0$, (b) $\lambda = 0.1$, (c) $\lambda = 0.5$, (d) $\lambda = 0.9$, (e) $\lambda = 1$ values for Salinas HSI

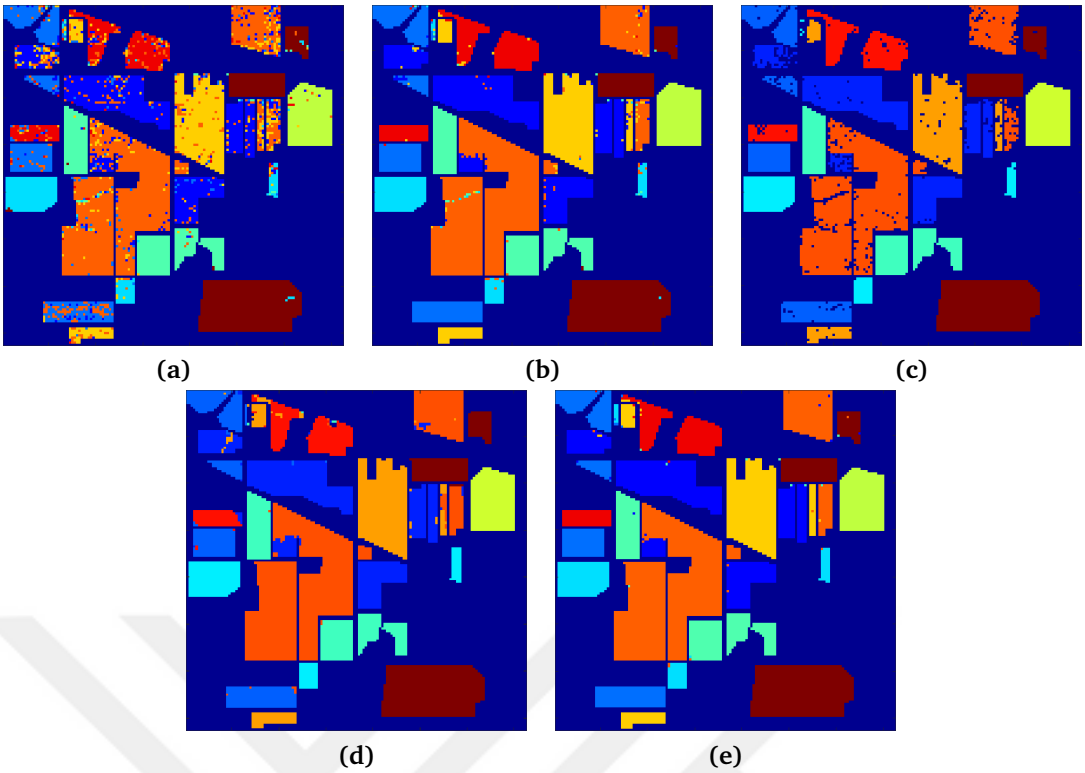

Figure 5.4 Classification maps obtained with (a) KELM, (b) CK-ELM, (c) MKBoost-D2, (d) MASR, and (e) Proposed HCKBoost (using $\lambda = 0.9$, 13×13 window size, $T = 10$, and 50% sub-sampling ratio) methods for Indian Pines HSI

Table 5.7 McNemar's Test Results for Indian Pines, Pavia University, and Salinas HSIs

McNemar's Test Results (Z value / selected hypothesis)			
Method Name	Indian Pines	Pavia University	Salinas
SVM	39.80/Yes	70.27/Yes	64.90/Yes
ELM	40.65/Yes	77.03/Yes	66.73/Yes
KSVM	38.73/Yes	52.69/Yes	65.98/yes
KELM	31.89/Yes	45.62/Yes	49.44/Yes
CK-SVM	31.16/Yes	36.00/Yes	50.65/Yes
CK-ELM	11.75/Yes	14.99/Yes	10.42/Yes
KOMP	51.16/Yes	127.36/Yes	83.13/Yes
KSOMP	41.46/Yes	92.39/Yes	61.76/Yes
MASR	5.25/Yes	31.66/Yes	**0.78/No**
SimpleMKL	27.09/Yes	126.43/Yes	145.79/Yes
L1MKL	25.50/Yes	126.36/Yes	146.26/Yes
L2MKL	22.29/Yes	126.04/Yes	146.13/Yes
SM1MKL	20.95/Yes	125.94/Yes	146.22/Yes
MKBoost-D1	22.12/Yes	126.55/Yes	146.12/Yes
MKBoost-D2	21.61/Yes	126.11/Yes	145.90/Yes
L1-MK-ELM	26.25/Yes	126.28/Yes	146.54/Yes
L2-MK-ELM	26.25/Yes	126.28/Yes	146.54/Yes

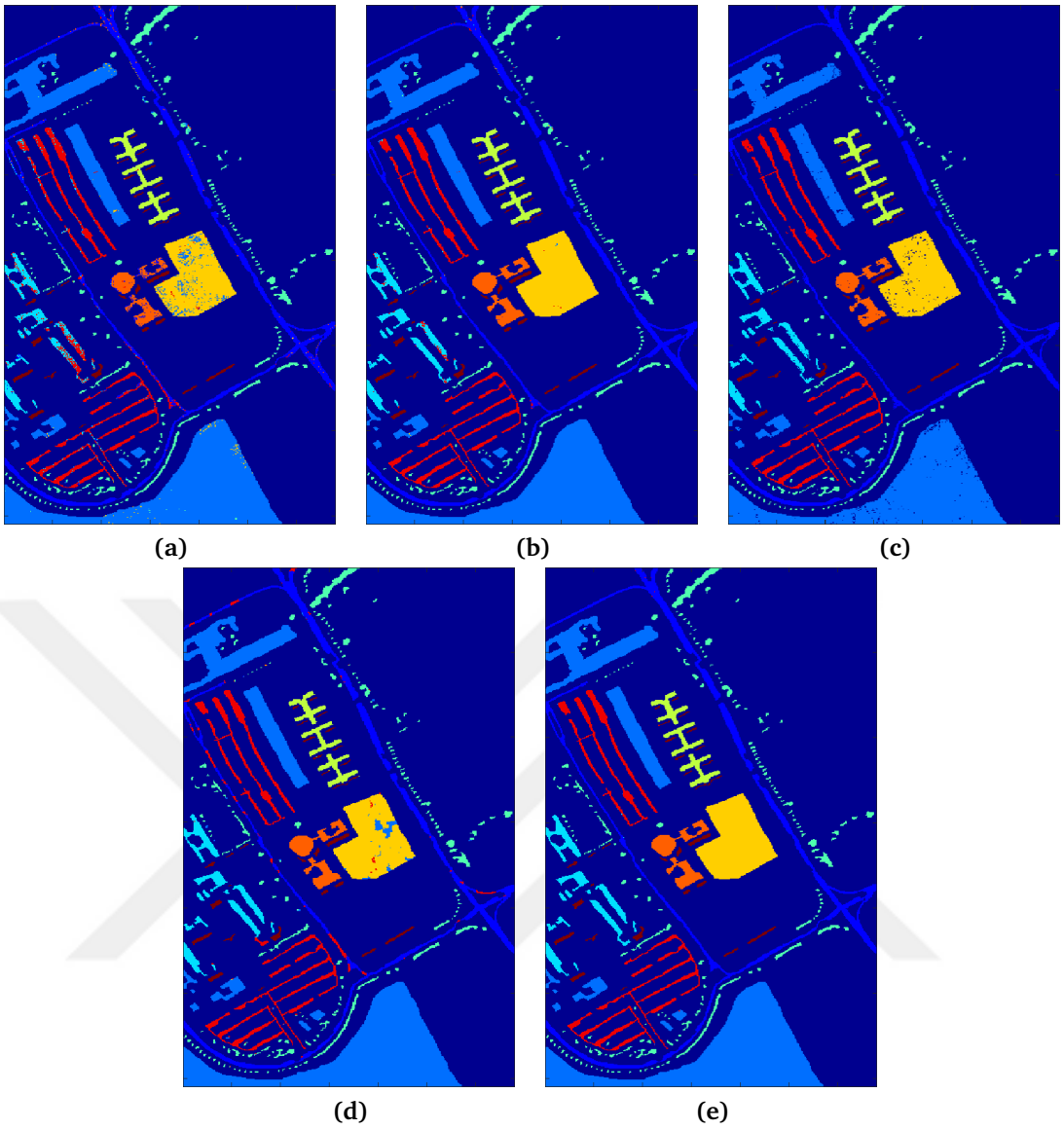

Figure 5.5 Classification maps obtained with (a) KELM, (b) CK-ELM, (c) MKBoost-D2, (d) MASR, and (e) Proposed HCKBoost (using $\lambda = 0.9$, 13×13 window size, $T = 10$, and 50% sub-sampling ratio) methods for Pavia University HSI

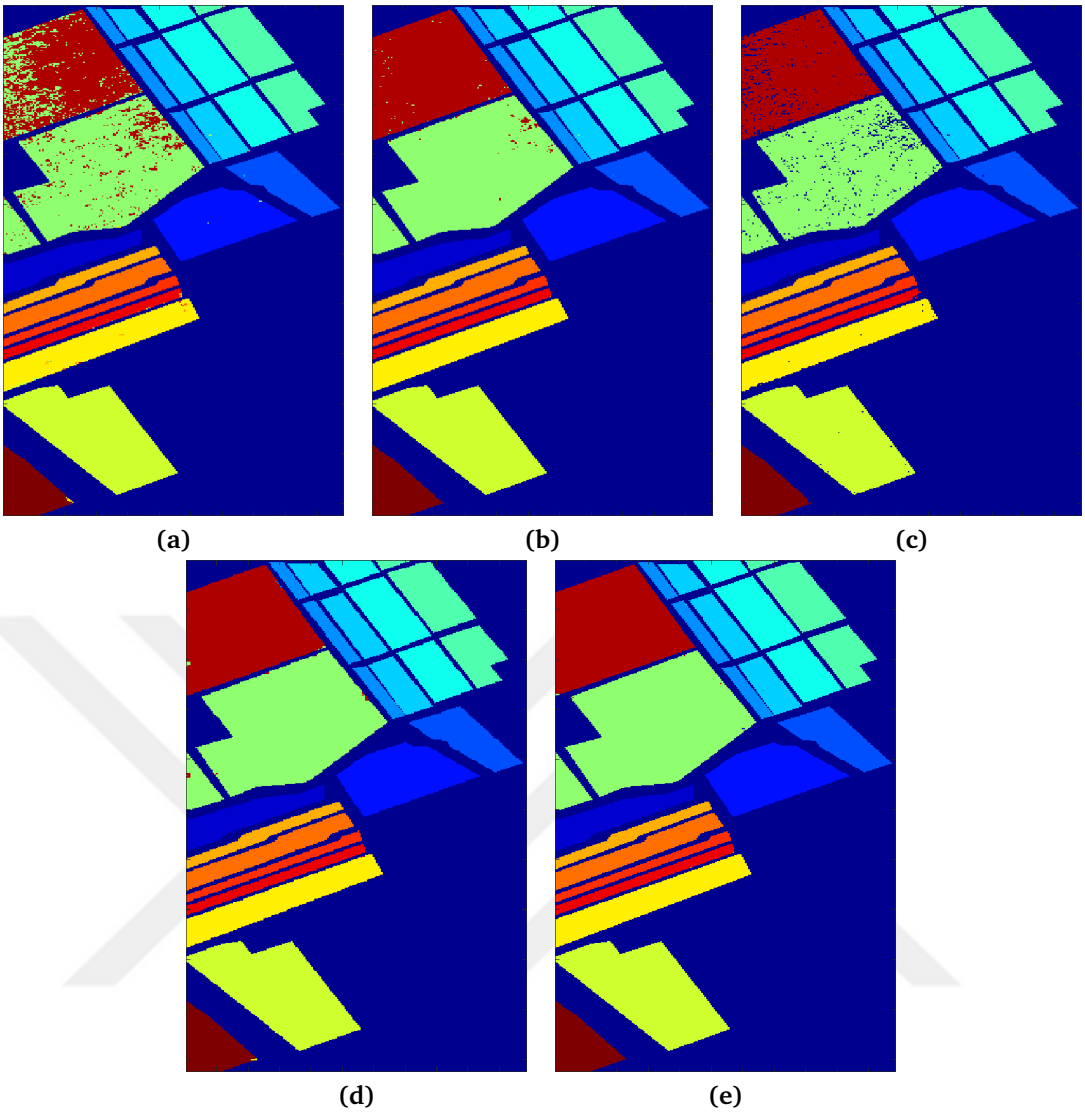

Figure 5.6 Classification maps obtained with (a) KELM, (b) CK-ELM, (c) MKBoost-D2, (d) MASR, and (e) Proposed HCKBoost (using $\lambda = 0.9$, 13×13 window size, $T = 10$, and 50% sub-sampling ratio) methods for Salinas HSI

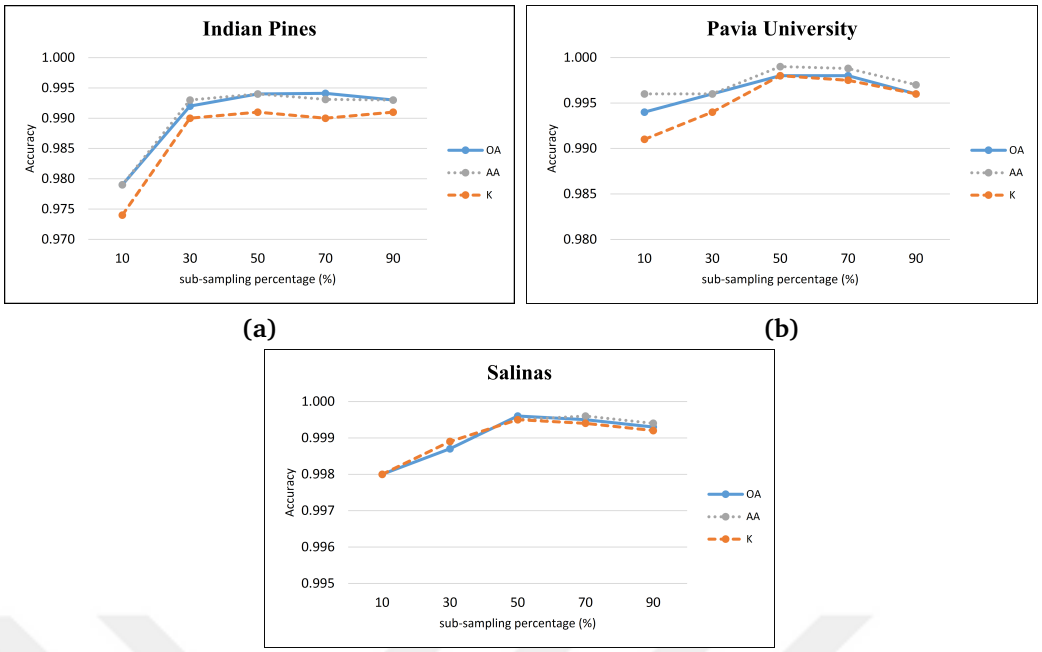

Figure 5.7 Overall Accuracy (OA), Average Accuracy (AA), and Kappa (κ) results of proposed HCKBoost method with respect to boosting sub-sampling ratio for (a) Indian Pines, (b) Pavia University, (c) Salinas HSIs (obtained using $\lambda = 0.9$, 13×13 window size, and $T = 10$).

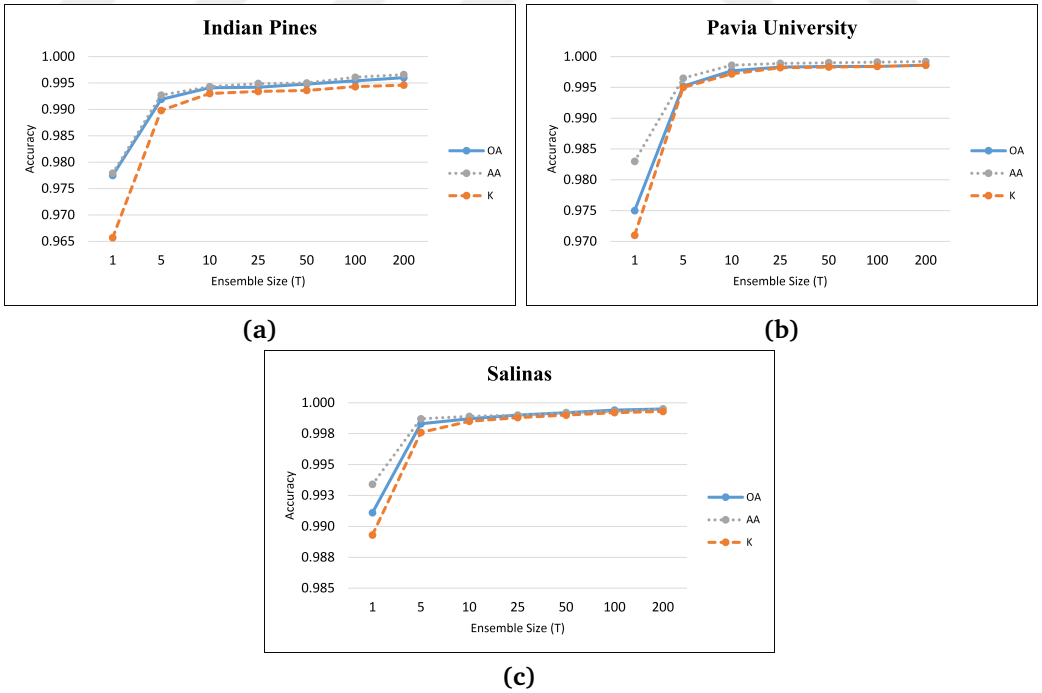

Figure 5.8 Overall Accuracy (OA), Average Accuracy (AA), and Kappa (κ) results of proposed HCKBoost method with respect to boosting ensemble size (T) for (a) Indian Pines, (b) Pavia University, (c) Salinas HSIs (obtained using $\lambda = 0.9$, 13×13 window size, and 50% sub-sampling ratio).

6
Multiple Composite Kernel Extreme Learning Machine

Multiple kernel (MK) learning (MKL) methods have significant impact on improving the classification performance. Besides that, composite kernel (CK) methods have high capability on the analysis of hyperspectral images (HSIs) due to making use of the contextual information. In this work, it is aimed to aggregate both CKs and MKs autonomously without the need of kernel coefficient adjustment manually. Convex combination of predefined kernel functions is implemented by using multiple kernel extreme learning machine (MK-ELM). Thus, complex optimization processes of standard MKL are disposed of and the facility of multi-class classification is profited. Different types of kernel functions are placed into MKs in order to realize hybrid kernel (HK) scenario. Proposed methodology is performed over Pavia University, Indian Pines, and Salinas hyperspectral scenes that have ground-truth information. Multiple composite kernels (MCKs) are constructed using Gaussian, polynomial, and logarithmic kernel functions with various parameters and then obtained results are presented comparatively along with the state-of-the-art standard machine learning, MKL, and CK methods.

6.1 Introduction

High-dimensional characteristics of hyperspectral images (HSIs) make them very compelling to work on [88]. Distorted signals, format errors, and spectral clutters are frequently encountered situations due to HSIs are remotely sensed from long distances [66]. In addition to them, inadequate and mislabeled samples make it a very challenging task to classify HSIs [89]. Kernel-based methods have been proposed on hyperspectral images in order to overcome these adverse situations [89], [90].

Kernel-based methods improve separation ability of a learning model on the linearly non-separable data by mapping data from original input space to high dimensional Hilbert space. Support vector machines (SVMs) and extreme learning machines (ELMs) are leading methods used in the field of kernel-based HSI classification in

recent years [122], [92], [95], [96]. Both methods provide remarkable performance and robust accuracy results. SVM is designed for binary classification through margin maximization operation and usually preferred for their large input handling and noisy sample dispatching capacity [94]. On the other hand, ELM stands out as a special sort of single hidden layer neural network that has very low run-time and provides an elegant solution for non-linear problems with least norm and least square solutions.

Assigning random weights to a neural network is a method that has been applied for a long time. An artificial neural network which is initialized with randomly assigned weights of the input layer and bias values of the hidden layer is proposed to be trained with using optimization process [123], [124]. So that, only hidden layer weight values need to be learned. Random link assignment based methods like "random vector functional link" show good generalization performance on different applications [125], [126]. On the other hand, randomly assigned network weights may cause instability. Thus, optimization of the random link values that are initially assigned to a feed forward neural network architecture is offered by using Fisher solution [127]. Another method named extreme learning machine that works with randomly assigned input weights and biases is claimed to have easy implementation, tendency to reach the smallest training error, and smallest norm of weights [50]. It is preferable due to its good generalization performance and high speed. ELM is also applicable to kernel base which provides more separability to a learning model.

Although, a kernel function provides effectiveness to a learning method, in the real-world problems, single predefined kernels are often inadequate to represent data obtained from heterogeneous sources or that have different representations. Multiple kernel learning (MKL) is a paradigm proposed to provide more accurate and flexible solutions to real-world problems by trying to find optimal combination of multiple kernels [54], [98]. In recent years, different variations of MKL have been proposed that have attempted to enhance accuracies on HSI analysis [99], [100]. Existing SVM based MKL methods have extensive usage. However, they have some negative aspects such as requirement of complicated optimization, low ability of scaling, and less effectiveness. Moreover, only binary classification ability of SVM reveals requirement of new strategies for multi-class classification. Multiple kernel extreme learning machine (MK-ELM) is proposed to eliminate the limitations of SVM based MKL approaches [118], [128]. In MK-ELM, optimal kernel is assumed to be obtained as a linear combination of base kernels and optimization process is carried out without the need of quadratic programming solver. In this paper, we have adopted the ELM based MK optimization process from the work in [118].

Considering spatial information along with spectral information brings added value

to hyperspectral image analysis. Performing spatial-spectral kernel combination on HSIs via SVM and kernel functions is reported with promising results and this combination is referred to as composite kernels (CKs) [61]. Since, HSIs usually contain high-volume data, SVM requires much more time to extrapolate. Thus, kernel ELM (KELM) based CK is proposed for the purpose of reducing run-time and increasing the classification performance on HSIs [108]. Remotely sensed HSI data is usually composed of compound distribution. Therefore, utilizing different kinds of kernel functions as much as contextual information may have crucial impact on HSI classification. Indeed, a hybrid kernel (HK) that consists of different type of kernels has been proposed and demonstrated its superiority against single type of kernels [60]. However, kernel coefficients need to be determined manually in this method.

In this chapter, extreme learning machine application of different types of kernels together with contextual information on HSIs is proposed. The MKL methodology for ELM is adapted to find optimal kernel combination. By taking contextual information into account through CKs, we named this method as multiple composite kernel extreme learning machine (MCK-ELM). The main contribution of this work can be summarized as follows: *1)* A new classification strategy is proposed for HSIs. Constructing multiple kernels with different types of kernel functions paves the way for exposing hidden features on a HSI scene. Incorporating spatial information with this procedure allows further inferences due to high correlation between neighboring pixels and enables to get promising results. *2)* Although, both HKs and CKs are beneficial on image analysis, manual kernel coefficient assignment makes usage of these methods difficult. Therefore, proposed methodology may thought to be an automated association of HKs and CKs for both individually and jointly. *3)* Opposite to standard MKL methods, computationally fast and effective ELM algorithm has been benefited. Results are obtained faster by using multi-class classification and by avoiding SVM-originated complicated optimization processes.

6.2 Multiple composite kernel extreme learning machine framework

When working with kernel based classifiers, kernel functions and parameters of these kernel functions have great impact on the results to be obtained. So far, the goal of the proposed MKL methods has been to find the optimal combination of the given P kernel functions ($\{\boldsymbol{K}_m(\cdot,\cdot)\}_{m=1}^{P}$). For this purpose, linear or nonlinear combination methods have been used. It is rational to choose linear combination methods in order to reduce complexity when using complex kernel functions such as radial based kernels [54]. As a linear method, configuration of weighted summation for multiple kernel is shown

in (6.1).

$$\boldsymbol{K}(\cdot,\cdot;\eta) = f_\eta(\{\boldsymbol{K}_m\}_{m=1}^P|\eta) = \sum_{m=1}^{P} \eta_m \boldsymbol{K}_m \tag{6.1}$$

In this expression, $\boldsymbol{K}(\cdot,\cdot;\eta)$ represents desired optimal kernel. f is hypothesis function learned from kernel based classifier. η is kernel weights and generally positive numbers are assigned to η during applications. Equation (6.1) can be expressed equivalently with feature mapping functions as in (6.2).

$$h(\cdot;\eta) = [\sqrt{\eta_1}h_1(\cdot),\cdots,\sqrt{\eta_m}h_m(\cdot),\cdots,\sqrt{\eta_P}h_P(\cdot)] \tag{6.2}$$

Various constraints are applied to η coefficients to ensure that the resulting multiple kernel is positive semi-definite. One of them is L_q norm and other one is being equal to or greater than zero constraint ($\eta_m \geq 0, \forall m$). In L_q norm, exponential coefficients of each η are set to q and sum of η values is equalized to 1 ($\sum \eta_m^q = 1$).

In multiple kernel extreme learning machine, structural parameters are learned by assuming that the optimal kernel is the linear combination of the base kernels. The objective function to be written with the imposed non-negative constraint for η coefficients and L_q norm is as in (6.3), where $q \geq 1$.

$$\begin{aligned} \min_{\eta} \min_{\boldsymbol{\beta},\xi} & \frac{1}{2}\sum_{m=1}^{P} \frac{||\tilde{\boldsymbol{\beta}}_m||_{\mathrm{F}}^2}{\eta_m} + \frac{C}{2}\sum_{i=1}^{n}||\xi_i||^2 \\ s.t. & \sum_{m=1}^{P} \tilde{\boldsymbol{\beta}}_m^\top h_m(\boldsymbol{x}_i) = \boldsymbol{y}_i - \xi_i, \quad \forall_i, \\ & \sum_{m=1}^{P} \eta_m^q = 1, \quad \eta_m \geq 0, \quad \forall_m \end{aligned} \tag{6.3}$$

Before this objective function is written, $\tilde{\boldsymbol{\beta}}_m = \sqrt{\eta_m}\boldsymbol{\beta}_m, m = 1,...,P$ conversion is made. Along with the specified constraints, Lagrangian of this objective function will be as in (6.4).

$$\begin{aligned} \mathscr{L}(\tilde{\boldsymbol{\beta}},\xi,\eta) = & \frac{1}{2}\sum_{m=1}^{P} \frac{||\tilde{\boldsymbol{\beta}}_m||_{\mathrm{F}}^2}{\eta_m} + \frac{C}{2}\sum_{i=1}^{n}||\xi_i||^2 \\ & - \sum_{t=1}^{T}\sum_{i=1}^{n} \alpha_{it}\Big(\sum_{m=1}^{P} \tilde{\boldsymbol{\beta}}_m^\top h_m(\boldsymbol{x}_i) - \boldsymbol{y}_i + \xi_i\Big) + \tau\Big(\sum_{m=1}^{P} \eta_m^q - 1\Big) \end{aligned} \tag{6.4}$$

In this equation, $\boldsymbol{\alpha}$ and τ represent Lagrange multipliers. Since, there is no possibility to η_m to become zero during the update process, transformation of inequality

constraint $\eta_m \geq 0$ is removed.

Finding dual form of Lagrangian of convex optimization problems allows reducing the number of coefficients to be optimized by turning the minimization problem into a maximization problem. In order to achieve this, Lagrangian needs to fulfill Karush-Kuhn-Tucker (KKT) optimality conditions. It is assumed that there is no duality gap in the convex optimization functions that satisfy the KKT optimality conditions. While stationary condition is given in (6.5), (6.6), and (6.7), primal feasibility condition is shown in (6.8) for Lagrangian given in (6.4).

$$\tilde{\boldsymbol{\beta}}_m = \eta_m \sum_{t=1}^{T} \sum_{i=1}^{n} \alpha_{it} h_m(\boldsymbol{x}_i), \forall_m \tag{6.5}$$

$$\xi_{ti} = \frac{\alpha_{ti}}{C}, \quad \forall_t \forall_i \tag{6.6}$$

$$\frac{1}{2} \frac{||\tilde{\boldsymbol{\beta}}_m||_{\mathrm{F}}^2}{\eta_m^2} - q\tau\eta_m^{q-1} = 0, \quad m = 1, ..., P \tag{6.7}$$

$$\sum_{m=1}^{P} \tilde{\boldsymbol{\beta}}_m^\top h_m(\boldsymbol{x}_i) = \boldsymbol{y}_i - \xi_i, \quad \forall_i \tag{6.8}$$

As can be seen, equations (6.5), (6.6), and (6.7) are calculated using gradients of Lagrangian. Equation (6.8) is obtained from differential which is taken with respect to $\boldsymbol{\alpha}$ multiplier. Equation (6.9) is acquired after applying (6.5) and (6.6) over the equation (6.8).

$$\sum_{t=1}^{T} \sum_{m=1}^{P} \alpha_{it} \left(\eta_m h_m(\boldsymbol{x}_i) h_m(\boldsymbol{x}_i) + \frac{1}{C} \right) = \boldsymbol{y}_i \tag{6.9}$$

This can be expressed in matrix form as in (6.10).

$$\left(\boldsymbol{K}(\cdot, \cdot; \eta) + \frac{\boldsymbol{I}}{C} \right) \boldsymbol{\alpha} = \boldsymbol{Y}^\top \tag{6.10}$$

$\boldsymbol{K}(\cdot, \cdot; \eta)$ represents the target multiple kernel. In this case, $\boldsymbol{\alpha}$ stands for a structural element of ELM and can be calculated by matrix inversion operation as in (6.11).

$$\boldsymbol{\alpha} = \left(\boldsymbol{K}(\cdot, \cdot; \eta) + \frac{\boldsymbol{I}}{C} \right)^{-1} \boldsymbol{Y}^\top \tag{6.11}$$

Equation (6.12) (which is needed for η coefficient update in every iteration) is obtained by combining equation (6.7) and $\sum_{m=1}^{P} \eta_m^q = 1$ equality constraint.

$$\eta_m' = \frac{||\tilde{\boldsymbol{\beta}}_m||_{\mathrm{F}}^{2/(1+q)}}{\left(\sum_{m=1}^{P} ||\tilde{\boldsymbol{\beta}}_m||_{\mathrm{F}}^{2q/(1+q)} \right)^{1/q}} \tag{6.12}$$

In this equation, η'_m expression shows updated coefficient to be used for m^{th} kernel and can be expressed as η_m^{t+1} alternatively. $||\tilde{\boldsymbol{\beta}}_m||_{\mathrm{F}}$ values are calculated according to (6.13).

$$||\tilde{\boldsymbol{\beta}}_m||_{\mathrm{F}} = \eta_m \sqrt{\boldsymbol{K}_m(\cdot,\cdot)\boldsymbol{\alpha}} \tag{6.13}$$

η' coefficients remain non-negative. Therefore, $\eta_m \geq 0$ inequality constraint in equation (6.3) is provided automatically. All iteration steps of L_q norm MCK-ELM are shown in algorithm 3.

Algorithm 3 L_q norm MCK-ELM

INPUTS:
$\{\boldsymbol{K}_m\}_{m=1}^{P} = \{\boldsymbol{K}_{m_w}^{w}, \boldsymbol{K}_{m_s}^{s}\}$: Predefined spatial ($\boldsymbol{K}^s$) and spectral ($\boldsymbol{K}^w$) kernels
$\boldsymbol{y}_i$: Output class values
q: L_q norm degree
C: Regulation parameter
γ: Threshold value for stop criterion
OUTPUTS:
$\boldsymbol{\alpha}$: ELM structural element
η: Multiple kernel coefficients
Initialization Parameters:
$\eta = \eta^0$ and $t = 0$:
repeat
    $\boldsymbol{K}(\cdot,\cdot;\eta) = \sum_{m=1}^{P} \eta_m^t \boldsymbol{K}_p$
    Update $\boldsymbol{\alpha}^t$ with equation (6.9)
    Update η^{t+1} with equation (6.10)
    $t = t + 1$
until $|\eta^{t+1} - \eta^t| \leq \gamma$

As can be seen from the algorithm 1, optimization process contains spatial and spectral predefined kernel collections as input in addition to standard MK-ELM. Both kernels are also hybridized with different type of kernel transfer functions and various parameters. Whole kernel formation details are discussed in the next section.

6.3 Experiments and results

In the experiments, proposed MCK-ELM methodology is compared with several state of the art methods and obtained classification results with different parameters are presented. Three different benchmark datasets are utilized to investigate performance

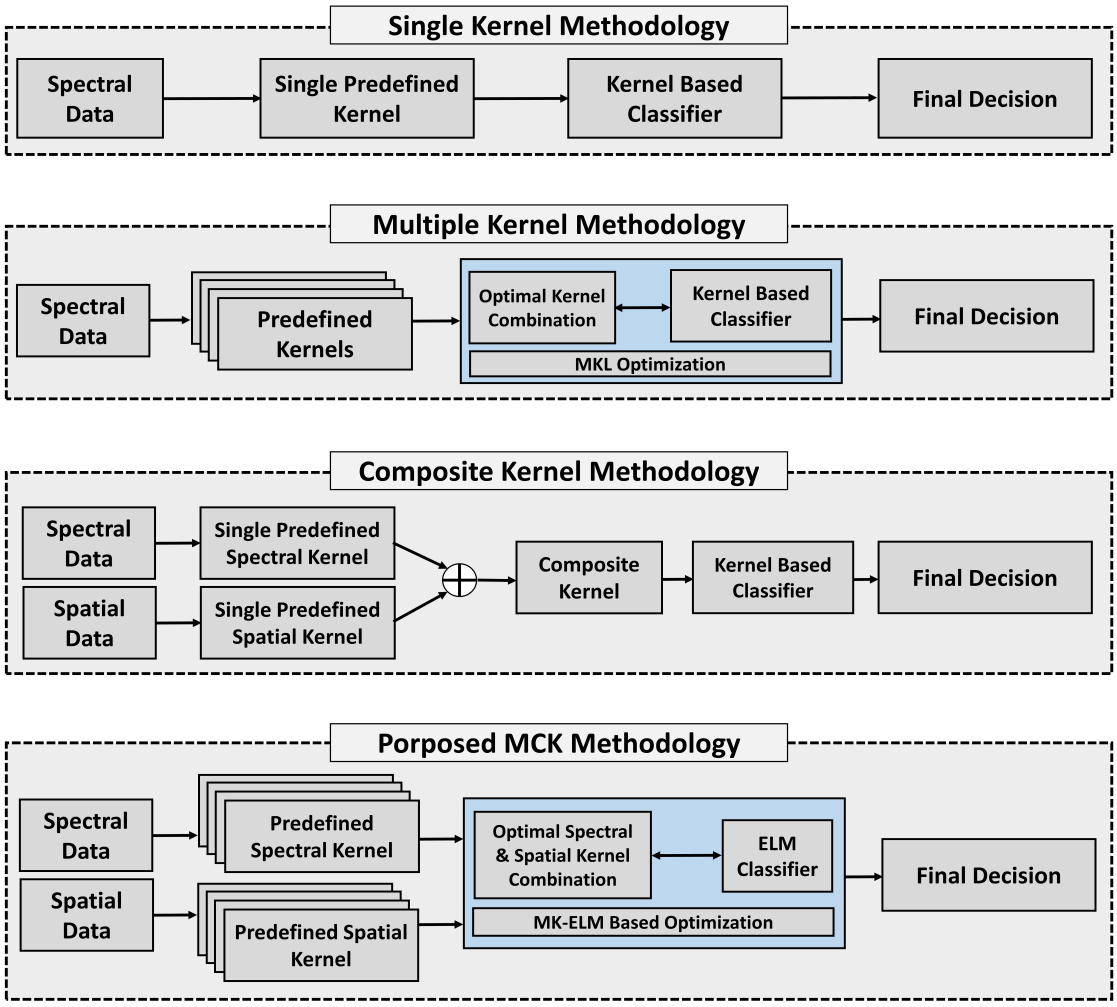

Figure 6.1 A general diagram for single kernel, multiple kernel, composite kernel, and proposed MCK methodologies

of the proposed method against state of the art methods. Pavia University, Indian Pines, and Salinas. Details of the experimental setup is discussed in the following subsections.

Table 6.1 Overall Accuracy (OA)(1 denotes %100 accuracy), Kappa (κ), and run Time(sec) results for Indian Pines, Pavia University, and Salinas HSIs

			Indian Pines			Pavia University			Salinas		
			OA ±*std*	κ ±*std*	Time(sec) ±*std*	OA ±*std*	κ ±*std*	Time(sec) ±*std*	OA ±*std*	κ ±*std*	Time(sec) ±*std*
Standard Machine Learning Methods		ELM	0.817	0.796	3.86	0.857	0.816	168.7	0.917	0.910	302.4
			±0.010	*±0.011*	*±0.185*	*±0.003*	*±0.004*	*±2.85*	*±0.003*	*±0.003*	*±22.86*
		SVM	0.827	0.807	68.45	0.868	0.832	2970	0.919	0.912	5108
			±0.010	*±0.011*	*±1.85*	*±0.016*	*±0.020*	*±48.65*	*±0.002*	*±0.002*	*±82.38*
		KELM	0.880	0.865	4.810	0.949	0.934	189.6	0.953	0.947	346.1
			±0.077	*±0.084*	*±0.250*	*±0.003*	*±0.041*	*±3.529*	*±0.013*	*±0.008*	*±26.39*
		KSVM	0.828	0.808	77.570	0.933	0.914	3409.0	0.918	0.910	5915.0
			±0.045	*±0.050*	*±2.010*	*±0.03*	*±0.039*	*±56.07*	*±0.02*	*±0.022*	*±93.01*
		SCN	0.826	0.805	236.2	0.889	0.857	203.7	0.922	0.914	1353.0
			±0.011	*±0.013*	*±5.67*	*±0.001*	*±0.002*	*±3.15*	*±0.002*	*±0.002*	*±34.22*
Composite Kernel Methods		CK-ELM	0.965	0.961	5.220	0.981	0.978	202.9	0.983	0.986	352.3
			±0.014	*±0.016*	*±0.630*	*±0.004*	*±0.005*	*±2.971*	*±0.004*	*±0.005*	*±31.03*
		CK-SVM	0.904	0.878	69.710	0.967	0.957	2373.0	0.951	0.947	4169.0
			±0.069	*±0.076*	*±1.370*	*±0.012*	*±0.016*	*±67.45*	*±0.007*	*±0.008*	*±31.88*
Multiple Kernel Methods		SimpleMKL	0.925	0.874	1947.0	0.973	0.874	9632.0	0.914	0.876	13233.0
			±0.097	*±0.152*	*±8.34*	*±0.034*	*±0.152*	*±380.8*	*±0.138*	*±0.184*	*±1973*
		SM1MKL	0.932	0.884	742.8	0.977	0.957	13794.0	0.974	0.954	7227.0
			±0.068	*±0.011*	*±5.12*	*±0.024*	*±0.044*	*±158.9*	*±0.019*	*±0.036*	*±1809*
		L1MKL	0.927	0.876	871.0	0.973	0.950	19185.0	0.973	0.953	11476.0
			±0.086	*±0.137*	*±5.43*	*±0.034*	*±0.061*	*±436.4*	*±0.021*	*±0.040*	*±231.5*
		L2MKL	0.931	0.882	882.0	0.976	0.955	21492.0	0.974	0.954	12056.0
			±0.082	*±0.132*	*±6.74*	*±0.026*	*±0.048*	*±553.24*	*±0.020*	*±0.037*	*±245.7*
		MKBoost-D1	0.929	0.879	348.1	0.975	0.953	7674.0	0.975	0.955	4102.0
			±0.085	*±0.134*	*±1.89*	*±0.037*	*±0.068*	*±197.2*	*±0.019*	*±0.036*	*±129.3*
		MKBoost-D2	0.931	0.882	378.8	0.977	0.956	8597.0	0.976	0.957	4805.0
			±0.074	*±0.119*	*±6.54*	*±0.027*	*±0.050*	*±219.1*	*±0.018*	*±0.034*	*±102.1*
		$L1$-MK-ELM	0.898	0.859	113.0	0.940	0.922	1871.0	0.941	0.935	4205.0
			±0.014	*±0.006*	*±4.967*	*±0.001*	*±0.001*	*±43.78*	*±0.001*	*±0.002*	*±167.3*
		$L2$-MK-ELM	0.917	0.866	101.4	0.943	0.926	1508.0	0.940	0.935	4387.0
			±0.009	*±0.003*	*±19.54*	*±0.001*	*±0.001*	*±33.45*	*±0.002*	*±0.002*	*±178.7*
Proposed		**$L1$-MCK-ELM**	0.981	0.978	113.0	0.990	**0.991**	1877.0	**0.998**	**0.998**	4057.0
			±0.024	*±0.027*	*±4.846*	*±0.009*	*±0.011*	*±31.06*	*±0.003*	*±0.004*	*±125.6*
		$L2$-MCK-ELM	**0.984**	**0.981**	96.3	**0.991**	0.989	1484.0	0.985	0.983	3971.0
			±0.001	*±0.001*	*±18.260*	*±0.008*	*±0.010*	*±18.26*	*±0.028*	*±0.031*	*±70.74*

Three different types of kernel transfer functions are taken into account during kernel formations: Gaussian transfer function (GTF) (equation (6.14)), polynomial transfer function (PTF) (equation (6.15)), and logarithmic transfer function (LTF) (equation (6.16)).

$$\boldsymbol{GTF}: K(\boldsymbol{u},\boldsymbol{v}) = exp\Big(-\frac{\|\boldsymbol{u}-\boldsymbol{v}\|^2}{2\sigma^2}\Big) \tag{6.14}$$

$$\boldsymbol{PTF}: K(\boldsymbol{u},\boldsymbol{v}) = (\lambda(\boldsymbol{u}-\boldsymbol{v})+r)^d, \lambda > 0 \tag{6.15}$$

$$\boldsymbol{LTF}: K(\boldsymbol{u},\boldsymbol{v}) = -\log\big(\|\boldsymbol{u}-\boldsymbol{v}\|^r + 1\big) \tag{6.16}$$

Proposed MCK-ELM methodology is compared with the various state-of-the-art composite kernel and multiple kernel algorithms. Also some basic machine

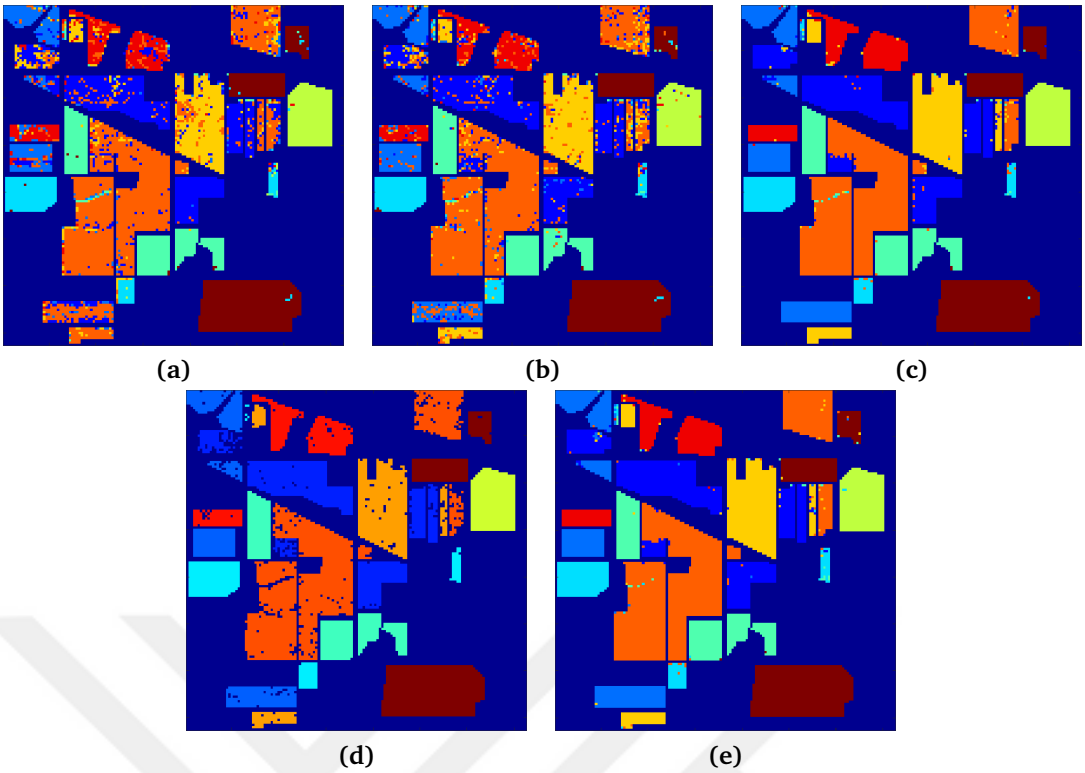

Figure 6.2 Classification maps for Indian Pines HSI acquired with using (a) SCN, (b) KELM, (c) CK-ELM, (d) MKBoost-D2, and (e) $L2$-MCK-ELM

learning algorithms such as SVM [111], ELM [50], KSVM [112], KELM [51], and stochastic configuration networks (SCN) [126] are inserted into the comparison scheme. CK-SVM [61] and CK-ELM [108] algorithms are the composite kernel based state-of-the-art methods. Multiple kernel based algorithms used for comparison are as follows: SimpleMKL [115], SM1MKL [116], LpMKL [117], MK-Boost [56], and MK-ELM [118]. General flowchart of all methodologies are demonstrated as a diagram in Figure 6.1.

Single Gaussian kernel is used for both basic kernel based classifiers (KSVM and KELM) and composite kernel based classifiers (CK-SVM and CK-ELM). Following 12 different kernels are defined for the MKL algorithms: Gaussian kernel that has 9 different widths $(2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4})$, polynomial kernels that have 2 different degrees (1 and 2), logarithmic kernel with the parameter $r = 2$. Since, logarithmic kernel is conditionally positive definite for $0 < r \leq 2$ range [120], r value is chosen between this range.

MCK-ELM consists of spatial and spectral parts. Each part is composed of different types of kernels. Thus, each part can be considered as a hybrid kernel in itself. Gaussian kernels with 4 different widths $(2^{-2}, 2^{-1}, 2^{1}, 2^{2})$, polynomial kernel with $d = 2$, and logarithmic kernel with $r = 2$ parameters are used to construct spatial and

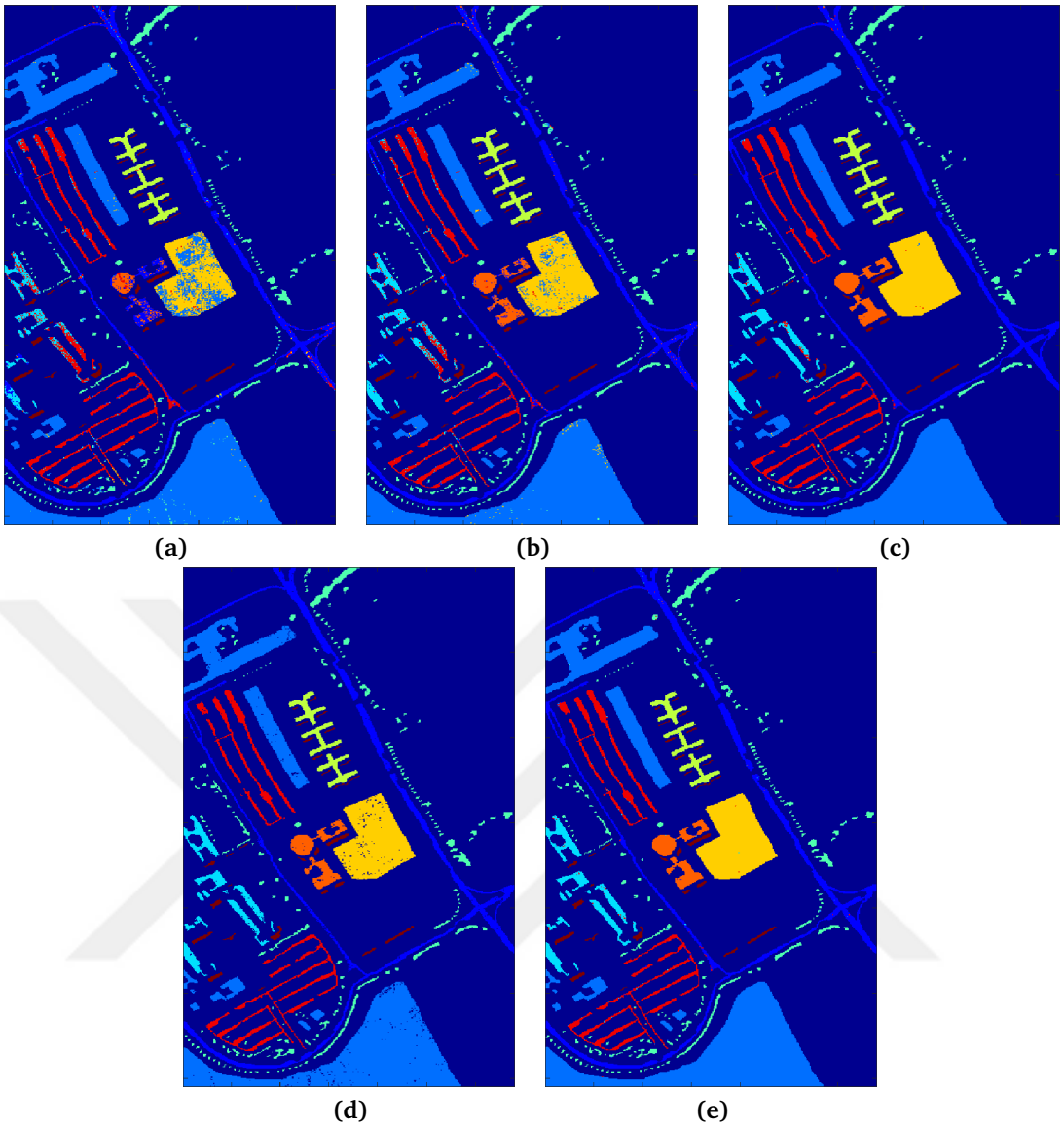

Figure 6.3 Classification maps for Pavia University HSI acquired with using (a) SCN, (b) KELM, (c) CK-ELM, (d) MKBoost-D2, and (e) $L2$-MCK-ELM

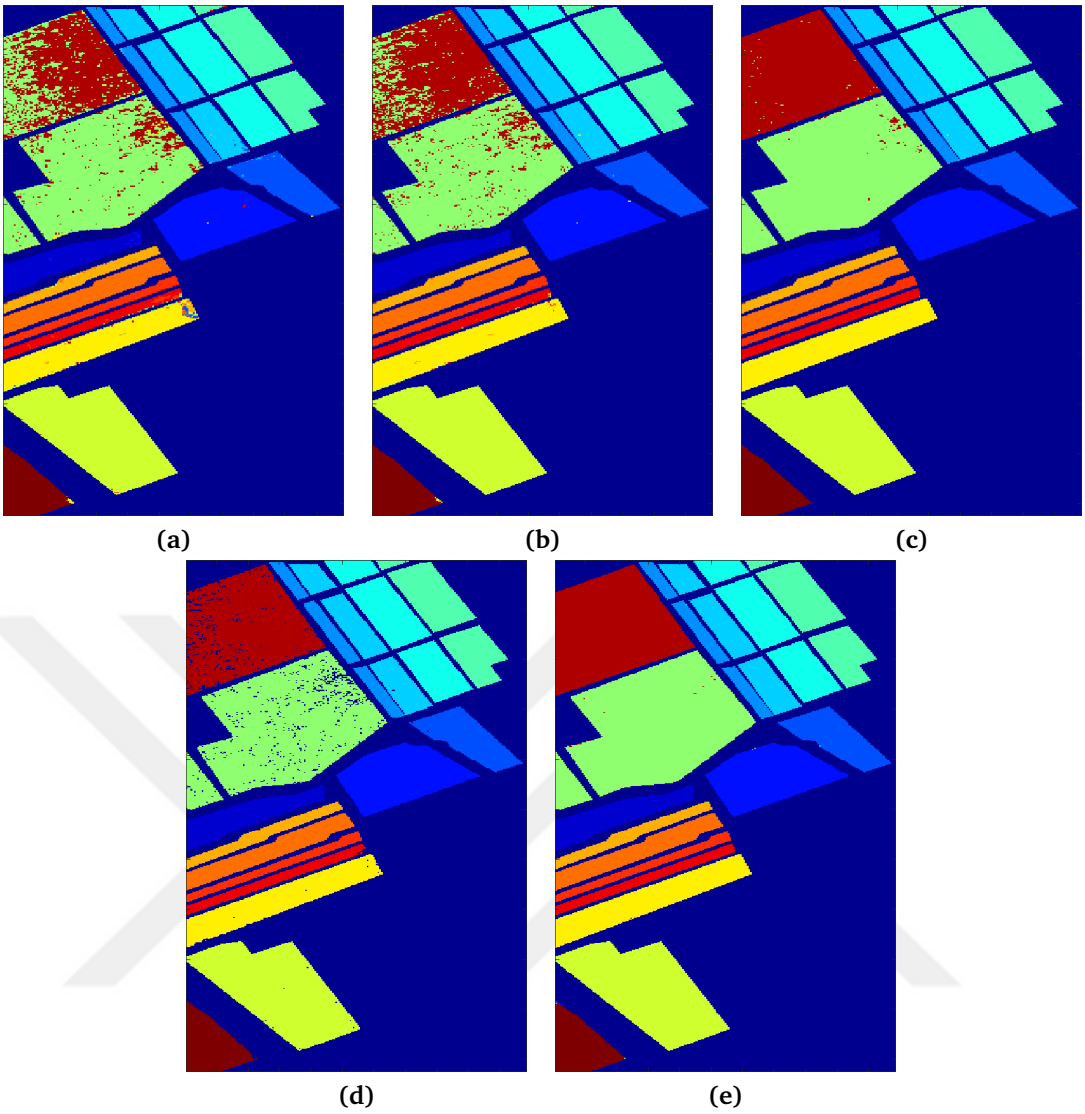

Figure 6.4 Classification maps for Salinas HSI acquired with using (a) SCN, (b) KELM, (c) CK-ELM, (d) MKBoost-D2, and (e) $L1$-MCK-ELM

spectral kernels individually. In summary, 6 kernels are utilized for both spatial and spectral kernels separately. As a result, 12 different kernels are defined for MCK-ELM just as in the other MKL algorithms.

Simple mean statistic is utilized as a spatial feature extraction method. To do this, windowing areas are defined for each pixel $\boldsymbol{x}_i \in \mathbb{R}^N$. Success of mean statistic in the sense of representing a group of pixel's mutual reflection has been shown in the previous CK based studies [61], [108]. In our case, 7×7 window is structured and applied on each HSI scene. Input weights and hidden nodes' bias values of ELM classifier are assigned randomly from uniform distribution between the range of [-1,1]. Also, k-fold cross validation is performed to divide data into testing and training. k value is set to 10 for simulations. In this way, we were able to run algorithm 10 times. Final result is obtained after averaging 10 results. Therewithal, occurrence of a potential system instability, on account of a classifier which has randomly assigned input weights and bias values, is tried to be eliminated.

6.3.1 Statistical evaluation

McNemar's test is an objective and statistical criterion, utilized for comparing the obtained classification results with the state-of-the-art methods. This test is a common way to understand whether classifiers' results are statistically different or not. We have chosen first degree of freedom and 5% significance level that corresponds to 1.96 value. That means, two classifiers' results are statistically different from each other if the obtained $|Z|$ value is bigger than 1.96. Obtained McNemar's test results for $L1$-MCK-ELM and $L2$-MCK-ELM are presented in Table 6.2 and Table 6.3 respectively.

6.3.2 Discussion

Overall Accuracy (OA) and Kappa (κ) statistic results of proposed method along with the results of state-of-the-art methods are demonstrated in Table 6.1 for Indian Pines, Pavia University, and Salinas HSIs respectively. It is clearly seen that MCK-ELM performs better in terms of accuracy against other algorithms. Improved results are observed in classification maps obtained by standard machine learning, CK, MKL, and MCK-ELM methods in Figure 6.2, Figure 6.3, and Figure 6.4 for related HSIs. Since, SVM is a binary classifier, one vs. all strategy is utilized in all SVM based MKL methods. Thus, some classification maps may contain unlabeled areas.

$L1$ and $L2$ norm constraints are applied to proposed method and referred them to as $L1$-MCK-ELM and $L2$-MCK-ELM on Table 6.1. These two constraints lead to different accuracy results. Since, $L1$ norm promotes sparsity, different data sets may

Table 6.2 McNemar's Test Results of $L1$-MCK-ELM for Indian Pines, Pavia University, and Salinas HSIs ($|Z|$ value / selected hypothesis)

Method Name	Indian Pines	Pavia University	Salinas
ELM	36.72/Yes	74.62/Yes	66.39/Yes
SVM	35.34/Yes	67.27/Yes	64.58/Yes
KELM	26.32/Yes	40.89/Yes	49.09/Yes
KSVM	34.89/Yes	48.74/Yes	65.66/Yes
SCN	94.84/Yes	65.49/Yes	62.34/Yes
CK-ELM	2.05/Yes	3.07/Yes	9.38/Yes
CK-SVM	25.93/Yes	30.06/Yes	50.37/Yes
SimpleMKL	20.72/Yes	124.72/Yes	145.67/Yes
SM1MKL	13.19/Yes	124.19/Yes	146.08/Yes
L1MKL	18.79/Yes	124.64/Yes	146.13/Yes
L2MKL	14.91/Yes	124.30/Yes	145.98/Yes
MKBoost-D1	14.47/Yes	124.78/Yes	145.99/Yes
MKBoost-D2	13.91/Yes	124.34/Yes	145.76/Yes
$L1$-MK-ELM	19.75/Yes	124.58/Yes	146.41/Yes
$L2$-MK-ELM	19.75/Yes	124.58/Yes	146.41/Yes

Table 6.3 McNemar's Test Results of $L2$-MCK-ELM for Indian Pines, Pavia University, and Salinas HSIs ($|Z|$ value / selected hypothesis)

Method Name	Indian Pines	Pavia University	Salinas
ELM	38.69/Yes	74.60/Yes	66.15/Yes
SVM	37.48/yes	67.36/Yes	64.32/Yes
KELM	29.22/Yes	40.97/Yes	48.68/Yes
KSVM	36.95/Yes	48.68/Yes	65.44/Yes
SCN	95.25/Yes	65.30/Yes	62.11/Yes
CK-ELM	5.79/Yes	3.14/Yes	7.85/Yes
CK-SVM	28.81/Yes	30.06/Yes	50.05/Yes
SimpleMKL	24.15/Yes	124.76/Yes	145.56/Yes
SM1MKL	17.15/Yes	124.24/Yes	145.97/Yes
L1MKL	22.45/Yes	124.69/Yes	146.02/Yes
L2MKL	18.71/Yes	124.33/Yes	145.87/Yes
MKBoost-D1	18.33/Yes	124.83/Yes	145.89/Yes
MKBoost-D2	17.72/Yes	124.35/Yes	145.68/Yes
$L1$-MK-ELM	23.27/Yes	124.60/Yes	146.30/Yes
$L2$-MK-ELM	23.27/Yes	124.60/Yes	146.30/Yes

produce different outcomes for this constraint. So that, $L1$-MCK-ELM shows better performance than $L2$-MCK-ELM on the Salinas HSI scene. That means, some of the utilized spatial and spectral kernel weights (η) vanish during $L1$ optimization. This constraint may be helpful in case of having so many kernels and not knowing which one is useful.

Algorithms' run time results are also shown in Table 6.1. Run time is measured according to total time consumption in training and testing phases. Standard machine learning and composite kernel algorithms usually reach the goal in only one step. So, these algorithms show relatively low run time compared to MKL methods. Since, our proposed algorithm's optimization phase is based on MK-ELM method, these two algorithms perform similar time consumption. Variations of the kernel constructions may cause small differences. Although, MKBoost method is proposed as an ensemble method, it does not operate on outstanding base. Of course, using larger ensembles may produce more accurate results. But, there is a trade-off between time and accuracy.

McNemar's test results on Table 6.2 and Table 6.3 indicate how much our proposed algorithm statistically differ from others. As shown in tables, null hypothesis is accepted for all cases. That means, proposed method has produced not only numerically but also statistically better results than the other methods. McNemar's test utilizes thematic maps for contingency matrix construction. Thus, most similar classification maps yield smaller $|Z|$ values. These circumstances can be shown for classification maps of the CK-ELM methods compared to proposed method's results in Figure 6.2, Figure 6.3, and Figure 6.4.

6.4 Conclusion

In this chapter, we proposed a new strategy which combines CKs and HKs via ELM based MKL algorithm. The proposed method is compared with numerous state-of-the-art MKL and CK methods. All results presented in the previous section show advantageous position of MCK-ELM compared to others in terms of accuracy. It is also a cost-effective solution as it does not require complex optimization tasks. In contrast to classical CK methods, manual arrangement of the spatial and spectral parts are not needed anymore. Joining these two domains is automated within the MCK-ELM. The spatial and spectral kernels are individually hybridized by using different type of kernels with various parameters. As a result of this, final kernel comes up with as a HK as much as a CK, naturally. Essentially, taking contextual information into account gave the fruit on HSI classification. Although, simple mean statistic is

used for spatial feature extraction via windowing structure, promising results have been achieved. Since, each HSI data have different pixel resolution, optimal window width may differ from one HSI data to another. Thus, window size is a parameter that needs to be adjusted and may be considered as a future work. Also, more sophisticated spatial feature extraction methods may provide some improvement.

7
Results And Discussion

In this thesis, HSI classification is addressed along with EnLe. Because HSI processing is a challenging task due to high-dimensional structure of images, processing raw data exclusively is usually inadequate to extrapolate. Therefore, some assisting methods need to be used, to which end approaches based on multiple instances, composite and hybrid kernels, and multiple kernels have been proposed.

Multiple classifier systems have become exceedingly popular in recent years due to their high performance compared to traditional learning systems based on single classifiers. Since labeling an image is a typically unreliable, expensive procedure, the ground-truth information of an image is generally insufficient. In addition, some valuable high-volume data in an image dataset such as in HSIs often remain unused. Using unlabeled data together with labeled ones, however, allows expanding the limited instance space and obtaining advanced analytical results. The proposed ensemble methods based on multiple instances for HSI yielded such enhanced classification results. Furthermore, random feature subspace selection seems to have reduced classification errors. Since the use of contextual information is pivotal in HSI analysis, taking spatial information into account via a windowing structure allows the use of unlabeled data in MIL methods. In that way, more sophisticated results can be obtained with the help of multiple classifier systems.

Diversity is another important factor in an ensemble classifier that needs to be ensured. Although bootstrap aggregation (i.e., bagging) and the random subspace methods are effective approaches for providing systems with diversity, boosting-based methods are more effective for classification. At the same time, the original feature space of data may not always possess linear separability. In that case, data transformation is needed to increase the linear separation ability of the classifier, for which kernel functions can be used. Although a kernel function improves the linear separation ability of a classifier, a single kernel function may not be sufficient for that task. Thus, systems with multiple kernels should be used to provide a more effective solution. Since hyperspectral data usually contain compound distribution, some kernel mapping

functions greatly influence classification. Furthermore, using more than one kernel function may become mandatory to ensure the maximum coverage of the distribution of the data. MKL seems suitable for that purpose, though it usually requires more complicated optimization processes, especially in SVM-based algorithms. To eliminate that unfavorable situation, a multiple kernel optimization approach based on ELMs has been proposed herein.

Although MKL seems able to represent compound data distribution, it usually requires a complicated optimization process. To reduce time complexity and to take different types of kernel functions into account, hybrid kernels have been proposed for use in HSI analysis. The convex combination of different kinds of kernels allow a less complex, more effective solution, as well as the use of spatial information. Using both hybrid and composite kernels has thus been proposed, both of which are used in hybridized composite kernel boosting and multiple composite kernel ELM approaches. In each proposed method, ELM was chosen as base learner to accelerate the completion of multiclass classification.

In most cases, the performance of an ensemble increases along with the number of base classifiers. However, increasing the number of trials can also require more time for computation. In that sense, there is a trade-off between time complexity and a system's success, which underscores the importance of having an ensemble system that yields robust results even fewer base classifiers are present. Therefore, the methods proposed herein may be regarded as examples of robust ensemble systems. Another factor that affects the success of HSI classification is spatial feature extraction. In all of our proposed methods, we have used simple mean statistics for spatial feature extraction, which, despite their simplicity, nevertheless yielded satisfactory results.

In sum, the use of contextual information heavily influences the effectiveness of HSI analysis. However, because parameters of the windowing function depend on the data, and that parameters need to be set appropriately on a case-by-case basis. More sophisticated methods of spatial feature extraction may improve the success of classification, which can be considered as future work. At the same time, using different types of kernels can be helpful when working with a data exhibiting compound distribution, in which case kernel-type selection plays an important role. In particular, multiple composite kernel ELM provides an elegant way of choosing kernels, by proposing $L1$ norm optimization, which is handy in the presence of numerous kernels whose usefulness is not known.

References

[1] L. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Hyperspectral remote sensing image subpixel target detection based on supervised metric learning," *IEEE transactions on geoscience and remote sensing*, vol. 52, no. 8, pp. 4955–4965, 2014.

[2] K. S. Pennington, E. S. Schlig, and J. M. White, *Apparatus for color or panchromatic imaging*, US Patent 4,264,921, 1981.

[3] C.-h. Chen, *Image processing for remote sensing*. CRC Press, 2007.

[4] H. Jianghua, B. Lianfa, K. Jie, L. Bin, W. Liping, and Z. Baomin, "Multispectral low light level image fusion technique," in *Proceedings of Third International Conference on Signal Processing (ICSP'96)*, IEEE, vol. 2, 1996, pp. 890–893.

[5] M.-C. Perkinson, D. Lobb, M. Cutter, and R. Renton, "Low cost hyperspectral imaging from space," *Photogrammetrie Fernerkundung Geoinformation*, pp. 47–50, 2002.

[6] J. M. Cathcart, R. D. Bock, and R. Campbell, "Analysis of soil and environmental processes on hyperspectral infrared signatures of landmines," in *Transformational Science And Technology For The Current And Future Force: (With CD-ROM)*, World Scientific, 2006, pp. 534–540.

[7] D. G. Ferris, R. A. Lawhead, E. D. Dickman, N. Holtzapple, J. A. Miller, S. Grogan, S. Bambot, A. Agrawal, and M. L. Faupel, "Multimodal hyperspectral imaging for the noninvasive diagnosis of cervical neoplasia," *Journal of Lower Genital Tract Disease*, vol. 5, no. 2, pp. 65–72, 2001.

[8] M. Maggioni, G. L. Davis, F. J. Warner, F. B. Geshwind, A. C. Coppi, R. A. DeVerse, and R. R. Coifman, "Hyperspectral microscopic analysis of normal, benign and carcinoma microarray tissue sections," in *Optical Biopsy VI*, International Society for Optics and Photonics, vol. 6091, 2006, p. 60910I.

[9] D. Smith, K. Lawewnce, and B. Park, "Detection of fertility and early development of hatching eggs with hyperspectral imaging," in *Proc. 11th European Symposium on the Quality of Eggs and Egg Products Netherlands: World's Poultry Science Association*, 2005, pp. 176–180.

[10] M. S. Kim, Y. Chen, and P. Mehl, "Hyperspectral reflectance and fluorescence imaging system for food quality and safety," *Transactions of the ASAE*, vol. 44, no. 3, p. 721, 2001.

[11] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[12] L. Rokach, "Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography," *Computational Statistics & Data Analysis*, vol. 53, no. 12, pp. 4046–4072, 2009.

[13] L. I. Kuncheva, *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004.

[14] D. S. Weld, *Machine learning iv ensembles*, `https://courses.cs.washington.edu/courses/cse473/04sp/schedule/18-ensembles.pdf`, Accessed: 2019-01-13.

[15] J. Xia, "Multiple classifier systems for the classification of hyperspectral data," PhD thesis, Universite de Grenoble, 2014.

[16] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE transactions on systems, man, and cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.

[17] R. Ranawana and V. Palade, "Multi-classifier systems: Review and a roadmap for developers," *International Journal of Hybrid Intelligent Systems*, vol. 3, no. 1, pp. 35–61, 2006.

[18] L. Rokach, *Pattern classification using ensemble methods*. World Scientific, 2010, vol. 75.

[19] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[20] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and systems magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[21] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[22] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[23] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 7, pp. 1088–1099, 2006.

[24] R. E. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, no. 2, pp. 197–227, 1990.

[25] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[26] R. E. Schapire, Y. Freund, P. Bartlett, W. S. Lee, *et al.*, "Boosting the margin: A new explanation for the effectiveness of voting methods," *The annals of statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.

[27] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 10, pp. 1619–1630, 2006.

[28] X. Huang and L. Zhang, "An svm ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery," *IEEE transactions on geoscience and remote sensing*, vol. 51, no. 1, pp. 257–272, 2013.

[29] X. Ceamanos, B. Waske, J. A. Benediktsson, J. Chanussot, M. Fauvel, and J. R. Sveinsson, "A classifier ensemble based on fusion of support vector machines for classifying hyperspectral data," *International Journal of Image and Data Fusion*, vol. 1, no. 4, pp. 293–307, 2010.

[30] M. Chi, Q. Kun, J. A. Benediktsson, and R. Feng, "Ensemble classification algorithm for hyperspectral remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 4, pp. 762–766, 2009.

[31] B. Waske, S. van der Linden, J. A. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 7, pp. 2880–2889, 2010.

[32] H. Kwon and P. Rauss, "Feature-based ensemble learning for hyperspectral chemical plume detection," *International journal of remote sensing*, vol. 32, no. 21, pp. 6631–6652, 2011.

[33] Y. Chen, X. Zhao, and Z. Lin, "Optimizing subspace svm ensemble for hyperspectral imagery classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1295–1305, 2014.

[34] P. Ramzi, F. Samadzadegan, and P. Reinartz, "Classification of hyperspectral data using an adaboostsvm technique applied on band clusters," *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 7, no. 6, pp. 2066–2079, 2014.

[35] Y. Gu and H. Liu, "Sample-screening mkl method via boosting strategy for hyperspectral image classification," *Neurocomputing*, vol. 173, pp. 1630–1639, 2016.

[36] S. Kawaguchi and R. Nishii, "Hyperspectral image classification by bootstrap adaboost with random decision stumps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 11, pp. 3845–3851, 2007.

[37] P. Gurram and H. Kwon, "Sparse kernel-based ensemble learning with fully optimized kernel parameters for hyperspectral classification problems," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 787–802, 2013.

[38] G. Matasci, D. Tuia, and M. Kanevski, "Svm-based boosting of active learning strategies for efficient domain adaptation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 5, pp. 1335–1343, 2012.

[39] A. Merentitis, C. Debes, and R. Heremans, "Ensemble learning in hyperspectral image classification: Toward selecting a favorable bias-variance tradeoff," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1089–1102, 2014.

[40] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, 2005.

[41] S. Rajan, J. Ghosh, and M. M. Crawford, "Exploiting class hierarchies for knowledge transfer in hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3408–3417, 2006.

[42] K. Y. Peerbhay, O. Mutanga, and R. Ismail, "Random forests unsupervised classification: The detection and mapping of solanum mauritianum infestations in plantation forestry using hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 3107–3122, 2015.

[43] S. R. Joelsson, J. A. Benediktsson, and J. R. Sveinsson, "Random forest classifiers for hyperspectral data," in *Geoscience and Remote Sensing Symposium, 2005. IGARSS'05. Proceedings. 2005 IEEE International*, IEEE, vol. 1, 2005, 4–pp.

[44] S. Amini, S. Homayouni, A. Safari, and A. A. Darvishsefat, "Object-based classification of hyperspectral data using random forest algorithm," *Geospatial Information Science*, vol. 21, no. 2, pp. 127–138, 2018.

[45] X. Cao, R. Li, Y. Ge, B. Wu, and L. Jiao, "Densely connected deep random forest for hyperspectral imagery classification," *International Journal of Remote Sensing*, pp. 1–16, 2018.

[46] J. Xia, P. Du, X. He, and J. Chanussot, "Hyperspectral remote sensing image classification based on rotation forest," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 1, pp. 239–243, 2014.

[47] J. Xia, J. Chanussot, P. Du, and X. He, "Spectral–spatial classification for hyperspectral data using rotation forests with local feature extraction and markov random fields," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 5, pp. 2532–2546, 2015.

[48] J. Xia, M. Dalla Mura, J. Chanussot, P. Du, and X. He, "Random subspace ensembles for hyperspectral image classification with extended morphological attribute profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 9, pp. 4768–4786, 2015.

[49] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, no. 1, pp. 31–71, 1997.

[50] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[51] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.

[52] A. Ben-Israel and T. N. Greville, *Generalized inverses: theory and applications*. Springer Science & Business Media, 2003, vol. 15.

[53] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, pp. 80–86, 2000.

[54] M. Gonen and E. Alpaydin, "Multiple kernel learning algorithms," *Jour. of Machine Learning Research*, vol. 12, no. Jul, pp. 2211–2268, 2011.

[55] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European Conference on Computational Learning Theory, EuroCOLT'95*, Springer, 1995, pp. 23–37.

[56] H. Xia and S. C. Hoi, "Mkboost: A framework of multiple kernel boosting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1574–1586, 2013.

[57] R. Wille, "Restructuring lattice theory: An approach based on hierarchies of concepts," in *Ordered Sets*, Springer, 1982, pp. 445–470.

[58] S. Ding, Y. Zhang, X. Xu, and L. Bao, "A novel extreme learning machine based on hybrid kernel function," *Journal of Computers*, vol. 8, no. 8, pp. 2110–2117, 2013.

[59] C.-C. Li, A.-l. Guo, and D. Li, "Combined kernel svm and its application on network security risk evaluation," in *IEEE Int. Symp. on Intelligent Information Technology Application Workshops, IITA'08*, 2008, pp. 36–39.

[60] U. Ergul and G. Bilgin, "Hyperspectral image classification with hybrid kernel extreme learning machine," in *IEEE 25th Signal Processing and Communications Applications Conference, SIU'17*, 2017, pp. 1–4.

[61] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 93–97, 2006.

[62] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[63] A. J. Viera, J. M. Garrett, *et al.*, "Understanding interobserver agreement: The kappa statistic," *Family Medicine*, vol. 37, no. 5, pp. 360–363, 2005.

[64] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1–30, 2006.

[65] G. M. Foody, "Thematic map comparison," *Photogrammetric Engineering & Remote Sensing*, vol. 70, no. 5, pp. 627–633, 2004.

[66] J. Bolton and P. Gader, "Application of multiple-instance learning for hyperspectral image analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 5, pp. 889–893, 2011.

[67] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Advances in Neural Information Processing Systems, NIPS'98*, Morgan Kaufmann Publishers, 1998, pp. 570–576.

[68] C. Leistner, A. Saffari, and H. Bischof, "Miforests: Multiple-instance learning with randomized trees," in *European Conference on Computer Vision*, Springer, 2010, pp. 29–42.

[69] P. Torrione, C. Ratto, and L. M. Collins, "Multiple instance and context dependent learning in hyperspectral data," in *IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, WHISPERS'09*, IEEE, 2009, pp. 1–4.

[70] J. T. Cobb, X. Du, A. Zare, and M. Emigh, "Multiple-instance learning-based sonar image classification," in *SPIE Defense+ Security*, International Society for Optics and Photonics, 2017, 101820H–101820H.

[71] J. A. Benediktsson, X. C. Garcia, B. Waske, J. Chanussot, J. R. Sveinsson, and M. Fauvel, "Ensemble methods for classification of hyperspectral data," in *IEEE International Geoscience and Remote Sensing Symposium, IGARSS'08*, vol. 1, 2008, pp. I62–I65.

[72] K. Tan, J. Hu, J. Li, and P. Du, "A novel semi-supervised hyperspectral image classification approach based on spatial neighborhood information and classifier combination," *ISPRS J. Photogram. Remote Sens.*, vol. 105, pp. 19–29, 2015.

[73] Z.-H. Zhou and M.-L. Zhang, "Ensembles of multi-instance learners," in *European Conference on Machine Learning, ECML'03*, Springer, 2003, pp. 492–502.

[74] X. Kang, D. Li, and S. Wang, "A multi-instance ensemble learning model based on concept lattice," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1203–1213, 2011.

[75] P. Viola, J. C. Platt, C. Zhang, *et al.*, "Multiple instance boosting for object detection," in *Advances in Neural Information Processing Systems, NIPS'05*, vol. 2, 2005, pp. 1417–1426.

[76] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, 2011.

[77] X.-S. Xu, X. Xue, and Z.-H. Zhou, "Ensemble multi-instance multi-label learning approach for video annotation task," in *19th ACM international Conference on Multimedia*, ACM, 2011, pp. 1153–1156.

[78] U. Ergul and G. Bilgin, "Multiple instance bagging approach for ensemble learning methods on hyperspectral images," in *IEEE 23th Signal Processing and Communications Applications Conference, SIU'15*, IEEE, 2015, pp. 403–406.

[79] ——, "Multiple instance bagging based ensemble classification of hyperspectral images," in *Signal Processing and Communication Application Conference (SIU), 2016 24th*, IEEE, 2016, pp. 757–760.

[80] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of machine learning research*, vol. 5, no. Jan, pp. 101–141, 2004.

[81] B. Babenko, "Multiple instance learning: Algorithms and applications," University of California San Diego, San Diego, Tech. Rep., 2008.

[82] J. Wang and J.-D. Zucker, "Solving multiple-instance problem: A lazy learning approach," in *Proc. 17th International Conference on Machine Learning, ICML'00*, Morgan Kaufmann, 2000, pp. 1119–1125.

[83] Y. Chevaleyre and J.-D. Zucker, "Solving multiple-instance and multiple-part learning problems with decision trees and rule sets. application to the mutagenesis problem," in *Conference of the Canadian Society for Computational Studies of Intelligence*, Springer, 2001, pp. 204–214.

[84] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in Neural Information Processing Systems, NIPS'15*, MIT Press, 2003, pp. 577–584.

[85] L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean, "Boosting algorithms as gradient descent.," in *Advances in Neural Information Processing Systems, NIPS'99*, 1999, pp. 512–518.

[86] T. K. Ho, "Nearest neighbors in random subspaces," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, Springer, 1998, pp. 640–648.

[87] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogram. Remote Sens.*, vol. 114, pp. 24–31, 2016.

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

[89] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 45–54, 2014.

[90] M. Han and B. Liu, "Ensemble of extreme learning machine for remote sensing image classification," *Neurocomputing*, vol. 149, pp. 65–70, 2015.

[91] G.-P. Yang, H.-Y. Liu, and X.-C. Yu, "Hyperspectral remote sensing image classification based on kernel fisher discriminant analysis," in *IEEE Int. Conf. on Wavelet Analysis and Pattern Recognition, ICWAPR'07*, vol. 3, 2007, pp. 1139–1143.

[92] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 3, pp. 247–259, 2011.

[93] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[94] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.

[95] M. Pal, A. E. Maxwell, and T. A. Warner, "Kernel-based extreme learning machine for remote-sensing image classification," *Remote Sensing Letters*, vol. 4, no. 9, pp. 853–862, 2013.

[96] C. Chen, W. Li, H. Su, and K. Liu, "Spectral-spatial classification of hyperspectral image based on kernel extreme learning machine," *Remote Sensing*, vol. 6, no. 6, pp. 5795–5814, 2014.

[97] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Jour. of Machine Learning Research*, vol. 7, no. Jul, pp. 1531–1565, 2006.

[98] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *Proceedings of the 21st International Conference on Machine Learning, ICML'04*, 2004, p. 6.

[99] Y. Gu, J. Chanussot, X. Jia, and J. A. Benediktsson, "Multiple kernel learning for hyperspectral image classification: A review," *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

[100] Q. Wang, Y. Gu, and D. Tuia, "Discriminative multiple kernel learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 7, pp. 3912–3927, 2016.

[101] Y. Gu, C. Wang, D. You, Y. Zhang, S. Wang, and Y. Zhang, "Representative multiple kernel learning for classification in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 7, pp. 2852–2865, 2012.

[102] C. Qi, Y. Wang, W. Tian, and Q. Wang, "Multiple kernel boosting framework based on information measure for classification," *Chaos, Solitons & Fractals*, vol. 89, pp. 175–186, 2016.

[103] C. Qi, Z. Zhou, L. Hu, and Q. Wang, "A framework of multiple kernel ensemble learning for hyperspectral classification," in *IEEE Conference on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)'16*, 2016, pp. 456–460.

[104] Y. Wang, Y. Gu, G. Gao, and Q. Wang, "Hyperspectral image classification with multiple kernel boosting algorithm," in *IEEE International Conference on Image Processing, ICIP'14*, 2014, pp. 5047–5051.

[105] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2013.

[106] P. Quesada-Barriuso, F. Arguello, and D. B. Heras, "Spectral–spatial classification of hyperspectral images using wavelets and extended morphological profiles," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1177–1185, 2014.

[107] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3681–3693, 2015.

[108] Y. Zhou, J. Peng, and C. P. Chen, "Extreme learning machine with composite kernels for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2351–2360, 2015.

[109] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, 1996.

[110] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recognition Letters*, vol. 54, pp. 11–17, 2015.

[111] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[112] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[113] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Transactions on Geoscience and Remote sensing*, vol. 51, no. 1, pp. 217–231, 2013.

[114] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification via multiscale adaptive sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7738–7749, 2014.

[115] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Jour. of Machine Learning Research*, vol. 9, no. Nov, pp. 2491–2521, 2008.

[116] X. Xu, I. W. Tsang, and D. Xu, "Soft margin multiple kernel learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 749–761, 2013.

[117] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "Lp-norm multiple kernel learning," *Jour. of Machine Learning Research*, vol. 12, no. Mar, pp. 953–997, 2011.

[118] X. Liu, L. Wang, G.-B. Huang, J. Zhang, and J. Yin, "Multiple kernel extreme learning machine," *Neurocomputing*, vol. 149, pp. 253–264, 2015.

[119] T. Eckhard, E. M. Valero, J. Hernández-Andrés, and V. Heikkinen, "Evaluating logarithmic kernel for spectral reflectance estimation—effects on model parametrization, training set size, and number of sensor spectral channels," *Journal of the Optical Society of America A*, vol. 31, no. 3, pp. 541–549, 2014.

[120] S. Boughorbel, J.-P. Tarel, and N. Boujemaa, "Conditionally positive definite kernels for svm based image recognition," in *IEEE Int. Conf. on Multimedia and Expo, ICME'05*, 2005, pp. 113–116.

[121] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "A spatial–spectral kernel-based approach for the classification of remote-sensing images," *Pattern Recognition*, vol. 45, no. 1, pp. 381–392, 2012.

[122] G. Camps-Valls, L. Gómez-Chova, J. Calpe-Maravilla, J. D. Martin-Guerrero, E. Soria-Olivas, L. Alonso-Chordá, and J. Moreno, "Robust support vector method for hyperspectral data classification and knowledge discovery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 7, pp. 1530–1542, 2004.

[123] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: Theory, system architecture, and functionalities," *Computer*, vol. 25, no. 5, pp. 76–79, 1992.

[124] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1320–1329, 1995.

[125] M. Li and D. Wang, "Insights into randomized algorithms for neural networks: Practical issues and common pitfalls," *Information Sciences*, vol. 382, pp. 170–178, 2017.

[126] D. Wang and M. Li, "Stochastic configuration networks: Fundamentals and algorithms," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3466–3479, 2017.

[127] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, "Feedforward neural networks with random weights," in *Proc. of 11th IAPR International Conference on Pattern Recognition, Vol. II. Conference B: Pattern Recognition Methodology and Systems*, IEEE, 1992, pp. 1–4.

[128] X. Li, W. Mao, and W. Jiang, "Multiple-kernel-learning-based extreme learning machine for classification design," *Neural Computing and Applications*, vol. 27, no. 1, pp. 175–184, 2016.

Publications From the Thesis

Contact Information: erguluur@gmail.com

Papers

1. **Ergul U.** and Bilgin G., "HCKBoost: Hybridized composite kernel boosting with extreme learning machines for hyperspectral image classification", *NEUROCOMPUTING*, vol.334, pp.100-113, 2019.
DOI:https://doi.org/10.1016/j.neucom.2019.01.010

2. **Ergul U.** and Bilgin G., "MCK-ELM: Multiple composite kernel extreme learning machine for hyperspectral images", *NEURAL COMPUTING & APPLICATIONS*, 2019. (In Press)
DOI:https://doi.org/10.1007/s00521-019-04044-9

3. **Ergul U.** and Bilgin G., "Multiple-instance ensemble learning for hyperspectral images", *JOURNAL OF APPLIED REMOTE SENSING*, vol.11, no.4, pp.1-17, 2017.
DOI:http://dx.doi.org/10.1117/1.JRS.11.045009

Conference Papers

1. **Ergul U.** and Bilgin G., "Classification of hyperspectral images with multiple kernel extreme learning machine." *Signal Processing and Communications Applications Conference (SIU)*, 2018 26th. IEEE, 2018.

2. **Ergul U.** and Bilgin G., "Hyperspectral image classification with hybrid kernel extreme learning machine." *Signal Processing and Communications Applications Conference (SIU)*, 2017 25th. IEEE, 2017.

3. **Ergul U.** and Bilgin G., "Multiple instance bagging based ensemble classification of hyperspectral images." *Signal Processing and Communication Application Conference (SIU)*, 2016 24th. IEEE, 2016.

4. **Ergul U.** and Bilgin G., "Multiple instance bagging approach for ensemble learning methods on hyperspectral images." *Signal Processing and Communications Applications Conference (SIU)*, 2015 23th. IEEE, 2015.

Projects

1. "Hiperspektral Görüntülerin Topluluk Öğrenme Yöntemleri ile Sınıflandırılması", BAP Doktora, 2016-04-01-DOP03, Araştırmacı, 2019.

2. "Yüksek Boyutlu İşaret ve Görüntülerin Grafik İşlem Birimleri ile Paralel Hesaplama Sistemi", BAP Arastırma Projesi, 2014-04-01-KAP01, Araştırmacı, 2016.